

RANCANG BANGUN ALAT PEMBELAJARAN *MICROCONTROLLER* BERBASIS ATMEGA 328 DI UNIVERSITAS SERANG RAYA

¹⁾Ahmad Fatoni, ²⁾Dhany Dwi Nugroho, ³⁾Agus Irawan

Sistem Komputer Fakultas Teknologi Informasi

Universitas Serang Raya Kota Serang Banten

ahmadfatoni101@gmail.com¹⁾, dhanydwinugroho@gmail.com²⁾, agus.irawan@unsera.ac.id³⁾

Abstrak - Perkembangan teknologi saat ini dapat dilihat sudah banyak alat yang diciptakan supaya mempermudah dalam pembelajaran microcontroller. Seiring dengan perkembangan zaman yang semakin pesat, kebutuhan akan efektifitas dan efisiensi sangat diutamakan dalam berbagai bidang. Hal tersebut mendorong manusia untuk berkreasi dan berinovasi dalam bidang teknologi untuk menciptakan suatu alat yang lebih efektif dan efisien. Sehubungan dengan hal tersebut maka Penulis menganalisa adanya peluang untuk dibuatkannya alat tersebut pada gedung UNSERA (Universitas Serang Raya). Penggunaan microcontroller Atmega 328 sebagai otak dari proses kendali disertai perangkat input dan output dengan ini mengambil “RANCANG BANGUN ALAT PEMBELAJARAN MICROCONTROLLER BERBASIS ATMEGA 328 DI UNIVERSITAS SERANG RAYA”.

Kata Kunci : *Microcontroller*, Alat Pembelajaran, Bahasa C

I. PENDAHULUAN

Seiring dengan perkembangan zaman yang semakin pesat, kebutuhan akan efektifitas dan efisiensi sangat diutamakan dalam berbagai bidang. Hal tersebut telah mendorong manusia untuk berkreasi dan berinovasi dalam bidang teknologi untuk menciptakan suatu alat yang lebih efektif dan efisien.

Perkembangan teknologi saat ini dapat dilihat sudah banyak alat yang diciptakan supaya mempermudah dalam pembelajaran microcontroller.. Penulis menganalisa adanya peluang untuk dibuatkannya alat tersebut pada gedung UNSERA (Universitas Serang Raya).

Penggunaan microcontroller ATmega 328 sebagai otak dari proses kendali disertai perangkat input dan output dengan ini mengambil judul“ PERANCANGAN ALAT PEMBELAJARAN MICROCONTROLLER BERBASIS ATMEGA 328 DI UNSERA ”.

1.1. Identifikasi Masalah

Penulis mengambil permasalahan dimana belum adanya alat pembelajaran microcontroller berbasis microcontroller ATmega 328 di UNSERA

1.2 Batasan Masalah

Sesuai dengan masalah tersebut, penulis memiliki batasan masalah dalam pembuatan alat ini, diantaranya :

1. alat menggunakan microcontroller ATmega 328
2. Bahasa Pemrograman yang dipakai menggunakan bahasa C

1.3. Rumusan Masalah

Dari masalah diatas dapat dirumuskan:

1. Bagaimana membuat alat pembelajaran microcontroller yang efektif berbasis ATmega 328?
2. Bagaimana belajar microcontroller berbasis program bahasa c?

1.4 Tujuan Penelitian

Tujuan dari penulisan dan penelitian alat ini adalah untuk membuat alat pembelajaran microcontroller yang efektifitas di UNSERA

1.5 Metode Penulisan

Adapun metode penulisan yang di pakai adalah sebagai berikut :

- a. **Metode Observasi**, yaitu dengan melakukan

penelitian terhadap komponen yang dipakai dan melakukan perbandingan terhadap fungsi dari komponen-komponen tersebut sehingga kami dapat mengambil kesimpulan dan prinsip kerja dari komponen dan alat tersebut.

- b. **Metode Studi Pustaka**, yaitu dengan mengambil materi-materi dari buku penunjang dan materi-materi lainnya dari internet ataupun yang lainnya.
- c. **Metode Kesimpulan**, yaitu dengan mengumpulkan ide dari masing-masing orang dan kemudian menjadikan suatu kesimpulan yang di tulis ke dalam suatu makalah.
- d. **Penelitian dan Eksperimen**, yaitu dengan melakukan penelitian, perancangan alat dan menguji alat tersebut.
- e. **Konsultasi**, yaitu dengan mendiskusikan serta menanyakan secara langsung kepada dosen pembimbing.

II. LANDASAN TEORI

2.1 Komponen yang digunakan

2.1.1 Resistor TidakTetap

Resistor tidak tetap yaitu resistor yang nilai resistansi atau hambatannya dapat diubah-ubah. Jenis dari resistor ini adalah potensiometer. Di dalam alat ini, tidak menggunakan potensiometer, tapi sekedar ingin tahu bahwa potensiometer adalah resistor yang nilai resistansinya dapat berubah-ubah dengan memutar porosnya dengan menggunakan tangan.



Gambar 1 Bentuk Fisik Potensiometer

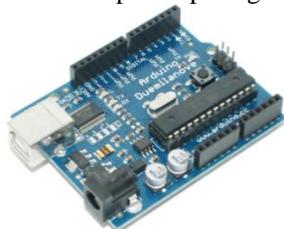


Gambar 2 Simbol Resistor Tidak Tetap

2.1.2 Mikrokontroler Arduino

Mikrokontroler *Arduino board* ialah modul yang menggunakan mikrokontroler ATmega 328 dan menggunakan seri yang lebih canggih, sehingga dapat digunakan untuk membangun system elektronika berukuran minimalis namun handal dan cepat. Berbagai modul dan sensor terkini dapat dipasang pada board ini dilengkapi dengan berbagai kode demo yang memuaskan.

Mikrokontroler *Arduino* terdiri dari beberapa *board*, yang dapat digunakan sesuai kebutuhan dan menggunakan *software open source* yang dapat dijalankan pada *Windows, Mac* dan *Linux*. Beberapa *board* yang terkenal ditampilkan pada gambar di bawah :



Gambar 3 Mikrokontroler Arduino Main Board

Mikrokontroler *Arduino* ini dilengkapi dengan konektor USB untuk memungkinkan pemrograman processor dari PC. *Arduino* juga dapat deprogram dengan menggunakan *In System Programming (ISP)*. 6 pin konektor ISP pemrograman di sisi berlawanan dari papan dari konektor USB.

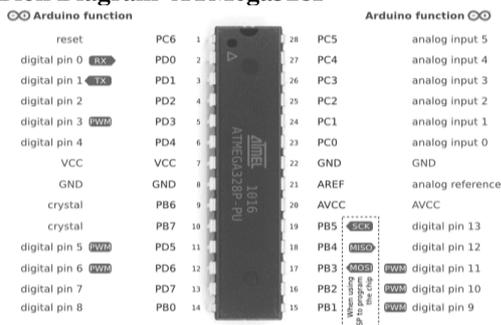
2.1.2.1 Fitur Mikrokontroler Arduino

Tabel 1 Spesifikasi Arduino

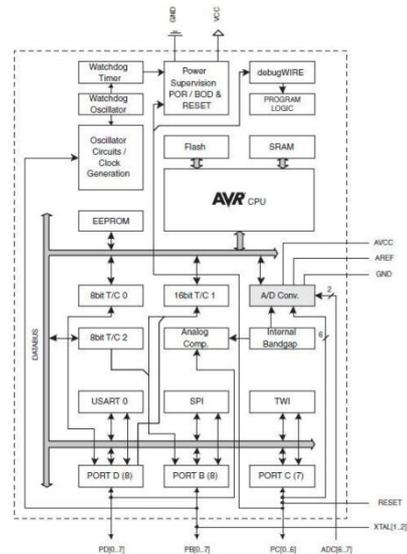
Microkontroler	ATMega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC current per I/O pin	40 mA
DC Current per 3.3V pin	50 Ma
Flash Memory	16 KB (Atmega168) or 32 KB (Atmega 328) of which 2 KB used by bootloader
SRAM	1 KB (Atmega 168) or 2 KB (ATMega328P)
EEPROM	512 bytes (ATMega328P)
Clock Speed	16 HZ

2.1.2.2 Konfigurasi pin ATMega328P

a. Blok Diagram ATMega328P

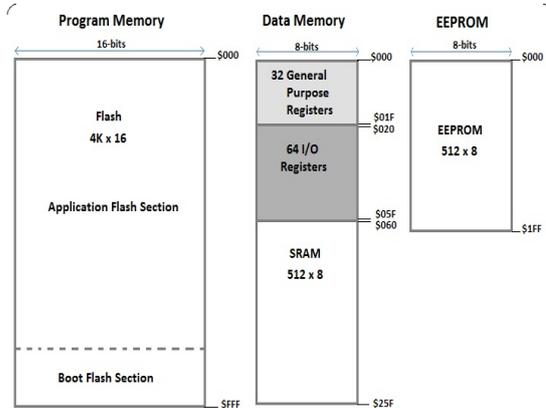


Gambar 4 Blok Diagram ATMega328P



Gambar 5 Blok Diagram ATMega328P

2.1.2.3 PetaMemori



Gambar 6 PetaMemori

2.1.3 Bahasa C

a. Tipe data. Variabel yang berfungsi menampung data sementara membutuhkan tipe data. Misal, variabel yang mampu menampung bilangan bulat harus menggunakan tipe data yang sesuai pada saat pendeklarasian variabel tersebut. Berikut table berbagai jenis tipe data di C/C++.

Tabel 2 Tipe data C++

Tipe Data	Byte*	Penjelasan	Kisaran
char	1	karakter atau integer dengan panjang 8 bit.	signed: -128 – 127 unsigned: 0 -225
short	2	integer dengan panjang 16 bit (bilangan bulat).	signed: -32768 - 32767 unsigned: 0 – 65535
long	4	integer dengan panjang 32 bit.	signed: -2147483648 - 2147483647 unsigned: 0 – 4294967295
int	*	integer. Panjangnya tergantung pada panjang dari tipe word pada system. Jika pada MSDOS sebesar 16 bit dan 32 bit pada system 32 bit seperti Windows.	Lihat short, long
float	4	Bilangan floating point.	3.4e +/- 38 (7 digit)

double	8	Bilangan double precision floating point.	1.7e + / -308 (15 digit)
long double	10	Bilangan long double precision floating point.	1.2e + / -4932 (19 digit)
bool	1	Nilai Boolean true atau false value.	true atau false
wchar_t	2	Karakterlebar, untuk menyimpan karakter internasional dari 2 byte set karakter.	Karakterlebar (wide characters)

- b. Konstanta. Konstanta merupakan suatu nilai yang tidak dapat diubah selama proses program berlangsung. Nilai konstanta selalu tetap. Konstanta harus didefinisikan terlebih dahulu di awal program. Konstanta dapat bernilai integer, pecahan, karakter, ataupun string.
- c. Variabel. Variabel adalah suatu pengenalan (*identifier*) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program . berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variabel bisa diubah-ubah sesuai kebutuhan.
- d. Deklarasi. Deklarasi diperlukan bila kita akan menggunakan identifier dalam program. Identifier dapat berupa variable, konstanta, fungsi.
- e. Operator. Operator meliputi operator penugasan (*assignment operator*) dengan tanda ("="), operator aritmatika dengan tanda (*) untuk perkalian ; (/) untuk pembagian; (%) untuk sisa pembagian; (+) untuk penambahan ; (-) untuk pengurangan, operator hubungan (perbandingan) untuk membandingkan hubungan antara dua buah operand / sebuah nilai atau variable, operator logika untuk membandingkan logika hasil dari operator-operator hubungan, operator bitwise untuk memanipulasi bit dari data yang ada di memori.

2.1.4 Software Fritzing

Fritzing adalah suatu software atau perangkat lunak gratis yang digunakan oleh desainer, seniman, dan para penghobi elektronika untuk perancangan berbagai peralatan elektronika. antarmuka *fritzing* dibuat seinteraktif dan semudah mungkin agar bisa digunakan oleh orang yang minim pengetahuannya tentang simbol dari perangkat elektronika. Di dalam *fritzing* sudah terdapat skema siap pakai dari berbagai mikrokontroler arduino serta shieldnya. Software ini memang khusus dirancang untuk perancangan dan pendokumentasian tentang produk kreatif yang menggunakan mikrokontroler arduino.

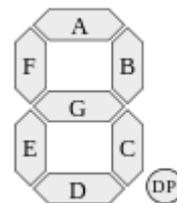
2.1.5 Diagram Alur (Flowchart)

Diagram alur merupakan salah satu cara penyajian suatu Algoritma. Sebelum sebuah program dibuat, alangkah baiknya kalau dibuat logika/ urutan-urutan instruksi program tersebut dalam suatu diagram yang disebut diagram alur (*flowchart*). Diagram alur dapat menunjukkan secara jelas arus pengendalian Algoritma, yakni bagaimana rangkaian pelaksanaan kegiatan. Suatu diagram alur memberi gambaran dua dimensi berupa simbol-simbol grafis.

2.1.6 Layar tujuh

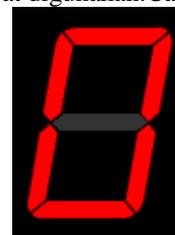
Seven-segment display (SSD)) adalah salah satu perangkat layar untuk menampilkan sistem angka desimal yang merupakan alternatif dari layar dot-matrix. Layar tujuh segmen ini seringkali digunakan pada jam digital, meteran elektronik, dan perangkat elektronik lainnya yang menampilkan informasi numerik.

Ide mengenai layar tujuh segmen ini sudah cukup tua. Pada tahun 1910 misalnya, sudah ada layar tujuh segmen yang diterangi oleh lampu pijar yang digunakan pada panel sinyal kamar ketel suatu pembangkit listrik. Konsep dan Struktur Tampilan



Gambar 7 Konsep dan struktur tampilan

Tujuh bagian dari layar dapat dinyalakan dalam bermacam-macam kombinasi untuk menampilkan angka Arab. Sering ketujuh segmen tersebut disusun dengan kemiringan tertentu, untuk memudahkan pembacaan. Pada sebagian besar penerapannya, ketujuh segmen ini memiliki bentuk dan ukuran yang hampir seragam (biasanya segienam panjang, walaupun trapesium dan pers egipanjang juga dapat digunakan. Cara Kerja



Gambar 8 Layar tujuh segmen berbasis LED

Layar tujuh segmen ini terdiri dari 7 buah LED yang membentuk angka 8 dan 1 LED untuk titik/DP. Angka yang ditampilkan di seven segmen ini dari 0-9. Cara kerja dari seven segmen disesuaikan dengan LED. LED merupakan komponen diode yang dapat memancarkan cahaya. kondisi dalam keadaan ON jika sisi anode mendapatkan sumber positif dari Vcc dan katode mendapatkan sumber negatif dari ground. Berdasarkan cara kerjanya, tujuh segmen dibagi menjadi 2 bagian:

- common katode
 Cara kerja dari seven segmen common katode akan aktif pada kondisi high "1" dan akan off pada kondisi low "0".

Tabel 3. Common Catode

ANGKA	h	g	f	e	d	c	b	a	HEXA
0	0	0	1	1	1	1	1	1	3FH
1	0	0	0	0	0	1	1	0	06H
2	0	1	0	1	1	0	1	1	5BH
3	0	1	0	0	1	1	1	1	4FH
4	0	1	1	0	0	1	1	0	66H
5	0	1	1	0	1	1	0	1	6DH
6	0	1	1	1	1	0	1	1	7DH
7	0	0	0	0	0	1	1	1	07H
8	0	1	1	1	1	1	1	1	7FH
9	0	1	1	0	1	1	1	1	6FH

- common anode

Cara kerja dari seven segmen common anode akan aktif pada kondisi low "0" dan akan off pada kondisi high "1".

Tabel 4. Pengaktifan common katode

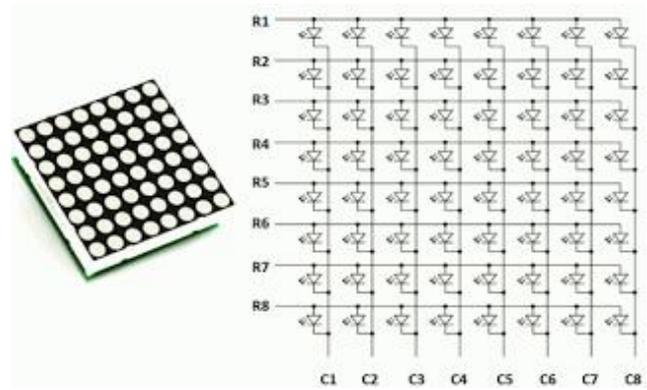
ANGKA	h	g	f	e	d	c	b	a	HEXA
0	1	1	0	0	0	0	0	0	C0H
1	1	1	1	1	1	0	0	1	F9H
2	1	0	1	0	0	1	0	0	A4H
3	1	0	1	1	0	0	0	0	B0H
4	1	0	0	1	1	0	0	1	99H
5	1	0	0	1	0	0	1	0	EDH
6	1	0	0	0	0	0	1	0	12H
7	1	1	1	1	1	0	0	0	F8H
8	1	0	0	0	0	0	0	0	10H
9	1	0	0	1	0	0	0	0	90H

2.1.6 Buzzer

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja buzzer hampir sama dengan loud speaker, jadi buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. Buzzer biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm).

2.1.8 LED Dot Matrix

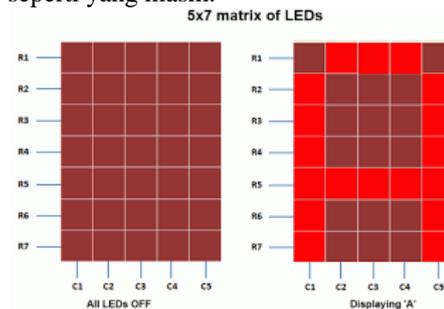
Teori LED dot matriks layar Dalam layar dot matrix, beberapa LED yang kabel bersama-sama dalam baris dan kolom. Hal ini dilakukan untuk meminimalkan jumlah pin yang diperlukan untuk mengusir mereka. Misalnya, 8 x 8 matriks LED (ditampilkan di bawah) akan membutuhkan 64 I / O pin, satu untuk setiap pixel LED. Dengan kabel semua anoda bersama-sama dalam baris (R1 melalui R8), dan katoda dalam kolom (C1 melalui C8), jumlah yang diperlukan pin I / O dikurangi menjadi 16. Setiap LED ditujukan demi baris dan jumlah kolom. Pada gambar bawah ini, jika R4 adalah ditarik tinggi dan C3 ditarik rendah, LED di baris keempat dan kolom ketiga akan dihidupkan. Karakter dapat ditampilkan oleh pemindaian cepat baik baris atau kolom. Tutorial ini akan membahas metode pemindaian kolom.



Gambar 9 Metode Pemindaian Kolom

2.1.2 Struktur dari 8x8 LED dot matriks

Matriks LED digunakan dalam percobaan ini adalah ukuran 5 x 7. Kita akan belajar bagaimana menampilkan karakter yang masih dalam format standar 5 7 x pixel. Gambar di bawah menunjukkan yang LED harus diaktifkan untuk menampilkan abjad Inggris 'A'. Para 7 baris dan 5 kolom dikendalikan melalui pin mikrokontroler. Sekarang, mari kita lihat secara detail cara kerjanya. Misalkan, kita ingin menampilkan A. alfabet Kami akan memilih kolom C1 (yang berarti C1 ditarik rendah dalam kasus ini), dan kolom hapus lainnya dengan memblokir jalan tanah mereka (salah satu cara untuk melakukan hal itu adalah dengan menarik C2 melalui C5 pin untuk tinggi logika). Sekarang, kolom pertama aktif, dan Anda perlu mengaktifkan LED di baris R2 melalui R7 kolom ini, yang dapat dilakukan dengan menerapkan tegangan bias maju untuk baris ini. Selanjutnya, pilih kolom C2 (dan hapus semua kolom lainnya), dan menerapkan bias maju ke R1 dan R5, dan sebagainya. Oleh karena itu, dengan memindai seluruh kolom cepat (> 100 kali per detik), dan menyalakan LED masing-masing di setiap baris kolom itu, kegigihan visi datang untuk bermain, dan kami merasakan tampilan gambar seperti yang masih.



Gambar 10 Sebuah standar 5x7 dot matriks layar LED struktur

Tabel di bawah ini memberikan tingkat logika untuk diterapkan ke R1 melalui R7 untuk setiap kolom untuk menampilkan abjad 'A'.

Column	R1	R2	R3	R4	R5	R6	R7
C1	0	1	1	1	1	1	1
C2	1	0	0	0	1	0	0
C3	1	0	0	0	1	0	0
C4	1	0	0	0	1	0	0
C5	1	0	0	0	1	0	0
C6	0	1	1	1	1	1	1

Gambar 11 Tabel Tingkat Logika

III. DESAIN DAN METODE PENELITIAN

3.1 Analisis Kebutuhan

Dalam Pembuatan alat penggerak solar cell ini membutuhkan beberapa perangkat keras (*hardware*), perangkat lunak (*software*) dan beberapa alat-alat pendukung antara lain:

3.1.1 Perangkat Keras (*Hardware*)

1. Spesifikasi perangkat keras pendukung pembuatan program mikrokontroler adalah:
 - a. Komputer, Sebagai media untuk merancang, membuat hingga meng-*input* program ke *chip* .
 - b. Mouse, Sebagai penggerak kursor pada komputer.
2. Spesifikasi komponen elektronik dan mikrokontroler yang digunakan adalah:
 - a. ATmega328p, Berfungsi untuk menyimpan program yang sudah dibuat, dan berfungsi mengatur jalur komponen yang sedang bekerja.
 - b. Resistor 220Ω, 1K,10K, Berfungsi sebagai penghambat arus listrik.
 - c. PCB, Berfungsi untuk tempat dimana komponen-komponen tersebut dipasangkan sesuai rancangan.
 - d. Konektor Power, Sebagai penghubung power ke alat.
 - e. Kapasitor, Kapasitor atau kondensator adalah alat (komponen) yang dibuat sedemikian rupa sehingga mampu menyimpan muatan listrik yang besar untuk sementara waktu. Sebuah kapasitor terdiri atas keping-keping logam yang disekat satu sama lain dengan isolator. Isolator penyekat disebut zat dielektrik. Simbol yang digunakan untuk menampilkan sebuah kapasitor dalam suatu rangkaian listrik adalah . Ada dua cara pemasangan kapasitor, yaitu tanpa memerhatikan kutub-kutubnya (untuk kapasitor nonpolar) dan dengan memerhatikan kutub-kutubnya (untuk kapasitor polar).
 - f. Kristal 16.000 Mhz, Kristal adalah resonator mekanik yang bergetar menstabilkan getaran elektronis, kristal stabil karena memiliki 'inersia' yang relatif besar. Jika menggunakan kristal maka akan didapatkan frekuensi clock yang stabil.
 - g. Kabel , Berfungsi sebagai komponen penghubung/penghantar listrik.
 - h. Socket IC, Berfungsi sebagai tempat pemasangan IC.

3.1.2 Perangkat Lunak (*Software*)

Aplikasi yang digunakan untuk menuliskan program yang akan dibuat sekaligus untuk memasukkan program yang sudah dibuat kedalam mikrokontroler *Arduino* adalah aplikasi *Arduino.exe*.

3.2 Analisa Biaya

Berikut kisaran anggaran biaya yang penulis butuhkan dalam menyelesaikan pembuatan alat penggerak solar cell ini:

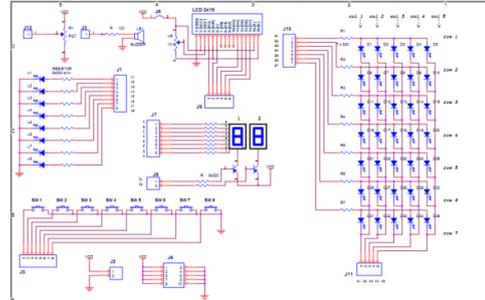
Tabel 5 Analisa Biaya yang dibutuhkan

Komponen	Biaya
LCD 2x16	Rp. 50.000
Arduino	Rp.250.000
Led Matrix	Rp.30.000
7 Segment	Rp.20.000
Switch	Rp.16000
Buzzer	Rp. 15.000
1 Keping PCB	Rp. 40.000

Solder + Timah	Rp. 75.000
3 bungkus Ferric Chloride	Rp. 15.000
Komponen Rangkaian Lainnya	Rp. 150.000
Total:	Rp. 661.000

3.3 Rangkaian utama Alat pembelajaran

Rangkaian lata ini dibuat terpisah antara input , proses dana output sehingga pengguna dapat memahami rangkaian dan jalura yang ada sehingga dapat menentukan proses dan aplikasi yang diinginkan



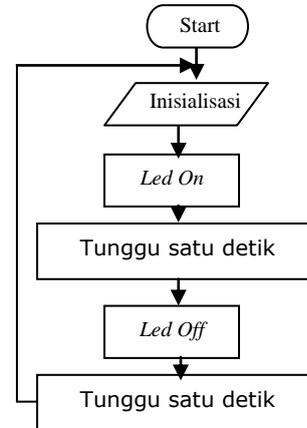
Gambar 12 Rangkaian utama alat

3.4 Diagram Alur

3.4.1 Control Led

Dalam pembelajaran ini mencoba dengan menghidupkan salah satu led agar mengetahui posisi pin yang ada dan program yang akan di gunakan untuk menghidupkan led tersebut, dipercobaan yang kedua dengan mencoba menyalakan led berkedip agar mengetahui pengaturan led dengan menggunakan fungsi delay, di percobaan ketiga bagaimana menyalakan led dengan memberikan fungsi control.

3.4.2 Flowchart control led



Gambar 13 Diagram Alur

3.3.3. listing program control led

```
//Program 1 : menghidupkan LED di pin 13 board
Arduino
void setup() {
  pinMode(13, OUTPUT);
}
void loop() {
  digitalWrite(13, HIGH);
}
//Program 2 : LED berkedip
void setup() {
  pinMode(13, OUTPUT);
}
void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
// Program 3 : kontrol 8 LED
void setup() {
  pinMode(2, OUTPUT);
```

```

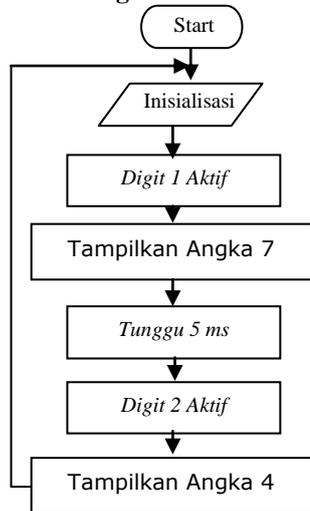
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
}
void loop() {
digitalWrite(2, 0);
digitalWrite(3, 1);
digitalWrite(4, 0);
digitalWrite(5, 1);
digitalWrite(6, 0);
digitalWrite(7, 1);
digitalWrite(8, 0);
digitalWrite(9, 1);
}

```

3.4.1 Control 7 segment

Alam pembelajaran ini mencoba untuk mengatur perangkat 7 segment yang merupakan gabungan beberapa led yang akan membentuk angka-angka yang biasanya digunakan untuk membuat jam digital

3.4.2 Flowchart 7 Segment



Gambar 14 Diagram Alur

3.4.3 listing program 7 Segment

```

//----- Program 5 : tampilan angka 3 pada digit
1
void setup() {
pinMode(2, OUTPUT);
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);
}
void loop() {
digitalWrite(10,0);
digitalWrite(11,1);
digitalWrite(2,0);
digitalWrite(3,0);
digitalWrite(4,0);
digitalWrite(5,0);
digitalWrite(6,1);
digitalWrite(7,1);
digitalWrite(8,0);
digitalWrite(9,1);
}
//----- Program 6 : tampilan angka 2 pada digit
2
void setup() {
pinMode(2, OUTPUT);

```

```

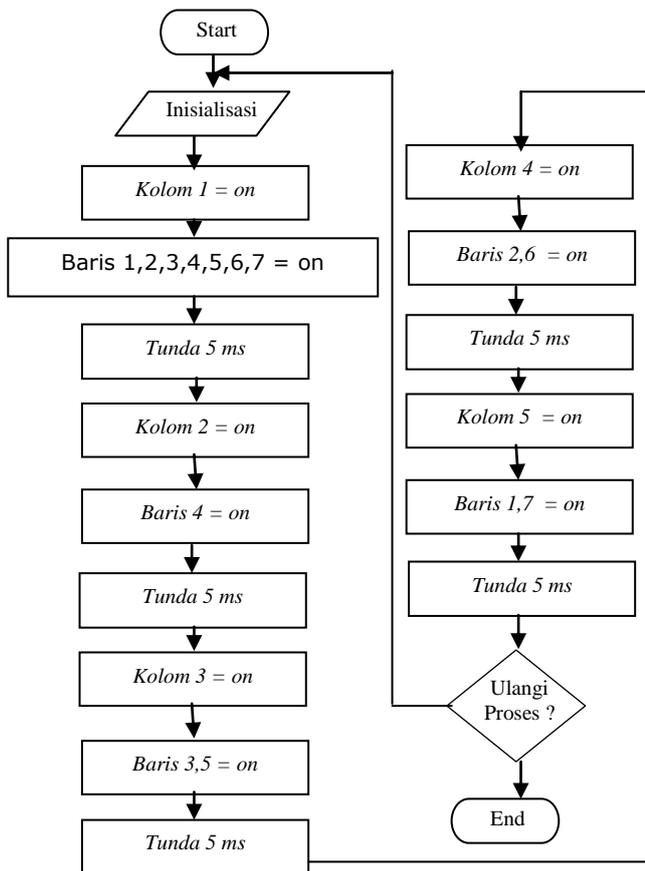
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);
}
void loop() {
digitalWrite(10,1);
digitalWrite(11,0);
digitalWrite(2,0);
digitalWrite(3,0);
digitalWrite(4,1);
digitalWrite(5,0);
digitalWrite(6,0);
digitalWrite(7,1);
digitalWrite(8,0);
digitalWrite(9,1);
}
//----- Program 7 : tampilan angka 74
void setup() {
pinMode(2, OUTPUT);
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);
}
void loop() {
digitalWrite(10,0);
digitalWrite(11,1);
//---tampilan angka 7
digitalWrite(2,0);
digitalWrite(3,0);
digitalWrite(4,0);
digitalWrite(5,1);
digitalWrite(6,1);
digitalWrite(7,1);
digitalWrite(8,1);
digitalWrite(9,1);
delay(5);
digitalWrite(10,1);
digitalWrite(11,0);
//---tampilan angka 4
digitalWrite(2,1);
digitalWrite(3,0);
digitalWrite(4,0);
digitalWrite(5,1);
digitalWrite(6,1);
digitalWrite(7,0);
digitalWrite(8,0);
digitalWrite(9,1);
delay(5);
}

```

3.5.1 Control Led Dot Matrix 5x7

Didalam pengaturan led matrix kita akan mengatur lampu led yang sudah tersusun yang nantinya akan menyalakan sebagian lampu led sehingga akan membentuk huruf yang kita inginkan.

3.5.2 Flowchart Led Dot Matrix 5x7



Gambar 15 Diagram Alur

listing program Led Matrix

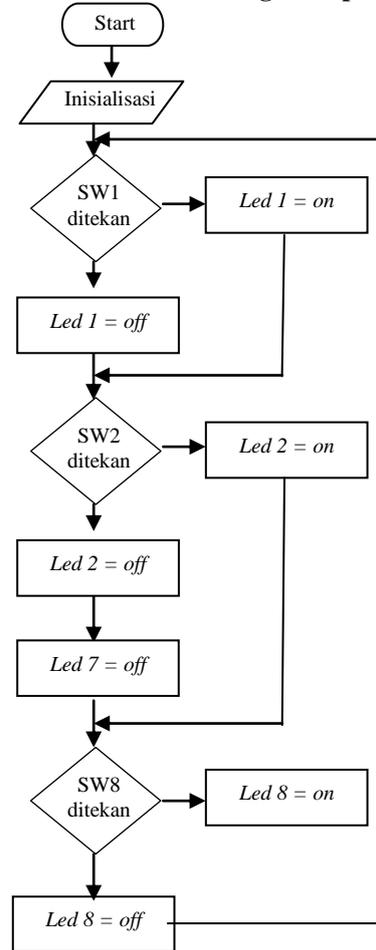
```
// Program 8 : membuat garis horisontal pada
baris ke-4
void setup()
{
  pinMode (2, OUTPUT);
  pinMode (3, OUTPUT);
  pinMode (4, OUTPUT);
  pinMode (5, OUTPUT);
  pinMode (6, OUTPUT);
  pinMode (7, OUTPUT);
  pinMode (8, OUTPUT);
  pinMode (9, OUTPUT);
  pinMode (10, OUTPUT);
  pinMode (11, OUTPUT);
  pinMode (12, OUTPUT);
  pinMode (13, OUTPUT);
}
void loop() {
  //-----
  digitalWrite (2,0);
  digitalWrite (3,0);
  digitalWrite (4,0);
  digitalWrite (5,1);
  digitalWrite (6,0);
  digitalWrite (7,0);
  digitalWrite (8,0);
  digitalWrite (9,0);
  digitalWrite (10,0);
  digitalWrite (11,0);
  digitalWrite (12,0);
  digitalWrite (13,0);
}
// Program 9 : membuat garis vertikal pada kolom
ke-2
void setup()
{
  pinMode (2, OUTPUT);
  pinMode (3, OUTPUT);
  pinMode (4, OUTPUT);
  pinMode (5, OUTPUT);
  pinMode (6, OUTPUT);
}
```

```
pinMode (7, OUTPUT);
pinMode (8, OUTPUT);
pinMode (9, OUTPUT);
pinMode (10, OUTPUT);
pinMode (11, OUTPUT);
pinMode (12, OUTPUT);
pinMode (13, OUTPUT);
}
void loop() {
  //-----
  digitalWrite (2,1);
  digitalWrite (3,1);
  digitalWrite (4,1);
  digitalWrite (5,1);
  digitalWrite (6,1);
  digitalWrite (7,1);
  digitalWrite (8,1);
  digitalWrite (9,1);
  digitalWrite (10,0);
  digitalWrite (11,1);
  digitalWrite (12,1);
  digitalWrite (13,1);
}
```

3.6.1 Control Digital Input

Didalam pembelajaran ini kita akan belajar bagaimana menegetahui proses didalam proses input data kemudian akan dimunculkan dalam komponen output.

3.6.2 Flowchart Digital Input



Gambar 16 Flowchar digital input

Listing program Digital Input

```
//Program 11 : membaca 8 tombol
byte tombol;
void setup()
{
  pinMode (2, INPUT); //pin 2 sebagai input
  pinMode (3, INPUT); //pin 3 sebagai input
  pinMode (4, INPUT); //pin 4 sebagai input
  pinMode (5, INPUT); //pin 5 sebagai input
  pinMode (6, INPUT); //pin 2 sebagai input
}
```

```

pinMode(7, INPUT); //pin 3 sebagai input
pinMode(8, INPUT); //pin 4 sebagai input
pinMode(9, INPUT); //pin 5 sebagai input
pinMode(10, OUTPUT); //pin 6 sebagai output
pinMode(11, OUTPUT); //pin 7 sebagai output
pinMode(12, OUTPUT); //pin 8 sebagai output
pinMode(13, OUTPUT); //pin 9 sebagai output
pinMode(A0, OUTPUT); //pin 6 sebagai output
pinMode(A1, OUTPUT); //pin 7 sebagai output
pinMode(A2, OUTPUT); //pin 8 sebagai output
pinMode(A3, OUTPUT); //pin 9 sebagai output
digitalWrite(2, HIGH); //pin 2 input pull up
digitalWrite(3, HIGH); //pin 3 input pull up
digitalWrite(4, HIGH); //pin 4 input pull up
digitalWrite(5, HIGH); //pin 5 input pull up
digitalWrite(6, HIGH); //pin 2 input pull up
digitalWrite(7, HIGH); //pin 3 input pull up
digitalWrite(8, HIGH); //pin 4 input pull up
digitalWrite(9, HIGH); //pin 5 input pull up
}
void loop()
{
  tombol=digitalRead(2);
  if (tombol==LOW) digitalWrite(10,1);
  else digitalWrite(10,0);
  tombol=digitalRead(3);
  if (tombol==LOW) digitalWrite(11,1);
  else digitalWrite(11,0);
  tombol=digitalRead(4);
  if (tombol==LOW) digitalWrite(12,1);
  else digitalWrite(12,0);
  tombol=digitalRead(5);
  if (tombol==LOW) digitalWrite(13,1);
  else digitalWrite(13,0);
  tombol=digitalRead(6);
  if (tombol==LOW) digitalWrite(A0,1);
  else digitalWrite(A0,0);
  tombol=digitalRead(7);
  if (tombol==LOW) digitalWrite(A1,1);
  else digitalWrite(A1,0);
  tombol=digitalRead(8);
  if (tombol==LOW) digitalWrite(A2,1);
  else digitalWrite(A2,0);
  tombol=digitalRead(9);
  if (tombol==LOW) digitalWrite(A3,1);
  else digitalWrite(A3,0);
}

```

3.7.1 Control LCD Display

Didalam pembelajaran ini kita akan mempelajari bagaimana kita mengaktifkan LCD display yang akan memberi keterangan lengkap informasi yang dimunculkan didalam LCD Display

3.7.2 listing program LCD Display

```

//Program 13 : menampilkan tulisan pada LCD
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
void setup() {
  lcd.begin(16, 2);
  lcd.print("Arduino UNO");
  lcd.setCursor(0, 1);
  lcd.print("Coba LCD 2x16");
}
void loop() {
}

```

IV. HASIL PENELITIAN DAN PEMBAHASAN

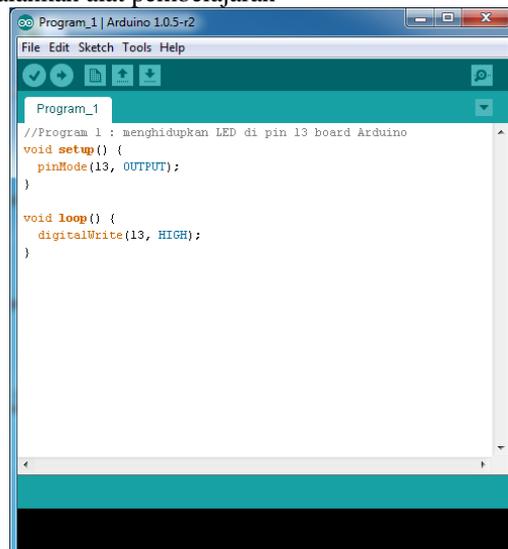
4.1 Pengoperasian

Prinsip kerja alat pembelajaran microcontroller adalah bagaimana memberikan kemudahan bagi orang yang awam sekalipun untuk mencoba melakukan pembelajaran microcontroller

Langkah-langkah untuk mengoperasikan Alat pembelajaran microcontroller adalah sebagai berikut:

1. Sambungkan alat pembelajaran dengan port USB

2. Buka program arduino
3. Masukkan program yang diinginkan kedalam arduino
4. Jalankan alat pembelajaran



Gambar 17

4.2 Pengujian

Alat diuji kan pada perangkat output led, lcd, 7 segment dan matrik sehingga bisa menghasilkan output yang kita inginkan

4.2.1 Tujuan Pengujian

Tujuan pengujian ini dilakukan agar alat dapat dengan mudah dipelajari dengan melihat input dan output yang kita inginkan, dengan memasukkan program kedalam memory di microcontroller akan melihat hasilnya langsung di komponen input maupun output.

4.2.2 Ruang Lingkup Pengujian

Pada pengujian ini, penulis menguji alat ini hanya pada kisaran komponen output yaitu lcd, matrik, led dan 7 segment yang banyak dibutuhkan dalam memunculkan hasil dari proses yang ada di microcontroller

4.2.3 Analisa dan Hasil Pengujian

Analisa yang dilakukan dengan menggunakan flowchart programming akan menghasilkan suatu tampilan output yang diinginkan sesuai dengan program yang diinput oleh user.

4.2.4 Hasil pengujian

Tabel 6 Hasil pengujian

No	Program	led	7 segment	Matrix
	Program 1	Led kuning menyala(13)		
	Program 2	Lampu led berkedip		
	Program 3	Led 1,5,7 padam Led 2,4,6,8 nyala		
	Program 4		Tampil 3 di seven segment digit ke 1	
	Program 5		Tampil 2 di seven segment digit ke 2	
	Program 6			Semua led baris ke 4 nyala
	Program			Semua

	7			led kolom ke 2 nyala
--	---	--	--	----------------------

4.3 Hasil Quisioner Penilaian Alat pembelajaran

Quisioner dilakukan untuk mengukur keberhasilan efektifitas alat sebagai media pembelajaran yang nantinya akan dipakai dalam pembelajaran khususnya sistemkomputer unsera dan umumnya seluruh lapisan masyarakat awam yang ingin mempelajari microcontroller dengan mudah.

Table 7 Hasil Rekapitulasi Quisioner

No	Penilaian	baik	cukup	kurang
1	Efektifitas	80 %	20%	0%
2	kemudahan	80 %	10%	10%
3	Pemahaman	50%	30 %	20%
4	Pengembangan	10%	40%	50%
5	Ketersediaan Alat	20%	40%	40%

V. KESIMPULAN

Setelah melakukan perencanaan dan pembuatan alat pembelajaran kemudian dilakukan pengujian, maka dapat diambil beberapa kesimpulan tentang alat pembelajaran microcontroller yaitu sebagai berikut:

1. Alat pembelajaran ini dapat mempermudah orang yang awam dalam dasar elektronika tetapi dapat

dengan cepat mempelajari microcontroller serta aplikasinya. Cara kerja alat ini berbasis mikrokontroller atmega328

2. Rangkaian alat pembelajaran ini dapat dikembangkan untuk pengembangan aplikasi yang lainnya.
3. Pemrograman yang digunakan yaitu program Bahasa C

VI. DAFTAR PUSTAKA

- [1] Bishop, Owen 2004, *Dasar-Dasar Elektronika*, Erlangga, Jakarta.
- [2] Danny Christianto, S.T., Kris Pusorini, S.T., M.T., 2004, *“Panduan Dasar Mikrokontroler Keluarga MCS-51”*, Innovative Elelectronics, Surabaya.
- [3] Habi, Miftahul, 2013, *Tugas Akhir: “ Perancangan alat Pembelajaran Microcontroller at89s2 di Universitas Serang Raya”* Sistem computer Universitas Serang Raya.
- [4] Sahrul, 2014, *Pemograman Mikrokontroler AVR Bahasa Assembly dan C*, Informatika, Bandung.
- [5] <http://en.wikipedia.org/wiki/ATmega328>
- [6] <http://arduino.cc>
- [7] http://en.wikipedia.org/wiki/SMART_criteria