

**LAPORAN RESMI
PRAKTIKUM 4 ARSITEKTUR KOMPUTER**

“SEVEN-SEGMENT”



**Disusun Oleh :
Izzuddin Ahmad Afif (2421600011)**

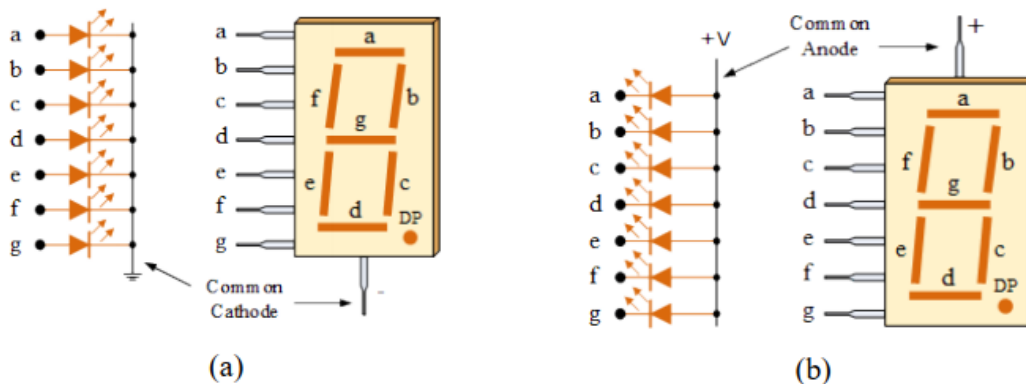
**Dosen :
Mohamad Ridwan S.T., M.T.**

**PROGRAM STUDI SARJANA TERAPAN TEKNOLOGI REKAYASA INTERNET
DEPARTEMEN TEKNIK ELEKTRO
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2021/2022**

BAB I PENDAHULUAN

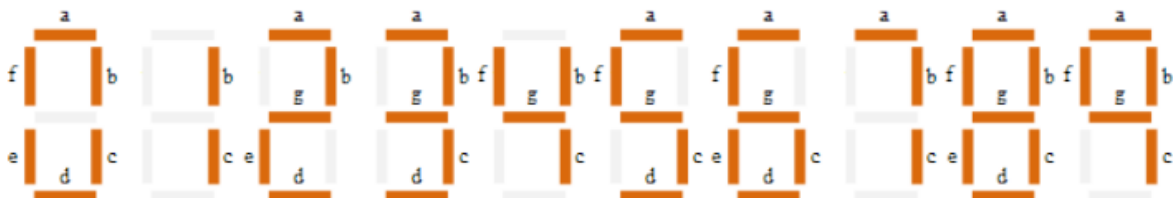
1. Dasar Teori

Tampilan 7-segmen, juga ditulis sebagai "seven segmen", terdiri dari tujuh LED (Light Emitting Diode) yang disusun sedemikian rupa, sehingga jika LED-LED itu menyala dapat merepresentasikan sebagai sebuah angka. Masing-masing dari tujuh LED disebut segmen karena ketika diterangi segmen tersebut membentuk bagian dari digit numerik (baik Desimal dan Hex) yang akan ditampilkan. LED ke-8 tambahan terkadang digunakan dalam paket yang sama sehingga memungkinkan indikasi titik desimal, (DP) ketika dua atau lebih tampilan 7-segmen dihubungkan bersama untuk menampilkan angka yang lebih besar dari sepuluh. Karena setiap LED memiliki dua pin penghubung, satu disebut "Anoda" dan yang lainnya disebut "Katoda", oleh karena itu ada dua jenis tampilan 7-segmen LED yang disebut: Common Cathode (CC) dan Common Anode (CA). Dalam tampilan common cathode (CC), semua sambungan katoda dari segmen LED digabungkan menjadi logika "0" atau ground. Setiap segment diberikan logic "HIGH", atau logika "1" melalui resistor pembatas arus untuk membias maju terminal Anoda setiap segment (a-g). Dalam tampilan Common Anode (CA), semua koneksi anoda dari segmen LED digabungkan menjadi satu dan diberikan logika "1" atau terhubung dengan supply (3V/5V). Setiap segment diberikan logic "LOW" atau logika "0" melalui resistor pembatas arus yang sesuai ke setiap segmen (a-g)



Gambar 1. (a) Common Cathode (CA); (b) Common Anode (CA)

Berikut adalah nyala setiap segmen untuk masing tampilan angka 0-9



BAB II

METODOLOGI PRAKTIKUM

1. Alat atau Bahan Praktikum

a. Tools :

1. Proteus Professional :

<https://downloadly.net/2020/13/3175/03/proteus/03/?#/3175-proteus-032127081430.html>

2. EMU8086 :

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

b. Bahan percobaan :

1. Datasheet Intel 8086

2. Instruction set for Intel 8086

3. Datasheet IC 8255 PPI

4. Datasheet IC 74HC373

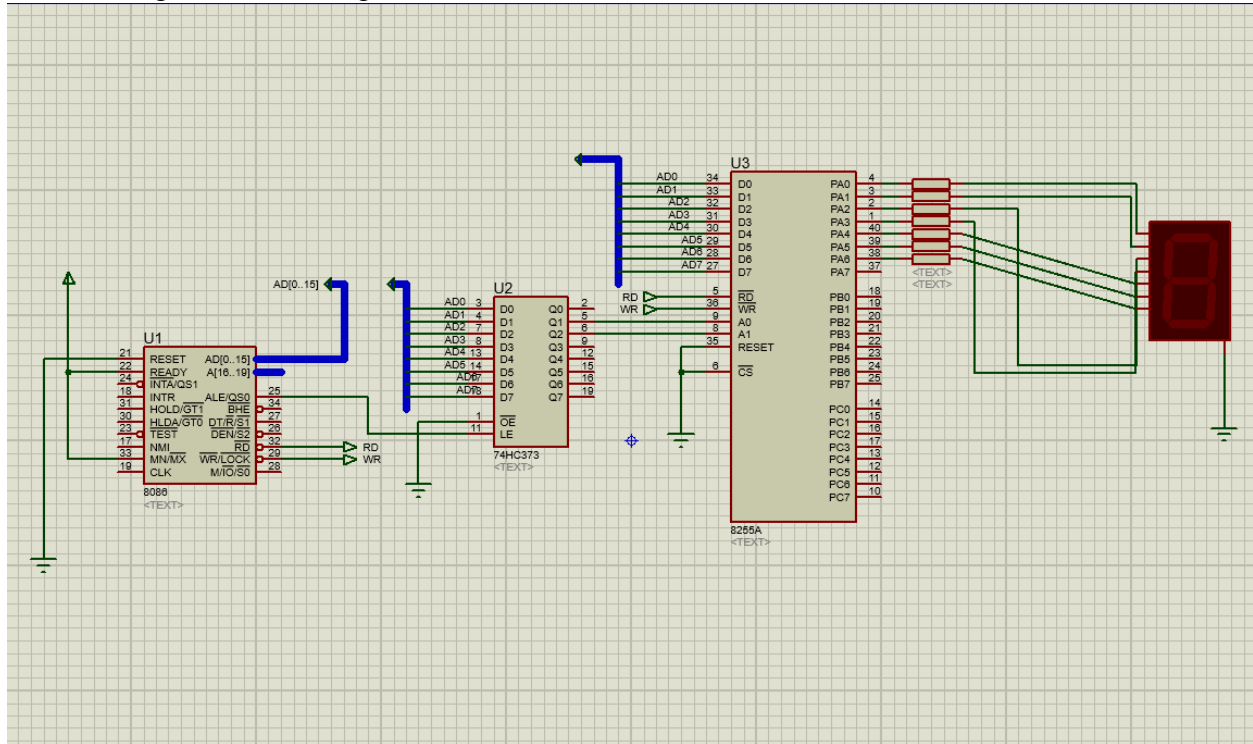
5. 7-Segment

BAB III

ANALISA DAN KESIMPULAN

3.1 Hasil Praktikum

1. Rangkaian seven-segment:



Source code awal:

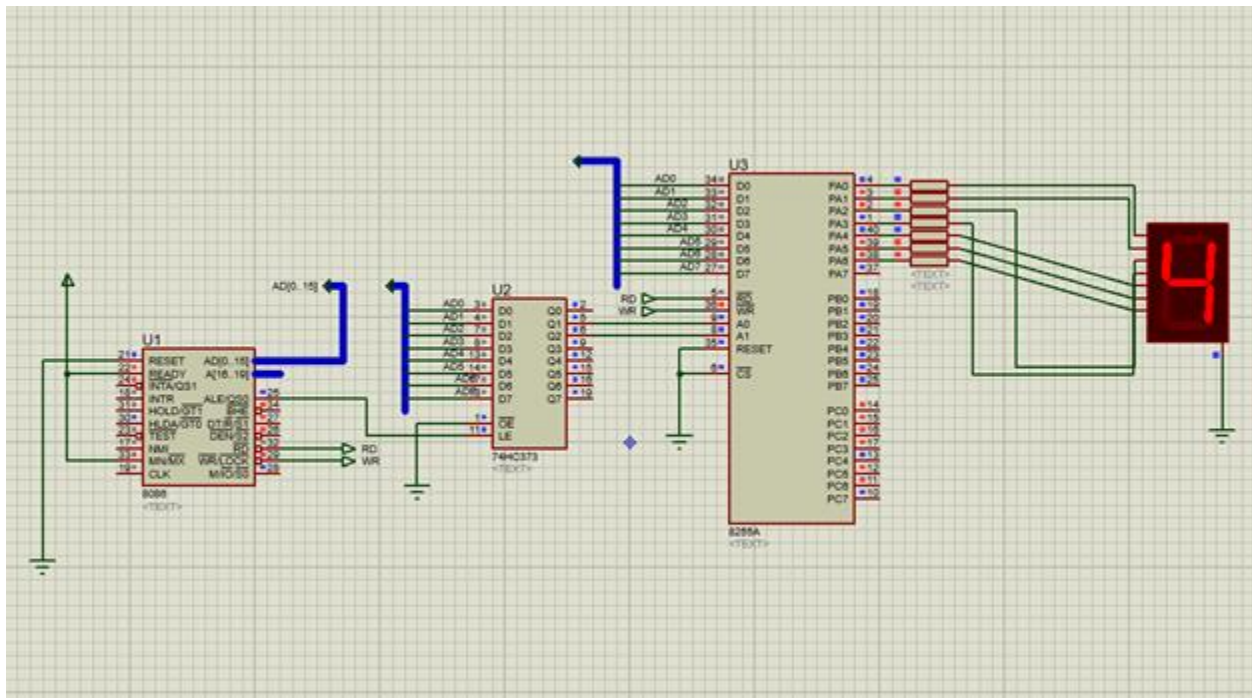
```
049 ;add your code here
050 START:
051
052 MOV DX, PORT_CON
053 MOV AL, 10000001B; PORT A as Output
054 OUT DX, AL
055
056 JMP XX
057
058 XX:
059
060 MOV AL, 00111111B ; displaying 0
061 MOV DX, PORTA
062 OUT DX, AL
063 MOV CX, 0DF36H; Delay
064 Delay0: loop Delay0
065
066 MOV AL, 0000110B ; displaying 1
067 MOV DX, PORTA
068 OUT DX, AL
069 MOV CX, 0DF36H; Delay
070 Delay1: loop Delay1
071
072 MOV AL, 01011011B ; displaying 2
073 MOV DX, PORTA
074 OUT DX, AL
075 MOV CX, 0DF36H; Delay
076 Delay2: loop Delay2
077
078 MOV AL, 01001111B ; displaying 3
079 MOV DX, PORTA
080 OUT DX, AL
081 MOV CX, 0DF36H; Delay
082 Delay3: loop Delay3
083
084 MOV AL, 01100110B ; displaying 4
085 MOV DX, PORTA
086 OUT DX, AL
087 MOV CX, 0DF36H; Delay
088 Delay4: loop Delay4
089
090 MOV AL, 01101101B ; displaying 5
091 MOV DX, PORTA
092 OUT DX, AL
093 MOV CX, 0DF36H; Delay
094 Delay5: loop Delay5
095
096 MOV AL, 01111101B ; displaying 6
097 MOV DX, PORTA
098 OUT DX, AL
099 MOV CX, 0DF36H; Delay
100 Delay6: loop Delay6
101
102 MOV AL, 00000111B ; displaying 7
103 MOV DX, PORTA
104 OUT DX, AL
105 MOV CX, 0DF36H; Delay
106 Delay7: loop Delay7
```

```

107
108 MOV AL, 01111111B ; displaying 8
109 MOV DX, PORTA
110 OUT DX, AL
111 MOV CX, 0DF36H; Delay
112 Delay8: loop Delay8
113
114 MOV AL, 01101111B ; displaying 9
115 MOV DX, PORTC
116 OUT DX, AL
117 MOV CX, 0DF36H; Delay
118 Delay9: loop Delay9|
119
120 JMP XX
121
122 JMP START
123
124 CODE ENDS
125 END
126
127
128 HLT ; halt!
129

```

Hasil Run source code awal:



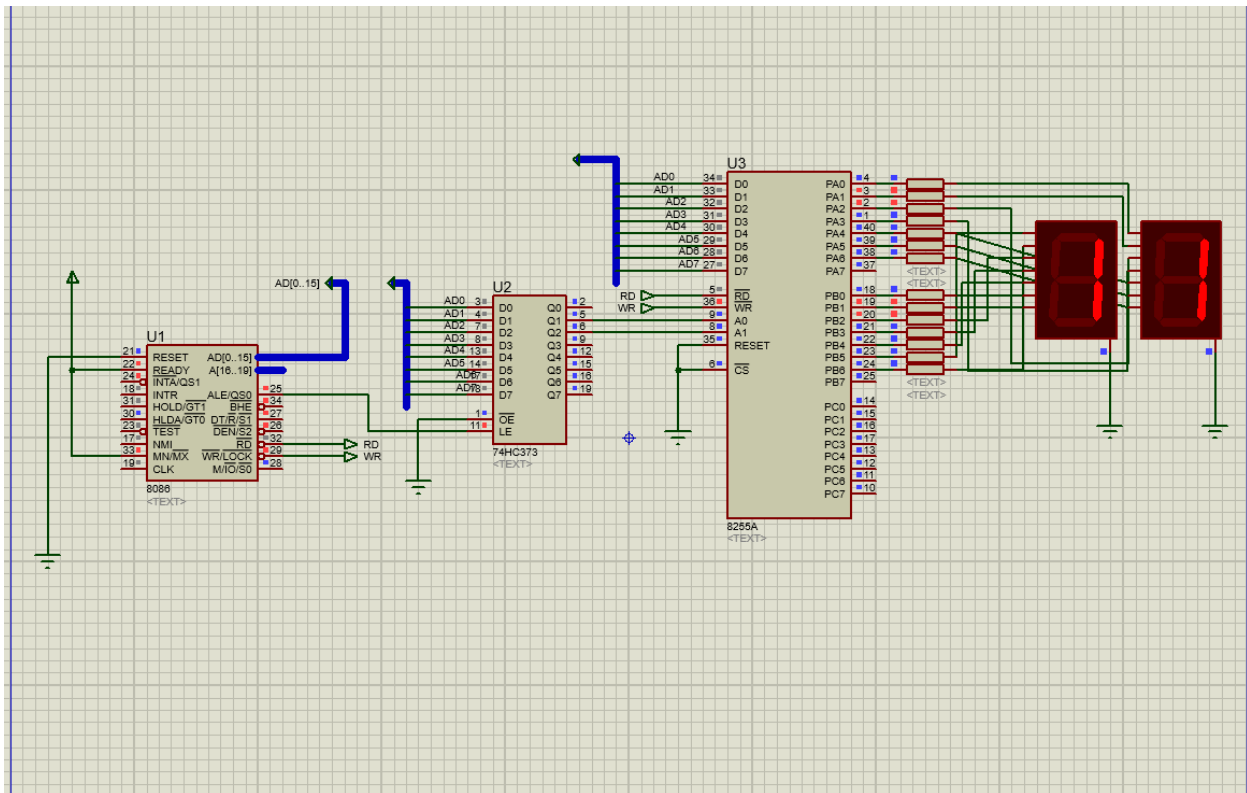
(seven segment menunjukkan angka 0-9 dengan delay 1 detik dalam loop)

```

49 ;add your code here
50 START:
51
52 MOV DX, PORT_CON
53 MOV AL, 10000000B; PORT A as Output
54 OUT DX, AL
55
56 JMP XX
57
58 XX:
59
60
61 MOV AL, 0000110B ; displaying 1
62 MOV DX, PORTA
63 OUT DX, AL
64
65 MOV AX, 0000110B ; displaying 1
66 MOV DX, PORTB
67 OUT DX, AX
68
69 JMP XX
70
71 CODE ENDS
72 END
73
74
75 HLT ; halt!
76

```

Hasil run code NRP:



3. Source code solusi tugas:

```
;add your code here
START:
MOV DX, PORT_CON
MOV AL, 10000000B; PORT A as Output
OUT DX, AL

XA:

MOV AL, 00111111B ; displaying 0
MOV DX, PORTB
OUT DX, AL
CALL XS

MOV AL, 00000110B ; displaying 1
MOV DX, PORTB
OUT DX, AL
CALL XS

MOV AL, 01011011B ; displaying 2
MOV DX, PORTB
OUT DX, AL
CALL XS

MOV AL, 01001111B ; displaying 3
MOV DX, PORTB
OUT DX, AL
CALL XS

MOV AL, 01100110B ; displaying 4
MOV DX, PORTB
OUT DX, AL
CALL XS

MOV AL, 01101101B ; displaying 5
MOV DX, PORTB
OUT DX, AL
CALL XS

MOV AL, 01111101B ; displaying 6
MOV DX, PORTB
OUT DX, AL
CALL XS

MOV AL, 00000111B ; displaying 7
MOV DX, PORTB
OUT DX, AL
CALL XS

MOV AL, 01111111B ; displaying 8
MOV DX, PORTB
OUT DX, AL
CALL XS

MOV AL, 01101111B ; displaying 9
MOV DX, PORTB
OUT DX, AL
CALL XS

JMP XA
```

```

108 XS PROC
109
110 MOV AL, 00111111B ; displaying 0
111 MOV DX, PORTA
112 OUT DX, AL
113 MOV CX, 0DF36H; Delay
114 Delay0: loop Delay0
115
116 MOV AL, 00000110B ; displaying 1
117 MOV DX, PORTA
118 OUT DX, AL
119 MOV CX, 0DF36H; Delay
120 Delay1: loop Delay1
121
122 MOV AL, 01011011B ; displaying 2
123 MOV DX, PORTA
124 OUT DX, AL
125 MOV CX, 0DF36H; Delay
126 Delay2: loop Delay2
127
128 MOV AL, 01001111B ; displaying 3
129 MOV DX, PORTA
130 OUT DX, AL
131 MOV CX, 0DF36H; Delay
132 Delay3: loop Delay3
133
134 MOV AL, 01100110B ; displaying 4
135 MOV DX, PORTA
136 OUT DX, AL
137 MOV CX, 0DF36H; Delay
138 Delay4: loop Delay4
139
140 MOV AL, 01101101B ; displaying 5
141 MOV DX, PORTA
142 OUT DX, AL
143 MOV CX, 0DF36H; Delay
144 Delay5: loop Delay5
145
146 MOV AL, 01111101B ; displaying 6
147 MOV DX, PORTA
148 OUT DX, AL
149 MOV CX, 0DF36H; Delay
150 Delay6: loop Delay6
151
152 MOV AL, 00000111B ; displaying 7
153 MOV DX, PORTA
154 OUT DX, AL
155 MOV CX, 0DF36H; Delay
156 Delay7: loop Delay7
157
158 MOV AL, 01111111B ; displaying 8
159 MOV DX, PORTA
160 OUT DX, AL
161 MOV CX, 0DF36H; Delay
162 Delay8: loop Delay8
163
164 MOV AL, 01101111B ; displaying 9
165 MOV DX, PORTA
166 OUT DX, AL
167 MOV CX, 0DF36H; Delay
168 Delay9: loop Delay9

```

170
171
172
173
174
175
176

The diagram illustrates a digital circuit using a 74HC373 8-bit D-type flip-flop (U2) interfaced with an 8255A PPI (U3) and two 7-segment displays. The 74HC373 is configured with its Q0-Q7 outputs connected to the data bus of the 8255A. The 8255A's PA0-PA7 and PB0-PB7 ports are connected to the two 7-segment displays. The 8255A's RESET pin is connected to the 74HC373's RD pin. The 8255A's CS pin is connected to ground. The 74HC373's AD0-AD7 inputs are connected to the 8255A's A0-A7 inputs. The 74HC373's OE and LE pins are connected to ground. The 74HC373's Q0-Q7 outputs are connected to the 8255A's PA0-PA7 and PB0-PB7 ports. The 8255A's PA0-PA7 and PB0-PB7 ports are connected to the two 7-segment displays. The 8255A's RESET pin is connected to the 74HC373's RD pin. The 8255A's CS pin is connected to ground. The 74HC373's AD0-AD7 inputs are connected to the 8255A's A0-A7 inputs. The 74HC373's OE and LE pins are connected to ground. The 74HC373's Q0-Q7 outputs are connected to the 8255A's PA0-PA7 and PB0-PB7 ports. The 8255A's PA0-PA7 and PB0-PB7 ports are connected to the two 7-segment displays.

3.2 Analisa Praktikum

1. Dalam percobaan pertama, output yang dihasilkan seperti keterangan gambar hasil run. Hal tersebut mungkin terjadi karena kode assembly yang sudah dibuat sebelumnya menentukan output pada address masing masing segment yang ditunjukkan dengan angka biner lalu address tersebut diatur pada port A, sehingga menunjukkan angka berurutan 0-9 serta diberi jeda output masing-masing angka sebesar 1 detik.
2. Dalam percobaan kedua, output yang dihasilkan adalah angka sebelas karena NRP saya adalah 2421600011. Output tersebut mungkin karena kode yang saya buat memerintahkan untuk menampilkan output angka 1 di kedua port dengan mengirim address segment yang menampilkan angka 1 ke kedua port A dan B. Lalu saya ulangi proses tersebut dengan JMP supaya angka 11 tetap menyala.
3. Dalam percobaan tugas, output yang dihasilkan seperti pada keterangan gambar hasil run. Hal tersebut dimungkinkan karena saya menggunakan fitur assembly yang bernama procedure, yang dalam bahasa pemrograman biasa disebut “fungsi”. Saya memanggil procedure pada akhir setiap code yang menyalakan segment bernilai puluhan. Sehingga ketika setiap segment puluhan masing-masing mulai menyala dari 0, saya panggil procedure untuk menjalankan segment yang menunjukkan satuan 0-9 dengan jeda 1 detik.

3.3 Kesimpulan

Kesimpulan yang bisa diambil dari praktikum kali ini:

1. Saya belajar untuk menyalakan seven-segment, dan saya dapat mengetahui masing-masing address binary untuk setiap angka pada segment.
2. Saya belajar memecahkan masalah baru dan mendapatkan solusi dengan menggunakan suatu fitur assembly yang belum saya ketahui, sehingga saya juga belajar hal baru dalam bahasa assembly.
3. Saya belajar untuk menggunakan procedure dalam assembly.