



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

BAGIAN 1

Instruction Set Mikrokontroler

Tujuan Pembelajaran Umum:

1. Mahasiswa mampu mendeskripsikan Instruction Set Mikrokontroler AT89S51

Tujuan Pembelajaran Khusus:

1. Mahasiswa memahami Instruction set sebuah mikrokontroler secara tuntas detail sebagai dasar pemahaman pemilihan instruksi pada saat pengembangan program
2. Mahasiswa dapat mengelompokkan Instruction set menjadi kelompok operasi aritmetika, operasi logika, transfer data, manipulasi variabel boolean, branching.
3. Mahasiswa memahami mode pengalamatan dan pemanfaatannya dalam instruction set

Instruction Set Mikrokontroler AT 89S51

Mikrokontroler AT89S51 memiliki sekitar 110 jenis instruksi. Keseluruhan instruksi dapat dikelompokkan menjadi :

- Kelompok Instruksi Operasi Aritmetika
- Kelompok Instruksi Operasi Logika
- Kelompok Instruksi Transfer Data
- Kelompok Instruksi Manipulasi Variabel Bolean dan
- Kelompok Operasi Branching



Instruction set mikrokontroler AT 89S51 selengkapnya dapat dilihat pada tabel berikut :

No	Mnemonik	Deskripsi	Simbol Operasi	
OPERASI ARITMETIKA				
1	ADD	A,Rn	Add isi Register ke Akumulator	$(A) \leftarrow (A) + (Rn)$
2	ADD	A,direct	Add direct byte ke Akumulator	$(A) \leftarrow (A) + (\text{direct})$
3	ADD	A, @Ri	Add indirek RAM ke Akumulator	$(A) \leftarrow (A) + ((Ri))$
4	ADD	A, #data	Add data Immediate ke Akumulator	$(A) \leftarrow (A) + \#data$
5	ADC	A,Rn	Add isi Register ke Akumulator sertakan Carry	$(A) \leftarrow (A) + (Rn) + (C)$
6	ADC	A,direct	Add direct byte ke Akumulator sertakan Carry	$(A) \leftarrow (A) + (\text{direct}) + (C)$
7	ADC	A, @Ri	Add indirek RAM ke Akumulator sertakan Carry	$(A) \leftarrow (A) + ((Ri)) + (C)$



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

No	Mnemonik	Deskripsi	Simbol Operasi
8	ADC	A, #data Add data Immediate ke Akumulator sertakan Carry	$(A) \leftarrow (A) + \#data + (C)$
9	SUBB	A,Rn Subtract Akumulator dengan register sertakan borrow	$(A) \leftarrow (A) - (Rn) - (C)$
10	SUBB	A,direct Subtract Akumulator dengan direct byte sertakan borrow	$(A) \leftarrow (A) - (\text{direct}) - (C)$
11	SUBB	A, @Ri Subtract Akumulator dengan indirect RAM sertakan borrow	$(A) \leftarrow (A) - ((Ri)) - (C)$
12	SUBB	A, #data Subtract Akumulator dengan data immediate sertakan borrow	$(A) \leftarrow (A) - \#data - (C)$
13	INC	A Increment Accumulator	$(A) \leftarrow (A) + 1$
14	INC	Rn Increment Register	$(A) \leftarrow (Rn) + 1$
15	INC	direct Increment direct byte	$(A) \leftarrow (\text{direct}) + 1$
16	INC	@Ri Increment Indirect RAM	$(A) \leftarrow ((Ri)) + 1$
17	DEC	A Decrement Accumulator	$(A) \leftarrow (A) - 1$
18	DEC	Rn Decrement Register	$(A) \leftarrow (Rn) - 1$
19	DEC	direct Decrement direct byte	$(A) \leftarrow (\text{direct}) - 1$
20	DEC	@Ri Decrement Indirect RAM	$(A) \leftarrow ((Ri)) - 1$
21	INC	DPTR Increment Data Pointer	$(DPTR) \leftarrow (DPTR) + 1$
22	MUL	AB Kalikan A dan B	$(A)_{7-0} \leftarrow (A) \times (B)$ $(B)_{15-8}$
23	DIV	AB Bagi A dengan B	$(A)_{15-8} \leftarrow (A)/(B)$ $(B)_{7-0}$
24	DA	A Decimal Adjust Accumulator	
OPERASI LOGIKA			
25	ANL	A, Rn AND Register ke Accumulator	$(A) \leftarrow (A) \wedge (Rn)$
26	ANL	A, direct AND byte direct ke Accumulator	$(A) \leftarrow (A) \wedge (\text{direct})$
27	ANL	A, @Ri AND indirect RAM ke Accumulator	$(A) \leftarrow (A) \wedge ((Ri))$
28	ANL	A, #data AND data immediate ke Accumulator	$(A) \leftarrow (A) \wedge \#data$
29	ANL	direct, A AND accumulator ke byte direct	$(\text{direct}) \leftarrow (\text{direct}) \wedge (A)$
30	ANL	direct, #data AND data immediate ke byte direct	$(\text{direct}) \leftarrow (\text{direct}) \wedge \#data$
31	ORL	A, Rn OR Register ke Accumulator	$(A) \leftarrow (A) \vee (Rn)$
32	ORL	A, direct OR byte direct ke Accumulator	$(A) \leftarrow (A) \vee (\text{direct})$
33	ORL	A, @Ri OR indirect RAM ke Accumulator	$(A) \leftarrow (A) \vee ((Ri))$
34	ORL	A, #data OR data immediate ke Accumulator	$(A) \leftarrow (A) \vee \#data$
35	ORL	direct, A OR accumulator ke byte direct	$(\text{direct}) \leftarrow (\text{direct}) \vee (A)$
36	ORL	direct, #data OR data immediate ke byte direct	$(\text{direct}) \leftarrow (\text{direct}) \vee \#data$
37	XRL	A, Rn Exclusive-OR Register ke Accumulator	$(A) \leftarrow (A) \oplus (Rn)$
38	XRL	A, direct Exclusive-OR byte direct ke Accumulator	$(A) \leftarrow (A) \oplus (\text{direct})$
39	XRL	A, @Ri Exclusive-OR indirect RAM ke Accumulator	$(A) \leftarrow (A) \oplus ((Ri))$
40	XRL	A, #data Exclusive-OR data immediate ke Accumulator	$(A) \leftarrow (A) \oplus \#data$
41	XRL	direct, A Exclusive-OR accumulator ke byte direct	$(\text{direct}) \leftarrow (\text{direct}) \oplus (A)$
42	XRL	direct, #data Exclusive-OR data immediate ke byte direct	$(\text{direct}) \leftarrow (\text{direct}) \oplus \#data$



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

No	Mnemonik		Deskripsi	Simbol Operasi
43	CLR	A	Clear Accumulator	$(A) \leftarrow 0$
44	CPL	A	Complement Accumulator	$(A) \leftarrow (A)^*$
45	RL	A	Rotate Accumulator Left	$(An+1) \leftarrow (An) ; n=0-6$ $(A0) \leftarrow (A7)$
46	RLC	A	Rotate Accumulator Left through Carry	$(An+1) \leftarrow (An) ; n=0-6$ $(A0) \leftarrow (C)$ $(C) \leftarrow (A7)$
47	RR	A	Rotate Accumulator Right	$(An) \leftarrow (An+1) ; n=0-6$ $(A7) \leftarrow (A0)$
48	RRC	A	Rotate Accumulator Right	$(An) \leftarrow (An+1) ; n=0-6$ $(A7) \leftarrow (C)$ $(C) \leftarrow (A0)$
49	SWAP	A	Rotate Accumulator Right through Carry	$(A3-0) \leftrightarrow (A7-4)$
TRANSFER DATA				
50	MOV	A, Rn	Move Register ke Accumulator	$(A) \leftarrow (Rn)$
51	MOV	A, direct	Move byte direct ke Accumulator	$(A) \leftarrow (direct)$
52	MOV	A, @Ri	Move indirect RAM ke Accumulator	$(A) \leftarrow ((Ri))$
53	MOV	A, #data	Move data immediate ke Accumulator	$(A) \leftarrow \#data$
54	MOV	Rn, A	Move Accumulator ke Register	$(Rn) \leftarrow (A)$
55	MOV	Rn, direct	Move byte direct ke Register	$(Rn) \leftarrow (direct)$
56	MOV	Rn, #data	Move data immediate ke Register	$(Rn) \leftarrow \#data$
57	MOV	direct, A	Move Accumulator ke byte direct	$(direct) \leftarrow (A)$
58	MOV	direct, Rn	Move Register ke byte direct	$(direct) \leftarrow (Rn)$
59	MOV	direct, direct	Move byte direct ke direct	$(direct) \leftarrow (direct)$
60	MOV	Direct, @Ri	Move Indirect RAM ke byte direct	$(direct) \leftarrow ((Ri))$
61	MOV	direct, #data	Move data immediate ke byte direct	$(direct) \leftarrow \#data$
62	MOV	@Ri, A	Move Accumulator ke indirect RAM	$((Ri)) \leftarrow (A)$
63	MOV	@Ri, direct	Move byte direct ke indirect RAM	$((Ri)) \leftarrow (direct)$
64	MOV	@Ri, #data	Move data immediate ke indirect RAM	$((Ri)) \leftarrow \#data$
65	MOV	DPTR, #data16	Load Data Pointer dengan data konstanta 16 bit	$(DPTR) \leftarrow \#data 15-0$ $DPH=data15-8, DPL=data7-0$
66	MOVC	A, @A+ DPTR	Move code byte relative pada DPTR ke Accumulator	$(A) \leftarrow ((A) + (DPTR))$
67	MOVC	A, @A + PC	Move code byte relative pada PC ke Accumulator	$(PC) \leftarrow (PC) + 1$ $(A) \leftarrow ((A) + (PC))$
68	MOVX	A, @Ri	Move eksternal RAM (8bit address) ke Accumulator	$(A) \leftarrow ((Ri))$
69	MOVX	A, @DPTR	Move eksternal RAM (16bit address) ke Accumulator	$(A) \leftarrow ((DPTR))$
70	MOVX	@Ri, A	Move Accumulator ke eksternal RAM (8bit address)	$((Ri)) \leftarrow (A)$



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

No	Mnemonik		Deskripsi	Simbol Operasi
71	MOVX	@DPTR, A	Move Accumulator ke eksternal RAM (16 bit adres)	$(DPTR) \leftarrow (A)$
72	PUSH	Direct	Push direct byte ke STACK	$(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (direct)$
73	POP	Direct	Pop direct byte dari STACK	$(direct) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$
74	XCH	A, Rn	Tukarkan isi Register dengan Accumulator	$(A) \leftrightarrow (Rn)$
75	XCH	A, direct	Tukarkan byte direct dengan Accumulator	$(A) \leftrightarrow (direct)$
76	XCH	A, @Ri	Tukarkan indirect RAM dengan Accumulator	$(A) \leftrightarrow ((Ri))$
77	XCHD	A, @Ri	Tukarkan Low Order Digit indirect RAM dengan Accumulator	$(A3-0) \leftrightarrow ((Ri3-0))$
MANIPULASI VARIABEL BOOLEAN				
78	CLR	C	Clear Carry	$(C) \leftarrow 0$
79	CLR	Bit	Clear direct bit	$(bit) \leftarrow 0$
80	SETB	C	Set Carry	$(C) \leftarrow 1$
81	SETB	Bit	Set direct bit	$(bit) \leftarrow 1$
82	CPL	C	Complemen Carry	$(C) \leftarrow (C)^*$
83	CPL	Bit	Complement direct bit	$(bit) \leftarrow (bit)$
84	ANL	C, bit	AND direct bit ke Carry	$(C) \leftarrow (C) \wedge (bit)$
85	ANL	C, /bit	AND complement direct bi ke Carry	$(C) \leftarrow (C) \wedge (bit)^*$
86	ORL	C, bit	OR direct bit ke Carry	$(C) \leftarrow (C) \vee (bit)$
87	ORL	C, /bit	OR complement direct bi ke Carry	$(C) \leftarrow (C) \vee (bit)^*$
88	MOV	C, bit	Move direct but ke Carry	$(C) \leftarrow (bit)$
89	MOV	Bit, C	Move Carry ke direct bit	$(bit) \leftarrow (C)$
90	JC	Rel	Jump jika Carry set C=1	$(PC) \leftarrow (PC) + 2$ Jika $(C) = 1$ then $(PC) \leftarrow (PC) + rel$
91	JNC	Rel	Jump jika Carry not Set C = 0	$(PC) \leftarrow (PC) + 2$ Jika $(C) = 0$ then $(PC) \leftarrow (PC) + rel$
92	JB	Bit, rel	Jump jika direct Bit set	$(PC) \leftarrow (PC) + 3$ Jika $(bit) = 1$ then $(PC) \leftarrow (PC) + rel$
93	JNB	Bit, rel	Jump jika direct bit Not set	$(PC) \leftarrow (PC) + 3$ Jika $(bit) = 0$ then $(PC) \leftarrow (PC) + rel$
94	JBC	Bit, rel	Jump jika direct Bit Set & Clear bit	$(PC) \leftarrow (PC) + 3$ Jika $(bit) = 1$ then $(bit) \leftarrow 0$ $(PC) \leftarrow (PC) + rel$
PROGRAM BRANCHING				
95	ACALL	Addr 11	Absolute Call	$(PC) \leftarrow (PC) + 2$ $(SP) \leftarrow (SP) + 1$



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

No	Mnemonik		Deskripsi	Simbol Operasi
				$((SP)) \leftarrow (PC\ 7-0)$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC15-8)$ $(PC10-0) \leftarrow \text{page address}$
96	LCALL	Addr16	Long Call	$(PC) \leftarrow (PC) + 3$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC\ 7-0)$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC15-8)$ $(PC10-0) \leftarrow \text{addr15-0}$
97	RET		Return	$(PC\ 15-8) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$ $(PC7-0) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$
98	RETI			$(PC\ 15-8) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$ $(PC7-0) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$
99	AJMP	Addr11	Absolute Jump	$(PC) \leftarrow (PC) + 2$ $(PC10-0) \leftarrow \text{page address}$
100	LJMP	Addr16	Long Jump	$(PC) \leftarrow (PC) + 3$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC\ 7-0)$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC15-8)$ $(PC10-0) \leftarrow \text{addr15-0}$
101	SJMP	Rel	Short Jump	$(PC) \leftarrow (PC) + 2$ $(PC) \leftarrow PC + \text{rel}$
102	JMP	@A+ DPTR	Jump indirect	$(PC) \leftarrow (A) + (DPTR)$
103	JZ	rel	Jump If Accumulator Zero	$(PC) \leftarrow (PC) + 2$ Jika (A) = 0 then $(PC) \leftarrow (PC) + \text{rel}$
104	JNZ	rel	Jump If Accumulatot Not Zero	$(PC) \leftarrow (PC) + 2$ Jika (A) <= > 0 then $(PC) \leftarrow (PC) + \text{rel}$
105	CJNE	A,direct, rel	Compare and Jump If Not Equal	$(PC) \leftarrow (PC) + 3$ jika (A) <> (direct) Then $(PC) \leftarrow (PC) + \text{relative}$ Jika (A) < (direct) then (C) $\leftarrow 1$ ELSE (C) $\leftarrow 0$
106	CJNE	A,#data, rel	Compare and Jump If Not Equal	$(PC) \leftarrow (PC) + 3$ jika (A) <> data Then $(PC) \leftarrow (PC) + \text{relative}$ Jika (A) < data then (C) $\leftarrow 1$ ELSE (C) $\leftarrow 0$
107	CJNE	Rn,#data, rel	Compare and Jump If Not Equal	$(PC) \leftarrow (PC) + 3$ jika (Rn) <> data Then (PC)



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

No	Mnemonik		Deskripsi	Simbol Operasi
				$\leftarrow (PC) + \text{relative}$ Jika $(Rn) < \text{data}$ then $(C) \leftarrow 1$ ELSE $(C) \leftarrow 0$
108	CJNE	@Ri, #data,rel	Compare and Jump If Not Equal	$(PC) \leftarrow (PC) + 3$ jika $((Ri)) \neq \text{data}$ Then $(PC) \leftarrow (PC) + \text{relative}$ Jika $((Ri)) < \text{data}$ then $(C) \leftarrow 1$ ELSE $(C) \leftarrow 0$
109	DJNZ	Rn,rel	Decrement and Jump If Not Zero	$(PC) \leftarrow (PC) + 2$ $(Rn) \leftarrow (Rn) - 1$ Jika $(Rn) > 0$ atau $(Rn) < 0$ Then $(PC) \leftarrow (PC) + \text{rel}$
110	DJNZ	Direct,rel	Decrement and Jump If Not Zero	$(PC) \leftarrow (PC) + 2$ $(\text{direct}) \leftarrow (\text{direct}) - 1$ Jika $(\text{direct}) > 0$ atau $(\text{direct}) < 0$ Then $(PC) \leftarrow (PC) + \text{rel}$
111	NOP		No Operation	

ALIH DATA (MOVING DATA)

Komputer atau sistim mikroprosesor atau mikrokontroler secara khusus memerlukan proses alih data dari suatu lokasi ke lokasi lainnya. Dalam mikrokontroler ada enam kemungkinan proses alih data yang dapat terjadi yaitu:

1. Alih data dari memori ke memori
2. Alih data dari register ke register
3. Alih data dari memori ke register
4. Alih data dari register ke memori
5. Alih data immediate ke register
6. Alih data immediate ke memori

Dalam proses alih data dari enam kemungkinan yang dapat terjadi, data sumber disebut sebagai "Source" dialihkan dalam hal ini secara nyata di copy ke tujuan yang disebut "Destinasi". Destinasi disebut atau ditulis lebih awal kemudian baru diikuti oleh Source. Alih data dalam sistim mikroprosesor pada umumnya menggunakan perintah-perintah MOV, LOAD, PUSH, POP, dan EXCHANGE. Pola umum penulisannya adalah :

- o **MOVE Destinasi, Source**
- o **PUSH Source atau POP Destinasi**
- o **XCH Destinasi, Source**



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

Cara khas pengalamatan alih data dalam mikrokontroler disebut dengan **Mode Pengalamatan** (*Addressing Modes*). Ada empat jenis mode pengalamatan yaitu:

- **Immediate addressing mode**
- **Register addressing mode**
- **Direct addressing mode**
- **Indirect Addressing mode**

Dalam mikrokontroler AT89S51 Opcode Perintah “MOVE” mencakup memori :

- **Internal RAM**
- **Internal SFR**
- **External RAM**
- **Internal dan eksternal ROM**

Tipe atau jenis perintah yang digunakan untuk alih data :

- **MOV**
- **MOVX**
- **MOVC**
- **PUSH dan POP**
- **XCH**

➤ **ADDRESSING MODE**

1. **Immediate Addressing Mode**

Immediate addressing mode merupakan cara yang paling sederhana untuk membangkitkan data pada destinasi dengan cara membuat data menjadi bagian dari opcode. Sumber data secara langsung dinyatakan sebagai bagian dari perintahnya. Pada saat AT89S51 mengeksekusi perintah ini, program counter secara otomatis naik satu digit untuk mengambil data secara langsung dari memori. Mnemonic untuk data immediate menggunakan tanda pagar “#” Pola immediate addressing mode dan contoh perintahnya adalah sebagai berikut :

Instruksi	Data
Opcode	Next byte

- | Mnemonic | Operasi |
|--------------------------|----------------------------------|
| ○ MOV A, #01h | copy data 01h ke Register A |
| ○ MOV R3, #1Ch | copy data 1Ch ke Register R3 |
| ○ MOV DPTR,#ABCDh | copy data ABCDh ke Register DPTR |

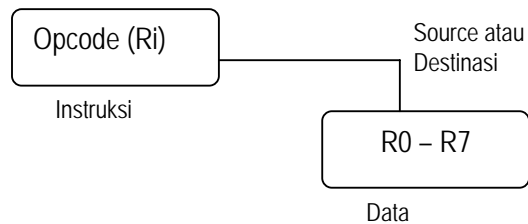


MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

2. Register Addressing Mode

Register addressing mode adalah model pengalamatan alih data dimana nama register (A, R0 – R7) digunakan sebagai bagian dari opcode mnemonik baik sebagai source atau sebagai destinasi. Model dan contoh register addressing mode sebagai berikut:



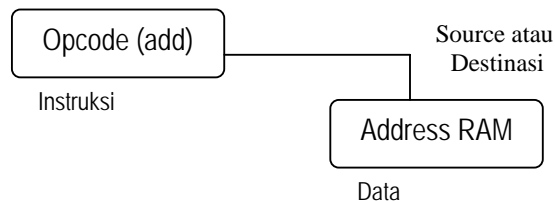
- **Mnemonic**
MOV A, R0
MOV R5, A
MOV R7,A

Operasi
copy data pada R0 ke register A
copy data pada A ke R5
copy data pada A ke R7

3. Direct Addressing Mode

Direct addressing mode menggunakan pengalamatan dengan penunjukan alamat secara langsung salah satu dari 128 byte alamat RAM internal (gambar 13) dan Special Function Register (SFR) gambar 18 Dalam satu waktu hanya satu bank atau 8 register yang dapat aktif dari 4 bank yang ada. Jika AT89S51 dalam keadaan reset bank 0 secara otomatis terpilih. Untuk memilih bank lainnya digunakan seting RS0 dan RS1 pada PSW.

Model dan contoh direc addressing mode sebagai berikut:



- **Mnemonic**
MOV A, 80h
MOV A, P0
MOV 80h, A
MOV P0 , A

Operasi
copy data dari Port 0 ke register A
copy data dari Port 0 ke register A
copy data dari register A ke Port 0
copy data dari register A ke Port 0



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

4. Indirect Addressing Mode

- Menggunakan register sebagai pencatat atau pemegang alamat aktual yang akan digunakan untuk memindahkan data
- Register itu sendiri bukan alamat
- Menggunakan R0 dan R1 sebagai Pointer data
- Menggunakan tanda “ @ ”

Mnemonic	Operasi
MOV A, @R0	copy isi data dari alamat yang dicatat oleh R0 ke register A
MOV @R1, A	copy data yang ada di register A ke alamat yang dicatat oleh R1
MOV @R0,80h	copy data dari Port 0 ke alamat yang tercatat oleh R0

➤ Pengalamatan eksternal menggunakan MOVX dan MOVC

MOVX (External Data Moves):

- Digunakan untuk pengalamatan eksternal
- R0, R1, dan DPTR digunakan untuk hold alamat dari byte data RAM eksternal
- R0 dan R1 dibatasi pada eksternal RAM alamat 00h – FFh, sedangkan DPTR dapat mengalami maksimum space 0000h-FFFFh.
- Alih data dari memori eksternal harus ke register A
- MOVX normalnya digunakan dengan alamat RAM eksternal atau alamat I/O eksternal

Mnemonic	Operasi
MOVX A, @R1	copy isi dari alamat eksternal R1 ke A
MOVX A, @R0	copy isi dari alamat eksternal R0 ke A
MOVX A, @DPTR	copy isi dari alamat eksternal DPTR ke A
MOVX @DPTR,A	copy data dari A ke alamat eksternal DPTR

MOVC (Code Memory Read-Only Data Moves)

- Digunakan alih data dari alamat sumber di ROM eksternal ke register A.

Mnemonic	Operasi
MOVC A, @A+DPTR1	copy byte kode pada ROM dengan alamat yang dibangun dari data A dengan data DPTR ke Register A
MOVC A, @A+PC	copy byte kode pada ROM dengan alamat yang dibangun dari data A dengan data PC ke Register A



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

Contoh :

MOV DPTR, #1234h	copy data immediate 1234h ke DPTR
MOV A, #56h	copy data immediate 56h ke Reg. A
MOVC A, @A+DPTR	copy data pada alamat 128Ah ke A
MOVC A, @A+PC	copy data pada alamat 4056h ke A jika PC = 4000

➤ **PUSH dan POP**

- Menggunakan register SP (Stack Pointer) sebagai pencatat alamat
- Data dialihkan dalam area RAM yang disebut dengan Stack
- Register SP berisi alamat Stack
- PUSH mengcopy data dari suatu source ke stack
- SP ditambah satu sebelum data dicopy ke RAM internal
- POP mengcopy data dari stack ke suatu destinasi
- SP diset sama dengan 07h pada saat ada Reset sehingga defaultnya perintah PUSH pertama menulis data ke R0 bank 1
- Jika SP mencapai FFh maka “rolls over”
- PUSH di atas 7Fh adalah ERROS karena alamat RAM maks 7Fh
- **Mnemonic** **Operasi**
PUSH add SP ditambah satu, copy data dari suatu alamat ke alamat internal RAM yang tercatat dalam SP
POP add SP dikurangi satu, copy data dari alamat internal RAM yang tercatat dalam SP ke suatu alamat Register

Contoh:

MOV 81h,#30h	copy data immediate 30h ke SP
MOV R0,#ACh	copy data immediate ACh ke Reg R0
PUSH 00h	SP=31h, alamat 31h berisi data ACh
PUSH 00h	SP=32h,alamat 32h berisi data ACh
POP 01h	SP=31, Register R1 sekarang berisi data ACh
POP 80h	SP=30h, Port 0 berisi data ACh

➤ **Data Exchanges XCH**

- **Mnemonic** **Operasi**
XCH A,Rr Pertukarkan data byte diantara register Rr dan A
XCH A,add Pertukarkan data byte diantara add dan A
XCH A,@Rp Pertukarkan data byte diantara A dan data yang ada dimemori yang alamatnya dicatat oleh register Rp
XCHD A,@Rp Pertukarkan data lower nibble A dengan data yang ada dimemori yang alamatnya dicatat oleh Rp



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

Contoh:

XCH A,R7	Pertukarkan data byte diantara register A dan register R7
XCH A,F0h	Pertukarkan data byte diantara register A dan register B
XCH A,@R1	Pertukarkan data byte diantara register A dan data alamat pada R1
XCHD A,@R1	Pertukarkan data lower nibble diantara register A dan data alamat pada R1

➤ Operasi Logika

➤ Operator Boolean

AND	<u>AT89S51 Mnemonic</u> ANL (AND logical)
OR	ORL (OR logical)
XOR	XRL (Exclusive OR logical)
NOT	CPL (complement)
RL	Rotate byte to Left
RLC	Rotate byte dan carry-bit to left

Operasi Logika Level Byte

Contoh:

Mnemonic	Operasi
MOV A,#FFh	A = FFh
MOV R0,#77h	R0 = 77h
ANL A,R0	A = 77h
MOV 15h, A	15h = 77h
CPL A	A = 88h
ORL 15h,#88h	15h = FFh
XRL A, 15h	A = 77h
XRL A, R0	A = 00h
ANL A,15h	A = 00h
ORL A, R0	A = 77h
CLR A	A = 00h

Operasi Logika Level Bit

RAM internal dan SFR dapat dialamati dalam dua mode yaitu mode pengalamatan byte dan mode pengalamatan bit. Mode pengalamatan bit sangat tepat digunakan jika anda hanya membutuhkan pengolahan salah satu bit dari suatu byte sebagai contoh dalam mengontrol register. Pengalamatan bit pada RAM dapat dilihat kembali pada gambar 13 halaman 31, sedangkan pengalamatan bit untuk SFR adalah seperti berikut:



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

SFR	Alamat Langsung	Alamat Bit
A	E0h	E0h – E7h
B	F0h	F0h – F7h
IE	A8h	A8h – AFh
IP	B8	B8h – BFh
P0	80h	80h – 87h
P1	90h	90h – 97h
P3	B0h	B0h -- B7h
PSW	D0h	D0h – D7h
TCON	88h	88h – 8Fh
TMOD	98h	98h – 9Fh

Dalam operasi Bolean Level Bit Carry Flag (C) pada PSW SFR bekerja sebagai destinasi.

Mnemonic	Operasi
ANL C,b	AND C dengan bit teralamat ; catat hasilnya di C
ANL C,/b	AND C dengan komplement dari bit yang teralamat; catat hasilnya di C; bit yang teralamat tidak berubah
ORL C,b	OR C dengan bit teralamat ; catat hasilnya di C
ORL C,/b	OR C dengan komplement dari bit yang teralamat; catat hasilnya di C; bit yang teralamat tidak berubah
CPL C	Komplemen flag Carry
CPL b	Komplemen bit teralamat
CLR C	Clear flag Carry menjadi 0
CLR b	Clear bit teralamat menjadi 0
MOV C,b	Copy data pada bit teralamat ke flag Carry
MOV b,C	Copy data pada flag Carry ke bit teralamat
SETB C	Set flag Carry menjadi 1
SETB b	Set bit teralamat menjadi 1

Contoh :

Mnemonic	Operasi
SETB 00h	Bit 0 dari RAM byte 20h = 1
MOV C,00h	Carry C = 1
MOV 7Fh,C	Bit 7 dari RAM alamat byte 2Fh = 1
ANL C,/00h	C = 0; bit 0 dari RAM byte 20h = 1
ORL C,00h	C = 1



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

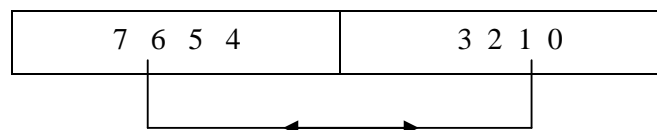
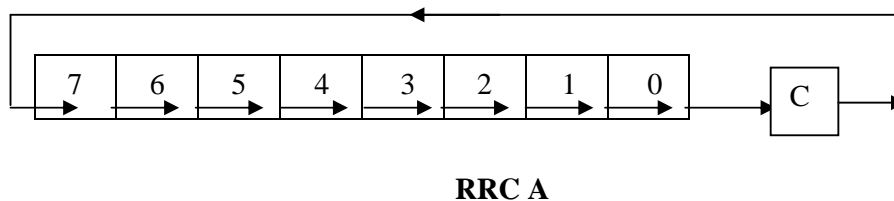
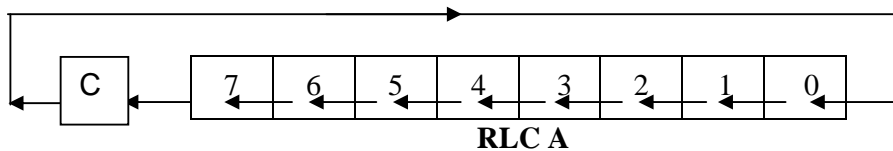
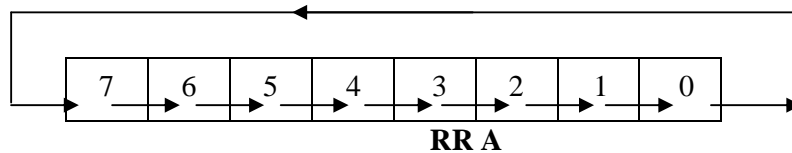
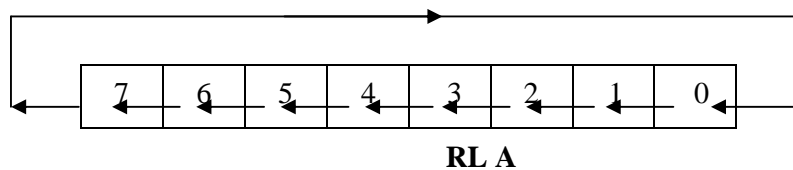
CPL 7Fh Bit 7 dari RAM alamat byte 2Fh = 0

CLR C C = 0

ORL C,/7Fh C = 1; bit 7 dari RAM byte 2Fh = 0

Operasi Rotate dan Swap

Rotate data sangat potensial sekali untuk mengolah dan membangkitkan data beraturan tanpa opcode khusus. Register A dapat digunakan untuk memutar satu posisi bit dengan melibatkan atau tidak melibatkan Carry. Sedangkan Swap bekerja mempertukarkan nibble tinggi dengan nibble rendah pada register A.





MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

Contoh :

Mnemonic	Operasi
MOV A, #0A5h	A = 1010 0101 = A5h
RR A	A = 1101 0010 = D2h
RR A	A = 0110 1001 = 69h
RR A	A = 1011 0100 = B4h
RR A	A = 0101 1010 = 5Ah
SWAP A	A = 1010 0101 = A5h
CLR C	C = 0; A = 1010 0101 = A5h
RRC A	C = 1; A = 0101 0010 = 52h
RRC A	C = 0; A = 1010 1001 = A9h
RL A	A = 0101 0011 = 53h
RL A	A = 1010 0110 = A6h
SWAP A	A = 0110 1010 = 6Ah

C. OPERASI ARITMETIKA

Aplikasi mikrokontroler sering membutuhkan perhitungan data matematika. Mikrokontroler dirancang tidak sebagai “pengolah angka” sebagaimana komputer untuk tujuan umum. Pokok dari pengembangan mikrokontroler adalah sebagai piranti kendali peristiwa yang berubah dalam waktu nyata. Kendati demikian opcode untuk operasi matematika yang cukup harus disediakan pada setiap mikrokontroler.

Pada AT 89S51 ada 24 opcode aritmetika yang dikelompokkan menjadi:

Mnemonic	Operasi
INC destinasi	Increment destination dengan 1
DEC destinasi	Decrement destination dengan 1
ADD/ADC dest,source	Add source ke destinasi tanpa/dengan carry
SUBB dest, source	Subtract dengan carry, source dari destinasi
MUL AB	Kalikan isi register A dan B
DIV AB	Bagi isi register A dengan isi register B
DA A	Decimal Adjust untuk register A



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

➤ FLAG

Flag C, AC, dan OV adalah flag aritmetika. Ketiga flag ini set 1 atau 0 secara otomatis, bergantung hasil operasi aritmetika sebelumnya.

Mnemonic Instruksi	Flag yang terpengaruh
ADD	C AC OV
ADC	C AC OV
ANL C,direct	C
CJNE	C
CLR C	C = 0
CPL C	C = C*
DA A	C
DIV	C = 0 OV
MOV C, direct	C
MUL	C = 0 OV
ORL C, direct	C
RLC	C
RRC	C
SETB C	C = 1
SUBB	C AC OV

➤ Increment dan Decreament

Mnemonic Instruksi	Operasi
INC A	Tambahkan satu nilai isi register A
INC Rr	Tambahkan satu nilai isi register Rr
INC add	Tambahkan satu nilai data pada alamat langsung
INC @Rp	Tambahkan satu nilai data pada alamat yang dicatat oleh register Rp
INC DPTR	Tambahkan satu nilai register 16 DPTR
DEC A	Kurangkan satu nilai isi register A
DEC Rr	Kurangkan satu nilai isi register Rr
DEC add	Kurangkan satu nilai data pada alamat langsung
DEC @Rp	Kurangkan satu nilai data pada alamat yang dicatat oleh register Rp



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

Contoh :

Mnemonik Instruksi	Operasi
MOV A,#3Ah	A = 3Ah
DEC A	A = 39h
MOV R0,#15h	R0 = 15h
MOV 15h,#12h	RAM internal alamat 15h = 12h
INC @R0	RAM internal alamat 15h = 13h
DEC 15h	RAM internal alamat 15h = 12h
INC R0	R0 = 16h
MOV 16h, A	RAM internal alamat 16h = 39h
INC @R0	RAM internal alamat 16h = 3Ah
MOV DPTR,#12FFh	DPTR = 12FFh
INC DPTR	DPTR = 1300h
DEC 83h	DPTR = 1200h

➤ **Penjumlahan**

Mnemonik Instruksi	Operasi
ADD A,#n	Jumlahkan A dengan data immediate n dan hasilnya disimpan di A
ADD A, Rr	Jumlahkan A dengan Reg Rr dan hasilnya disimpan di A
ADD A,add	Jumlahkan A dengan data alamat langsung dan hasilnya disimpan di A
ADD A,@Rp	Jumlahkan A dengan data alamat yang dicatat Rp dan hasilnya disimpan di A
ADC A,#n	Jumlahkan A dengan data immediate n dan carry ; hasilnya disimpan di A
ADC A, Rr	Jumlahkan A dengan Reg Rr dan Carry ; hasilnya disimpan di A
ADC A,add	Jumlahkan A dengan data alamat langsung dan Carry ; hasilnya disimpan di A
ADC A,@Rp	Jumlahkan A dengan data alamat yang dicatat Rp dan Carry hasilnya disimpan di A

Contoh :

Mnemonik Instruksi	Operasi
ADC A, #1Ch	A = 1Ch
MOV R5,#0A1h	R5 = A1h
ADD A, R5	A = BDh; C = 0, OV = 0
ADD A, R5	A = 5Eh; C = 1, OV = 1
ADC A,#10h	A = 6Fh; C = 0, OV = 0
ADC A,#10h	A = 7Fh; C = 0, OV = 0



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

➤ Pengurangan

Mnemonik Instruksi

SUBB A,#n

SUBB A, Rr

SUBB A,add

SUBB A,@Rp

Operasi

Kurangkan A dengan data immediate n dan carry; hasilnya disimpan di A

Kurangkan A dengan Reg Rr dan Carry ; hasilnya disimpan di A

Kurangkan A dengan data alamat langsung dan Carry ; hasilnya disimpan di A

Kurangkan A dengan data alamat yang dicatat Rp dan Carry; hasilnya disimpan di A

Contoh :

Mnemonik Instruksi

MOV 0D0h , #00h

MOV A,#3Ah

MOV 45h,#13h

SUBB A,45h

SUBB A,45h

SUBB A,#80h

SUBB A,#22h

SUBB A,#0FFh

Operasi

Carry = 0

A = 3Ah

Alamat 45h = 13h

A = 27h ; C = 0 , OV = 0

A = 14h ; C = 0 , OV = 0

A = 94h ; C = 1 , OV = 1

A = 71h ; C = 0 , OV = 0

A = 72h ; C = 1 , OV = 0

➤ Perkalian

Mnemonik Instruksi

MOV A , #7Bh

MOV 0F0h,#02h

MUL AB

MOV A, #0FEh

MUL AB

Operasi

A = 7Bh

B = 02h

A = 00h dan B = F6h ; OV = 0

A = FEh

A = 14h dan B = F4h; OV = 1

➤ Pembagian

Mnemonik Instruksi

MOV A , #0FFh

MOV 0F0h,#2Ch

DIV AB

DIV AB

Operasi

A = FFh (255d)

B = 2Ch (44d)

A = 05h dan B = 23h ; 255d=(5x44)+35

A = 00h dan B = 00h



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

➤ Aritmetika Desimal

Contoh :

Mnemonik Instruksi	Operasi
MOV A, #42h	A = 42 BCD
ADD A, #13h	A = 55h ; C = 0
DA A	A = 55h ; C = 0
ADD A, #17h	A = 6Ch; C = 0
DA A	A = 72 BCD; C = 0
ADC A, #34h	A = A6h; C = 0
DA A	A = 06 BCD; C = 1
ADC A, #11h	A = 18 BCD; C = 0
DA A	A = 18 BCD; C = 0

D. OPERASI JUMP DAN CALL

Instruksi Jump dan call adalah kode-kode pengambilan keputusan dalam mengatur pencabangan aliran program berdasarkan perubahan isi Program Counter. Jump merubah isi PC secara permanen sedangkan Call merubah isi PC secara temporer.

Jenis-jenis Jump dan Call :

- Jump on bit condition
- Compare Bytes and Jump if Not equal
- Decrement Byte and Jump if Not Zero
- Call a Subroutine
- Return from a Subroutine

Perintah Jump dan call bekerja mengganti isi PC dengan alamat baru sehingga menyebabkan eksekusi program berlangsung pada alamat baru pada PC.

➤ Jump Bit

- Bekerja berdasarkan status flag carry pada PSW atau status lokasi bit teralamati.

Mnemonik Instruksi	Operasi
JC radd	Jump relatif jika carry set 1
JNC radd	Jump relatif jika carry reset 0
JB b, radd	Jump relatif jika bit teralamati set 1



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

JNB b, radd

Jump relatif jika bit teralamati reset 0

JBC b,radd

Jump relatif jika bit teralamati set 1, dan clear bit teralamati menjadi 0

Contoh :

Alamat	Mnemonik	Komentar
Loop:	MOV A, #10h	; A = 10h
	MOV R0,A	; R0 = 10h
Adda:	ADD A,R0	; A = A + R0
	JNC Adda	: Lompat ke Adda jika carry = 0, terus jika C = 1
	MOV A,#10h	; A = 10h
Addr:	ADD A, R0	; A = A + R0
	JNB 0D7h, Addr	; Lompat ke Addr jika bit carry = 0, terus jika C = 1;
	JBC 0D7h,Loop	; Lompat ke Loop jika bit carry = 1, dan C = 0

➤ Jump Byte

- o Instruksi yang mengetes byte data

Mnemonik Instruksi	Operasi
CJNE A,add, radd	Bandingkan isi register A dengan data suatu alamat langsung; jika tidak sama lompat ke alamat relatif; set carry 1 jika A kurang dari isi alamat langsung “n”, keadaan lain set cary 0.
CJNE A,#n, radd	Bandingkan isi register A dengan data immediate n; jika tidak sama lompat ke alamat relatif; set carry 1 jika A kurang dari “n”, , keadaan lain set cary 0.
CJNE Rn,#n, radd	Bandingkan isi register Rn dengan data immediate n; jika tidak sama lompat ke alamat relatif; set carry 1 jika Rn kurang dari “n”, keadaan lain set cary 0.
CJNE @Rp,#n, radd	Bandingkan isi/data suatu alamat yang dicatat register Rp dengan data immediate n; jika tidak sama lompat ke alamat relatif; set carry 1 jika data pada alamat yang dicatat Rp kurang dari “n”, keadaan lain set cary 0
DJNZ Rn,radd	Kurangi satu isi register Rn dan lompat ke alamat relatif jika Rn tidak sama nol; Rn=0 kontinyu/lanjut.
DJNZ add,radd	Kurangi satu isi data alamat lansung add dan lompat ke alamat relatif jika data pada alamat langsung add tidak sama nol; data pada alamat langsung add =0 kontinyu/lanjut.
JZ,radd	Lompat ke alamat relatif jika A=0
JNZ,radd	Lompat ke alamat relatif jika A tidak =0



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

➤ Jump Tanpa Kondisi

Mnemonik Instruksi

JMP@A+DPTR

AJMP sadd

LJMP ladd

SJMP radd

Operasi

Lompat ke alamat yang dibangun oleh A+DPTR.

Lompat ke alamat absolut range pendek

Lompat ke alamat absolut range panjang

Lompat ke alamat relatif range pendek

Contoh :

Alamat	Mnemonik Instruksi	Komentar
	ORG 0100h	
Mulai:	MOV A,#30h	; A = 30h
	MOV 50h,#00h	;RAM lokasi 50h = 00h
Putar:	CJNE A,50h, Bawah	; lompat ke bawah A=30h tidak sama dengan data alamat 50 = 00
	SJMP Berikut	; lompat jika (50) = 30
Tengah:	DJNZ 50h, Putar	
	NOP	

➤ Perintah CALL

- Digunakan untuk memanggil sub routine layanan program

Mnemonik Instruksi

ACALL, sadd

LCALL, ladd

Operasi

Call sub routine alamat pendek add

Call sub routine alamat panjang add



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

BAGIAN 2

PETUNJUK KERJA

A. PETUNJUK PRE-TEST

1. Kerjakan soal dan latihan pre-test yang ada pada Modul 3 dengan mengisi tanda cek.
2. Isi dengan sebenarnya sesuai keadaan saudara
3. Jika saudara telah memiliki kompetensi seperti yang dinyatakan dalam pre test kerjakan soal-soal Post-Test
4. Jika saudara belum memiliki kompetensi seperti yang dinyatakan dalam pre test pelajari materi pada bagian satu dari Modul ini

B. PETUNJUK POST-TEST

I. UMUM

Dalam tugas ini, pada akhirnya saudara akan memiliki kompetensi terkait dengan :

1. Memahami Instruction set sebuah mikrokontroler secara tuntas detail sebagai dasar pemahaman pemilihan instruksi pada saat pengembangan program
2. Mengelompokkan Instruction set menjadi kelompok operasi aritmetika, operasi logika, transfer data, manipulasi variabel boolean, branching.
3. Memahami mode pengalamatan dan pemanfaatannya dalam instruction set

II. KHUSUS

1. Jika saudara belum memiliki data sheet mikrokontroler upayakan mencari baik di Internet atau sumber lainnya.



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

BAGIAN 3

PRE-TEST

Subkompetensi	Pernyataan	Saya memiliki kompetensi ini	
		Tidak	Ya
3. Mendeskrripsi kan Instruction Set Mikrokontroler AT89S51	Saya memahami Instruction set sebuah mikrokontroler AT89S51 secara tuntas detail sebagai dasar pemahaman pemilihan instruksi pada saat pengembangan program		
	Saya dapat mengelompokkan Instruction set mikrokontroler AT89S51 menjadi kelompok operasi aritmetika, operasi logika, transfer data, manipulasi variabel boolean, branching.		
	Saya memahami mode pengalamatan dalam instruction set mikrokontroler AT89S51		



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

BAGIAN 4

POST-TEST

A. Pilihlah salah satu jawaban yang saudara anggap paling benar

1. Mode pengalamatan yang paling sederhana untuk membangkitkan data pada destinasi disebut
 - a. Register addressing mode
 - b. Immediate addressing mode
 - c. Direct addressing mode
 - d. Indirect addressing mode
2. MOV A,#64h termasuk mode pengalamatan
 - a. Register addressing mode
 - b. Immediate addressing mode
 - c. Direct addressing mode
 - d. Indirect addressing mode
3. MOV A, R0 termasuk mode pengalamatan
 - a. Register addressing mode
 - b. Immediate addressing mode
 - c. Direct addressing mode
 - d. Indirect addressing mode
4. Perintah berikut termasuk perintah direct addressing mode kecuali
 - a. MOV A,80h
 - b. MOV A,P0
 - c. MOV #80h,A
 - d. MOV P0,A

Program berikut untuk kasus soal no 5 s/d 8

1. MOV A,#3Ah
2. DEC A
3. MOV R0,#15h
4. MOV 15h,#12h
5. INC @R0
6. DEC 15h
7. INC R0
8. MOV 16h, A
9. INC @R0
10. MOV DPTR,#12FFh
11. INC DPTR
12. DEC 83h

5. Sampai pada langkah ke 2 kondisi berikut yang benar
 - a. A = 3A
 - b. A = 39
 - c. A = 3B
 - d. A = 38



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

6. Sampai pada langkah 6 kondisi berikut yang benar adalah
 - a. R0 = 15h alamat 15h = 13 A = 38
 - b. R0 = 15h alamat 15h = 12 A = 3A
 - c. R0 = 15h alamat 15h = 13 A = 3B
 - d. R0 = 15h alamat 15h = 12 A = 39
7. Sampai pada langkah 9 kondisi berikut yang benar adalah
 - a. R0 = 15h alamat 16h = 38 A = 38
 - b. R0 = 16h alamat 16h = 39 A = 3A
 - c. R0 = 16h alamat 16h = 3A A = 39
 - d. R0 = 15h alamat 16h = 39 A = 39
8. Sampai pada langkah 12 kondisi berikut yang benar adalah
 - a. R0 = 16h alamat 16h = 3A A = 38 DPTR = 1200
 - b. R0 = 16h alamat 16h = 39 A = 3A DPTR = 12FF
 - c. R0 = 16h alamat 16h = 3A A = 39 DPTR = 1300
 - d. R0 = 15h alamat 16h = 39 A = 39 DPTR = 1213

Program berikut untuk kasus soal no 9 s/d 10

1. MOV A, #1Ch
 2. MOV R5, #0A1h
 3. ADD A, R5
 4. ADD A, R5
 5. ADC A, #10h
 6. ADC A, #10h
9. Sampai pada langkah 3 kondisi berikut yang benar adalah
 - a. A = BD ; Cy = 0; OV = 0 ; R5 = A1
 - b. A = 1C ; Cy = 0; OV = 0 ; R5 = A1
 - c. A = 1C ; Cy = 1; OV = 0 ; R5 = A1
 - d. A = BD ; Cy = 1; OV = 0 ; R5 = A1
 10. Sampai pada langkah 6 kondisi berikut yang benar adalah
 - a. A = 6F ; Cy = 0; OV = 0 ; R5 = A1
 - b. A = 5E ; Cy = 1; OV = 1 ; R5 = A1
 - c. A = 7F ; Cy = 0; OV = 0 ; R5 = A1
 - d. A = BD ; Cy = 0; OV = 0 ; R5 = A1



MODUL 3

Mendeskrripsikan Instruction Set Mikrokontroler

BAGIAN 5

KUNCI JAWABAN

A. Pilihan ganda

1. b
2. b
3. a
4. c
5. b
6. d
7. c
8. a
9. a
10. c