



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

PRAKTIKUM MIKROPROSESOR DAN MIKROKONTROLER

SEMESTER : GANJIL

TAHUN : 2025/2026

BAB V

JUDUL BAB : LIQUID CRYSTAL DISPLAY
NAMA : Muh. Fadhlir Rudiyanto
NIM : 245150300111025
TANGGAL PELAKSANAAN : 04 November 2025
ASISTEN : - ALVIN RIFKY WAHYUDI
 - DOS HANSEL SIHOMBING
KOORDINATOR ASISTEN : DOS HANSEL SIHOMBING
DOSEN PENGAMPU : Dr. Ir. HERU NURWARSITO, M.Kom

Pengesahan

Tanggal:

Asisten



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

BAB : LIQUID CRYSTAL DISPLAY
NAMA : Muh. Fadhlir Rudiyanto
NIM : 245150300111025
TANGGAL : 04 November 2025
ASISTEN : - ALVIN RIFKY WAHYUDI
- DOS HANSEL SIHOMBING

LAPORAN

PERCOBAAN 5.1

Jalankan program di bawah ini dan lakukan analisis!

Program5.1

```
1  #define F_CPU 1000000UL
2  #include <avr/io.h>
3  #include <util/delay.h>
4
5  #define LCD_Dir DDRB
6  #define LCD_Port PORTB
7  #define RS PB0
8  #define EN PB1
9
10 void LCD_Command( unsigned char cmnd )
11 {
12     // letakkan 4 bit MSB pada cmnd ke LCD_Port[7:4]:
13     LCD_Port = (LCD_Port & 0x0F) | (cmnd & 0xF0);
14     // (X & 0x0F)      : 4 bit MSB di-0-kan
15     // (cmnd & 0xF0)    : mengabaikan 4 bit LSB di "cmnd"
16     // 0 | X = X      : sifat identitas OR
17     // andai tidak di-0-kan dulu,
18     // misal awalnya 1, di-OR-kan dengan suatu nilai 0 di cmnd, menjadi 1 (tidak sesuai cmnd yang nilainya
19
20     LCD_Port &= ~(1<<RS);
21     // di-AND-kan dengan 11111110 agar bit RS = 0
22
23     LCD_Port |= (1<<EN);
24     _delay_us(1);
25     LCD_Port &= ~(1<<EN);
26     _delay_us(200);
27
28     // letakkan 4 bit LSB pada cmnd ke LCD_Port[7:4]:
29     LCD_Port = (LCD_Port & 0x0F) | (cmnd << 4);
30     // (cmnd << 4) : cmnd diletakkan mulai dari bit 4
31     // agar 4 bit LSBnya menempati slot 4 bit MSB
32
33     LCD_Port |= (1<<EN);
34     _delay_us(1);
35     LCD_Port &= ~(1<<EN);
36     _delay_ms(2);
37 }
38
39 void LCD_Char( unsigned char data )
40 {
41     LCD_Port = (LCD_Port & 0x0F) | (data & 0xF0);
42
43     LCD_Port |= (1<<RS);
44     // di-OR-kan dengan 00000001 agar bit RS = 1
45
46     LCD_Port|= (1<<EN);
47     _delay_us(1);
48     LCD_Port &= ~(1<<EN);
49     _delay_us(200);
50
51     LCD_Port = (LCD_Port & 0x0F) | (data << 4);
52
53     LCD_Port |= (1<<EN);
54     _delay_us(1);
55     LCD_Port &= ~(1<<EN);
56     _delay_ms(2);
57 }
58 }
```

```

59 void LCD_Init (void)
60 {
61     LCD_Dir = 0xFF;
62     _delay_ms(20);
63     LCD_Command(0x02); //return home mengembalikan ke tampilan bagian awal
64     LCD_Command(0x28); //2 baris, 5x8 Matriks, mode 4bit
65     LCD_Command(0x0c); //Menyalakan display, Cursor mati
66     LCD_Command(0x06); //Geser kursor ke kanan
67     LCD_Command(0x01); //clrscreen (bersihin layar)
68     _delay_ms(2);
69 }
70
71 void LCD_String (char *str)
72 {
73     int i;
74     for(i = 0; str[i] != 0; i++)
75     {
76         LCD_Char( str[i] );
77     }
78 }
79
80 void LCD_String_xy (char row, char pos, char *str)
81 {
82     if (row == 0 && pos < 8)
83         LCD_Command((pos & 0x0F) | 0x80);
84     else if (row == 1 && pos < 8)
85         LCD_Command((pos & 0x0F) | 0xC0);
86     LCD_String(str);
87 }
88
89 void LCD_Clear() {
90     LCD_Command (0x01);
91     _delay_ms(2);
92     LCD_Command (0x80); //Meletakkan kursor di baris pertama
93 }
94
95 int main()
96 {
97     LCD_Init();
98
99     LCD_String("Sismik");
100    LCD_Command(0xC0); //Meletakkan kursor di baris kedua
101    LCD_String("Prak");
102    _delay_ms(2000);
103    LCD_Clear();
104
105    char A[8]="BAB";
106    char B[8]="LCD";
107    LCD_String_xy(0,1,A);
108    LCD_String_xy(1,4,B);
109    while(1);
110 }

```

Analisis 5.1

1. Jalankan program pada mikrokontroler lalu amati dan ceritakan apa yang muncul pada LCD!

Jawab:

Saat Program 5.1 dijalankan, urutan tampilan di LCD adalah sebagai berikut:

- Pertama, layar langsung menampilkan tulisan Sismik di baris atas (baris 0) dan Prak di baris bawah (baris 1).
- Tampilan ini bertahan selama 2 detik (2000 ms).
- Setelah 2 detik, layar langsung bersih (clear).
- Kemudian, langsung muncul tampilan baru:
 - a. Baris atas menampilkan teks BAB (dari variabel A), dimulai dari kolom ke-1 (bukan kolom ke-0).
 - b. Baris bawah menampilkan teks LCD (dari variabel B), dimulai dari kolom ke-4.
- Tampilan teks BAB dan LCD ini bertahan seterusnya karena program masuk ke while(1);.

2. Ubah nilai argumen kedua pada pemanggilan LCD_String_xy() di baris ke-98 menjadi 3, 2, atau 1! Jelaskan perbedaan apa yang terjadi!

Jawab:

Argumen kedua pada fungsi LCD_String_xy(char row, char pos, char *str) berfungsi untuk mengatur posisi kolom (koordinat X) tempat teks akan mulai ditampilkan.

- Pada kode aslinya, LCD_String_xy(1, 4, B) artinya menampilkan teks LCD di baris 1, mulai dari kolom ke-4.
- Jika diubah menjadi 3 (LCD_String_xy(1, 3, B)), maka teks LCD akan bergeser ke kiri dan mulai tampil dari kolom ke-3.
- Jika diubah menjadi 2 atau 1, teks akan semakin bergeser ke kiri.

Jadi, mengubah nilai argumen kedua ini akan mengubah posisi horizontal (kiri-kanan) dari teks yang ditampilkan.

3. Jelaskan kegunaan bagian-bagian dari kode program!

Jawab:

- #define RS PB0 dan #define EN PB1 adalah preprocessor yang memberi nama alias. RS untuk pin PB0 dan EN untuk pin PB1. Ini membuat kode lebih mudah dibaca daripada harus menghafal nomor pin.
- void LCD_Command(unsigned char cmd) adalah fungsi untuk mengirim perintah ke LCD. Fungsi ini mengatur pin RS ke nilai 0 (logika rendah) yang menandakan bahwa data yang dikirim adalah instruksi (seperti 'clear screen' atau 'pindah kursor'), bukan karakter.
- void LCD_Char(unsigned char data) adalah fungsi untuk mengirim data karakter ke LCD. Fungsi ini mengatur pin RS ke nilai 1 (logika tinggi) yang menandakan data yang dikirim adalah karakter yang ingin ditampilkan di layar.
- void LCD_Init(void) adalah fungsi inisialisasi atau "pemanasan". Fungsi ini dijalankan pertama kali untuk mengatur mode operasi LCD, seperti:
 - Mengatur mode 4-bit (0x28)
 - Menyalakan display dan mematikan kursor (0x0c)
 - Mengatur kursor agar bergeser ke kanan setiap kali menulis karakter (0x06)
 - Membersihkan layar (0x01)
- LCD_Command(0xC0) adalah pemanggilan fungsi perintah. Nilai 0xC0 adalah kode heksadesimal yang spesifik untuk memberi perintah "pindahkan kursor ke awal baris kedua".
- LCD_String_xy(0, 1, A) adalah pemanggilan fungsi untuk menulis string. Artinya: "tampilkan string yang ada di variabel A pada baris 0 dan mulai dari kolom 1

PERCOBAAN 5.2

Jalankan program di bawah ini dan lakukan analisis!

Program5.2

```
1  #define F_CPU 1000000UL
2  #include <avr/io.h>
3  #include <util/delay.h>
4
5  #define LCD_Dir DDRB
6  #define LCD_Port PORTB
7  #define RS PB0
8  #define EN PB1
9
10 void LCD_Command( unsigned char cmnd )
11 {
12     LCD_Port = (LCD_Port & 0x0F) | (cmnd & 0xF0);
13
14     LCD_Port &= ~(1<<RS);
15
16     LCD_Port |= (1<<EN);
17     _delay_us(1);
18     LCD_Port &= ~(1<<EN);
19     _delay_us(200);
20
21     LCD_Port = (LCD_Port & 0x0F) | (cmnd << 4);
22
23     LCD_Port |= (1<<EN);
24     _delay_us(1);
25     LCD_Port &= ~(1<<EN);
26     _delay_ms(2);
27 }
28
29 void LCD_Char( unsigned char data )
30 {
31     LCD_Port = (LCD_Port & 0x0F) | (data & 0xF0);
32
33     LCD_Port |= (1<<RS);
34
35     LCD_Port |= (1<<EN);
36     _delay_us(1);
37     LCD_Port &= ~(1<<EN);
38     _delay_us(200);
39
40     LCD_Port = (LCD_Port & 0x0F) | (data << 4);
41
42     LCD_Port |= (1<<EN);
43     _delay_us(1);
44     LCD_Port &= ~(1<<EN);
45     _delay_ms(2);
46 }
47
48 void LCD_Init( void )
49 {
50     LCD_Dir = 0xFF;
51     _delay_ms(20);
52     LCD_Command(0x02); //return home mengembalikan ke tampilan bagian awal
```

```

53     LCD_Command(0x28); //2 baris, 5x8 Matriks, mode 4bit
54     LCD_Command(0x0c); //Menyalakan display, Cursor mati
55     LCD_Command(0x06); //Geser kursor ke kanan
56     LCD_Command(0x01); //clrscreen (bersihin layar)
57     _delay_ms(2);
58 }
59
60 void LCD_String (char *str)
61 {
62     int i;
63     for(i = 0; str[i] != 0; i++)
64     {
65         LCD_Char( str[i] );
66     }
67 }
68
69 void LCD_String_xy (char row, char pos, char *str)
70 {
71     if (row == 0 && pos < 8)
72         LCD_Command((pos & 0x0F) | 0x80);
73     else if (row == 1 && pos < 8)
74         LCD_Command((pos & 0x0F) | 0xC0);
75     LCD_String(str);
76 }
77
78 void LCD_Clear() {
79     LCD_Command (0x01);
80     _delay_ms(2);
81     LCD_Command (0x80); //Meletakkan kursor di baris pertama
82 }
83
84 int main()
85 {
86     int shift, i;
87     LCD_Init();
88     LCD_String("Text");
89     _delay_ms(2000);
90     shift = 4;
91     while(1)
92     {
93         for(i = 0; i < shift; i++)
94         {
95             LCD_Command(0x1c); //Geser seluruh tampilan ke kanan
96             _delay_ms(500);
97         }
98     }
99
100     return 0;
101 }

```

Analisis 5.2

1. Jalankan program pada mikrokontroler lalu amati dan ceritakan apa yang muncul pada LCD!

Jawab:

Saat program dijalankan, LCD akan menampilkan tulisan Text di sebelah kiri.

Tulisan ini akan diam selama 2 detik. Setelah itu, tulisan akan mulai bergeser ke kanan 4 langkah, di mana setiap langkah diberi jeda 0.5 detik. Karena proses ini berada di dalam loop abadi while(1), maka setelah 4 langkah selesai, program akan mengulang lagi pergeseran 4 langkah ke kanan. Akibatnya, tulisan Text akan terus bergerak ke kanan secara bertahap hingga hilang dari layar.

2. Ceritakan apa yang terjadi jika baris ke-95 "LCD_Command(0x1c)" diubah menjadi "LCD_Command(0x18)"!

Jawab:

Jika baris ke-95 diubah menjadi LCD_Command(0x18), maka hasilnya akan berbalik. Saat program dijalankan, LCD akan menampilkan tulisan Text di sebelah kiri dan diam selama 2 detik. Setelah itu, program akan mengeksekusi perintah geser ke kiri (0x18) sebanyak 4 kali, dengan jeda 0.5 detik setiap langkahnya. Ini akan menyebabkan tulisan Text bergeser ke kiri hingga hilang dari sisi kiri layar. Karena loop while(1) terus mengulang proses geser kiri ini pada teks yang sudah hilang, layar LCD akan tetap kosong setelahnya.

3. Jelaskan kegunaan baris-baris kode program di dalam fungsi main()!

Jawab:

- LCD_Init(); Memanggil fungsi LCD_Init untuk melakukan inisialisasi (persiapan awal) pada LCD, seperti mengatur mode 4-bit, menyalakan display, dan membersihkan layar.
- LCD_String("Text"); Memanggil fungsi LCD_String untuk menampilkan tulisan Text ke layar LCD.
- _delay_ms(2000); Memberikan jeda atau menunda program selama 2000 milidetik (2 detik). Ini bertujuan agar tulisan Text terlihat diam dulu di layar sebelum bergeser.
- shift = 4; Memberi nilai 4 ke dalam variabel shift. Nilai ini akan digunakan untuk menentukan batas perulangan pada for loop.
- while(1) Memulai sebuah loop tak terbatas (infinite loop). Semua kode di dalam kurung kurawal ini akan diulang terus-menerus selamanya.
- for(i = 0; i < shift; i++) Memulai sebuah for loop yang akan berulang selama nilai i kurang dari nilai shift (yaitu 4 kali). Loop ini akan berjalan untuk i=0, i=1, i=2, dan i=3.
- LCD_Command(0x1c); Perintah di dalam for loop yang memanggil fungsi LCD_Command untuk mengirimkan instruksi 0x1c (geser seluruh tampilan ke kanan) ke LCD.
- _delay_ms(500); Memberikan jeda 500 milidetik (0.5 detik) setiap kali setelah perintah geser kanan dieksekusi. Ini membuat pergeseran terlihat perlahan.
- return 0; Mengakhiri fungsi main dan mengembalikan nilai 0 ke sistem. Namun, karena program terjebak di dalam loop while(1), baris ini tidak akan pernah dieksekusi.



BAB : LIQUID CRYSTAL DISPLAY
NAMA : Muh. Fadhlir Rudiyanto
NIM : 245150300111025
TANGGAL : 04 November 2025
ASISTEN : - ALVIN RIFKY WAHYUDI
- DOS HANSEL SIHOMBING

TUGAS

5.5 Tugas

Tuliskan nama salah satu anggota kelompokmu pada layar LCD (Min: 4 – Max: 5 karakter) di salah satu baris, tulisan tersebut akan shifting dari kiri layar sampai ujung kanan dan kemudian shifting balik ke ujung kiri, lalu shifting balik ke kanan, begitu seterusnya!

Jawab:



Program5.5

```
1  #define F_CPU 1000000UL
2  #include <avr/io.h>
3  #include <util/delay.h>
4
5  #define LCD_Dir DDRB
6  #define LCD_Port PORTB
7  #define RS PB0
8  #define EN PB1
9
10 void LCD_Command( unsigned char cmnd )
11 {
12     LCD_Port = (LCD_Port & 0x0F) | (cmnd & 0xF0);
13
14     LCD_Port &= ~(1<<RS);
15
16     LCD_Port |= (1<<EN);
17     _delay_us(1);
18     LCD_Port &= ~(1<<EN);
19     _delay_us(200);
20
21     LCD_Port = (LCD_Port & 0x0F) | (cmnd << 4);
22
23     LCD_Port |= (1<<EN);
24     _delay_us(1);
25     LCD_Port &= ~(1<<EN);
26     _delay_ms(2);
27 }
```

```

28
29 void LCD_Char( unsigned char data )
30 {
31     LCD_Port = (LCD_Port & 0x0F) | (data & 0xF0);
32
33     LCD_Port |= (1<<RS);
34
35     LCD_Port|= (1<<EN);
36     _delay_us(1);
37     LCD_Port &= ~(1<<EN);
38     _delay_us(200);
39
40     LCD_Port = (LCD_Port & 0x0F) | (data << 4);
41
42     LCD_Port |= (1<<EN);
43     _delay_us(1);
44     LCD_Port &= ~(1<<EN);
45     _delay_ms(2);
46 }
47
48 void LCD_Init (void)
49 {
50     LCD_Dir = 0xFF;
51     _delay_ms(20);
52     LCD_Command(0x02); //return home mengembalikan ke tampilan bagian awal
53     LCD_Command(0x28); //2 baris, 5x8 Matriks, mode 4bit
54     LCD_Command(0x0c); //Menyalakan display, Cursor mati
55     LCD_Command(0x06); //Geser kursor ke kanan
56     LCD_Command(0x01); //clrscreen (bersihin layar)
57     _delay_ms(2);
58 }
59
60 void LCD_String (char *str)
61 {
62     int i;
63     for(i = 0; str[i] != 0; i++)
64     {
65         LCD_Char( str[i] );
66     }
67 }
68
69 void LCD_String_xy (char row, char pos, char *str)
70 {
71     if (row == 0 && pos<8)
72         LCD_Command((pos & 0x0F) | 0x80);
73     else if (row == 1 && pos < 8)
74         LCD_Command((pos & 0x0F) | 0xC0);
75     LCD_String(str);
76 }
77
78 void LCD_Clear() {
79     LCD_Command (0x01);
80     _delay_ms(2);
81     LCD_Command (0x80); //Meletakkan kursor di baris pertama
82 }
83
84 int main()

```

```

85  {
86      int shift, i;
87      LCD_Init();
88      LCD_String("BIRU");
89      _delay_ms(2000);
90      shift = 4;
91      while(1)
92      {
93          for(i = 0; i < shift; i++)
94          {
95              LCD_Command(0x1c); //Geser seluruh tampilan ke kanan
96              _delay_ms(500);
97          }
98          for(i = 0; i < shift; i++)
99          {
100             LCD_Command(0x18); //Geser seluruh tampilan ke kanan
101             _delay_ms(500);
102         }
103     }
104     return 0;
105 }
106

```

5.6 Kesimpulan

Setelah mengikuti praktikum bab Liquid Crystal Display (LCD), berikan kesimpulan Anda dari setiap percobaan dan analisis yang telah Anda kerjakan!

Jawab:

- Liquid Crystal Display (LCD) adalah modul yang bisa diprogram untuk menampilkan karakter. LCD yang dipakai di modul DT-COMBO AVR-51 adalah tipe 8x2 (8 karakter, 2 baris).
- Komunikasi dengan LCD (yang standar HD44780) berjalan lewat beberapa pin kontrol, utamanya RS (Register Select) dan E (Enable), plus pin data (D0-D7). Di praktikum ini, kita pakai mode 4-bit, yang cuma memakai pin D4-D7.
- Ada dua jenis pengiriman data ke LCD:
 1. Perintah (Command): Tugasnya mengatur operasi LCD (misalnya clear screen, atur posisi kursor, mode operasi). Caranya pakai LCD_Command() dengan pin RS di-set '0'.
 2. Data (Character): Tugasnya mengirim karakter yang mau ditampilkan. Caranya pakai LCD_Char() dengan pin RS di-set '1'.
- Percobaan 5.1 menunjukkan cara menampilkan teks diam (statis) di posisi tertentu. Posisi kursor bisa diatur manual pakai perintah seperti 0x80 (baris 1) atau 0xC0 (baris 2), atau bisa juga pakai fungsi yang lebih gampang seperti LCD_String_xy().
- Percobaan 5.2 menunjukkan cara membuat efek animasi teks. Caranya, kita kirim perintah geser (seperti 0x1C untuk geser kanan atau 0x18 untuk geser kiri) berulang-ulang di dalam loop. Dengan diberi jeda waktu, teks di LCD jadi kelihatan bergerak.