

**LAPORAN PROJEK ARSITEKTUR KOMPUTER :
DIGITAL DOOR LOCK DENGAN KEYPAD**



DOSEN :

Norma Ningsih S.S.T.,M.T.

DISUSUN OLEH :

Akmal Muhammad Hafizh Hibatullah (2423600036)

R. Gusti Aryakusuma Dewa Wijaya (2423600058)

Ilyasa Ayasy El Ghofiqi (2423600060)

**PROGRAM STUDI TEKNOLOGI REKAYASA INTERNET
DEPARTEMEN TEKNIK ELEKTRO
POLITEKNIK ELELTRONIKA NEGERI SURABAYA**

2023

A. TUJUAN

Setelah menyelesaikan percobaan ini mahasiswa diharapkan mampu

1. Dapat memahami cara kerja doorlock dengan menggunakan keypad.
2. Memahami prinsip kerja rangkaian doorlock menggunakan 8086.
3. Mampu merangkai Prototipe Doorlock menggunakan Proteus dengan mikroprosesor 8086.

B. LANDASAN TEORI

1. Doorlock



Doorlock menggunakan keypad adalah sebuah sistem pengaman pintu yang memungkinkan pengguna untuk membuka pintu tanpa menggunakan kunci fisik. Keypad pada doorlock ini dapat digunakan untuk memasukkan kode pengaman yang telah ditentukan sebelumnya. Selain itu, terdapat doorlock menggunakan keypad juga dapat terhubung ke jaringan dan dikontrol dari jarak jauh melalui antarmuka smartphone atau komputer. Doorlock menggunakan keypad yang cocok digunakan di rumah atau kantor yang membutuhkan sistem pengamanan yang lebih canggih daripada kunci tradisional.

Salah satu kelebihan Doorlock menggunakan keypad adalah selain tidak menggunakan kunci fisik, adalah pintu dapat dibuka oleh siapapun yang mengetahui pin dari doorlock tersebut. Saat ini doorlock menggunakan keypad telah di gunakan untuk mengamankan seperti kotak brankas, ruangan rahasia,

dan lain – lain yang memerlukan keamanan tingkat tinggi. Saat ini doorlock telah berkembang dan menghadirkan beberapa opsi lain untuk membuka pintu, contohnya adalah kartu RFID, dan Fingerprint. Beberapa opsi tersebut tentunya memiliki kelebihan masing – masing.

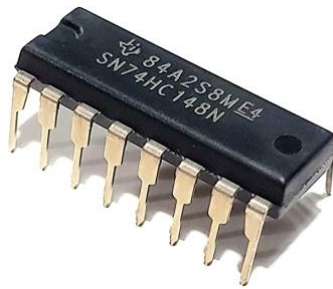
Sistem kerja dari doorlock sendiri terbilang cukup sederhana, yaitu pengguna memasukkan pin atau passkey dengan menekan keypad, lalu sistem mengoreksi apakah pin yang dimasukkan benar atau tidak, lalu sistem akan membuka pintu dengan memberikan arus listrik ke pengunci pintu untuk membuka pintu tersebut.

2. Microprosesor 8086



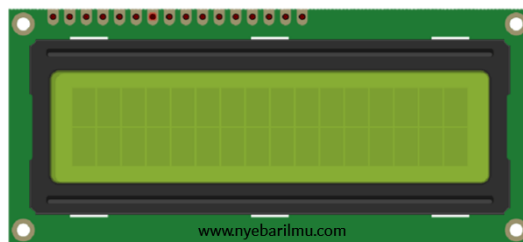
Intel 8086 merupakan sebuah chip mikroprosesor 16-bit rancangan Intel pada tahun 1978 yang membangkitkan penggunaan arsitektur x86. Tidak lama kemudian, Intel 8088 diperkenalkan dengan bus 8-bit external, yang memungkinkan penggunaan chipset yang murah. 8086 dirancang berdasarkan Intel 8080 dan Intel 8085 dengan set register yang mirip, tetapi dikembangkan menjadi 16 bit. "Bus Interface Unit" memberikan rangkaian instruksi ke "Execution Unit" melalui sebuah prefetch queue 6 byte, jadi pemberian dan pelaksanaan dilakukan bersamaan- sebuah bentuk pipelining primitif (instruksi 8086 bervariasi dari 1 sampai 4 byte). Proyek ini kami membuat rangkaian prototipe Doorlock menggunakan mikroprosesor 8086 dengan Proteus.

3. IC Encoder 74HC138



IC 74HC138 adalah multiplexer 8-input yang dirancang untuk memilih dan mentransfer data antara input dan output eksternal. IC ini termasuk dalam keluarga gerbang logika TTL 74xx. IC ini adalah decoder 8-to-3 line yang dirancang untuk aplikasi encoding atau multiplexing. IC ini memiliki 8 input dan 3 output. Dalam sistem memori berkinerja tinggi, encoder ini dapat digunakan untuk meminimalkan penggunaan pin input pada I/O 8255A. IC ini akan digunakan sebagai enkoder yang akan menerjemahkan input dari keypad menjadi biner yang mempermudah dalam pemrograman processor 8086 serta menjadi lebih efisien.

4. LCD 16x2



LCD singkatan dari Liquid Crystal Display. Modul ini terdiri dari dua baris di mana setiap baris dapat menghasilkan 16 karakter. Modul tampilan elektronik yang digunakan dalam berbagai aplikasi seperti berbagai rangkaian & perangkat seperti ponsel, kalkulator, komputer, perangkat TV, dan lain – lain.

Berbeda dengan 7-segment yang hanya dapat menampilkan angka dan beberapa huruf alfabet, LCD 16x2 ini dapat menampilkan hampir semua huruf dari angka, alfabet, hingga karakter khusus dan karakter spesial, maka dari itu

LCD sangat cocok digunakan sebagai display dengan sistem minimum yang kami gunakan untuk project ini.

5. Buzzer



Buzzer adalah perangkat elektronik yang digunakan untuk menghasilkan suara atau bunyi. Buzzer piezoelektrik menghasilkan suara dengan memanfaatkan getaran kristal piezoelektrik, sedangkan buzzer elektromagnetik menghasilkan suara dengan memanfaatkan medan magnet. Buzzer biasanya digunakan dalam aplikasi yang memerlukan peringatan suara, seperti alarm, timer, dan konfirmasi input pengguna seperti klik mouse atau tombol keyboard. Dengan project ini Buzzer digunakan sebagai alarm kepada pengguna sebagai konfirmasi bahwa passkey yang telah dimasukan benar atau salah.

6. DC-Motor



DC motor adalah perangkat elektronik yang mengubah energi listrik menjadi energi mekanik. DC motor biasanya digunakan dalam aplikasi yang memerlukan pergerakan mekanik, seperti mesin industri, robot, dan kendaraan listrik. Dalam project ini DC-Motor digunakan sebagai penggerak sistem pengunci dan pembuka pintu, disamping itu project ini lebih logis jika menggunakan solenoid sebagai sistem pengunci pintu karena lebih mudah digunakan, kuat serta lebih aman.

7. Transistor



Transistor adalah perangkat semikonduktor yang digunakan untuk menguatkan atau mengalihkan sinyal listrik dan daya. Transistor terdiri dari bahan semikonduktor, biasanya dengan setidaknya tiga terminal untuk terhubung ke rangkaian elektronik. Transistor dalam project ini digunakan sebagai pengatur tegangan yang akan keluar menuju output DC-Motor dan LED. Alternatif lain yang dapat digunakan adalah Relay, berbeda dengan transistor yang mengatur sinyal listrik dengan cara sirkuit, Relay mengatur keluaran listrik secara fisik diibaratkan seperti saklar yang dikontrol secara otomatis menggunakan aliran listrik yang lebih kecil.

8. RGB LED-CC



LED RGB adalah dioda pemancar cahaya yang dapat menghasilkan tiga warna dasar yaitu merah, hijau, dan biru. Warna dari LED ini dapat diatur dari setiap warna dasar yang telah disebutkan dengan mengatur besar dan kecil tegangan yang diberikan. RGB LED ini digunakan untuk memberikan penglihatan kepada pengguna jika password yang diberikan benar akan mengeluarkan cahaya hijau, sedangkan jika salah akan mengeluarkan cahaya merah.

C. Tools dan Bahan

Tools :

1. Proteus Professional :

<https://downloadly.net/2020/13/3175/03/proteus/03/?#/3175-proteus032127081430.html>

2. EMU8086 :

https://drive.google.com/drive/folders/1OPVhsYiHJm3_rfvUWiqL9yJW5Wn7S3LU

Bahan percobaan :

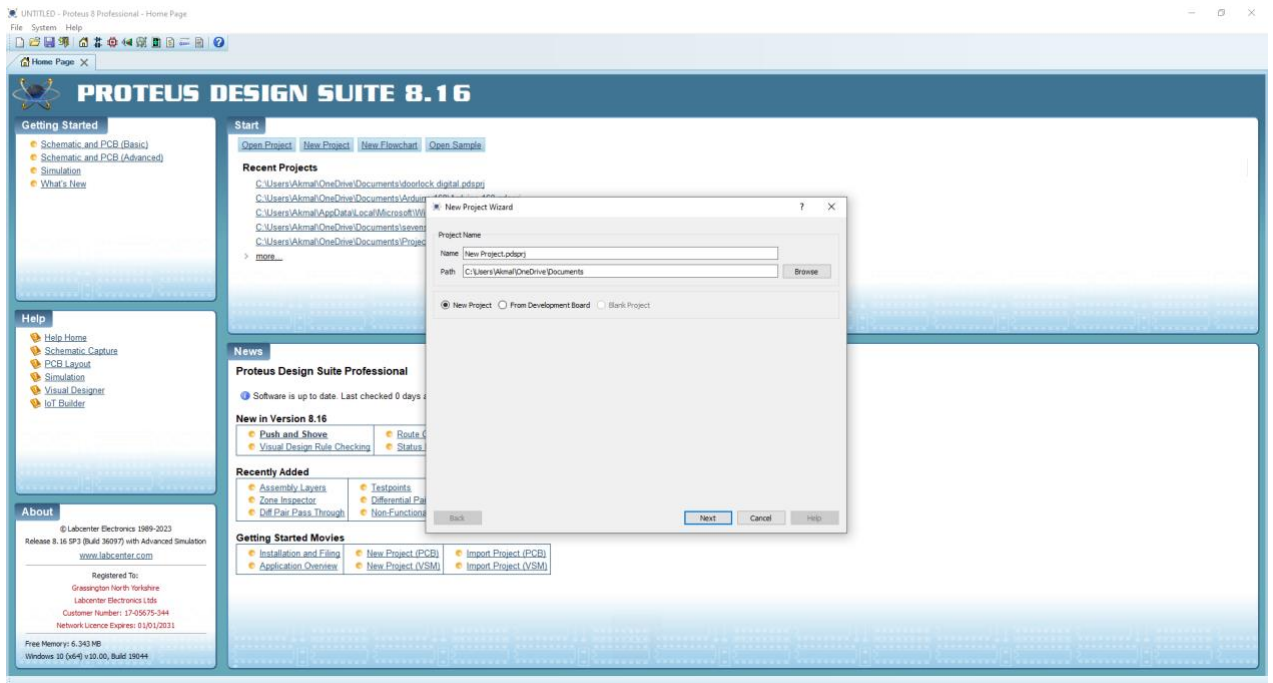
1. Datasheet Intel 8086
2. Instruction set for Intel 8086
3. Datasheet Encoder 74HC148
4. Datasheet IC 8255A
5. Datasheet IC 74HC373
6. Datasheet LCD 16×2

Komponen :

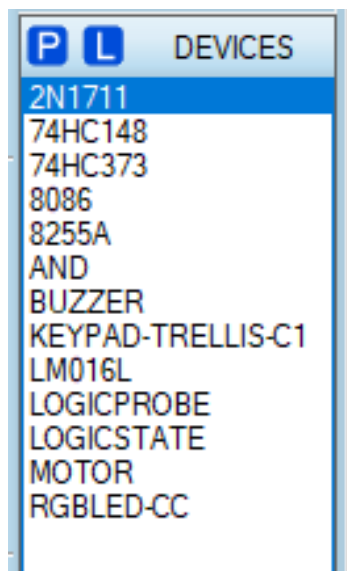
1. Microprocessor Intel 8086
2. IC 74HC373
3. IC Encoder 74HC138(2)
4. IC 8255A
5. LCD16×2
7. Gerbang AND (3)
8. Transistor
9. RGB LED-CC
10. Buzzer
11. Keypad-Trellis-C1
12. DC-Motor

D. Langkah Percobaan

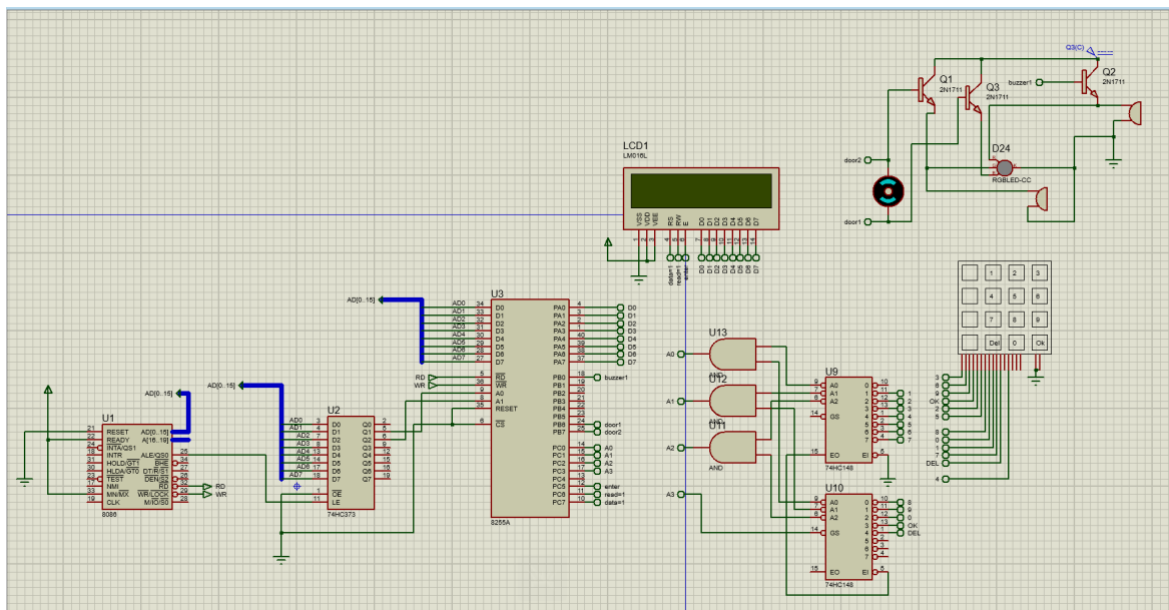
1. Buka proteus dan klik New Project, berikan nama pada project kemudian klik Next



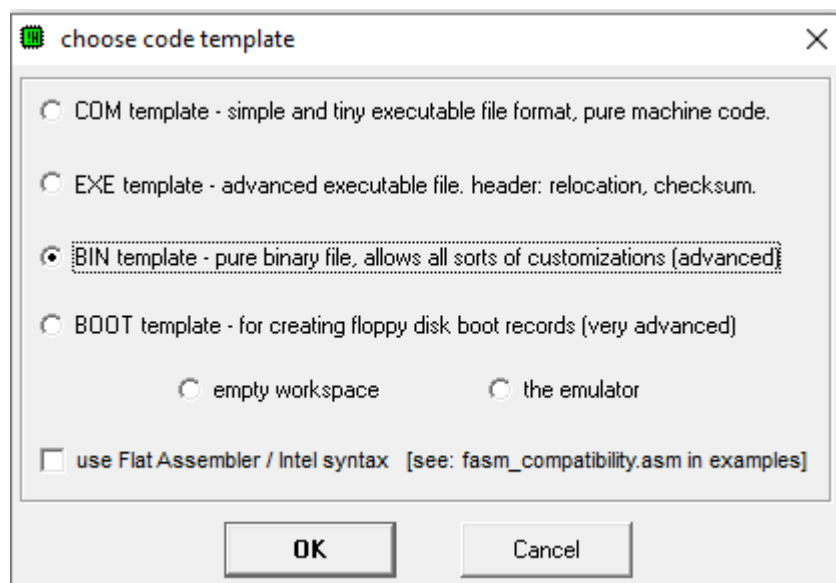
2. Kemudian klik Device (P) dan pilih alat dan bahan yang di butuhkan



3. Lalu rangkai komponen tersebut menjadi seperti di gambar berikut.



4. Kemudian buka EMU8086 buat file baru , pilih BIN template, klik ok



5. Ketik kode assembly berikut

```

0001 org 0000h
0002 JMP INIT
0003
0004 PORTA EQU 00H
0005 PORTB EQU 02H
0006 PORTC EQU 04H
0007 PORT_CON EQU 06H
0008
0009 NUM DB '#1234567890'
0010 PASS DB '123'
0011 KEY DB 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0012
0013 OPEN DB ' DOOR OPENED$'
0014 OPEN3 DB ' DOOR LOCKS IN\ 3$'
0015 OPEN2 DB ' DOOR LOCKS IN\ 2$'
0016 OPEN1 DB ' DOOR LOCKS IN\ 1$'
0017 LOCKED DB ' DOOR LOCKED$'
0018
0019 ENTER DB ' ENTER PASSWORD\$'
0020 WRONG DB ' WRONG PASSWORD\ TRY AGAIN$'
0021
0022 MAX EQU 8
0023
0024
0025 A: ; ARGUMENT<BX:NUMBER, SI:CURSOR-POSITION>
0026 CMP SI, MAX
0027 JE KEYPAD
0028
0029 MOV AL, NUM[BX]
0030 MOV KEY[SI], AL
0031
0032 OUT PORTA, AL
0033 CALL ENTER_DATA
0034 CALL DELAY
0035 CALL DELAY
0036 CALL DELAY
0037
0038 INC SI
0039 JMP KEYPAD
0040
0041
0042
0043 ENTER_DATA:
0044 MOV AL, 10100000B
0045 OUT PORTC, AL
0046
0047 MOV AL, 10000000B
0048 OUT PORTC, AL
0049 RET
0050
0051 ENTER_INS:
0052 MOV AL, 00100000B
0053 OUT PORTC, AL
0054
0055 MOV AL, 00000000B
0056 OUT PORTC, AL
0057 RET
0058
0059
0060
0061 DELAY:
0062 MOV CX, 01F36H
0063 MYLOOP: LOOP MYLOOP
0064 RET
0065 DELAY1:
0066 MOV CX, 0DF36H
0067 MYLOOP1: LOOP MYLOOP
0068 RET
0069

```

```

071 CHECK:
072     MOV SI,0
073
074     LOOPCHECK:
075         MOV AL, KEY[SI]
076         CMP AL, PASS[SI]
077         MOV KEY[SI],0
078         JNE SALAH
079         INC SI
080         CMP SI, MAX
081         JNE LOOPCHECK
082     JMP BENAR
083
084     CLEAR:
085         MOV SI,0
086
087     LOOPCLEAR:
088         MOV KEY[SI],0
089         INC SI
090         CMP SI, MAX
091         JNZ LOOPCLEAR
092
093         MOV AL, 00000001B
094         OUT PORTA, AL
095         CALL ENTER_INS
096
097         MOV SI,0
098     RET
099
100     PRINT: ;ARGUMENT(DI: TEXT)
101         MOV AL,[DI]
102
103         CMP [DI],'\ '
104         JNE SKIP
105         CALL BR
106         JE NEXT
107         SKIP:
108
109         OUT PORTA,AL
110         CALL ENTER_DATA
111
112         NEXT:
113         INC DI
114         CMP [DI], '$ '
115         JNE PRINT
116         RET
117
118     BR:
119         MOV AL, 11000000B
120         OUT PORTA, AL
121         CALL ENTER_INS
122     RET
123
124
125
126     BENAR:
127
128         ;MOTOR ON
129         MOV AL,10000000B
130         OUT PORTB,AL
131
132         CALL CLEAR
133         LEA DI,OPEN
134         CALL PRINT
135
136         CALL DELAY1
137         CALL CLEAR
138
139
140         ;MOTOR OF
141         MOV AL,00000000B
142         OUT PORTB,AL

```

```

144     LEA DI,OPEN3
145     CALL PRINT
146
147     CALL DELAY1
148     CALL CLEAR
149
150     LEA DI,OPEN2
151     CALL PRINT
152
153     CALL DELAY1
154     CALL CLEAR
155
156     LEA DI,OPEN1
157     CALL PRINT
158
159     CALL DELAY1
160     CALL CLEAR
161
162     ;MOTOR ON
163     MOV AL,01000000B
164     OUT PORTB,AL
165
166     CALL DELAY1
167
168     ;MOTOR OF
169     MOV AL,00000000B
170     OUT PORTB,AL
171
172     LEA DI,LOCKED
173     CALL PRINT
174     CALL DELAY1
175     JMP START
176
177
178     SALAH:
179         ;BUZZER ON
180         MOV AL,00000001B
181         OUT PORTB,AL
182
183         CALL CLEAR
184         LEA DI,WRONG
185         CALL PRINT
186
187         CALL DELAY
188         ;BUZZER OFF
189         MOV AL,0B
190         OUT PORTB,AL
191
192         CALL DELAY1
193     JMP START
194
195     INIT:
196         MOV DX, PORT_CON
197         MOV AL, 10000001B
198         OUT DX, AL
199
200         ;ON CURSOR
201         MOV AL, 00001100B
202         OUT PORTA, AL
203         CALL ENTER_INS
204
205         ;2 LINE MODE
206         MOV AL, 00111100B
207         OUT PORTA, AL
208         CALL ENTER_INS
209
210     START:
211         CALL CLEAR
212         LEA DI,ENTER
213         CALL PRINT
214
215         MOV SI,0

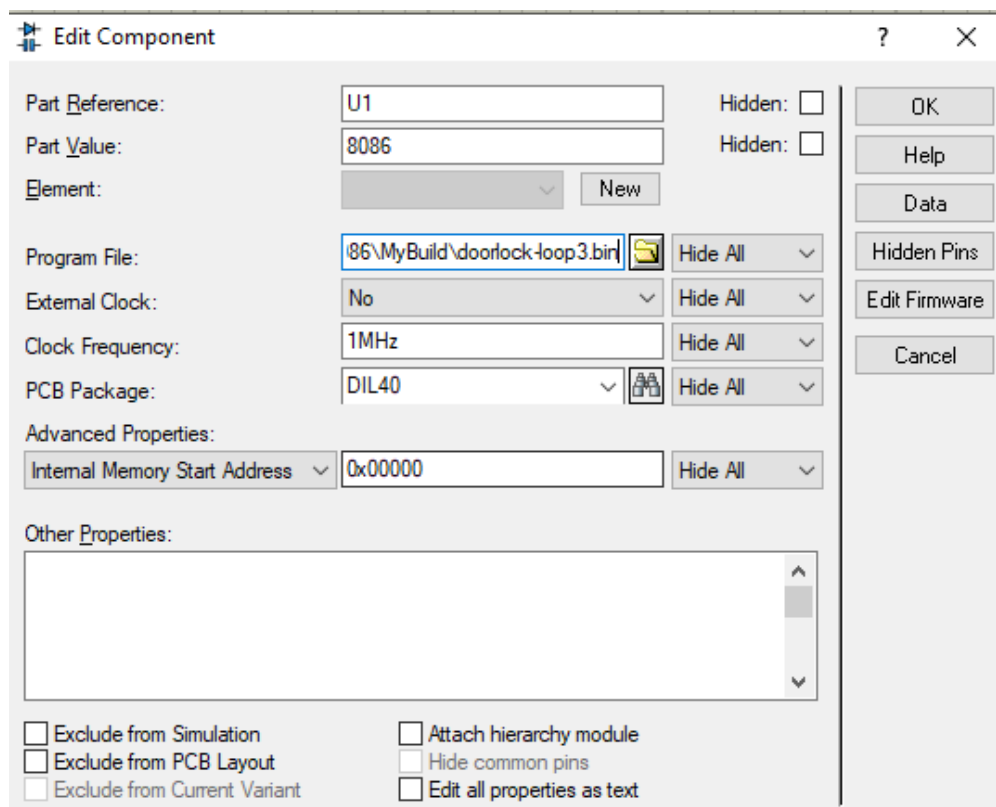
```

```

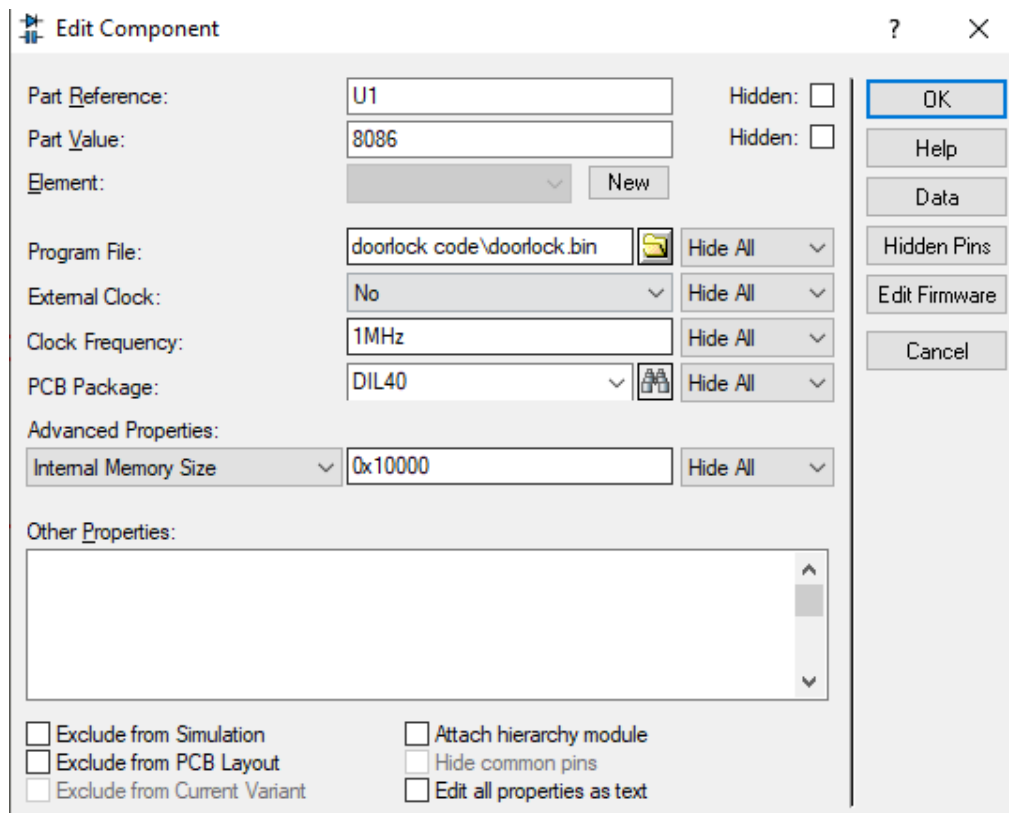
217 KEYPAD:
218     MOV BX,0
219     MOV AL,0B
220     OUT PORTC,AL
221
222 KEYPADLOOP:
223     INC BL
224     IN AL, PORTC
225     XOR AL,00001111B
226     CMP AL, BL
227     JE A
228     CMP BL,10
229     JNE KEYPADLOOP
230
231
232     IN AL, PORTC
233     XOR AL,00001111B
234     CMP AL, 11
235     JE CHECK
236
237     IN AL, PORTC
238     XOR AL,00001111B
239     CMP AL, 12
240     JNE KEYPAD
241     JE START
242
243 JMP KEYPAD
244
245
246 HLT

```

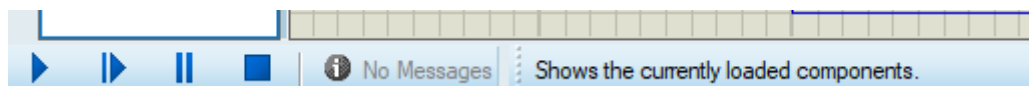
6. Compile program yang telah dibuat dan simpan file *.bin pada directory yang diinginkan.
7. Masukkan file *.bin yang telah digenerate pada simulator proteus, arahkan ke directory dimana file *.bin tersimpan



8. Kemudian klik pada advanced properties klik tanda panah kebawah klik internal memory size kemudian ganti seperti yang ada di gambar dibawah



9. Klik ok dan kemudian jalankan dengan menekan tombol play yang ada di bagian bawah



E. ANALISA

Prototipe Door Lock dengan Keypad ini dibuat menggunakan Bahasa Assembly yang di tenagai oleh mikroprosesor 8086. Secara dasar komponen input yakni keypad sendiri menggunakan keypad Trellis dikarenakan lebih mudah dikodekan dibanding menggunakan keypad standar. Masukan keypad tersebut lalu di kodekan dengan IC encoder 74HC148, yang mengubah input per pin menjadi bilangan binary.

Penjelasan Kode:



Deklarasi

1. Deklarasi Variabel

Array2 yg fungsinya berkaitan dgn password

```
NUM    DB  '#1234567890'  
PASS   DB  '1453'  
KEY     DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Num: Karakter yang akan ditampilkan LCD saat keypad ditekan, karakter yang ditampilkan pada setiap tombol di keypad sesuai dengan urutan index pada array tersebut.

Pass : Kunci password yang benar

Key : Array yang digunakan untuk menyimpan password yang dimasukan user

Berikut Alur Sistem pengecekan Passkey

- Key no 1 ditekan -> maka program akan mencari indeks ke-1 dari array NUM-> karakter '1' ditampilkan ke lcd -> karakter '1' disimpan pada KEY indeks ke-0
- Key no 4 ditekan -> maka program akan mencari indeks ke-4 dari array NUM-> karakter '4' ditampilkan ke lcd -> karakter '4' disimpan pada KEY indeks ke-1
-dst
- Ketika tombol OK ditekan -> satu persatu karakter pada KEY dicek apakah sama dengan PASS

Array yang berkaitan dengan teks – teks yang akan ditampilkan di LCD

```
OPEN      DB  ' DOOR OPENED$'  
OPEN3     DB  ' DOOR LOCKS IN\ 3$'  
OPEN2     DB  ' DOOR LOCKS IN\ 2$'  
OPEN1     DB  ' DOOR LOCKS IN\ 1$'  
LOCKED    DB  ' DOOR LOCKED$'  
  
ENTER     DB  ' ENTER PASSWORD\$',  
WRONG     DB  ' WRONG PASSWORD\ TRY AGAIN$'
```

'\$' karakter penanda sebagai akhir dari string ditampilkan

'\` karakter untuk ganti baris kebawah atau enter

Maximal digit password yg bisa diinputkan user

MAX EQU 8

Jika digit yang diinputkan user berlebih maka digit yang lebih akan diabaikan.

2. Deklarasi Fungsi

```
A: ; ARGUMENT(BX:NUMBER, SI:CURSOR-POSITION)
  CMP SI, MAX
  JE KEYPAD

  MOV AL, NUM[BX]
  MOV KEY[SI], AL

  OUT PORTA, AL
  CALL ENTER_DATA
  CALL DELAY
  CALL DELAY
  CALL DELAY

  INC SI
  JMP KEYPAD
```

Fungsi diatas berfungsi untuk menangani ketika keypad ditekan, mencetak karakter ke LCD dan menyimpan nilai nya di array KEY

Utility Function

Fungsi yang melakukan tugas – tugas umum yang sering digunakan sehingga dapat menghindari redundansi kode dan memudahkan untuk dibaca

```
ENTER_DATA:
  MOV AL, 10100000B
  OUT PORTC, AL

  MOV AL, 10000000B
  OUT PORTC, AL
  RET

ENTER_INS:
  MOV AL, 00100000B
  OUT PORTC, AL

  MOV AL, 00000000B
  OUT PORTC, AL
  RET

DELAY:
  MOV CX, 01F36H
MYLOOP: LOOP MYLOOP
  RET
DELAY1:
  MOV CX, 0DF36H
MYLOOP1: LOOP MYLOOP1
  RET
```

Enter Data fungsi yang menghasilkan output perintah khusus untuk ‘meng-enter’ karakter tertentu agar tampil pada LCD

Enter Data fungsi yang menghasilkan output perintah khusus untuk ‘meng-enter’ instruksi tertentu pada LCD

Delay Fungsi yang untuk memberikan jeda sekian detik / milidetik


```

CHECK:
    MOV SI, 0

LOOPCHECK:
    MOV AL, KEY[SI]
    CMP AL, PASS[SI]
    MOV KEY[SI], 0
    JNE SALAH
    INC SI
    CMP SI, MAX
    JNE LOOPCHECK
    JMP BENAR

```

Fungsi untuk melakukan pengecekan apakah password yang dimasukkan benar atau tidak, dengan melakukan membandingkan satu per satu karakter dari array KEY dan PASS. Jika ditemukan ada karakter yang beda maka break dan loncat ke fungsi salah. Jika tidak ditemukan 1 karakter pun yang berbeda panggil fungsi benar

```

CLEAR:
    MOV SI, 0

LOOPCLEAR:
    MOV KEY[SI], 0
    INC SI
    CMP SI, MAX
    JNZ LOOPCLEAR

    MOV AL, 00000001B
    OUT PORTA, AL
    CALL ENTER_INS

    MOV SI, 0
    RET

```

Fungsi untuk menghapus semua karakter yang telah diinputkan user

```

PRINT: ;ARGUMENT(DI: TEXT)
    MOV AL, [DI]

    CMP [DI], '\n'
    JNE SKIP
    CALL BR
    JE NEXT
    SKIP:

    OUT PORTA, AL
    CALL ENTER_DATA

    NEXT:
    INC DI
    CMP [DI], '$'
    JNE PRINT
    RET

BR:
    MOV AL, 11000000B
    OUT PORTA, AL
    CALL ENTER_INS
    RET

```

Fungsi untuk mencetak teks, dengan melakukan perulangan pada string sampai ditemukan karakter '\$', dan jika ditemukan karakter '\n' maka ganti baris

BENAR:

```
;MOTOR ON
MOV AL,10000000B
OUT PORTB,AL

CALL CLEAR
LEA DI,OPEN
CALL PRINT

CALL DELAY1
CALL CLEAR

;MOTOR OF
MOV AL,00000000B
OUT PORTB,AL

LEA DI,OPEN3
CALL PRINT

CALL DELAY1
CALL CLEAR

LEA DI,OPEN2
CALL PRINT

CALL DELAY1
CALL CLEAR

LEA DI,OPEN1
CALL PRINT

CALL DELAY1
CALL CLEAR

;MOTOR ON
MOV AL,01000000B
OUT PORTB,AL

CALL DELAY1
```

Fungsi yang akan melakukan tindakan jika password yang dimasukkan benar, fungsi ini akan menampilkan teks dan menyalakan motor (membuka pintu)

SALAH:

```
;BUZZER ON
MOV AL,00000001B
OUT PORTB,AL

CALL CLEAR
LEA DI,WRONG
CALL PRINT

CALL DELAY
;BUZZER OFF
MOV AL,0B
OUT PORTB,AL

CALL DELAY1
JMP START
```

Fungsi yang akan melakukan tindakan jika password yang dimasukkan benar, fungsi ini akan menampilkan teks dan menyalakan buzzer.

Main Program

```

INIT:
MOV DX, PORT_CON
MOV AL, 10000001B
OUT DX, AL

;ON CURSOR
MOV AL, 00001100B
OUT PORTA, AL
CALL ENTER_INS

;2 LINE MODE
MOV AL, 00111100B
OUT PORTA, AL
CALL ENTER_INS

```

Kode yang akan dijalankan sekali saat program baru saja dimulai. Kode ini berguna untuk mengatur port, menyalakan display pada LCD, dan mengatur LCD ke mode 2 baris

```

START:
CALL CLEAR
LEA DI, ENTER
CALL PRINT

MOV SI, 0

KEYPAD:
MOV BX, 0
MOV AL, 0B
OUT PORTC, AL

KEYPADLOOP:
INC BL
IN AL, PORTC
XOR AL, 00001111B
CMP AL, BL
JE A
CMP BL, 10
JNE KEYPADLOOP

IN AL, PORTC
XOR AL, 00001111B
CMP AL, 11
JE CHECK

IN AL, PORTC
XOR AL, 00001111B
CMP AL, 12
JNE KEYPAD
JE START

JMP KEYPAD

```

Kode tersebut akan dijalankan berulang2 menunggu keypad ditekan oleh pengguna.

F. KESIMPULAN

Door Lock dengan Keypad merupakan perangkat yang memiliki tujuan mempermudah pengguna untuk mengamankan sebuah pintu / ruangan. Hanya dengan memasukan PIN pengguna dapat mengakses pintu yang diamankan, tanpa menggunakan kunci fisik. Komponen utama yang digunakan adalah keypad, LCD dan Buzzer sebagai interface output ke pengguna.

Alasan utama proyek ini dibuat adalah, kami ingin membuat sesuatu yang mudah dibuat atau rekayasa, dapat dikembangkan tanpa batas, serta telah banyak digunakan umat manusia.

Door Lock dengan Keypad menggunakan prosesor Intel 8086 merupakan suatu hal yang menantang bagi kami, bahasa assembly yang tergolong rumit, serta keterbatasan fitur yang tidak bisa kami tambahkan dalam proyek ini. Selain itu pembuatan proyek dengan berdasarkan aplikasi proteus juga menambah tantangan, simulasi yang sering gagal dan kurangnya komponen – komponen yang diperlukan.

Akan tetapi kami tidak menyerah dan mencari solusi yang terbaik dalam pembuatan proyek ini, salah satunya adalah penggunaan LCD 16x2 yang setelah dipelajari jauh lebih mudah digunakan dan memiliki banyak fitur dibandingkan menggunakan 7-Segment. Selain itu dari proyek ini kami dapat mempelajari cara kerja sistem mekanisme dari Door Lock dan cara kerja password yang dimasukan di periksa dan memberikan output benar atau salah.