

BAB II

LANDASAN TEORI

II.1. Perancangan

Defenisi perancangan menurut Al-Bahra Bin Ladjamudin (2005 : 51) perancangan adalah kemampuan untuk membuat berupa alternatif pemecahan masalah. Sedangkan menurut Azhar Susanto (2004 : 332) perancangan adalah spesifikasi umum dan terinci dari pemecahan masalah berbasis komputer yang telah dipilih selama tahap analisis. Berdasarkan defenisi tersebut dapat disimpulkan bahwa perancangan adalah kemampuan untuk membuat alternatif pemecahan masalah berbasis komputer selama tahap analisis.

II.1.1. UML

Unified modeling language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML mendefenisikan notasi dan syntax.

Fowler (2005) memberikan daftar UML 2 terdiri dari 13 jenis diagram resmi seperti dalam tabel II.1 berikut ini:

Tabel II.1. Jenis Diagram Resmi UML

Diagram	Kegunaan
<i>Activity</i>	<i>Behaviour</i> prosedural dan paralel
<i>Class</i>	<i>Class</i> , fitur, dan hubungan-hubungan
<i>Communication</i>	Interaksi antar objek, penekanan pada jalur
<i>Component</i>	Struktur dan koneksi komponen
<i>Composite structure</i>	Dekomposisi <i>runtime</i> sebuah <i>class</i>
<i>Deployment</i>	Pemindahan artifak ke node

<i>Interaction overview</i>	Campuran <i>sequence</i> dan <i>activity</i> diagram
<i>Object</i>	Contoh konfigurasi
<i>Package</i>	Struktur hirarki <i>compile-time</i>
<i>Sequence</i>	Interaksi antar object, penekanan pada <i>sequence</i>
<i>State machine</i>	Bagaimana even mengubah objek selama aktif
<i>Timing</i>	Interaksi antar objek, penekanan pada <i>timing</i>
<i>Use case</i>	Bagaimana pengguna berinteraksi dengan sebuah sistem

(Sumber : Agustinus Mujilan, 2013)

Bagi perancang pemula disarankan oleh Fowler (2005) untuk memusatkan perhatian pada bentuk-bentuk diagram dasar *class diagram* dan *sequence diagram*, diagram tersebut adalah bentuk yang paling umum dan diagram yang paling berguna. Kuske et al. (2009) dalam artikelnya juga menyampaikan bahwa cara yang cukup berarti dalam mengaplikasikan UML adalah menggunakan diagram *class*, *state*, dan *interaction* sebagai dasar penjelasan sistem. *Class diagrams* menjelaskan dasar penjelasan struktur objek, *state diagram* untuk menjalankan objek *behaviour*, dan *interaction diagrams* menjelaskan bagaimana objek-objek berinteraksi dalam suatu kolaborasi.

II.2. Robot

Robot berasal dari bahasa Cheko, “Robota” yang berarti pekerja atau kuli yang tidak mengenal lelah atau bosan. Robot merupakan alat mekanik yang diciptakan untuk meringankan pekerjaan manusia. Robot sendiri memiliki beberapa jenis dan fungsi. Hal tersebut tergantung pada program yang ditanamkan dalam dirinya. (Pram : 2013).



Gambar II.1. Robot

(sumber : www.zonaelektro.net)

II.2.1. Karakteristik Dasar Robot

Robot memiliki empat karakteristik dasar, sehingga kita bisa lebih mudah menentukan apakah suatu benda merupakan robot atau bukan dengan mengetahui karakteristik dasar dari benda tersebut. Empat karakteristik tersebut adalah:

1. Robot memiliki sensor

Sensor merupakan peralatan yang berguna untuk mengukur ataupun merasakan sesuatu pada lingkungan diluar robot, layaknya indera pada makhluk hidup, dan memberi laporan hasilnya kepada robot. Dengan adanya sensor, robot bisa memiliki suatu pertimbangan dalam mengambil keputusan..

2. Robot memiliki (kontrol) sistem kecerdasan

Sistem kecerdasan bekerja dengan memproses data masukan berupa keadaan ataupun kejadian yang sedang terjadi dari luar lingkungan. Selanjutnya sistem menghasilkan keluaran berupa instruksi ataupun keputusan pada robot untuk melakukan suatu tindakan tertentu. Sistem ini secara umum memiliki

prinsip kerja seperti otak pada makhluk hidup, yang berfungsi untuk berpikir dan memutuskan tindakan apa yang perlu diambil pada suatu waktu tertentu.

3. Robot memiliki (aktuator) peralatan mekanik

Peralatan mekanik berfungsi untuk membuat robot dapat melakukan suatu tindakan tertentu dan berinteraksi dengan lingkungannya. Contohnya seperti adanya roda bermotor untuk bergerak, lengan untuk mengambil objek, dan lain-lain.

4. Robot memiliki (power) sumber daya

Seperti halnya makhluk hidup yang membutuhkan makanan untuk hidup, robot juga memerlukan sumber tenaga untuk menggerakkan komponen elektrik dan mekanika yang terpasang. Sumber energy pada robot mencakup penyedia tenaga listrik seperti baterai, dan sistem pengatur transmisi yang bertugas mengkonversi tenaga listrik sesuai kebutuhan setiap komponen.

Sensor pada robot modern seringkali berkaitan dengan ilmu elektronika. Dalam ilmu elektronika dipelajari hal-hal yang berkaitan dengan komponen elektronik, sirkuit analog, sirkuit digital, dan juga mikrokontroler. Sebuah sensor dapat tersusun dari rangkaian analog atau rangkaian digital. Bersamaan dengan meningkatnya teknologi komputer dan elektronika maka meningkat pula pengembangan sensor yang dapat difabrikasikan dengan ukuran mini.

Sistem kecerdasan pada robot modern dikembangkan pada sebuah piranti lunak. Kecerdasan buatan dapat dirancang menggunakan algoritma yang memungkinkan robot bergerak secara otomatis, dengan mempertimbangkan informasi sekitar yang dibaca dari sensor yang ada. Selain itu terdapat layer

keterhubungan antara sistem kecerdasan dengan sensor dan actuator. Dengan demikian, data dapat ditransfer antar komponen yang dikelola oleh perangkat keras menggunakan piranti lunak yang dibuat untuk robot tersebut.

II.3. Sejarah dan Perkembangan rubik

Rubik's Cube atau kubus rubik adalah sebuah teka-teki (*puzzle*) mekanis dalam bentuk tiga dimensi, yang ditemukan pada tahun 1974 oleh seorang pemahat dan profesor arsitektur dari Hungaria bernama Erno Rubik. Pada pertengahan tahun 70-an, Erno Rubik ingin membuat sebuah model sebagai alat pembelajaran untuk membantu murid - muridnya memahami geometri tiga dimensi dan akhirnya tercipta menjadi sebuah mainan teka-teki yang paling laris didunia.

Rubik memberi nama hasil temuannya itu "*Magic Cube*" - Kubus Ajaib, yang kemudian mendapatkan hak paten Hungaria dan dijual pertama kali melalui perusahaan *Ideal Toy Corporation*. Pada tahun 1980, perusahaan *Ideal Toy* mengubah nama kubus ajaib tersebut menjadi "*Rubik's Cube*". Dan hingga saat ini, lebih dari 300 juta *Rubik's Cube* telah dijual di seluruh dunia.

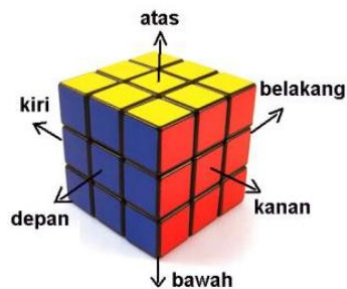
Rubik masuk ke Indonesia sekitar tahun 90-an dan mulai berkembang pesat ketika masuk keabad millennium yaitu tahun 2000 ke atas. Pertengahan tahun 2009, masuknya permainan kubik dengan nama *speedcubing*. *Speedcubing* adalah menyelesaikan Kubus Rubik dalam waktu secepat mungkin dengan dihitung waktunya. Dari situlah munculah komunitas - komunitas pencinta speedrubik muncul, salah satunya Nusantara Speedcubing Association. Visi mereka adalah memopulerkan speedcubing agar sejajar dengan olahraga yang lain

seperti catur, billiard, renang, dan sebagainya. Melalui misi mengadakan kompetisi dan kegiatan – kegiatan speedcubing baik yang bertaraf local maupun internasional.

II.3.1. Mengenal Rubik

Terminologi yang biasa dipakai untuk menjelaskan suatu algoritma *Rubik's Cube* sebagai berikut:

1. *Rubik's Cube* terdiri atas enam buah sisi: depan (*front*), belakang (*back*), kiri (*left*), kanan (*right*), atas (*up*), dan bawah (*down*) seperti pada gambar II.2 yang menjelaskan sisi-sisi pada rubik.



Gambar II.2. Penamaan sisi-sisi pada *Rubik's Cube*

2. Gerakan memutar sisi *Rubik's Cube* searah jarum jam disimbolkan dengan huruf pertama dari nama sisi tersebut dalam bahasa Inggris. Untuk yang berlawanan arah dengan jarum jam, ditambahkan tanda petik ('). Contohnya, F berarti memutar sisi depan (*front*) searah jarum jam, sedangkan L' berarti memutar sisi kiri (*left*) berlawanan arah dengan jarum jam.
3. Gerakan rotasi rubik tidak memiliki notasi yang tetap, sehingga pada laporan ini menggunakan X, Y, dan Z, dimana posisi tersebut adalah sumbu rotasi rubik.

II.3.2. Algoritma Dalam Permainan Rubik

Sebelum mempelajari pada pengenalan notasi atau algoritma rubik, salah satu hal yang perlu diperhatikan dalam memahami karakter rubik 3x3x3 adalah pembagian kategori piece atau kubus kecil. Pembagian piece ini terdiri dari :

1. **Center Piece** : *Piece* (kubus kecil) yang terdapat ditengah disetiap sisi rubik.

Center piece berjumlah 6 buah sesuai dengan warna yang terdapat disetiap sisi.



Gambar II.3. Center Piece Rubik

2. **Corner Piece** : *Piece* yang terdapat pada setiap sudut disemua sisi rubik, tentu jumlah seluruhnya 8 buah dimana sebuah *corner piece* terdiri dari 3 warna stiker.



Gambar II.4. Corner Piece Rubik

3. **Edge Piece** : *Piece* yang mengelilingi *center piece* (bukan *corner piece*), sehingga apabila dilihat dari atas akan membentuk *cross*. *Piece* ini berjumlah 12 buah dimana sebuah *edge piece* terdiri dari 2 warna stiker.



Gambar II.5. Edge Piece Rubik

Dalam menyelesaikan rubik, ada yang disebut dengan notasi. Notasi ini mengindikasikan bagian dari *cube* yang harus diputar. Beberapa notasi yang digabungkan menjadi satu set gerakan akan memberi efek tertentu pada rubik. Gabungan notasi inilah yang disebut “algoritma”. Efek yang diberikan oleh algoritma adalah efek yang dibutuhkan, sehingga sangat dibutuhkan untuk menyelesaikan *cube*. Cara memutar *cube* ditulis dalam notasi khusus menggunakan huruf F, B, U, D, L, R, dan seterusnya. Notasi tersebut digunakan secara internasional oleh para *cuber* di seluruh dunia. Agar lebih jelas, perhatikan penjelasan dibawah ini :

























1. *Single Turn* (Putaran Tunggal)

Notasi *single turn* adalah notasi yang digunakan untuk memutar sebuah sisi (*layer*) dari rubik sebesar seperempat putaran (90 derajat). Notasi ini memiliki beberapa macam yang ditunjukkan dengan huruf, antara lain :

- a. **F** (*front* atau sisi depan)
- b. **B** (*back* atau sisi belakang)
- c. **U** (*up* atau sisi atas)
- d. **D** (*down* atau sisi bawah)
- e. **R** (*right* atau sisi kanan)
- f. **L** (*left* atau sisi kiri)

Implementasi gerakan notasi huruf pada rubik dapat dilihat pada Tabel II.2 sebagai berikut:

Tabel II.2. *Single Turn* Rubik (Putaran Tunggal Rubik)













Notasi	Gerakan	Efek
F		
F'		
B		
B'		
U		
U'		
D		
D'		
R		
R'		
L		
L'		













Secara umum notasi - notasi di atas dibagi menjadi dua kelompok. Kelompok pertama yaitu: F, B, R, L, U, D yang gerakannya adalah searah jarum jam. F, berarti gerakan Front atau depan searah jarum jam. Kelompok kedua adalah: F', B', R', L', U', dan D' yang gerakannya adalah berlawanan dengan arah jarum jam. Untuk mendapatkan notasi-notasi di atas rubik harus diletakkan di depan pemain. Setiap notasi yang ditambahkan angka 2 dibelakangnya, berarti sisi/ *layer* tersebut diputar 180 derajat atau setengah putaran.

2. *Double Turn* (Putaran Ganda)

Notasi *double turn* adalah notasi yang digunakan untuk memutar dua sisi (*layer*) dari rubik sekaligus sebesar seperempat putaran (90 derajat). Implementasi gerakan notasi huruf pada rubik dapat dilihat pada Tabel II.3 sebagai berikut :

Tabel II.3. *Double Turn* Rubik (Putaran Ganda Rubik)

Notasi	Gerakan	Efek
f		
f'		
b		
b'		
u		
u'		

d		
d'		
r		
r'		
l		
l'		



3. *Slice Turn* (Putaran Iris)











Notasi slice turn adalah notasi yang digunakan untuk memutar sisi tengah rubik. Adapun notasi – notasi tersebut menggunakan huruf :

- a. M (*middle*)
- b. E (*Equator*)
- c. S (*Stand*)

Implementasi gerakan notasi huruf pada rubik dapat dilihat pada Tabel II.4 sebagai berikut :

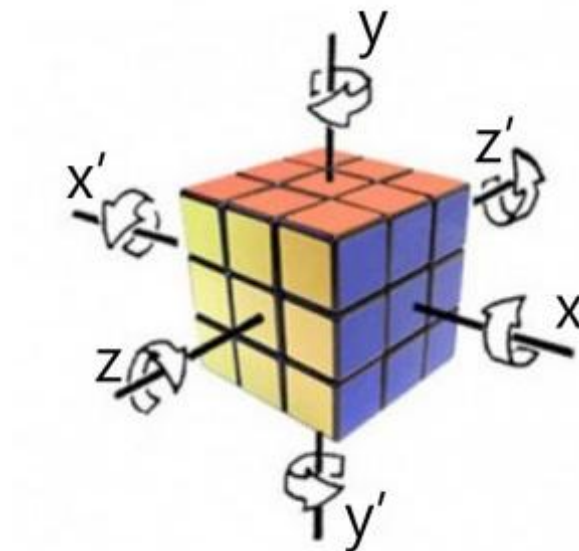
Tabel II.4. *Slice Turn* Rubik (Putaran Iris Rubik)

Notasi	Gerakan	Efek
M		

M'		
E		
E'		
S		
S'		

4. *Rotation* (Rotasi)

Pada rotasi, rubik yang diputar bukanlah lapis atau *layer*, melainkan *cube* secara keseluruhan. Untuk lebih jelas perhatikan Gambar II.6 dibawah ini :



Gambar II.6. *Rotation* Rubik (Putaran Rotasi Rubik)

II.3.3. Metode Penyelesaian Rubik

Metode yang sangat dikenal dalam penyelesaian rubik yaitu Metode Jessica Fridrich, Jessica Fridrich adalah seorang dosen bergelar Professor yang berhasil menemukan metode tercepat dalam menyelesaikan rubik. Metode ini dikenal dengan sebutan CFOP (*Cross-F2L-OLL-PLL*). Ada 4 tahap yang perlu dilalui untuk menyamakan keenam sisi Rubiks *cube* dari posisi yang paling acak sekalipun. Berikut pembahasannya:

1. *Cross*

Hal pertama yang harus dilakukan untuk menyelesaikan Rubiks *cube* adalah membuat *Cross*. *Cross* adalah tanda silang (atau tanda +) pada salah satu sisi warna *Rubik's cube*. *Cross* yang benar bisa dilihat di Gambar II.7 berikut:



Gambar II.7. Cross

Dari Gambar II.7 bisa kita lihat bahwa *cross* dibuat pada sisi putih. Dari gambar tersebut juga terlihat warna merah dan hijau (serta orange dan biru pada sisi sebaliknya) tergabung dengan center. Untuk membuat *cross* pada posisi *cube* teracak murni menggunakan logika dan latihan secara kontinyu. Pada awalnya seseorang pemula memerlukan waktu hingga 1 menit untuk membuat *cross*. Lambat laun proses ini bisa diturunkan hingga waktu ideal 2-3 detik.

2. F2L

Merupakan singkatan dari *First two layer* atau tahap dimana kita menyelesaikan 2 *layer* pertama. Dalam metode full fridrich kita membentuk pasangan lalu memasukkannya ke 4 slot yang tersedia untuk menyelesaikan *layer* 1 dan 2 sekaligus.



Gambar II.8. F2L

Cross dan F2L digunakan untuk menyelesaikan *layer* 1 dan *layer* 2. Pada metode pemula untuk menyelesaikan *layer* 1 dan 2 dibagi dalam 3 tahap yakni membuat *cross*, menyelesaikan *layer* pertama (dengan menempatkan 4 corner ke tempat asalnya) lalu menyelesaikan *layer* kedua (memasukkan edge ke tempat yang sesuai). Perbedaan dengan F2L adalah, pada F2L corner dan edge digabungkan terlebih dulu, lalu dimasukkan ketempatnya sehingga lebih menghemat langkah dan waktu. Sementara itu untuk menyelesaikan *layer* terakhir dibagi dalam 2 tahap, yakni mengoreksi arah (OLL) dan mengoreksi tempat (PLL).

3. OLL

Merupakan singkatan dari *Orientating Last layer*. Segera setelah *layer* 1 dan 2 selesai, akan terbentuk pola tertentu dan tujuan kita adalah menyamakan warna sehingga "*top face*" akan memiliki warna yang sama (seringnya warna kuning, jika *cross* dilakukan pada warna putih dengan skema warna standar). Ada

57 kasus pada OLL yang menyebabkan kita perlu menghafal 57 langkah yang berbeda. Waktu eksekusi ideal untuk tahap ini adalah 2-3 detik. Bagi mereka yang belum mampu 27 menghafal 57 langkah sekaligus bisa melakukan OLL dalam 2 tahap (sebagai konsekuensinya waktu akan lebih lambat), yakni membetulkan seluruh edge dan disusul membetulkan seluruh corner. Hal ini dikenal sebagai 2 look OLL. Hanya perlu mempelajari 1-2 langkah untuk membetulkan orientasi edge dan 7 langkah untuk membetulkan orientasi corner.



Gambar II.9. OLL

4. PLL

Merupakan singkatan dari *Permutation Last layer*. Disini kita saling menukar posisi *corner* dan *edge* ke tempat semula. Ada 21 langkah pada tahap ini yang secara ideal rata rata diseksekusi dalam 2-4 detik. Apabila sulit dalam menghafal 21 langkah bisa melakukannya dalam 2 tahap (dikenal sebagai 2 look PLL) yaitu mengoreksi letak *corner* disusul *edge*.



Gambar II.10. PLL

Pada prinsipnya pemecahan *cube* dengan metode Jessica Fridrich selalu melalui tahapan berikut:

1. Selesaikan *layer* pertama/atas
 - a. Bentuk *cross* disisi atas, sesuaikan dengan warna keempat sisi di samping
 - b. Isi keempat sudut atas dengan kubus yang sesuai.
2. Selesaikan *layer* kedua/tengah
 - a. Isi keempat kubus pada *layer* kedua dengan kubus yang sesuai.
3. Selesaikan *layer* ketiga/bawah
 - a. Bentuk *cross* di sisi bawah, tanpa merusak kedua *layer* di atas.
 - b. Tempatkan keempat sudut bawah di tempat sebenarnya, sesuaikan dengan warna dari ketiga sisi yang bersisian dengannya.
 - c. Bentuk supaya sisi bawah memiliki warna yang uniform.
 - d. Pertukarkan kubus-kubus yang belum sesuai pada tempatnya.

II.4. Mikrokontroler

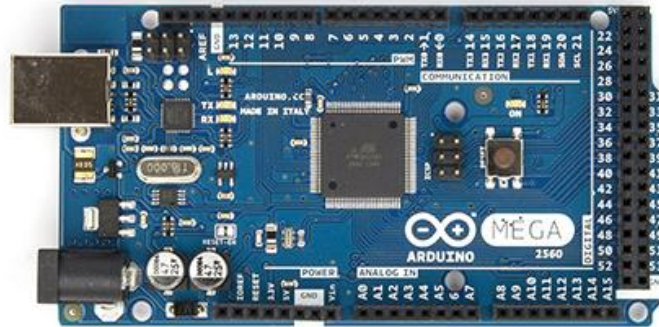
Mikrokontroler sering juga disebut sebagai mikrokomputer atau *embedded system*. Mikrokontroler dapat dipandang sebagai suatu system yang terdiri atas input, program, dan output. Perancang dapat mengatur perilaku mikrokontroler melalui program. Proses memasukkan program ke dalam mikrokontroler disebut proses *download* dan alat yang digunakan disebut *downloader*. Seperti sistem computer, nilai tambah sistem mikrokontroler dapat dilipatgandakan melalui program.

Kelebihan utama dari mikrokontroler ialah tersedianya RAM dan peralatan I/O pendukung sehingga ukuran board mikrokontroler menjadi sangat ringkas.

Dengan perkembangan terakhir, yaitu generasi AVR (*Aif Vegard's Risc Processor*), para desainer sistem elektronika telah diberi suatu teknologi yang memiliki kapabilitas yang amat maju, tetapi dengan biaya ekonomis yang cukup minimal. AVR berteknologi RISC (*Reduced Instruction Set Computing*). Secara umum, AVR dapat dikelompokkan menjadi 4 (empat) kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATMega, dan AT86RFxx. Pada dasarnya yang membedakan masing - masing kelas adalah memori, peripheral dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, mereka bisa dikatakan hampir sama. Oleh karena itu, dipergunakan salah satu AVR produk ATMEL, yaitu ATMega2560.

II.4.1. Arduino Mega2560

Arduino Mega2560 adalah papan mikrokontroler berbasis ATmega2560. Arduino Mega2560 memiliki 54 pin digital input/output, dimana 15 pin dapat digunakan sebagai output PWM, 16 pin sebagai analog, dan 4 pin sebagai UART (port serial hardware), 16 MHz Kristal osilator, koneksi USB, jack power, header ISCP, dan tombol reset. Ini semua yang diperlukan untuk mendukung mikrokontroler. Cukup dengan menghubungkannya ke computer melalui kabel USB atau power dihubungkan dengan adaptor AC-DC atau baterai untuk memulai mengaktifkannya. Arduino Mega2560 kompatibel dengan sebagian besar shield yang dirancang untuk Arduino Duemilanove atau Arduino Diecimila. Arduino Mega2560 adalah versi terbaru yang menggantikan versi Arduino Mega.



Gambar II.11 Arduino Mega2560 tampak depan

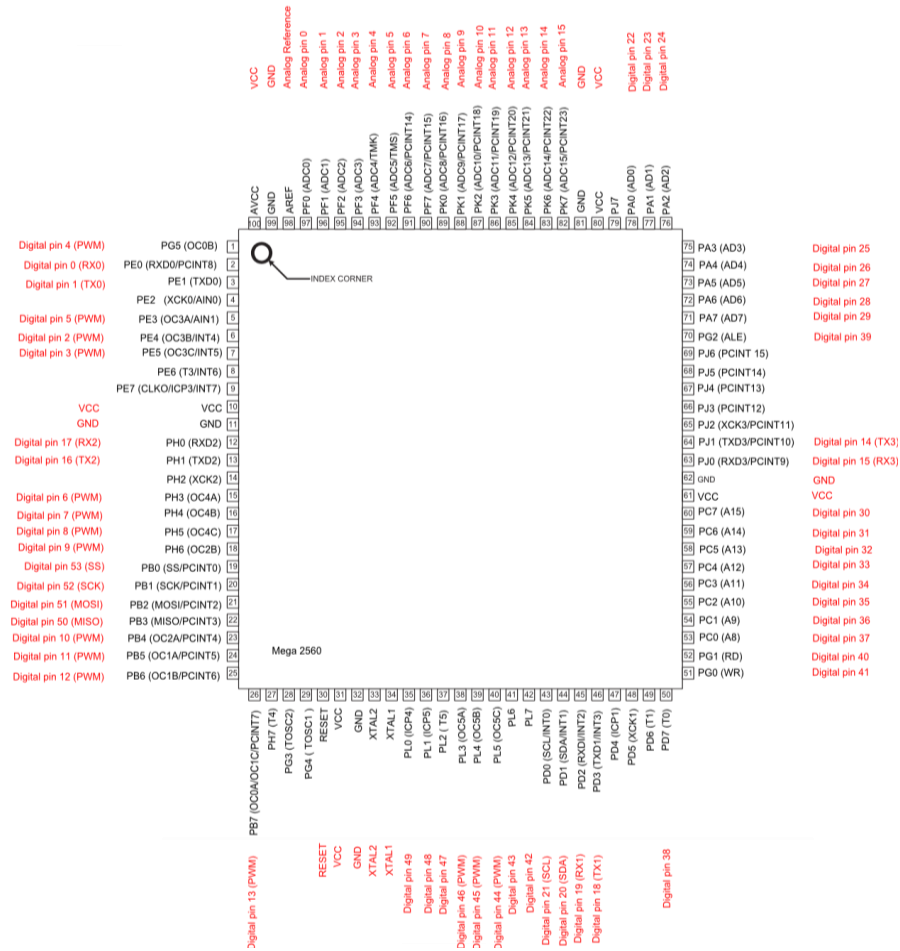
(sumber : www.hendriono.com)

Ringkasan Spesifikasi dari Arduino Mega2560 adalah sebagai berikut:

Mikrokontroler	: ATmega2560
Tegangan Operasi	: 5V
Input Voltage (disarankan)	: 7-12V
Input Voltage (limit)	: 6-20V
Pin Digital I/O	: 54 (yang 15 pin digunakan sebagai output PWM)
Pins Input Analog	: 16
Arus DC per pin I/O	: 40 mA
Arus DC untuk pin 3.3V	: 50 mA
Flash Memory	: 256 KB (8 KB digunakan untuk bootloader)
SRAM	: 8 KB
EEPROM	: 4 KB
Clock Speed	: 16 MHz

II.4.2. Konfigurasi Pin Arduino Mega2560

Dibawah ini konfigurasi pin ATmega2560 dengan Arduino Mega2560:



Gambar II.12. Konfigurasi pin ATmega2560 dengan Arduino Mega2560

(sumber : www.hendriono.com)

Berikut adalah tabel pemetaan pin-pin pada ATmega2560 dengan Arduino Mega2560:

Table II.5. Pemetaan pin Arduino Mega2560

Pin	Nama Pin	Peta Nama Pin
1	PG5 (OC0B)	Digital pin 4 (PWM)

2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)
3	PE1 (TXD0)	Digital pin 1 (TX0)
4	PE2 (XCK0/AIN0)	
5	PE3 (OC3A/AIN1)	Digital pin 5 (PWM)
6	PE4 (OC3B/INT4)	Digital pin 2 (PWM)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)
8	PE6 (T3/INT6)	
9	PE7 (CLKO/ICP3/INT7)	
10	VCC	VCC
11	GND	GND
12	PH0 (RXD2)	Digital pin 17 (RX2)
13	PH1 (TXD2)	Digital pin 16 (TX2)
14	PH2 (XCK2)	
15	PH3 (OC4A)	Digital pin 6 (PWM)
16	PH4 (OC4B)	Digital pin 7 (PWM)
17	PH5 (OC4C)	Digital pin 8 (PWM)
18	PH6 (OC2B)	Digital pin 9 (PWM)
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)
21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)
22	PB3 (MISO/PCINT3)	Digital pin 50 (MISO)
23	PB4 (OC2A/PCINT4)	Digital pin 10 (PWM)
24	PB5 (OC1A/PCINT5)	Digital pin 11 (PWM)
25	PB6 (OC1B/PCINT6)	Digital pin 12 (PWM)
26	PB7 (OC0A/OC1C/PCINT7)	Digital pin 13 (PWM)
27	PH7 (T4)	
28	PG3 (TOSC2)	
29	PG4 (TOSC1)	
30	RESET	RESET
31	VCC	VCC
32	GND	GND

33	XTAL2	XTAL2
34	XTAL1	XTAL1
35	PL0 (ICP4)	Digital pin 49
36	PL1 (ICP5)	Digital pin 48
37	PL2 (T5)	Digital pin 47
38	PL3 (OC5A)	Digital pin 46 (PWM)
39	PL4 (OC5B)	Digital pin 45 (PWM)
40	PL5 (OC5C)	Digital pin 44 (PWM)
41	PL6	Digital pin 43
42	PL7	Digital pin 42
43	PD0 (SCL/INT0)	Digital pin 21 (SCL)
44	PD1 (SDA/INT1)	Digital pin 20 (SDA)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)
47	PD4 (ICP1)	
48	PD5 (XCK1)	
49	PD6 (T1)	
50	PD7 (T0)	Digital pin 38
51	PG0 (WR)	Digital pin 41
52	PG1 (RD)	Digital pin 40
53	PC0 (A8)	Digital pin 37
54	PC1 (A9)	Digital pin 36
55	PC2 (A10)	Digital pin 35
56	PC3 (A11)	Digital pin 34
57	PC4 (A12)	Digital pin 33
58	PC5 (A13)	Digital pin 32
59	PC6 (A14)	Digital pin 31
60	PC7 (A15)	Digital pin 30
61	VCC	VCC
62	GND	GND
63	PJ0 (RXD3/PCINT9)	Digital pin 15 (RX3)

64	PJ1 (TXD3/PCINT10)	Digital pin 14 (TX3)
65	PJ2 (XCK3/PCINT11)	
66	PJ3 (PCINT12)	
67	PJ4 (PCINT13)	
68	PJ5 (PCINT14)	
69	PJ6 (PCINT 15)	
70	PG2 (ALE)	Digital pin 39
71	PA7 (AD7)	Digital pin 29
72	PA6 (AD6)	Digital pin 28
73	PA5 (AD5)	Digital pin 27
74	PA4 (AD4)	Digital pin 26
75	PA3 (AD3)	Digital pin 25
76	PA2 (AD2)	Digital pin 24
77	PA1 (AD1)	Digital pin 23
78	PA0 (AD0)	Digital pin 22
79	PJ7	
80	VCC	VCC
81	GND	GND
82	PK7 (ADC15/PCINT23)	Analog pin 15
83	PK6 (ADC14/PCINT22)	Analog pin 14
84	PK5 (ADC13/PCINT21)	Analog pin 13
85	PK4 (ADC12/PCINT20)	Analog pin 12
86	PK3 (ADC11/PCINT19)	Analog pin 11
87	PK2 (ADC10/PCINT18)	Analog pin 10
88	PK1 (ADC9/PCINT17)	Analog pin 9
89	PK0 (ADC8/PCINT16)	Analog pin 8
90	PF7 (ADC7)	Analog pin 7
91	PF6 (ADC6)	Analog pin 6
92	PF5 (ADC5/TMS)	Analog pin 5
93	PF4 (ADC4/TMK)	Analog pin 4
94	PF3 (ADC3)	Analog pin 3

95	PF2 (ADC2)	Analog pin 2
96	PF1 (ADC1)	Analog pin 1
97	PF0 (ADC0)	Analog pin 0
98	AREF	Analog Reference
99	GND	GND
100	AVCC	VCC

Berikut ini adalah fungsi dari pin-pin pada Arduino Mega2560:

1. VIN

Adalah input tegangan untuk papan Arduino ketika menggunakan sumber daya eksternal (sebagai 'saingan' tegangan 5 Volt dari koneksi USB atau sumber daya ter-regulator lainnya). Anda dapat memberikan tegangan melalui pin ini, atau jika memasok tegangan untuk papan melalui jack power, kita bisa mengakses/mengambil tegangan melalui pin ini.

2. 5V

Sebuah pin yang mengeluarkan tegangan ter-regulator 5 Volt, dari pin ini tegangan sudah diatur (ter-regulator) dari regulator yang tersedia (built-in) pada papan. Arduino dapat diaktifkan dengan sumber daya baik berasal dari jack power DC (7-12 Volt), konektor USB (5 Volt), atau pin VIN pada board (7-12 Volt). Memberikan tegangan melalui pin 5V atau 3.3V secara langsung tanpa melewati regulator dapat merusak papan Arduino.

3. 3V3

Sebuah pin yang menghasilkan tegangan 3,3 Volt. Tegangan ini dihasilkan oleh regulator yang terdapat pada papan (on-board). Arus maksimum yang dihasilkan adalah 50 mA.

4. GND

Pin Ground atau Massa.

5. IOREF

Pin ini pada papan Arduino berfungsi untuk memberikan referensi tegangan yang beroperasi pada mikrokontroler. Sebuah perisai (shield) dikonfigurasi dengan benar untuk dapat membaca pin tegangan IOREF dan memilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan (voltage translator) pada output untuk bekerja pada tegangan 5 Volt atau 3,3 Volt.

6. Input dan Output

Masing-masing dari 54 digital pin pada Arduino Mega dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()` , `digitalWrite()` , dan `digitalRead()`. Arduino Mega beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor pull-up internal (yang terputus secara default) sebesar 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus, antara lain:

7. Serial

0 (RX) dan 1 (TX); **Serial 1** : 19 (RX) dan 18 (TX); **Serial 2** : 17 (RX) dan 16 (TX); **Serial 3** : 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Pins 0 dan 1 juga terhubung ke pin chip ATmega16U2 Serial USB-to-TTL.

8. Eksternal Interupsi

Pin 2 (interrupt 0), pin 3 (interrupt 1), pin 18 (interrupt 5), pin 19 (interrupt 4), pin 20 (interrupt 3), dan pin 21 (interrupt 2). Pin ini dapat dikonfigurasi untuk

memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubah nilai.

9. SPI

Pin 50 (MISO), pin 51 (MOSI), pin 52 (SCK), pin 53 (SS). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI. Pin SPI juga terhubung dengan header ICSP, yang secara fisik kompatibel dengan Arduino Uno, Arduino Duemilanove dan Arduino Diecimila.

10. LED

Pin 13. Tersedia secara built-in pada papan Arduino ATmega2560. LED terhubung ke pin digital 13. Ketika pin diset bernilai HIGH, maka LED menyala (ON), dan ketika pin diset bernilai LOW, maka LED padam (OFF).

11. TWI

Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan perpustakaan *Wire*.

12. AREF

Referensi tegangan untuk input analog. Digunakan dengan fungsi `analogReference()`.

13. RESET

Jalur LOW ini digunakan untuk me-reset (menghidupkan ulang) mikrokontroler. Jalur ini biasanya digunakan untuk menambahkan tombol reset pada shield yang menghalangi papan utama Arduino.

II.5. Keypad

Keypad adalah bagian penting dari suatu perangkat elektronika yang membutuhkan interaksi manusia. *Keypad* berfungsi sebagai *interface* antara perangkat (mesin) elektronik dengan manusia atau dikenal dengan HMI (*human machine interface*). Matrik *keypad* 4X4 ini merupakan salah satu contoh *keypad* yang dapat digunakan untuk berkomunikasi antara manusia dengan mikrokontroler. (Sugiarto dan Roby Chaerulloh : 2014).



Gambar II.13. Keypad

(Sumber Sugiarto dan Roby Chaerulloh : 2014)

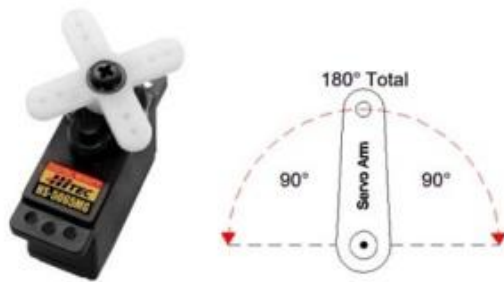
II.6. Motor Servo

Motor servo atau biasa disingkat servo saja merupakan sebuah perangkat yang merupakan gabungan dari beberapa komponen berikut:

- a. Motor DC brush.
- b. Reduksi gear untuk meningkatkan torsi.
- c. Sensor posisi dan rangkaian kontrol elektrik.

Kebanyakan motor servo berputar sekitar 90-180 derajat. Namun beberapa diantaranya ada yang dapat berputar 360 derajat. Hanya saja motor ini tidak dapat berputar terus menerus sehingga tidak dapat digunakan untuk mengendalikan roda (kecuali dimodifikasi).

Motor servo dapat dimanfaatkan pada pembuatan robot, salah satunya sebagai penggerak lengan robot, karena memiliki tenaga atau torsi yang besar, sehingga dapat menggerakkan beban yang berat. Pada umumnya motor servo yang digunakan sebagai penggerak pada robot adalah motor servo 180⁰.



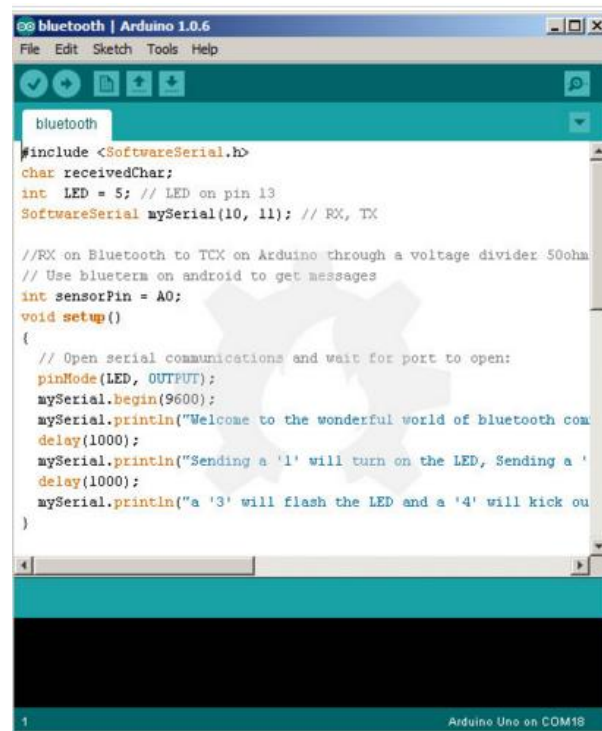
Gambar II.14. Motor Servo

(Sumber : www.zonaelektro.net)

Untuk menjalankan atau mengendalikan motor servo berbeda dengan motor DC. Karena untuk mengendalikan motor servo perlu diberikan sumber dan sinyal kontrol. Besarnya sumber tegangan tergantung dari spesifikasi motor servo yang digunakan.

II.7. Perangkat Lunak dan Bahasa Pemrograman

Agar mikrokontroler dapat bekerja maka digunakan perangkat lunak dan pemrograman sebagai pengkondisian dan perintah-perintah yang diinginkan oleh pembuat alat. Perangkat lunak yang penulis gunakan adalah Arduino IDE. Arduino IDE (*Integrated Development Environment*) digunakan untuk menulis dan meng-*upload* kode dari komputer ke papan fisik (*hardware*) Arduino. Program yang ditulis menggunakan *Software* Arduino IDE disebut *sketch*. *Sketch* ditulis di teks editor dan disimpan dengan *extension* .ino.



Gambar II.15. Tampilan Software Editor Arduino

Arduino IDE menggunakan bahasa pemrograman C++ dengan versi yang telah disederhanakan, sehingga lebih mudah dalam belajar pemrograman. Arduino menggunakan *software processing* yang digunakan untuk menulis program kedalam Arduino. *Processing* sendiri merupakan penggabungan antara bahasa C++ dan Java. *Software* Arduino ini dapat di-install diberbagai *operating system* (OS) seperti: LINUX, Mac OS, Windows. *Software* IDE Arduino terdiri dari 3 (tiga) bagian:

1. Editor program, untuk menulis dan mengedit program dalam bahasa *processing*.

2. *Compiler*, modul yang berfungsi mengubah bahasa *processing* (kode program) kedalam kode biner karena kode biner adalah satu-satunya bahasa program yang dipahami oleh mikrokontroler.
3. *Uploader*, modul yang berfungsi memasukkan kode biner kedalam memori mikrokontroler.

Struktur perintah pada Arduino secara garis besar terdiri dari 2 (dua) bagian yaitu *void setup* dan *void loop*. *Void setup* berisi perintah yang akan dieksekusi hanya satu kali sejak Arduino dihidupkan sedangkan *void loop* berisi perintah yang akan dieksekusi berulang-ulang selama arduino dinyalakan.