

# BAB 7

---

## PEMROGRAMAN MIKROPROSESOR KASUS PADA MPF-1

Kompetensi memprogram sebuah mikroprosesor sangat ditentukan oleh penguasaan set instruksi, arsitektur internal khususnya susunan dan jenis register, arsitektur sistem mikroprosesor dan juga jam terbang penggunaan mikrokomputer. Untuk lebih meningkatkan kemampuan penguasaan pemrograman dalam bahasa assembly berikut disajikan kasus-kasus program dan penyelesaiannya. Dimulai dari penjabaran arsitektur rangkaian mikrokomputer, program subrutin, dan pemrograman mikrokomputer MPF-1.  
Kata kunci: program, mikroprosesor

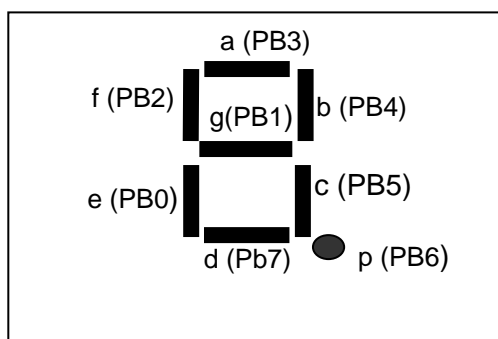
### 1. Monitor dan Keyboard MPF-1

Mikrokomputer MPF-1 merupakan salah satu jenis mikrokomputer lama yang paling banyak digunakan untuk belajar memahami sistem mikroprosesor. Mikrokomputer jenis MPF-1 menggunakan mikroprosesor Z-80 CPU buatan Zilog. Dilihat dari kapasitas yang terpakai MPF-1 termasuk kategori sistem minimum. Sistem mikrokomputer MPF-1 sangat membantu digunakan dalam pembelajaran sistem mikroprosesor. MPF-1 dilengkapi dengan monitor seven segment, keyboard, speaker, memori yang cukup dan I/O untuk keperluan interfacing.

Monitor MPF-1 menggunakan 6 buah seven segment dan keyboard menggunakan konfigurasi matrik 6x6 diatas sebuah peripheral PPI 8255. Bentuk konfigurasi rangkaian monitor dan keyboard MPF-1 ditunjukkan pada Gambar 7.2.

Dari Gambar 7.2. terlihat dengan jelas MPF-1 menggunakan komponen utama I/O berupa PPI 8255 yang dapat diprogram, enam saluran pada Port A (PA5 s/d PA0) digunakan sebagai input untuk matrik keyboard, delapan buah saluran Port B (PB7 s/d PB0) digunakan untuk saluran pengendalian segment dan enam buah saluran Port C (PC5 s/d PC0) digunakan untuk pemilihan LED. Monitor MPF-1 menggunakan LED seven segment jenis common cathoda. Monitor seven segment merupakan konfigurasi LED yang paling sederhana untuk membentuk karakter. Seven segment cukup baik dipakai sebagai pembentuk karakter. Elemen seven segment yang terbatas hanya tujuh membuat pembentukan karakter tidak bisa sempurna untuk semua jenis karakter. Karakter K, R sebagai contoh tidak bisa dibentuk dengan baik dalam seven segment.

Konfigurasi LED seven segment tersusun seperti Gambar 7.1. Ada delapan LED, tujuh LED digunakan untuk pembentuk karakter dan satu LED untuk segment dot. Tujuh segment dari seven segment diberi kode segment a, b, c, d, e, f, dan g. Rancangan rangkaian seven segment sebagai monitor pada mikrokomputer MPF-1 dikonfigurasi dengan PPI 8255 melalui port B dengan susunan seperti Gambar 7.2. Anoda LED masing-masing segment dihubungkan ke port B.



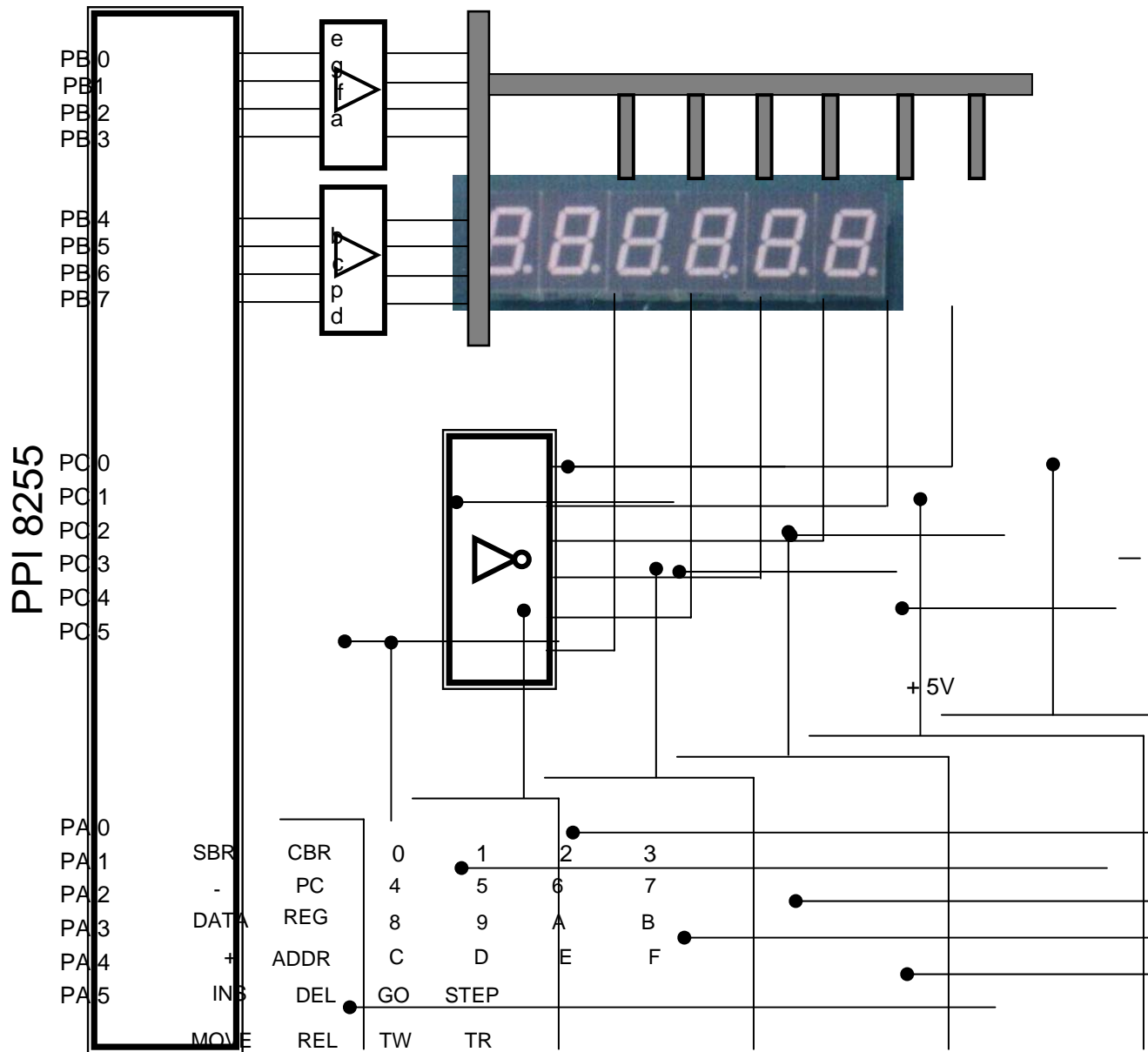
Gambar 7.1. Konfigurasi LED Seven segment terhadap Port B.

Dari Gambar 7.1. dan Gambar 7.2. dapat dipetakan hubungan atau konfigurasi segment sebagai berikut:

- Segment a terhubung dengan PB3
- Segment b terhubung dengan PB4
- Segment c terhubung dengan PB5
- Segment d terhubung dengan PB7
- Segment e terhubung dengan PB0
- Segment f terhubung dengan PB2
- Segment g terhubung dengan PB1
- Segment p terhubung dengan PB6

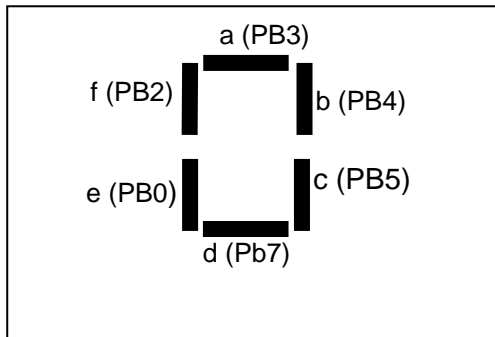
Konfigurasi Gambar 7.1. dan Gambar 7.2. penting diperhatikan sebagai dasar pengembangan data-data karakter yang diperlukan pada saat pemrograman. Konfigurasi ini juga sangat penting sebagai bahan dalam pengembangan antar muka mikroprosesor. Hubungan antara segment dengan port B tidak selalu harus seperti Gambar 7.1. Bisa saja dalam rangkaian lain susunan hubungan port berbeda. Yang penting menjadi perhatian adalah konsistensi hubungan port B dengan segment karena akan membentuk data karakter yang berbeda.

Jenis LED seven segment juga akan berpengaruh terhadap pembentukan karakter. Ada dua jenis LED seven segment yaitu common anoda dan common cathoda. Diantara dua jenis LED seven segment ini akan memberi data karakter berbeda. Pada LED common cathoda segmen akan berlaku aktif menyala jika anoda segment berlogika 1 dan common LED seven segment berlogika 0. Untuk keadaan lain LED seven segment akan padam. Sedangkan untuk LED seven segment common anoda LED akan aktif jika segment cathoda berlogika 0 dan common LED seven segment berlogika 1. Keadaan ini akan menentukan data-data karakter yang dibentuk dalam setiap penyalaan monitor.



Gambar 7.2. Konfigurasi rangkaian monitor – keyboard MPF-1

Dari Gambar 7.1 dan Gambar 7.2 dapat ditetapkan bahwa LED akan menyala jika anoda segment berlogika 1 dan katoda berlogika 0. Untuk keadaan sebaliknya segment LED akan padam. Kondisi ini dapat digunakan sebagai dasar pembentukan data untuk masing-masing karakter. Untuk membangun data penyalan LED diperlukan data nyala sesuai dengan konfigurasi Port. Sebagai contoh untuk menyalakan karakter angka 0 pada LED seven segment konfigurasinya seperti Gambar 7.3.



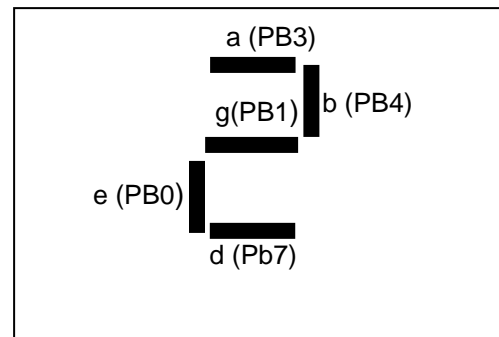
Gambar 7.3. Konfigurasi karakter 0 pada seven segment

Dari Gambar 7.3. terlihat segment yang aktif adalah segment a, b, c, d, e, dan f. Segment g dan p ada dalam posisi off. Dengan demikian data penyalan dapat dinyatakan sebagai berikut:

- ✦ PB7 (segment d) berlogika 1 : on
- ✦ PB6 (segment p) berlogika 0 : off
- ✦ PB5 (segment c) berlogika 1 : on
- ✦ PB4 (segment b) berlogika 1 : on
- ✦ PB3 (segment a) berlogika 1 : on
- ✦ PB2 (segment f) berlogika 1 : on
- ✦ PB1 (segment g) berlogika 0 : off
- ✦ PB0 (segment e) berlogika 1 : on

Kemudian akan membentuk data penyalan karakter angka 0 yang jika disusun datanya adalah 1011 1101b atau sama dengan **BDH**.

Untuk kasus yang lain misalnya jika membentuk karakter angka dua maka konfigurasi LED seven segment dapat digambarkan seperti Gambar 7.4.



Gambar 7.4. Konfigurasi karakter angka 2 pada seven segment

Dari Gambar 7.4. terlihat segment yang aktif adalah segment a, b, d, e, dan g. Segment b dan f ada dalam posisi off. Dengan demikian data penyalan dapat dinyatakan sebagai berikut:

- ✦ PB7 (segment d) berlogika 1 : on
- ✦ PB6 (segment p) berlogika 0 : off
- ✦ PB5 (segment c) berlogika 0 : on
- ✦ PB4 (segment b) berlogika 1 : on
- ✦ PB3 (segment a) berlogika 1 : on
- ✦ PB2 (segment f) berlogika 0 : on
- ✦ PB1 (segment g) berlogika 1 : off
- ✦ PB0 (segment e) berlogika 1 : on

Kemudian akan membentuk data penyalan karakter angka 2 yaitu 1001 1011b atau sama dengan **9BH**.

Seperti disajikan dalam Bab 1 bahwa sistim mikroprosesor dilengkapi dengan peralatan input dan peralatan output. Peralatan input yang paling sederhana adalah key pad matrik dan peralatan monitor dalam bentuk seven segment. Untuk menyajikan atau men-display-kan sebuah atau sejumlah karakter di monitor maka data penyalaan karakter disusun sesuai konfigurasi rangkaian seperti dijelaskan pada Gambar 7.3. dan Gambar 7.4. Dengan konfigurasi rangkaian interface seperti Gambar 7.1 dan 7.2 maka data penyalaan karakter dapat dijabarkan seperti contoh-contoh diatas. Penjabaran data karakter sangat penting karena akan digunakan sebagai data penyalaan pada saat pemrograman. Untuk memudahkan penjabaran data penyalaan segment untuk tiap karakter maka formula Tabel 7.1. dapat digunakan sebagai model analisis.

Tabel 7.1. Data penyalaan karakter seven segment

PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	DATA	NYALA
<b>d</b>	<b>p</b>	<b>c</b>	<b>b</b>	<b>a</b>	<b>f</b>	<b>g</b>	<b>e</b>		
0	0	0	0	0	0	0	0	00H	padam
1	0	1	1	1	1	0	1	BDH	0
0	0	1	1	0	0	0	0	30H	1
1	0	0	1	1	0	1	1	9BH	2
1	0	1	1	1	0	1	0	BAH	3
0	0	1	1	0	1	1	0	36H	4
1	0	1	0	1	1	1	0	AEH	5
1	0	1	0	1	1	1	1	AFH	6
0	0	1	1	1	0	0	0	38H	7

PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	DATA	NYALA
<b>d</b>	<b>p</b>	<b>c</b>	<b>b</b>	<b>a</b>	<b>f</b>	<b>g</b>	<b>e</b>		
1	0	1	1	1	1	1	1	BFH	8
0	0	1	1	1	1	1	0	BEH	9
0	0	1	1	1	1	1	1	3FH	A
1	0	1	0	0	1	1	1	A7H	B
1	0	0	0	1	1	0	1	8DH	C
1	0	1	1	0	0	1	1	B3H	D
1	0	0	0	1	1	1	1	8FH	E
0	0	0	0	1	1	1	1	0FH	F
1	0	1	0	1	1	0	1	ADH	G
0	0	1	1	0	1	1	1	37H	H
1	0	0	0	1	0	0	1	89H	I
1	0	1	1	0	0	0	1	B1H	J
1	0	0	1	0	1	1	1	97H	K
1	0	0	0	0	1	0	1	85H	L
0	0	1	0	1	0	1	1	2BH	M
0	0	1	0	0	0	1	1	23H	N
1	0	1	0	0	0	1	1	A3H	O
0	0	0	1	1	1	1	1	1FH	P
0	0	1	1	1	1	1	0	3EH	Q
0	0	0	0	0	0	1	1	03H	R
1	0	1	0	0	1	1	0	A6H	S
1	0	0	0	0	1	1	1	87H	T
1	0	1	1	0	1	0	1	B5H	U
1	0	1	1	0	1	1	1	B7H	V
1	0	1	0	1	0	0	1	A9H	W
0	0	0	0	0	1	1	1	07H	X
1	0	1	1	0	1	1	0	B6H	Y
1	0	0	0	1	0	1	0	8AH	Z
1	0	0	0	0	0	1	1	83H	(
1	0	1	0	0	0	1	0	A2H	)
0	0	1	1	0	0	1	0	32H	+
0	0	0	0	0	0	1	0	02H	-

Tabel 7.1. menunjukkan data-data penyalaaan untuk tiap-tiap karakter. Berdasarkan Tabel 7.1. dapat dijelaskan pertama hubungan antara port B dengan segment. Berdasarkan konfigurasi rangkaian PB0 dihubungkan dengan segment e, PB1 dihubungkan dengan segment g, PB2 dihubungkan dengan segment f, PB3 dihubungkan dengan segment a, PB4 dihubungkan dengan segment b, PB5 dihubungkan dengan segment c, PB6 dihubungkan dengan segment p, PB7 dihubungkan dengan segment d. Konfigurasi ini membentuk struktur seperti Tabel 7.1. Dengan konfigurasi aktif high atau logika 1 maka untuk setiap penyalaaan segment dalam pembentukan karakter data binernya adalah 1 dan untuk mematikan segment data binernya adalah 0. Dengan cara ini sebagaimana diuraikan dalam dua contoh penyalaaan angka 0 dan angka 2 seluruh karakter dapat diuraikan data penyalaaannya seperti tabel 7.1.

Dalam monitor seven segment MPF-1 ada enam LED yang terpasang secara paralel. Masing-masing segment terkoneksi mejadi satu. Segment a untuk LED1, LED2, LED3, LED4, LED5, dan LED6 terhubung menjadi satu ke PB3. Segment b untuk LED1, LED2, LED3, LED4, LED5, dan LED6 terhubung menjadi satu ke PB4. Segment e untuk LED1, LED2, LED3, LED4, LED5, dan LED 6 terhubung menjadi satu ke PB5. Segment d untuk LED1, LED2, LED3, LED4,

LED5, dan LED 6 terhubung menjadi satu ke PB7. Segment e untuk LED1, LED2, LED3, LED4, LED5, dan LED6 terhubung menjadi satu ke PB0. Segment f untuk LED1, LED2, LED3, LED4, LED5, dan LED6 terhubung menjadi satu ke PB2. Segment g untuk LED1, LED2, LED3, LED4, LED5, dan LED6 terhubung menjadi satu ke PB1. Dan segment p untuk LED1, LED2, LED3, LED4, LED5, dan LED6 terhubung menjadi satu ke PB6. Hubungan paralel seperti ini menyebabkan segment untuk semua LED aktif bersamaan. Masalah selanjutnya adalah bagaimanakah cara melakukan pemilihan LED yang aktif diantara 6 LED.

Untuk memilih LED yang mana untuk aktif dari 6 buah LED seven segment apakah LED1, LED2, LED3, LED4, LED5 atau LED6 atau seluruhnya caranya adalah dengan memilih aktif atau tidak aktif common cathoda dari masing-masing LED seven segment. Jika common cathoda LED berlogika 0 maka LED akan aktif menyala. Sebaliknya jika common cathoda LED berlogika 1 maka LED akan mati.

Kembali ke Gambar 7.2. konfigurasi rangkaian common cathoda LED dihubungkan melalui inverter atau NOT gate terhadap port C PPI 8255. Konfigurasi ini menyebabkan adanya pembalikan kondisi logika antara common cathoda dan port C. Artinya untuk menyalakan LED diperlukan data 1 pada port C.

Pada Gambar 7.2. konfigurasi rangkaian port C adalah sebagai berikut:

- ✦ PC0 digunakan untuk mengendalikan LED6 (LED paling kanan)
- ✦ PC1 digunakan untuk mengendalikan LED5
- ✦ PC2 digunakan untuk mengendalikan LED4
- ✦ PC3 digunakan untuk mengendalikan LED3
- ✦ PC4 digunakan untuk mengendalikan LED2
- ✦ PC5 digunakan untuk mengendalikan LED1 (led paling kiri)
- ✦ PC6 tidak terpasang
- ✦ PC7 tidak terpasang

Dengan konfigurasi Port C seperti itu dimana sebelum masuk ke common cathoda LED masing-masing dilewatkan melalui sebuah buffer inverter maka port pemilih LED berkeadaan status aktif tinggi atau logika 1. Dengan demikian dapat disusun data pemilihan LED seperti Tabel 7.2.

Tabel 7.2. Data pemilihan LED

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	DATA	NYALA
B7	B6	B5	B4	B3	B2	B1	B0		
1	1	0	0	0	0	0	1	C1H	LED 6
1	1	0	0	0	0	1	0	C2H	LED 5
1	1	0	0	0	1	0	0	C4H	LED 4
1	1	0	0	1	0	0	0	C8H	LED 3
1	1	0	1	0	0	0	0	D0H	LED 2
1	1	1	0	0	0	0	0	E0H	LED 1
1	1	1	1	1	1	1	1	FFH	SEMUA LED

Dari Tabel 7.2. terlihat untuk mengaktifkan LED6 yang paling kanan diperlukan data C1H, untuk mengaktifkan LED5 diperlukan data C2H, untuk mengaktifkan LED4 diperlukan data C4H, untuk mengaktifkan LED3 diperlukan data C8H, untuk mengaktifkan LED2 diperlukan data D0H, dan untuk mengaktifkan LED1 diperlukan data D0H. Untuk mengaktifkan semua LED diperlukan data FFH.

Kedua data yang telah diturunkan yaitu data pembentuk karakter dan data pemilih LED dapat digunakan sebagai data pengendalian monitor pada MPF-1. Data ini merupakan data dasar penyalan yang berhubungan dengan jenis karakter dan posisi LED yang akan dinyalakan. Berikut ini disajikan beberapa contoh program untuk memudahkan memahami dan menganalisa cara pemrograman penyalan karakter di monitor.

#### Contoh 1:

Penyalan LED1 dengan karakter angka 2. LED1 pada kasus ini adalah LED yang paling kiri. Kondisi yang diharapkan adalah:

- ✦ LED 1 paling kiri menyala angka 2
- ✦ LED 2 padam
- ✦ LED 3 padam
- ✦ LED 4 padam
- ✦ LED 5 padam
- ✦ LED 6 padam

Untuk itu diperlukan dua data yaitu data penyalan angka 2 dan data pengaktifan

LED1 atau LED paling kiri dan LED lain padam. Kedua data itu adalah:

- Data penyalan angka 2 adalah 9BH
- Data pemilihan LED1 paling kiri adalah E0H

Untuk mewujudkan keperluan penyalan tersebut dapat dibuat program dengan algoritma sebagai berikut:

1. Isi register A dengan data angka 2 (9B)
2. Keluarkan data pada register A ke Port B
3. Isi register A dengan data pemilih LED1 (E0)
4. Keluarkan data pada register A ke Port C

Bentuk program penyalan angka 2 pada LED paling kiri adalah seperti Gambar 7.5.

Addr	Op-Code	No.	Label	Assembly	Keterangan
1800	3E 9B	1.		LD A, 9BH	9BH= data nyala angka 2 dikeluarkan ke PORT B
1802	D3 01	2.		OUT PORTB, A	
1804	3E E0	3.		LD A, E0H	E0H= data pemilih LED1 dikeluarkan ke PORT C
1806	D3 02	4.		OUT PORTC, A	
1808	76	5.		HALT	Berhenti

Gambar 7.5. Program penyalan LED1 dengan karakter angka 2

Dari Gambar 7.5. terlihat pertama register A diisi data 9BH sebagai data penyalan angka 2 kemudian dikeluarkan ke Port B sebagai port pembentuk nyala karakter. Langkah selanjutnya register A diisi data E0H sebagai data pemilih LED 1

yang aktif dan dikeluarkan lewat port C. Dengan empat langkah ini LED1 yang paling kiri akan menyala angka 1 dan LED lainnya semua padam.

### Contoh 2:

Penyalan LED2 dengan karakter angka 4

- ✦ LED 1 padam (paling kiri) dan
- ✦ LED 2 menyala angka 4
- ✦ LED 3 padam
- ✦ LED 4 padam
- ✦ LED 5 padam
- ✦ LED 6 padam

Data penyalan dan pemilihan LED karakter angka 4 pada LED2 adalah:

- Data penyalan angka 4 adalah 36H
- Data pemilihan LED2 adalah D0H

Program penyalan LED2 dengan karakter angka 4.

Addr	Op-Code	No	Label	Assembly	Keterangan
1800	3E 36	1		LD A, 36H	36H = data nyala angka 4 dikeluarkan ke PORT B
1802	D3 01	2		OUT PORTB, A	
1804	3E D0	3		LD A, D0H	D0H = data pemilih LED2 dikeluarkan ke PORT C
1806	D3 02	4		OUT PORTC, A	
1808	76	5		HALT	Berhenti

Gambar 7.6. Program penyalan LED2 dengan karakter angka 4



**Contoh 3:**

Penyalan LED3 dengan karakter angka 1.

- ✦ LED 1 padam (paling kiri) dan
- ✦ LED 2 padam
- ✦ LED 3 menyala angka 1
- ✦ LED 4 padam
- ✦ LED 5 padam
- ✦ LED 6 padam

Data penyalan dan pemilihan LED adalah:

- Data penyalan angka 1 adalah 30H
- Data pemilihan LED3 adalah C8H

Program penyalan LED 3 dengan karakter angka 1.

Addr	Op-Code	No	Label	Assembly	Keterangan
1800	3E 30	1		LD A, 30H	30H= data nyala angka 1
1802	D3 01	2		OUT PORTB, A	dikeluarkan ke PORT B
1804	3E C8	3		LD A, C8H	C8H = data pemilih LED 3
1806	D3 02	4		OUT PORTC, A	dikeluarkan ke PORT C
1808	76	5		HALT	Berhenti

Gambar 7.7. Program penyalan LED3 dengan karakter angka 1

**Contoh 4:**

Penyalan LED4 dengan karakter angka 9

- ✦ LED 1 padam (paling kiri) dan
- ✦ LED 2 padam
- ✦ LED 3 padam
- ✦ LED 4 menyala angka 9
- ✦ LED 5 padam
- ✦ LED 6 padam

Data penyalan dan pemilihan LED adalah:

- Data penyalan angka 9 adalah BEH
- Data pemilihan LED4 adalah C4H

Program penyalan LED4 dengan karakter angka 9

Addr	Op-Code	No	Label	Assembly	Keterangan
1800	3E BE	1		LD A, BEH	BEH = data nyala angka 9
1802	D3 01	2		OUT PORTB, A	dikeluarkan ke PORT B
1804	3E C4	3		LD A, C4H	E0H = data pemilih LED4
1806	D3 02	4		OUT PORTC, A	dikeluarkan ke PORT C
1808	76	5		HALT	Berhenti

Gambar 7.8. Program penyalan LED4 dengan karakter angka 9

**Contoh 5:**

Penyalan LED5 dengan karakter angka 6

- ✦ LED 1 padam (paling kiri) dan
- ✦ LED 2 padam
- ✦ LED 3 padam
- ✦ LED 4 padam
- ✦ LED 5 menyala angka 6
- ✦ LED 6 padam

Data penyalan dan pemilihan LED adalah:

- Data penyalan angka 6 adalah AFH
- Data pemilihan LED5 adalah C2H

Program penyalaaan LED5 dengan karakter angka 6

Addr	Op-Code	No	Label	Assembly	Keterangan
1800	3E AF	1		LD A, AFH	AFH = data nyala angka 6
1802	D3 01	2		OUT PORTB, A	dikeluarkan ke PORT B
1804	3E C2	3		LD A, C2H	C2H = data pemilih LED 5
1806	D3 02	4		OUT PORTC, A	dikeluarkan ke PORT C
1808	76	5		HALT	Berhenti

Gambar 7.9. Program penyalaaan LED5 dengan karakter angka 6

#### Contoh 6:

Penyalaaan LED6 dengan karakter angka 4.

- ✦ LED 1 padam (paling kiri) dan
- ✦ LED 2 padam
- ✦ LED 3 padam
- ✦ LED 4 padam
- ✦ LED 5 padam
- ✦ LED 6 menyala angka 4

Maka data penyalaaan dan pemilihan LED adalah:

- Data penyalaaan angka 4 adalah 36H
- Data pemilihan LED6 adalah C1H

Program penyalaaan LED6 dengan karakter angka 4

Addr	Op-Code	No	Label	Assembly	Keterangan
1800	3E 36	1		LD A, 36H	36H = data nyala angka 4
1802	D3 01	2		OUT PORTB, A	dikeluarkan ke PORT B
1804	3E C1	3		LD A, C1H	C1H = data pemilih LED 6
1806	D3 02	4		OUT PORTC, A	dikeluarkan ke PORT C
1808	76	5		HALT	Berhenti

Gambar 7.10. Program penyalaaan LED6 dengan karakter angka 4

#### Contoh 7:

Penyalaaan LED1, LED2, LED3, LED4, LED5, LED6 dengan karakter P

- ✦ LED 1 menyala angka P
- ✦ LED 2 menyala angka P
- ✦ LED 3 menyala angka P
- ✦ LED 4 menyala angka P
- ✦ LED 5 menyala angka P
- ✦ LED 6 menyala angka P

Maka data penyalaaan dan pemilihan LED adalah:

- Data penyalaaan huruf P adalah 1FH
- Data pemilihan LED6 adalah C1H

Program penyalaaan semua LED dengan karakter huruf P

Addr	Op-Code	No Urut	Label	Assembly	Keterangan
1800	3E 1F	1		LD A, 1FH	1FH = data nyala huruf P dikeluarkan ke PORT B
1802	D3 01	2		OUT PORTB, A	
1804	3E FF	3		LD A, FFH	FFH = data pemilih semua LED1 dikeluarkan ke PORT C
1806	D3 02	4		OUT PORTC, A	
1808	76	5		HALT	Berhenti

Gambar 7.11. Program penyalaaan LED1, LED2, LED3, LED4, LED5, LED6 dengan karakter huruf P

Tujuh contoh program di atas memberi suatu model penyalaaan statis yaitu suatu bentuk penyalaaan dimana dalam satu waktu hanya ada satu jenis karakter yang dapat disajikan. Hal ini terjadi karena masing-masing segment pada setiap LED terhubung menjadi satu yaitu pada Port B.

Agar penyalaaan LED dapat menyajikan karakter yang berbeda, maka tiap LED harus dinyalakan secara bergantian dengan perioda nyala tertentu. Jika perioda nyala LED cukup lama maka LED akan nampak nyala bergantian. Apabila perioda nyala LED dipercepat dengan

frekuensi yang cukup maka LED seakan-akan nampak menyala bersama-sama namun sesungguhnya nyala bergantian.

Cara penyalaaan LED bergantian dengan perioda yang tepat agar nyala LED cukup terang disebut penyalaaan dinamik dan penyalaaan tersebut disebut penyalaaan secara multiplek. Pola penyalaaan multiplek dapat dijelaskan menggunakan algoritma sebagai berikut :

- LED6 (paling kanan) dinyalakan selama perioda tertentu, sedangkan LED lainnya tetap padam selama perioda itu.
- Kemudian LED5 dinyalakan selama perioda tertentu sedangkan LED lainnya tetap padam pada perioda itu.
- Selanjutnya LED4 dinyalakan selama perioda tertentu sedangkan LED lainnya tetap padam pada perioda itu.
- Selanjutnya LED3 dinyalakan selama perioda tertentu sedangkan LED lainnya tetap padam pada perioda itu.
- Selanjutnya LED2 dinyalakan selama perioda tertentu sedangkan LED lainnya tetap padam pada perioda itu.
- Selanjutnya LED1 dinyalakan selama perioda tertentu sedangkan LED lainnya tetap padam pada perioda itu.

Bentuk program penyalaaan multiplek adalah seperti Gambar 7.12.

Addr	Op-Code	No Urut	Label	Assembly	Keterangan
		1	Contoh Program		
		2	Penyalaaan LED		
		3	PORT B	EQU 01H	
		4	PORT C	EQU 02H	
		5		ORG 1800	
1800	3E AF	6	LOOP	LD A, AFH	LED 6 dinyalakan dengan Karakter 6, yang lain Padam
1802	D3 01	7		OUT PORTB, A	
1804	3E C1	8		LD A, C1H	
1806	D3 02	9		OUT PORT C,A	
1808	CD 50 18	10		CALL PERIODA	Tunda selama 1 periode
		11			
180B	3E AE	12		LD A, AEH	LED 5 dinyalakan dengan Karakter 5, yang lain Padam
180D	D3 01	13		OUT PORTB, A	
180F	3E C2	14		LD A, C2H	
1811	D3 02	15		OUT PORT C,A	
1813	CD 50 18	16		CALL PERIODA	Tunda selama 1 periode
		17			
1816	3E 36	18		LD A, 36H	LED 4 dinyalakan dengan Karakter 4, yang lain Padam
1818	D3 01	19		OUT PORTB, A	
181A	3E C4	20		LD A, C4H	
181C	D3 02	21		OUT PORT C,A	
181E	CD 50 18	22		CALL PERIODA	Tunda selama 1 periode
		23			
1821	3E	24		LD A,	LED 3

Addr	Op-Code	No Urut	Label	Assembly	Keterangan	
	BA			BAH	dinyalakan dengan Karakter 3, yang lain Padam	
1823	D3 01	25		OUT PORTB, A		
1825	3E C8	26		LD A, C8H		
1827	D3 02	27		OUT PORT C,A		
1829	CD 50 18	28		CALL PERIODA	Tunda selama 1 periode	
		29				
182C	3E 9B	30		LD A, 9BH	LED 2 dinyalakan dengan Karakter 2, yang lain Padam	
182E	D3 01	31		OUT PORTB, A		
1830	3E D0	32		LD A, D0H		
1832	D3 02	33		OUT PORT C,A		
1834	CD 50 18	34		CALL PERIODA	Tunda selama 1 periode	
		35				
1837	3E 30	36		LD A, 30H	LED 1 dinyalakan dengan	
1839	D3 01	37		OUT PORTB, A	Karakter 1, yang lain	
183B	3E E0	38		LD A, E0H	Padam	
183D	D3 02	39		OUT PORT C,A		
183F	CD 50 18	40		CALL PERIODA	Tunda selama 1 periode	
		41				
1842	C3 00 18	42		JP LOOP	Ulangi mulai dari awal	
		43				
		44	Subrutin PERIODA			
		45		ORG 1850		
1850	F5	46	PERIOD	PUSH AF		

Addr	Op-Code	No Urut	Label	Assembly	Keterangan
			A		
1851	C5	47		PUSH BC	
1852	D5	48		PUSH DE	
		49			
1853	0E 10	50		LD C, 10H	Angka penentu periode
1855	11 10 10	51	LOOP2	LD DE, 1010H	
		52			
1858	1D	53	LOOP1	DEC E	
1859	C2 58 18	54		JP NZ, LOOP1	
		55			
185C	15	56		DEC D	
185D	C2 58 18	57		JP NZ, LOOP1	
		58			
1860	0D	59		DEC C	
1861	C2 55 18	60		JP NZ, LOOP2	
		61			
1864	D1	62		POP DE	
1865	C1	63		POP BC	
1866	F1	64		POP AF	
1867	C9	65		RET	

Gambar 7.12. Penyalaaan LED secara multiplek

Program Gambar 7.12. cukup panjang dan baik digunakan untuk menjelaskan proses penyalaaan LED secara multiplek karena step-step penyalaaannya terurai dengan jelas dan mudah. Akan tetapi nantinya program yang diperlukan adalah program yang pendek bukan program yang panjang.

Untuk mempercepat laju pergeseran penyalaaan LED, kecilkan isi register C baris 50 (paling kecil 01H). Kemudian kecilkan isi register E (paling kecil 01H) baris 51, selanjutnya kecilkan isi register D paling kecil 01H baris 51. Eksperimen ini akan menghasilkan penyalaaan multiplek berupa penyajian angka **1 2 3 4 5 6** menyala bersama secara multiplek.

Susunan program tersebut belum efisien karena disusun dalam konsep berfikir yang masih sederhana/bodoh. Untuk mendapatkan efisiensi yang lebih baik dapat dilaksanakan dengan memesan memori khusus yang digunakan untuk menyimpan data nyala angka dan huruf, sehingga perubahan nyala tinggal mengubah data di memori ini. Untuk menyingkat pengulangan instruksi sejenis dilakukan dengan operasi logik sebagai berikut:

- ✓ Mulai dari menyimpan data C1h ke Register E: LD E, C1H
- ✓ Data C1 = 1100 0001 ditransfer ke register A: LD A, E
- ✓ Dioperasikan AND dengan 3FH = 0011 1111h: AND 3FH
- ✓ Menghasilkan data A = 01H = 0000 0001
- ✓ Dioperasikan RLCA: RLCA
- ✓ Hasilnya 02H = 0000 0010
- ✓ Operasi OR dengan C0H: OR C0H
- ✓ Hasilnya A = C2H = 1100 0010

C2 adalah data pemilih LED nomor 5 yang akan diaktifkan. Selanjutnya jika diulang kembali prosesnya mulai dengan data C2H maka akan didapat nilai C4 yaitu data pemilihan LED ke 3. Demikian seterusnya sampai data pemilihan LED ke 1.

Pola operasi logik dan aritmetik tersebut dapat digunakan sebagai pola pengaturan program untuk penyalaan LED dinamis yang lebih baik dari program 1. Berikut program Gambar 7.13 yaitu program dengan arah yang sama dengan program Gambar 7.12.

Addr	Op-Code	No. Urut	Label	Assembly	Ket.
		1	Program 02		
		2	Menampilkan angka 1 2 3 4 5 6		
		3	Input :		
		4	Data nyala angka ditempatkan berurutan di memori yang alamatnya dicatat oleh Reg. IX . Data untuk LED paling kiri ditaruh pada alamat tertinggi dan data LED paling kanan di taruh pada alamat terendah		
		5	Port A Equ 00H, Port B Equ 01, Port C Equ 02		
		6			
		7			
		8			
		9		ORG 1800	
1800	DD 21 21 18	10	Saji	LD IX, BuffHur	IX digunakan pencatat Alamat buffer data
		11			
1804	1E C1	12		LD E, C1H	C1 = data awal
1806	26 06	13		LD H, 06H	Jumlah LED 6
		14			
1808	7B	15	Olah	LD A, E	Pengaktifan LED 1, 2, 3,
1809	D3 02	16		OUT (PORTC), A	LED 4, 5, 6
180B	DD 7E 00	17		LD A, (IX+00)	Dengan nyala multiplek
180E	D3 01	18		OUT (PORTB), A	

Addr	Op-Code	No. Urut	Label	Assembly	Ket.
1810	CD 27 18	19		CALL PERIODA	
		20			
1813	DD 23	21		INC IX	Pencatatan adres
1815	7B	22		LD A, E	
1816	E6 3F	23		AND , 3FH	
1818	07	24		RLCA	
1819	F6 C0	25		OR C0H	
181B	5F	26		LD E, A	
		27			
181C	25	28		DEC H	
181D	20 E9	29		JR NZ, Olah	
181F	18 DF	30		JR Saji	
		31			
1821	AF	32		DB AFH	Data nyala angka 6
1822	AE	33		DB AEH	Data nyala angka 5
1823	36	34		DB 36H	Data nyala angka 4
1824	BA	35		DB BAH	Data nyala angka 3
1825	9B	36		DB 9BH	Data nyala angka 2
1826	30	37		DB 30H	Data nyala angka 1
		38			
		39		Subrutin PERIODA	
1827	F5	40	PERIODA	PUSH AF	
1828	C5	41		PUSH BC	
1829	D5	42		PUSH DE	
182A	E5	43		PUSH HL	
		44			
182B	0E 10	45		LD C, 10H	
182D	11 05 05	46	LOOP2	LD DE, 0505H	
1830	1D	47	LOOP1	DEC E	
1831	C2 2F 18	48		JPNZ LOOP1	
1834	15	49		DEC D	
1835	C2 30 18	50		JPNZ, LOOP1	
1838	0D	51		DEC C	
1839	C2 2D 18	52		JPNZ, LOOP2	
		53			
183C	E1	54		POP HL	
183D	D1	55		POP DE	
183E	C1	56		POP BD	
183F	F1	57		POP AF	
1840	C9	58		RET	
				END	

Gambar 7.13. Penyalaan LED secara multiplek

Periode atau waktu tunda diatur menggunakan register C dan DE dimana isi register C sebagai MSB. Untuk memperkecil waktu tunda atau mempercepat animasi diperlukan pengaturan isi register C minimum = 01H, kemudian isi register DE dengan nilai terkecil 0001.

## 2. Key Board Encoder

Pengendalian dan pemrograman MPF-1 menggunakan tombol tekan dan tombol-tombol itu secara fisik tidak ada bedanya satu sama lain. (lihat Gambar 7.2). Jika tombol diketik atau ditekan maka kontak tombol tersambung, dan jika dilepas kontak tombol terputus.

Dengan menggunakan sebagian dari saluran PORT A sebagai input dan sebagian saluran PORT C sebagai saluran pembentuk matrik identifikasi maka dapat disusun program identifikasi khusus masing-masing tombol. Port A dan Port C membentuk matrik silang 6 x 6 sehingga setiap titik silang dapat ditempatkan satu tombol tekan. Jumlah tombol tekan yang dapat diimplementasikan adalah 36 tombol.

Setiap tombol diberi nomor kode yang disebut nomor Kode Posisi yang dapat digambarkan seperti Gambar 7.14.

	PC5	PC4	PC3	PC2	PC1	PC0
PA0	1E SBR	18 CBR	12 0	0C 1	06 2	00 3
PA1	1F -	19 PC	13 4	0D 5	07 6	01 7
PA2	20 DATA	1A REG	14 8	0E 9	08 A	02 B
PA3	21 +	1B ADDR	15 C	0F D	09 E	03 F
PA4	22 INS	1C DEL	16 GO	10 STEP	0A	04
PA5	23 MOVE	10 RELA	17 TPWR	11 TPRD	0B	05

Gambar 7.14. Kode Posisi Keyboard MPF-1

Nomor titik silang menggunakan angka heksadesimal, misalnya tombol angka 5 ditempatkan pada titik silang nomor 0DH, tombol REG pada titik silang 1A, dan seterusnya. Nomor titik silang tersebut digunakan dalam program keyboard scanning dijadikan kode posisi tombol. Caranya dengan selalu memberi logic 1 pada PORT PA0, PA1, PA2, PA3, PA4, dan PA5, serta secara bergilir diberi keadaan nol melalui Port C. Bit Carry flag mencatat adanya tombol yang ditekan dan register C mencatat nomor posisi titik silang yang dijadikan kode posisi titik silang.

SCAN pertama memberi keadaan nol pada saluran PC0 sedangkan saluran PC lainnya diberi keadaan satu. Register C diisi 00H, dan jika tombol pada nomor titik silang 00 ditekan maka carry flag dijadikan nol (direset) melalui saluran Port A. Sebaliknya jika tombol tidak ditekan maka bit carry flag

tetap 1, isi register C tidak dikutipkan ke register A, dan isi register A dan F tidak dikutipkan ke A' dan F'.

Dalam keadaan PC0 tetap nol, isi register C ditambah 1 dan scan kedua dilaksanakan. Bila tombol titik silang pada posisi 01 ditekan maka carry flag dijadikan nol. Sebaliknya jika tombol tidak ditekan maka bit carry flag tetap 1, isi register C tidak dikutipkan ke register A, dan isi register A dan F tidak dikutipkan ke A' dan F'. Jumlah scanning melalui PC0 ada 6 kali, dan masing-masing scan diselingi dengan penyajian tulisan pada monitor LED.

Enam scan berikutnya menggunakan keadaan nol melalui Port C1 dengan saluran PC lainnya diberi keadaan 1. Jika salah satu tombol yang terletak pada titik silang saluran PC1 ditekan maka carry flag akan menjadi nol.

Register C tetap menyimpan kelanjutan catatan nomor posisi titik silang dan nomor ini dijadikan kode posisi tombol yang diketik. Penyimpanan kode posisi tombol yang ditekan tetap dengan cara yang sama seperti scan pertama.

Scan tersebut bergilir berurutan mulai dari saluran PC0 berkeadaan nol dan saluran PC1, PC2, PC3, PC4, dan PC5 berkeadaan 1. Urutan kedua saluran PC1 diberi logika nol sedangkan saluran PC lainnya diberi 1. Demikian seterusnya hingga saluran PC5.

Pada MPF-I program yang menjalankan satu siklus Scanning di beri nama SCAN1. dan cuplikan naskah programnya adalah sebagai berikut.

```

SCF; bit carry flag dijadikan 1
EX AF, AF'; menggunakan alternatif register AF'
EXX ; Isi semua register disimpan di Alt Reg.
;-----
;Menyalakan Tulisan LED secara Multiplex
;-----
LD C, 00H      ; Kode posisi tombol mulai dari 00H
LD E, 1100 0001 ; mulai dari LED paling kanan
LD H, 06H      ; Jumlah LED ada 6
;
Kolom: LD A, (IX+00) ; A diisi data segment nyala LED
        OUT (PORTB), A ; Keluarkan ke PORTB
        LD A, E        ; Siapkan data pemilih LED
        OUT (PORTC), A ;
;

```



```

LD B, C9H      ; konstanta waktu tunda
Perioda:      DJNZ Perioda ; tunda waktu
;
XOR A
OUT (PORTB), A ; Segment dipadamkan
;-----
;Mulai melakukan Scanning tombol-tombol MPF-1
;-----
LD A, E
CPL
OR 1100 0000
OUT (PORTC), A ; saluran PC diberi nol, mulai dari PC0
LD B, 06H      ; jumlah cacahan setiap kolom ada 6
;
NoKey:        IN A, (PORTA) ; baca Port A untuk mengetahui titik
;             ; silang tombol ditekan
LD D,A        ; jika tombol ditekan bit carry flag menjadi
KoRow         RR D          ; kan nol , jika tidak ditekan Cy = 1
JR C, NoKey   ; dan isi AF tidak disimpan di AF'
;
LD A,C        ; Jika tombol ditekan kode posisi disimpan
EX AF,AF'    ; di register A, dan isi AF disimpan di AF'
INC C        ; catatan kode posisi ditambah 1
DJNZ KoRow   ; Ulangi 5 kali hingga scanning untuk Satu
;             ; saluran PC0 lengkap 6 kali
INC IX       ; IX mencatat lokasi data penyalaan
;             ; segment berikutnya
;-----
; Operasi Logik untuk mendapatkan Data Pemilih LED berikutnya
;-----
LD A,E
AND 0011 1111B
RLC A
DEC H        ; lanjutkan penyalaan LED berikutnya
JR NZ, Kolom ; hingga 6 LED mendapat giliran
;
LD DE, 0006  ;
ADD IX,DE    ; kembalikan isi IX semula
EXX         ; Kembalikan isi Register BC,DE, HL
EX AF,AF'   ; Kembalikan isi register AF
RET

```

Program SCAN1 dalam melakukan satu siklus scanning berlangsung selama 9,97 milli detik. SCAN1 bekerja menyajikan nyala 6 buah LED dan melakukan scanning 36 buah tombol MPF-I masing-masing satu kali dalam satu putaran.

Dalam menyajikan penyalaaan 6 buah LED diperlukan data. Data nyala segment LED (data angka dan data huruf) berjumlah 6 byte dan harus ditempatkan berurutan di memori dengan pengaturan Data LED segment yang paling kanan ditempatkan pada alamat memori terendah dan Data LED segment yang paling kiri ditempatkan pada alamat tertinggi. Alamat awal atau adres terendah dicatat menggunakan register 16 bit yaitu register IX. Pencatat alamat awal harus menggunakan register IX tidak bisa menggunakan register 16 bit lainnya. Program SCAN1 disusun dalam bentuk program SUBRUTIN tersimpan permanen di memori program monitor yaitu ROM. Subrutin SCAN1 dapat digunakan dengan menggunakan perintah CALL. Spesifikasi subrutin SCAN 1 adalah seperti disajikan dikolom samping berikut:

**Nama Subrutin: SCAN1**

**Alamat Awal : 0624H**

*Fungsi : Menyajikan tulisan LED secara multiplex dan melakukan scanning tombol keyboard dalam satu siklus, meliputi 36 buah tombol dan 6 buah LED seven segment*

*Jangka Waktu : 9,97 mili detik*

*Input Data : Data karakter penyalaaan segment LED berjumlah 6 byte disimpan berurutan pada memori Data Led segment yang paling kanan ditempatkan pada alamat terendah dan Data Led segment yang paling kiri ditempatkan pada alamat tertinggi.*

*Alamat awal tempat simpan data karakter harus dicatat oleh register IX.*

*Output Data : Jika tombol tidak ditekan , bit carry flag CY=1. Jika tombol ditekan CY=0 dan memberi kode posisi tombol pada pada register A.*

Untuk memahami kerja subrutin SCAN1 berikut disajikan contoh-contoh penggunaan subrutin SCAN1.

Contoh Program 7.14.

Addr	Op-Code	No. Urut	Label	Assembly	Keterangan
		1	Program 7.14 : Menyalakan Tulisan 123456 Selama 1 detik		
		2			
		3	SCAN1	EQU 0624H	
		4		ORG 1800	
1800	DD 21 0C 18	5		LD IX, ANGKA	
1804	06 64	6		LD B, 64H	64H=100D > Pengali
		7			
1806	CD 24 06	8	Nyala	CALL SCAN1	
1809	10 FB	9		DJNZ Nyala	
180B	76	10		HALT	
		11			
180C		12	ANGKA	DB AFH	Data angka 6
180D		13		DB AEH	Data angka 5
180E		14		DB 36H	Data angka 4
180F		15		DB BAH	Data angka 3
1810		16		DB 9BH	Data angka 2
1911		17		DB 30H	Data angka 1
		18		END	

Gambar 7.15. Program penyalakan angka 1,2,3,4,5,6 selama satu detik

Contoh Program 7.15.

Addr	Op-Code	No. Urut	Label	Assembly	Keterangan
		1	Program 7.15: Menyalakan Tulisan 123456 Berkedip tiap 1 detik		
		2			
		3	SCAN1	EQU 0624H	
		4		ORG 1800	
1800	21 13 18	5		LD HL, ANGKA	
1803	E5	6		PUSH HL	
1804	DD 21 19 18	7		LD IX, PADAM	
1808	DD E3	8	Kedip	EX (SP), IX	
180A	06 64	9		LD B, 64H	64H=100D > Pengali
		10			
180C	CD 24 06	11	Tampil	CALL SCAN1	
180F	10 FB	12		DJNZ Tampil	
1811	18 F5	13		JR Kedip	
		14			
1813	AF	15	ANGKA	DB AFH	Data angka 6
1814	AE	16		DB AEH	Data angka 5
1815	36	17		DB 36H	Data angka 4
1816	BA	18		DB BAH	Data angka 3
1817	9B	19		DB 9BH	Data angka 2
1818	30	20		DB 30H	Data angka 1
1819	00	21	PADAM	DB 00H	Data PADAM
181A	00	22		DB 00H	Data PADAM
181B	00	23		DB 00H	Data PADAM
181C	00	24		DB 00H	Data PADAM
181D	00	25		DB 00H	Data PADAM
181E	00	26		DB 00H	Data PADAM
		27		END	

Gambar 7.16. Program penyalakan angka 1,2,3,4,5,6 berkedip setiap satu detik

Keadaan tombol-tombol pada MPF-1 selalu harus diamati, misalnya karena beberapa permasalahan berikut:

- ✓ Ada tombol yang kontakannya selalu sambung dan tidak mau lepas. Jika hal ini terjadi maka harus ada bagian program yang melakukan penilikan dan memberi tanda peringatan.
- ✓ Keadaan tombol semuanya baik, tetapi pengguna menekan salah satu tombol terus menerus cukup lama dan setelah itu baru dilepaskan. Jika hal ini terjadi maka harus ada bagian program yang menunggu sehingga jika tombol dilepas maka hitungan menekannya hanya satu kali tekan.
- ✓ Keadaan tombol semuanya baik, tetapi pengguna tidak segera menekan tapi dibiarkan cukup lama. Jika hal ini terjadi maka harus ada bagian program yang menunggu terus menerus hingga tiba saatnya ada tombol yang ditekan.
- ✓ Untuk mencakup ketiga permasalahan di atas pada MPF-I disusun program subrutin yang disebut dengan **SCAN**. Sub rutin SCAN secara terus menerus menyajikan nyala tulisan di LED dan melakukan scanning tombol hingga ada salah satu tombol yang ditekan.
- ✓ Subrutin SCAN menggunakan surutin SCAN1. Yang membedakan subrutin SCAN dengan SCAN1 adalah SCAN tidak terbatas waktu sedangkan SCAN1 terbatas waktu. Disamping itu

SCAN menghasilkan Kode yang disebut dengan **KODE INTERNAL**. Spesifikasi SCAN adalah sebagai berikut :

**Nama Subrutin : SCAN**

**Alamat Awal : 05FEH**

*Fungsi* : Menyajikan tulisan LED secara multiplek dan melakukan scanning tombol keyboard dalam satu siklus, meliputi 36 buah tombol dan 6 buah LED seven segment

*Jangka Waktu* : Terus menerus menunggu hingga salah satu tombol ditekan

*Input Data* : Data karakter penyalan segment LED berjumlah 6 byte disimpan berurutan pada memori Data Led segment yang paling kanan ditempatkan pada alamat terendah dan Data Led segment yang paling kiri ditempatkan pada alamat tertinggi.

Alamat awal tempat simpan data karakter harus dicatat oleh register IX.

*Output Data* : Jika tombol tidak ditekan , bit carry flag CY=1. Jika tombol ditekan CY=0 dan memberi kode posisi tombol pada pada register A.

Kode internal adalah hasil ubahan dari kode posisi oleh Subrutin SCAN , yang menggunakan Subrutin SCAN1. Tabel hubungan tombol dengan kode internal adalah seperti Gambar 7.17:

	PC5	PC4	PC3	PC2	PC1	PC0
<b>PA0</b>	15 SBR	1A CBR	00 0	01 1	02 2	03 3
<b>PA1</b>	11 -	18 PC	04 4	05 5	06 6	07 7
<b>PA2</b>	14 DATA	1B REG	08 8	09 9	0A A	0B B
<b>PA3</b>	10 +	19 ADDR	0C C	0D D	0E E	0F F
<b>PA4</b>	16 INS	17 DEL	16 GO	10 STEP	0A	04
<b>PA5</b>	1C MOVE	1D RELA	17 TPWR	11 TPRD	0B	05

Gambar 7.17. Kode Internal Keyboard MPF-1

Kode internal dipilih untuk dijadikan selisih alamat (offset address) yang efektif untuk konversi dari data biner angka heksa desimal menjadi format nyala angka pada LED seven segment..

Selisih address tersebut dapat digunakan untuk membaca data konversi dari data biner angka heksadesimal ke format nyala angka LED. Hal ini akan dapat dipahami lebih jelas dari contoh-controh program berikut secara berkelanjutan. Contoh Penggunaan Subrutin SCAN adalah sebagai berikut.

Contoh Program 7.17.

Addr	Op-Code	No. Urut	Label	Assembly	Keterangan
		1			Program 7.17: Menyajikan Tulisan SCAN-- Penyajian berhenti jika tombol STEP ditekan
		2			
		3	SCAN	EQU 05FEH	
		4		ORG 1800	
1800	DD 21 0C 18	5		LD IX, TULISAN	
1804	CD FE 05	6	Nyala	CALL SCAN	
1807	FE 13	7		CP STEP	
1809	20 F9	8		JR NZ, Nyala	
180B	76	9		HALT	
180C	02	10	TULISAN	DB 02H	Data Tulisan -
180D	02	11		DB 02H	Data Tulisan -
180E	23	12		DB 23H	Data Tulisan N
180F	BB	13		DB BBH	Data Tulisan A
1810	83	14		DB 83H	Data Tulisan C
1811	AE	15		DB AEH	Data Tulisan S
		16		END	

Gambar 7.18. Program menyajikan tulisan SCAN

Program tersebut menyajikan tulisan SCAN—dan menunggu ada tombol ditekan. Jika tombol STEP yang ditekan dimana kode internal tombol tersebut adalah 13 maka pada proses CP 13 akan menghasilkan kondisi Z=1 yang berarti program akan kontinyu ke perintah HALT. Selain tombol STEP yang ditekan perbandingan CP 13 akan menghasilkan Z=0 dan program akan mencabang ke label "Tampil". Pprogram ini mulai mengenalkan proses pengendalian penggunaan keyboard.

Dua subrutin dapat digunakan untuk mengembangkan berbagai program misalnya bagaimanakah program bisa bekerja seperti pengetikan angka-angka pada Kalkulator. Angka pertama diketikkan mempunyai bobot tertinggi dan angka berikutnya mempunyai bobot berikutnya dengan penyajian bergeser ke kiri.

Kembali pada penjelasan subrutin SCAN, jika ada tombol yang ditekan, maka Register A akan berisi kode internal tombol tersebut, sementara bit carry flag  $Cy=0$ . Keadaan ini dapat digunakan sebagai dasar pengendalian keyboard.

Agar dapat melayani ketikan tombol 0 s/d 9 maka

- ✓ Tombol harus di saring, hanya tombol angka 0 sampai dengan tombol angka 9 saja yang diproses.
- ✓ Setiap kode internal tombol angka yang ditekan harus diubah menjadi data penyalaan angka LED yang sesuai, sehingga pada LED yang sesuai disajikan nyala angka yang benar.
- ✓ Proses tersebut memerlukan sejumlah memori tempat simpan untuk menaruh data nyal LED yang berasal dari ubahan kode internal. Jumlah memori tergantung dari jumlah angka yang diketikkan.

Perhatikan tabel kode internal , Tombol angka "0" memiliki kode interna "00". Demikian juga dengan tombol angka lainnya. Pada MPF-1 data nyala disimpan pada ROM mulai alamat 07F0 hingga 07FF. Address awal 07FF dapat langsung dicatat oleh register HL atau register lainnya. Dengan menggunakan data kode internal dari tombol yang ditekan sebagai offset maka data nyala LED pada ROM dapat diambil secara benar. Untuk lebih jelasnya analisis program 7.19 berikut:

Program 7.18.

Addr	Op-Code	No. Urut	Label	Asse mby	Keterangan
		1			Program 06 Menyajikan Tulisan Seperti Kalkulator, Ketikan hanya angka 0 s/d 9, Berhenti jika Tombol STEP ditekan
		2			
		3	SCAN	EQU 05FEH	
		4	BUFDISP	EQU 1A00	
		5		ORG 1800	
1800	DD 21 05 1A	6		LD IX, BUFDIS+5	Membersihkan memori pada alamat 1800 – 1805 sehingga LED pada awal padam
1804	06 06	7		LD B, 06H	
1806	DD 36 00 00	8	Hapus	LD (IX+00), 00H	
180A	DD 2B	9		DEC IX	
180C	10 F8	10		DJNZ Hapus	
		11			
180E	DD 21 00 1A	12		LD IX, BUFDIS	
1812	CD FE 05	13	Ketik	CALL SCAN	
1815	FE 0A	14		CP 0A	
1817	30 07	15		JR C, BCTabel	

Addr	Op-Code	No. Urut	Label	Asse mbly	Keterangan
		16			
1819	FE 13	17		CP STEP	
181B	CA 2D 18	18		JPZ, selesai	
181E	18 F2	19		JR Ketik	
		20			
1820	21 F0 07	21	BCTabel	LD HL, 07F0	
1823	85	22		ADD A, L	
1824	6F	23		LD L, A	
1825	7E	24		LD A, (HL)	
1826	DD 2B	25		DEC IX	
1828	DD 77 00	26		LD (IX+00), A	
182B	18 E5	27		JR Ketik	
182D	FF	28	selesai		

Gambar 7.19. Program menyajikan tulisan seperti kalkulator

Pada baris 13 ditunjukkan mulai ada proses pendeteksian tombol yang ditekan. Pada langkah 14 kode internal tombol disaring hanya angka 0 sampai dengan 9 yang diloloskan untuk diproses. Jika tombol yang ditekan adalah tombol angka 0 sampai dengan angka 9 maka program akan menyajikan nilai tombol di monitor. Jika tombol yang ditekan selain tombol angka 0 sampai angka 9 maka program mengabaikan dan tidak diproses. Jika tombol STEP yang ditekan maka program akan berhenti bekerja.

### 3. Subrutin Pelayanan pada MPF-1

Subrutin SCAN dan SCAN1 yang telah dijelaskan adalah subrutin pelayanan yang telah disediakan pada MPF-1. Disamping itu ada subrutin lainnya yaitu :

- ❖ HEX7 : untuk mengubah data 4 bit di register A menjadi data nyala angka LED
- ❖ HEX7SG : untuk mengubah data 8 bit di register A menjadi data nyala angka LED
- ❖ RAMCHK : men-cheek adanya RAM pada suatu Address dengan menguji sekaligus
- ❖ TONE : Suara nada dimana Frekuensi dan Lama susra dapat dipilih
- ❖ TONE1K : Suara nada pada Frekuensi 1 KHz
- ❖ TONE2K : Suara nada pada Frekuensi 2 KHz

#### 3.1. Spesifikasi Subrutin HEX7

*Nama Subrutin* : HEX7

*Alamat Awal* : 0689H

*Fungsi* : Mengubah data 4 bit (nibble rendah) di register A menjadi data nyala karakter pada LED.

*Input Data* : Data pada Register A (nibble rendah)

*Output Data* : Data nyala karakter untuk LED tercatat di register A

Cara menggunakan :

*LD A, n*

*CALL HEX7*

### 3.2. Spesifikasi Subrutin HEX7SG

Nama Subrutin : *HEX7SG*

Alamat Awal : *0678H*

Fungsi : *Mengubah data 8 bit di register A menjadi dua data nyala karakter pada LED.*

Input Data : *Data pada Register A*

Output Data : *Data nyala karakter untuk LED disimpan pada memori data nyala LED untuk nibble rendah pada alamat (HL) dan data nyala LED untuk nibble tinggi di (HL+1).*

*Setelah itu isi register  
HL = HL+2*

Cara menggunakan :

*LD HL, alamat awal  
tempat simpan*

*LD A, n*

*CALL HEX7SG*

### 3.3. Spesifikasi Subrutin RAMCHK

Nama Subrutin : *RAMCHK*

Alamat Awal : *05F6H*

Fungsi : *Men-cek adanya RAM pada suatu alamat dengan menguji pelaksanaan simpannya.*

Output Data : *Jika RAM ada dan pelaksanaan simpannya baik bit flag Z=1, jika pada address tersebut pelaksanaan simpannya jelek atau tidak ada RAM maka flag Z=0.*

Cara menggunakan :

*LD HL, address awal RAM yang akan di Cek  
LD BC, jumlah lokasi RAM yang akan di Cek  
CekR CALL RAMCHK  
JR Z, Cek  
HALT  
Cek CP I  
JP PE, CekR  
RST 00H*

### 3.4. Spesifikasi Subrutin TONE

Nama Subrutin : *TONE*

Alamat Awal : *05E4H*

Fungsi : *Meberikan nada dengan Frekuensi dan lama suara yang dapat kita pilih.*

Input : *Register C diisi bilangan yang menentukan frekuensi nada dengan rumus :*



$$F = \frac{200}{10 + 3 (\text{isi Register C})} \text{ KHz}$$

$$T = (44 + 13 (\text{Isi Register C})) \times 2 \times 0,56 \text{ mikro detik}$$

Register HL diisi bilangan 16 bit dari 0000h - 7FFFH dan isi HL menentukan lama suara

Output Data : Nada di Loud Speaker

Cara menggunakan :

LD HL, salah satu bilangan 0000-7FFFH

LD C, salah satu bilangan 00H - FFH

CALL TONE

### 3.5. Spesifikasi Subrutin TONE1K

Nama Subrutin : TONE1K

Alamat Awal : 05DEH

Fungsi : Meberikan nada dengan Frekuensi 1 KHz

Input : Register HL diisi bilangan 16 bit dari 0000h - 7FFFH untuk menentukan lama suara

Output Data : Nada 1 KHz di Loud Speaker

Cara menggunakan :

LD HL, salah satu bilangan 0000-7FFFH

CALL TONE1K

Program subrutin TONE ada pada ROM monitor. Pembangkitan nada suara melalui saluran PC7 pada Port C PPI 8255 menggunakan cara sederhana, yaitu dengan cara membangkitkan Square Wave pada frekuensi dan lama suara dipilih.

Didasarkan pada dasar nada isi register C = 82 = DO maka tangga nada suara 2 oktaf dapat disusun seperti Gambar 7.20.

No.	Nada	DATA
1.	DO	82H
2.	RE	76H
3.	MI	68H
4.	FA	62H
5.	SOL	57H
6.	LA	48H
7.	SI	45H
8.	DO	40H
9.	RE	39H
10.	MI	33H
11.	FA	31H
12.	SOL	2BH
13.	LA	26H
14.	SI	21H
15.	DO	20H

Gambar 7.20. Data nada

Tangga nada tersebut dapat dibunyikan dengan program 7.20 sebagai berikut:

### Program 7.20

Addr	Op-Code	No. Urut	Label	Assembly	Keterangan
		1	Program 7.20 Membunyikan tangga nada menggunakan subrutin TONE		
		2			
		3	TONE	EQU 05E4H	
		4			
		5		ORG 1800	
1800	DD 21 1E 18	6		LD IX, TABEL	Membaca table frekuensi nada
1804	DD 4E 00	7		LD C, (IX+00)	
1807	21 00 01	8	Bunyi	LD HL, 0100H	
180A	CD E4 05	9		CALL TONE	
		10			
180D	0E 01	11		LD C, 01H	
180F	21 00 01	12		LD HL, 0100H	
1812	CD E4 05	13		CALL TONE	
		14			
1815	AF	15		XOR A	
1816	DD 23	16		INC IX	Melanjutkan membaca data dan berhenti jika ketemu data 00H
1818	DD BE 00	17		CP (IX+00)	
181B	20 E7	18		JR NZ, Bunyi	
181D	76	19		HALT	
		20			
181E	82 76 68	21	TABEL	DB 82H, 76H, 68H	
1821	62 57 6D	22		DB 62H, 57H, 6DH	
1824	45 41 39	23		DB 45H, 41H, 39H	
1827	33 31 2B	24		DB 33H, 31H, 2BH	

Addr	Op-Code	No. Urut	Label	Assembly	Keterangan
182A	26 21 20	25		DB 26H, 21H, 20H	
182D	20 21 26	26		DB 20H, 21H, 26H	
1830	2B 31 33	27		DB 2BH, 31H, 33H	
1833	39 41 45	28	SELESAI	DB 39H, 41H, 45H	
1836	4D 57 62	29		DB 4DH, 57H, 62H	
1839	68 76 83	30		DB 68H, 76H, 83H	
183C	00 00	31		DB 00H, 00H	
		32	END		

Gambar 7.21. Program tangga nada

Untuk lebih memahami cara kerja program berikut disajikan contoh-contoh program aplikasi. Contoh program ini disajikan sebagai bahan untuk menganalisis struktur dan kerja dari program sistem mikroprosesor.

## 6. CONTOH-CONTOH PROGRAM:

### PROGRAM 7.21.

Addr	Op-Code	No. Urut	Label	Assembly	Keterangan
		1	Program 7.21: Menyajikan Ketikan Angka Seperti Kalkulator		
		2	Angka bobot tinggi bergeser ke kiri		
		3	SCAN	EQU 05FEH	
		4	HEX7	EQU 0689H	
		5		ORG 1900	
1900	DD 21 F9 18	6		LD IX, SIMDIS	
1904	CD FE 05	7	Tombol	CALL SCAN	
1907	CD 89 06	8		CALL HEX7	
190A	DD 2B	9		DEC IX	
190C	DD 77 00	10		LD (IX+00), A	
190F	18 F3	11		JR Tombol	
		12			
18F9	00	13	SIMDIS	DB 00H	
18FA	00	14		DB 00H	
18FB	00	15		DB 00H	
18FC	00	16		DB 00H	
18FD	00	17		DB 00H	
18FE	00	18		DB 00H	

Gambar 7.22. Program menyajikan ketikan angka kalkulator

### PROGRAM 7.22

Addr	Op-Code	No. Urut	Label	Assembly	Keterangan
		1	Program 09: Menyajikan Ketikan Angka Seperti Kalkulator		
		2	Angka 0 s/d 9 bobot tinggi bergeser ke kiri		
		3	SCAN	EQU 05FEH	
		4	HEX7	EQU 0689H	
		5		ORG 1900	
1900	DD 21 F9 18	6		LD IX, SIMDIS	

Addr	Op-Code	No. Urut	Label	Assembly	Keterangan
1904	CD FE 05	7	Tombol	CALL SCAN	Terima Ketikan Jika tombol yang ditekan bukan 0 s/d 9 ulang
1907	FE 0A	8		CP 0AH	
1909	30 F9	9		JR NC, Tombol	
190B	CD 89 06	10		CALL HEX7	
190E	DD 2B	11		DEC IX	
1910	DD 77 00	12		LD (IX+00), A	
1913	18 EF	13		JR Tombol	
		14			
18F9	00	15	SIMDIS	DB 00H	
18FA	00	16		DB 00H	
18FB	00	17		DB 00H	
18FC	00	18		DB 00H	
18FD	00	19		DB 00H	
18FE	00	20		DB 00H	

Gambar 7.23. Program menyajikan ketikan angka kalkulator

### PROGRAM 7.23

Addr	Op-Code	No. Urut	Label	Assembly	Keterangan
		1	Program 10: Menyajikan Ketikan Kode Posisi Tombol		
		2			
		3	SCAN1	EQU 0624H	
		4	HEX7SG	EQU 0678H	
		5		ORG 1900	
1900	DD 21 11 19	6		LD IX, SIMDAT	
1904	CD 24 06	7	ScanPos	CALL SCAN1	Terima Ketikan Jika tombol yang ditekan
1907	38 FB	8		JR C, ScanPos	
1909	21 11 19	9		LD HL, SIMDAT	
190C	CD 78 06	10		CALL HEX7SG	
190F	18 F3	11		JR ScanPos	
		12			

Addr	Op-Code	No. Urut	Label	Assembly	Keterangan
1911	00	13	SIMDAT	DB 00H	
1912	00	14		DB 00H	
1913	00	15		DB 00H	
1914	00	16		DB 00H	
1915	00	17		DB 00H	
11916	00	18		DB 00H	

Gambar 7.24. Program menyajikan ketikan Kode Posisi

## PROGRAM 7.24

Addr	Op-Code	No. Urut	Label	Assembly	Keterangan
		1	Program 7.24: Menyajikan Ketikan Kode Internal Tombol		
		2			
		3	SCAN	EQU 05FEH	
		4	HEX7SG	EQU 0678H	
		5		ORG 1900H	
1900	DD 21 11 19	6		LD IX, SIMDAT	
1904	CD FE 05	7	ScanInt	CALL SCAN	Terima Ketikan Jika tombol yang ditekan
1907	38 FB	8		JR C, ScanInt	
1909	21 11 19	9		LD HL, SIMDAT	
190C	CD 78 06	10		CALL HEX7SG	
190F	18 F3	11		JR ScanPos	
		12			
1911	00	13	SIMDAT	DB 00H	
1912	00	14		DB 00H	
1913	00	15		DB 00H	
1914	00	16		DB 00H	
1915	00	17		DB 00H	
11916	00	18		DB 00H	

Gambar 7.25. Program menyajikan ketikan Kode Internal

## PROGRAM 7.25

Addr	Op-Code	No. Urut	Label	Assembly	Keterangan
		1	Program 7.25: Pembangkitan Data Desimal 00 -99		
		2	Pada alamat awal 1900		
1800	0E 64	3		LD C, 64H	
1802	DD 21 00 19	4		LD IX, 1900	
1806	3E 00	5		LD A, 00	
1808	DD 77 00	6	Catat	LD (IX+00) A	
180B	C6 01	7		ADD A, 01H	
180D	27	8		DAA	
180E	DD 23	9		INC IX	
1810	0D	10		DEC C	
1811	20 F5	11		JRNZ Catat	
1813	FF	12		RST 38	

Gambar 7.26. Program Pembangkitan data desimal 00-99

## PROGRAM 7.26

Addr	Op-Code	No. Urut	Label	Assembly	Keterangan
		1	Program 7.26: Pembangkitan Data Desimal 99 - 00		
		2	Pada alamat awal 1900		
1800	0E 64	3		LD C, 64H	
1802	DD 21 00 19	4		LD IX, 1900	
1806	3E 00	5		LD A, 99	
1808	DD 77 00	6	Catat	LD (IX+00) A	
180B	C6 01	7		SUB, 01H	
180D	27	8		DAA	
180E	DD 23	9		INC IX	
1810	0D	10		DEC C	
1811	20 F5	11		JRNZ Catat	
1813	FF	12		RST 38	

Gambar 7.27. Program pembangkitan data desimal 99-00

## PROGRAM 7.27

Addr	Op-Code	No. Urut	Label	Assembly	Keterangan
		1			Program 7.27: Perkalian , Bilangan Pengali dalam Heksa Desimal di memori (1901) ; Bil yang dikalikan di memori (1900) hasil kali dimemori (1902) dan (1903)
		2			
		3			
		4			
1800	DD 21 00 19	5		LD IX , 1900	
1804	3E 00	6		LD A, 00	
1806	DD 46 00	7		LD B, (IX+00)	
1809	DD 4E 01	8		LD C, (IX+01)	
180C	1E 00	9		LD E, 00H	
180E	80	10	LOOP	ADD A, B	
180F	27	11		DAA	
1810	57	12		LD D, A	
1811	3E 00	13		LD A, 00H	
1813	8B	14		ADC A, E	
1814	27	15		DAA	
1815	5F	16		LD E, A	
1816	7A	17		LD A, D	
1817	0D	18		DEC C	
1818	C2 0E 18	19		JP NZ LOOP	
181B	DD 77 02	20		LD (IX+02), A	
181E	DD 73 03	21		LD (IX+03), E	
1811	FF	22		RST 38	

Gambar 7.28. Program perkalian