

MODUL PRAKTIKUM SISTEM TERTANAM



Robotic and Embedded System Laboratory Departemen Teknik Komputer Universitas Andalas



MODUL 2

ASSEMBLY AVR 1.1 CPMK

Mahasiswa mampu menggunakan register, set instruksi dan pemrograman AVR dengan bahasa Assembly pada mikrokontroler AVR. (CP-3) SUB-CPMK :

- a) Mengenal dan mencobakan pemrograman AVR dengan menggunakan bahasa Assembly.
- b) Mencobakan register dalam mikrokontroler AVR.
- c) Mampu menggunakan set instruksi dalam mikrokontroler AVR.
- d) Mampu membedakan register PIN, register DDR dan register PORT pada mikrokontroler AVR.

1.2 ALAT DAN BAHAN

- 1. Laptop
- 2. Arduino Uno
- 3. LED
- 4. Push Button
- 5. Resistor
- 6. Breadboard
- 7. Aplikasi Arduino IDE



1.3 LANDASAN TEORI

Mikrokontroler AVR ATMega328p



Gambar 3.1 ATMega328p [1]

Mikrokontroler AVR (*Alf and Vegaard's Risc Processor*) ATMega328p adalah seri mikrokontroler *Complementary Metal Oxide Semiconductor* (CMOS) 8-bit buatan Atmel berbasis arsitektur RISC (*Reduced Instruction Set Computer*). ATMega328p mempunyai *throughput* mendekati 1 *Millions Instruction Per Second* (MIPS) per MHz, sehingga membuat konsumsi daya menjadi rendah terhadap kecepatan proses eksekusi perintah. [1]

AVR Instruction Set

1. LDI (Load Immediate)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
LDI	Rd, K	Load immediate	Rd ← K	None	1

Gambar 3.2 Instruksi load immediate [2]

LDI (Load Immediate) termasuk kedalam jenis instruksi pada transfer data. LDI digunakan untuk meng-*copy* nilai konstanta 8-bit (1 byte) ke dalam suatu register (R16 – R31).

2. OUT



Gambar 3.3 Instruksi out [2]

Out port berfungsi untuk menulis data dari register ke I/O port.



3. BRNE

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRNE	k	Branch if not equal	if (Z = 0) then PC \leftarrow PC + k + 1	None	1/2

Gambar 3.4 Instruksi branch if not equal [2]

Branch If Not Equal menguji nilai dari flag Z (Zero). Apabila bit tersebut *clear* (0) melompat ke label tertentu.

4. RCALL

Mnemonics	Operands	Description	Operation	Flags	#Clocks
RCALL	k	Relative subroutine call	PC ← PC + k + 1	None	3

Gambar 3.5 Instruksi *relative subroutine call* [2] *Relative Subroutine Call* berfungsi untuk memanggil subrutin.

5. RET

Mnemonics	Operands	Description	Operation	Flags	#Clocks
RET		Subroutine return	PC ← STACK	None	4

Gambar 3.6 Instruksi subroutine return [2] Subroutine Return berfungsi untuk keluar dari subroutine.

General Purpose Register



Gambar 3.7 General purpose register AVR [2]

General Purpose Register adalah register dalam CPU yang dimanfaatkan untuk berbagai macam kepentingan. Misalnya untuk indeks, alamat, perhitungan dan operasi logika. Pada AVR terdapat 32 alamat register dan yang bisa digunakan pada *assembly* yaitu pada R16-R25.



Arduino UNO



Gambar 3.8 Arduino Uno [3]

Arduino Uno adalah *board* mikrokontroler yang berbasis pada mikrokontroler AVR ATmega328 [3]. Memiliki 14 pin *input* dari *output* digital dimana 6 pin input tersebut dapat digunakan sebagai *output* PWM dan 6 pin *input* analog, 16 MHz osilator kristal, koneksi USB, *jack power*, ICSP *header*, dan tombol reset.

Direct Port Register Addressing

Metode ini juga disebut sebagai *Direct Port Register Addressing*. Dalam teknik ini menggunakan metodologi pemrograman mikrokontroler AVR. Jadi untuk memahami metode ini, dapat dilihat dasar-dasar pemrograman AVR. Ingat bahwa perintah C & C++ digunakan dalam AVR. *Pinout* dari ATMega328p diberikan di bawah ini.



Gambar 3.9 ATMega328p pin mapping [4]



Arduino UNO menggunakan mikrokontroler ATMega328p yang mana memiliki 3 PORT I/O yaitu: PORTB, PORTC & PORTD.

1. PORTB (8-pin)	: PB7 hingga PB0).	
2. PORTC (6-pin)	: PC5 hingga PC0).	
3. PORTD (8-pin)	: PD7 hingga PD0).	
		Digital pins 0 - PD0 1 - PD1 2 - PD2 3 - PD3 4 - PD4 5 - PD5 6 - PD6 7 - PD7	Digital pins 8 – PB0 9 – PB1 10 – PB2 11 – PB3 12 – PB4 13 – PB5 Analog pins A0 – PC0 A1 – PC1 A2 – PC2 A3 – PC3
	PC0 PC0 PC2 PC3 PC3		A4 – PC4 A5 – PC5

Gambar 3.10 *Pinout* pada Arduino Uno [4]

Pada gambar 3.10 diatas, ditampilkan mengenai letak *port* mikrokontroler ATMega328p terhadap pin pada Arduino Uno. Dapat disimpulkan bahwa PortB berada pada pin 8-13 pada Arduino Uno. PortC berada pada pin A1-A5 pada Arduino Uno. Kemudian untuk PortD berada pada pin 0-7 pada Arduino Uno.

Register Addressing

ki m	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Name	Address
	PORTD0	PORTD1	PORTD2	PORTD3	PORTD4	PORTD5	PORTD6	PORTD7	PORTD	0x0B (0x2B)
C	DDD0	DDD1	DDD2	DDD3	DDD4	DDD5	DDD6	DDD7	DDRD	0x0A (0x2A)
	PIND0	PIND1	PIND2	PIND3	PIND4	PIND5	PIND6	PIND7	PIND	0x09 (0x29)
	PORTC0	PORTC1	PORTC2	PORTC3	PORTC4	PORTC5	PORTC6	-	PORTC	0x08 (0x28)
C	DDC0	DDC1	DDC2	DDC3	DDC4	DDC5	DDC6	-	DDRC	0x07 (0x27)
	PINC0	PINC1	PINC2	PINC3	PINC4	PINC5	PINC6	-	PINC	0x06 (0x26)
	PORTB0	PORTB1	PORTB2	PORTB3	PORTB4	PORTB5	PORTB6	PORTB7	PORTB	0x05 (0x25)
α	DDB0	DDB1	DDB2	DDB3	DDB4	DDB5	DDB6	DDB7	DDRB	0x04 (0x24)
	PINB0	PINB1	PINB2	PINB3	PINB4	PINB5	PINB6	PINB7	PINB	0x03 (0x23)

Tabel 3.1 Register addressing AVR [2]

Sesuai dengan konsep mikrokontroller ATMega328p, register yang didukung hanya 8bit. Untuk penempatan register dapat dilihat pada tabel 3.1 diatas. Dengan 8bit register ini kita mampu mengatur setidaknya 8 pin pada Arduino Uno untuk setiap Port (PortB, PortC, PortD).



Arduino Uno Register Control



Gambar 3.11 PORTX device register [2]

Pada gambar 3.11 diatas, ditampilkan gambaran rangkaian pada pin Arduino Uno. Setiap pin diatur dengan 3 register yaitu register PORT, DDR dan PIN.

a. Register PORT

Register PORT berfungsi untuk mengatur *high* atau *low* sebuah pin. *High* dan *low* diatur menggunakan nilai pada isi register.

b. Register DDR

Register DDR (*Data Direct Register*) berfungsi untuk mengatur *input / output* sebuah pin. DDR bertugas untuk mengatur *enable* pada rangkaian. Saat nilai *enable* bernilai 1 (*High*), maka nilai dari register PORT akan diteruskan ke *out* sehingga rangkaian pada pin bernilai *high*.

c. Register PIN

Register PIN berfungsi untuk menyimpan nilai input dari pin Arduino Uno.

Contoh sederhana saat ingin membuat pin 13 pada Arduino Uno menjadi *high*, nilai yang ada pada register untuk mengatur pin 13 harus diberikan nilai 1(*high*).



Gambar 3.12 Nilai register PORTB saat kondisi low [2]



Sebelumnya cari tahu terlebih dahulu digital pin 13 pada Arduino Uno berada pada PB berapa pada mikroprosessor ATMega328p. Sesuai dengan *datasheet*, pin digital 13 berada pada PB5. Setelah mengetahui posisi pin pada ATMega328p, berikan nilai pada register PORTB dengan nilai 1.



Gambar 3.13 Nilai register PORTB saat kondisi high [2]

Konsep Delay

Dalam *datasheet*-nya, mikrokontroler ATMega328p memiliki *clock speed* sebesar 16MHz. ATMega328p mempunyai *throughput* mendekati 1 *Millions Instruction Per Second* (MIPS) per MHz. Artinya dalam 1 detik terdapat 16 juta instruksi yang dapat diproses. Maka dari itu, untuk membuat Arduino Uno *delay* selama 1 detik, dibutuhkan perulangan yang mengeksekusi 16 juta instruksi.

Dalam hal ini dibuat 3 perulangan karena nilai maksimal dalam 8bit hanya 255.

Delay Delay Count	(Sec) (Sec)	= 255 x 255 x count x 3 x (1 / 16 = 0.0122 x count = 1 / 0.0122	.000.000)
Count		= 82	
• • •			
delay: LDI 11: LDI 12: MOV 5: I3: DEC 6 BRNE 7 DEC 8 9 18 18 10 11 11 11 12 11 12 12 13 14 15 16 17 18 11 11 12	R20, 255 R21, 255 R22, 82 R22 13 R21 12 R20 11	<pre>;3-level nested loop subroutine ;outer loop ;mid loop ;inner loop counter to give 0.5s delay ;decrement inner loop ;loop if not zero ;decrement mid loop ;loop if not zero ;decrement outer loop ;loop if not zero ;return to caller</pre>	

Gambar 3.14 Program assembly delay



Dari gambar 3.14 diatas terdapat *source code* bahasa *assembly* untuk *delay* selama 1 detik. Nilai *count* yang telah didapatkan sebelumnya dimasukkan pada register R22. Kemudian dilakukan pengurangan secara bertahap menggunakan instruksi DEC dan dideteksi menggunakan instruksi BRNE apakah register tersebut seudah bernilai 0 atau belum. Jika belum akan dilakukan pengurangan secara terus menerus.

Dari perhitungan diatas, dapat dicari nilai untuk menentukan delay pada count.

Delay (Sec)	Count
0.25	20
0.50	41
1	82
3.11	255

Tabel 3.2 Nilai count pada masing-masing delay

Assembly pada Arduino Uno

a. pinMode pada Arduino Uno

pinMode (1, Input)
pinMode (2, Output)

Gambar 3.15 Pin mode

Dari gambar 3.15 diatas dapat diketahui bahwa struktur dari perintah pinMode yaitu:

pinMode (Pin ke-Arduino, Input/Output)

Pin Mode pada Arduino Uno digunakan untuk mengatur input atau *output* pada sebuah pin. Input atau *output* pada *assembly* diatur menggunakan register DDR (*Data Direct Register*).

Contoh: Mengatur pin 4 Arduino Uno sebagai output

- Mengetahui bahwa pin 4 berada pada PD3.
- Jika pin 4 dijadikan *output*, maka nilai pada register DDRD harus 1.



Register DDRD

Pin Arduino	7	6	5	4	3	2	1	0
PortD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Bit	0	0	0	1	0	0	0	0

Gambar 3.16 Register DDRD

Jadi bahasa assembly untuk menjadikan pin 4 sebagai output yaitu

LDI	R16,	0b00010000	;Load	nilai	binne	er ke	e re	egister 1	16
OUT	DDRD,	R16	;Pinda	ahkan	nilai	R16	ke	register	DDRD

Gambar 3.17 Source code pinMode

b. digitalWrite pada Arduino Uno

```
digitalWrite(1, High)
digitalWrite(2, Low)
```

Gambar 3.18 *Digital write*

Dari gambar 3.18 diatas dapat diketahui bahwa struktur dari perintah digitalWrite yaitu:

```
digitalWrite (Pin ke-Arduino, High/Low)
```

Digital write pada Arduino Uno digunakan untuk mengatur *high* atau *low* pada sebuah pin. *High* atau *low* pada *assembly* diatur menggunakan register Port.

Contoh: Mengatur pin 4 Arduino Uno bernilai high.

- Mengetahui bahwa *pin* 4 berada pada PD3.
- Jika pin 4 dijadikan *output*, maka nilai pada register PORTD harus *high*.

Register PORTD

Pin Arduino	7	6	5	4	3	2	1	0
PortD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Bit	0	0	0	1	0	0	0	0

Gambar 3.19 Register PORTD

Bahasa assembly untuk menjadikan pin 4 bernilai high yaitu:

LDI R17, 0b00010000 ;Load nilai binner ke register 17 OUT PORTD, R17 ;Pindahkan nilai R17 ke register PORTD

Gambar 3.20 Source code digitalWrite



1.4 TUGAS PENDAHULUAN

- 1. Jelaskan langkah-langkah instalasi aplikasi Arduino IDE beserta *screenshot* instalasi pribadi!
- 2. Jelaskan perbedaan antara assembly pada ARM dengan assembly pada AVR!
- 3. Jelaskan cara kerja dan fungsi dari masing-masing register PIN, register DDR dan register PORT pada ATMega 328p!
- 4. Jelaskan konsep delay pada mikrokontroler AVR!
- 5. Jelaskan assembly dari program Arduino berikut!
 - LDI R16,0b11111111 OUT DDRX, R16
 - IN R22, PINX OUT PORTX, R22
 - PinMode (5, Output) digitalWrite (5, High)

1.5 PERCOBAAN

A. PROSEDUR PERCOBAAN

- 1. Hidupkan laptop
- 2. Buka aplikasi Arduino IDE pada laptop
- 3. Buat direktori baru dengan nama Sister_III pada file explorer
- 4. Persiapkan Arduino Uno
- 5. Hubungkan kabel USB-B pada Arduino Uno ke USB laptop
- 6. Persiapkan komponen yang dibutuhkan saat praktikum
- 7. Ikuti langkah-langkah percobaan praktikum

B. PERCOBAAN 1 : Mengatur *Output* Arduino Uno dalam *Assembly*

- 1. Rangkailah LED pada pin Arduino Uno dengan ketentuan:
 - a. Kelompok 1: pin 4
 - b. Kelompok 2: pin 11
 - c. Kelompok 3: pin A5
 - d. Kelompok 4: pin 9





Gambar 3.21 Contoh rangkaian LED

- Buat sebuah file baru pada Arduino IDE kemudian simpan pada direktori Sister_III dengan format nama file: Nama_Nobp_Percobaan1
- 3. Ketikan program .ino yang diberikan saat praktikum.
- Selanjutnya buat file baru menggunakan notepad. Ketikkan program assembly yang diberikan saat praktikum. Selanjutnya simpan dengan nama file Nama_Nobp_Percobaan1.S (Perhatikan ekstensi S menggunakan huruf kapital) pada folder Nama Nobp Percobaan1
- 5. Lakukan *screenshot* pada program yang dibuat.
- 6. Ambil gambar rangkaian yang telah dijalankan.
- 7. Buat analisa jalannya program pada laporan.

C. PERCOBAAN 2: Mengatur Input Arduino Uno dalam Assembly

- 1. Rangkaian pada percobaan 2 sama dengan rangkaian percobaan 1. Buatlah waktu delay untuk rangkaian dengan ketentuan berikut
 - a. Kelompok 1: 2 detik
 - b. Kelompok 2: 4 detik
 - c. Kelompok 3: 3 detik
 - d. Kelompok 4: 5 detik
- Buat sebuah file baru pada Arduino IDE kemudian simpan pada direktori Sister_III dengan format nama file: Nama_Nobp_Percobaan2
- 3. Ketikan program .ino yang diberikan saat praktikum.



- Selanjutnya buat file baru menggunakan notepad. Ketikkan program assembly yang diberikan saat praktikum. Selanjutnya simpan dengan nama file Nama_Nobp_Percobaan2.S (Perhatikan ekstensi S menggunakan huruf kapital) pada folder Nama_Nobp_Percobaan2
- 5. Lakukan *screenshot* pada program yang dibuat.
- 6. Ambil gambar rangkaian yang telah dijalankan.
- 7. Buat analisa jalannya program pada laporan.

D. PERCOBAAN 3: Mengatur Delay Arduino Uno dalam Assembly

- 1. Rangkailah *push button* dengan ketentuan:
 - a. Kelompok 1: PORT D
 - b. Kelompok 2: PORT B
 - c. Kelompok 3: PORT C
 - d. Kelompok 4: PORT B



Gambar 3.22 Contoh rangkaian push button

- Rangkailah LED dengan ketentuan lokasi Port ATMega328p sesuai dengan Arduino Uno dengan ketentuan:
 - a. Kelompok 1: PORT B
 - b. Kelompok 2: PORT D
 - c. Kelompok 3: PORT B
 - d. Kelompok 4: PORT D



- Buat sebuah file baru pada Arduino IDE kemudian simpan pada direktori Sister_III dengan format nama file: Nama_Nobp_Percobaan3
- 4. Ketikan program .ino yang diberikan saat praktikum.
- Selanjutnya buat file baru menggunakan notepad. Ketikkan program assembly yang diberikan saat praktikum. Selanjutnya simpan dengan nama file Nama_Nobp_Percobaan3.S (Perhatikan ekstensi S menggunakan huruf kapital) pada folder Nama_Nobp_Percobaan3
- 6. Lakukan *screenshot* pada program yang dibuat.
- 7. Ambil gambar rangkaian yang telah dijalankan.
- 8. Buat analisa jalannya program pada laporan.



1.6 DAFTAR PUSTAKA

- [1] Author, "Apa Itu Mikrokontroller AVR ATMEGA328P ? Dan Bagaimana konfigurasi pinnya?," Inforbes.com, 12-Apr-2022. [Online]. Available: <u>http://www.inforbes.com/2017/07/apa-itu-mikrokontroller-avr-atmega328p.html</u>. [Accessed: 05-Nov-2023].
- [2] ATmega328P Datasheet, 7810D-AVR Atmel, 2015. [Online]. Available: <u>https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-</u> <u>AutomotiveMicrocontrollers-ATmega328P Datasheet.pdf</u>
- [3] R. B. Pradana, "SISTEM KEAMANAN RUMAH DENGAN PEMBERITAHUAN MELALUI SMS BERBASIS ARDUINO," thesis, 2017.
- [4] Author, "Port Register Control," Electronoobs, 01-Nov-2020. [Online]. Available: https://electronoobs.io/tutorial/201. [Accessed: 05-Nov-2023].