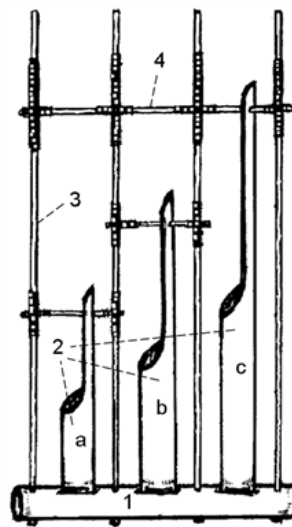


BAB II

TEORI PENUNJANG

2.1 Angklung

Angklung merupakan alat musik tradisional yang seluruh bagian tubuhnya terbuat dari bambu. Asal-usul kata angklung berasal dari kata angka (nada) dan lung (patah/hilang), maka tidak aneh apabila ketika mendengar bunyi dari alat musik angklung itu terdengar seperti terpatah-patah. Karakter bunyi yang khas dari angklung adalah “klung..klung..” Bunyi yang dihasilkan sedemikian rupa bersumber dari pergerakan antara tabung-tabung bambu yang disusun secara vertikal dengan tabung bambu yang dijadikan sebagai penopangnya. Sumber bunyi alat musik ini terletak pada lubang tabung bambu yang telah diraut menyerong pada bagian badannya [1].



Gambar 2.1 Bentuk Angklung

(Sumber : <http://widyasunda-permana.blogspot.com/2012/12/angklung.html>)

Angklung terdiri atas susunan dua tabung bambu dengan perbedaan ukuran panjang yaitu tabung bambu panjang dan pendek. Tabung-tabung tersebut disusun secara vertikal dengan arah lubang resonansi yang searah. Dalam perkembangannya angklung dikembangkan dengan susunan hingga tiga tabung bambu.

Angklung di Jawa Barat terdiri atas dua tabung dan tiga tabung, dan ditingkai sedemikian rupa untuk menempatkan tabung-tabung angklung. Tabung tersebut terdiri atas tabung besar dan tabung kecil yang mempunyai jarak interval satu oktaf. Sumber bunyi dalam angklung yaitu tabung yang bergerak ke kanan dan kekiri dalam sebuah bingkai yang digoyang-goyangkan (*shake*).

Belum ada sumber yang jelas mengenai kapan alat musik angklung muncul. Namun Prier (1991: 18) menyatakan bahwa:

“Pada waktu orang hindu datang ke Jawa, mereka telah menemukan macam-macam alat musik. Dalam relief candi Borobudur terdapat alat musik lokal maupun alat musik yang di impor dari India yaitu gendang, termasuk gendang dari tanah dengan kulit hanya di satu sisi, kledi, suling, *angklung*, alat tiup semacam hobo, xylophone (bentuknya setengah gambang setengah calung), sapek, sitar, harpa, dengan 10 dawai, lonceng dari perunggu dalam macam-macam ukuran, gong, saron dan bonang.”

Pada saat ini, sebagian besar kesenian angklung yang ada hanya disajikan untuk kebutuhan hiburan saja. Padahal dimasa lalu angklung pernah digunakan sebagai upacara ritual. Musik angklung secara fungsional bergeser menjadi seni pertunjukan, meskipun di beberapa tempat masih berfungsi sebagai bagian dari upacara. Angklung juga kerap di hubungkan dengan kepercayaan masyarakat dan padi. Ritual penanaman padi tersebut bertujuan untuk menghaturkan rasa syukur dan menghargai sosok Nyi Pohaci Sanghyang Asri oleh masyarakat Kanéhés (golongan masyarakat Sunda), karena diyakini bahwa peranNya berpengaruh pada kelangsungan daur hidup mereka. Bahwa pada awalnya angklung merupakan salah satu alat bunyi-bunyian yang digunakan untuk upacara-upacara yang berhubungan dengan padi. Dari pernyataan tersebut nampak bahwa adanya suatu transformasi yang terjadi pada angklung dari berupa medium dalam suatu ritual penanaman padi hingga menjadi alat musik pertunjukan.

Angklung merupakan alat musik perkusi bernada, yang dilihat dari sumber bunyinya termasuk ke dalam jenis *idiophone*. Sachs, Mahillon dan Horbostel bahwa *idiophone* adalah jenis alat musik yang sumber bunyinya dari sentuhan atas badan alat musik itu sendiri. Menurut Kawit dalam Masunah, kesenian yang termasuk ke

dalam rumpun angklung di Jawa Barat berjumlah 21 jenis. Sampai saat ini banyak ragam jenis angklung di Jawa Barat yang masih bertahan, namun beberapa lainnya dalam kondisi hampir punah. Ragam jenis angklung yang tersebar di Jawa Barat yang masih bertahan sampai saat ini diantaranya adalah sebagai berikut :

Tabel 2.1 Jenis-jenis angklung yang berkembang di Jawa barat

Jenis Angklung	Tempat	Peralatan	Fungsi	Ciri Musik
Angklung <i>Dogdog</i> Lojor	Ciptarasa, Cisorolok Kabupaten Sukabumi	Dua <i>dogdog</i> lojor dan empat buah angklung	Ritual pertanian	1. <i>Salendro</i> 2. Terdapat vocal 3. Bersifat ritmis 4. <i>Interlocking</i> (carukan/imbal)
Angklung Badeng	Desa Sanding, Malangbong Garut	Tiga Vokal, Sembilan buah angklung, empat <i>dogdog</i> biasa dan dua buah terbang	Penyebaran agama Islam dan tontonan	1. <i>Salendro</i> 2. Vokal lebih mandiri 3. Lebih ritmis 4. <i>Interlocking</i> (carukan/imbal)
Angklung Buncis	Desa baros, Arjasari, Bandung	Tiga <i>dogdog</i> , satu tarompet, Sembilan angklung, song/kempul, <i>sinden (vocalis)</i>	Seni pertunjukan, hiburan	1. Salendro 2. Vokal lebih mandiri 3. Kadang-kadang ada melodi tarompet 4. Pukulan gong lebih menonjol 5. Lebih melodis 6. <i>Interlocking</i> (carukan/imbal)
Angklung Badud	Dusun Margajaya, Desa Margacinta, Cijulang Pangandaran	Enam <i>dogdog</i> , delapan angklung	Ritual dan sekarang berkembang menjadi bersifat tontonan	1. Salendro 2. Bersifat lebih melodis 3. <i>Interlocking</i> (carukan/imbal)
Angklung Pa Daeng	Jawa Barat	Angklung ansambel berjumlah 17-100 lebih, ditambah alat musik lain.	Tontonan, hiburan untuk pariwisata	1. Diatonis 2. Bersifat lebih melodis 3. Lebih orkestral

(Sumber: Masunah (2003: 4-5))

Angklung secara umum terbagi menjadi dua menurut jenisnya yaitu angklung buhun (tradisional) dan angklung Pa Daeng (modern). Dari sisi bentuk, angklung Pa Daeng tidak ada perbedaan yang mendasar dengan angklung buhun, hanya saja tangga nadanya berbeda. Angklung Pa Daeng menggunakan tangga nada *diatonis kromatis* sedangkan angklung buhun menggunakan tangga nada *pentatonis salendro* [2].

Angklung modern tersebut merupakan hasil inovasi dari Daeng Soetigna (Pa Daeng) yang dikembangkan dengan bekal keilmuan musik Baratnya. Dengan konsep angklung bertangga nada *diatonis kromatis* tersebut membuat kiprah angklung menjadi semakin populer di khalayak umum, karena menjadikannya dapat dimainkan oleh siapapun bahkan jangkauan materi lagunya lebih luas. Alat musik sederhana ini tidak hanya populer di Indonesia, melainkan juga di kalangan mancanegara. Di mancanegara, angklung diterima dengan baik dan senang oleh banyak pihak. Atas prestasi yang telah dirajut oleh angklung baik di aspek pertunjukan maupun pendidikan, Pemerintah Indonesia menetapkan angklung sebagai alat musik pendidikan. Hal tersebut dinyatakan pada tanggal 23 Agustus 1968 dalam SK Menteri Pendidikan dan Kebudayaan Np.082/1986 yang ditandatangani oleh Sekretaris Jenderal atas nama Menteri Pendidikan dan Kebudayaan.

Ada juga angklung jenis lain yang sudah di modifikasi dari segi rangka tempat angklungnya yaitu angklung toel ciptaan kang Yayan Udjo, satu set angklung bisa dimainkan seorang diri saja. angklung toel ini terdiri dari sebuah rangka kayu yang mewadahi 30 angklung dari nada G3 – C6. Angklung dipasang berjejer dalam 2 sap. Sap bawah (dekat pemain) adalah nada-nada penuh (G, A, B, C, dst), sementara sap atas adalah nada-nada kromatis (G#, A#, C#, dst).



Gambar 2.2 Bentuk Angklung Toel
(Sumber : <http://klungbot.com/angklung-toel/>)

Setiap angklung dipasang terbalik. Di bawah, topi angklung diikat ke suatu palang miring dengan kawat. Sementara itu di sebelah atas, tabung dasar angklung diikat dengan tali elastis ke palang mendatar. Dulu, waktu model pertama, tali elastis ini memakai karet gelang yang disambung-sambung. Saat ini yang digunakan sudah karet hitam yang bagus. Tali elastis ini sangat krusial dalam angklung toel, dan sampai saat ini masih ingin dicari bahan yang lebih baik.

Selanjutnya, pada tabung dasar ini dipasang pasak sekitar 5 cm dari bambu, atau kayu. Jika pasak ini di-toel (ditarik maju sedikit, lalu dilepas), maka angklung akan bergetar beberapa kali akibat ditarik tali elastis. Nah, dengan demikian seorang pemain bisa memakai dua tangannya untuk bergantian memainkan salah satu angklung.

Dari keseluruhan jenis angklung yang ada, angklung Pa Daeng dan angklung toel ciptaan kang Yayan Udjo dipilih sebagai fokus objek penelitian ini. Angklung Pa Daeng menjadi angklung yang secara umum dapat dimainkan berbagai kalangan dan telah digemari dimana-mana baik sebagai media pertunjukan maupun pendidikan.

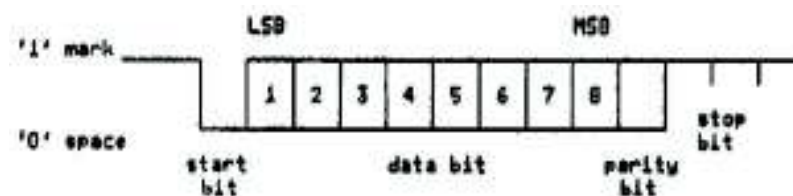
2.2 Komunikasi Serial

Komunikasi serial adalah metode komunikasi data yang dimana hanya satu bit data yang dikirimkan melalui kabel pada suatu waktu tertentu. Komunikasi serial ada dua macam, yaitu *Synchronous* dan *Asynchronous*. *Synchronous* adalah tipe

komunikasi serial dimana hanya ada satu pihak pengirim atau penerima menghasilkan *clock* dan mengirimkan *clock* tersebut bersama-sama dengan data sedangkan tipe *Asynchronous*, yaitu kedua pihak (pengirim dan penerima) masing-masing menghasilkan *clock*, namun hanya data yang ditransmisikan (tanpa *clock*). Agar data yang dikirim sama dengan data yang diterima, maka kedua frekuensi *clock* harus sama dan harus terdapat sinkronisasi. Setelah adanya sinkronisasi, pengirim akan mengirimkan datanya sesuai dengan frekuensi *clock* pengirim dan penerima akan membaca data sesuai dengan frekuensi *clock* penerima [3].

2.2.1 Transmisi Data *Asynchronous*

Pada transmisi data *asynchronous* setiap data karakter mempunyai satu bit yang berfungsi sebagai *start* bit dan satu atau dua bit yang berfungsi sebagai *stop* bit. Selain itu setiap data karakter juga dilengkapi dengan bit *parity* yang berfungsi untuk mendeteksi kesalahan data yang terjadi pada saat transmisi. Karena setiap karakter masing-masing diidentifikasi sendiri, karakter dapat dikirimkan setiap saat (*asynchronously*) tidak tergantung dengan karakter yang lainnya. Gambar 2.3 menunjukkan bentuk umum format data transmisi serial *asynchronous* [4].



Gambar 2.3 Format Data *Asynchronous*

Bilamana tidak ada data yang dikirim, sinyal data akan tetap berlogika '1', keadaan ini dinamakan *marking*. Awal dari suatu data karakter ditunjukkan dengan adanya transisi dari keadaan *marking* menuju logika '0' selama waktu '1' bit. Bit inilah yang dinamakan *start* bit. Setelah *start* bit, bit-bit data dikirimkan satu persatu secara bergantian dimulai dengan *least significant bit (LSB)* dahulu. Data dapat berisi 5, 6, 7, atau 8 bit bergantung pada sistem yang ada. Bit *parity* dikirimkan setelah bit-bit data selesai ditransmisikan untuk mendeteksi bilamana ada kesalahan saat transmisi berlangsung. Sinyal data kemudian akan berlogika '1'

selama paling sedikit waktu 1 bit untuk menunjukkan bahwa karakter sudah berakhir. Bit yang selalu berlogika '1' ini dinamakan *stop* bit.

Transmisi data *serial asynchronous* ini, kecepatan transmisi dinyatakan dengan istilah *baud rate*. Dimana *baud rate* ini didefinisikan sebagai seperwaktu yang diperlukan untuk 1 sel bit. Bilamana waktu yang diperlukan 1 sel bit sebesar 3,33 ms maka *baud rate* adalah $1/3,33$ atau sebesar 300 bps. *Baud rate* yang umum digunakan pada komunikasi serial adalah 110, 300, 1200, 2400, 4800, 9600, 19200 bps. Untuk membuat suatu interface antara suatu komputer mikro dengan jalur data *serial*, maka pada sisi kirim data harus diubah dari *parallel* menjadi *serial* dan pada sisi terima data harus diubah dari serial menjadi *parallel*. Untuk itu diperlukan suatu '*parallel-in serial-out shift register*' pada sisi kirim dan '*serial-in paralelout shift register*' pada sisi terima. Selain itu juga diperlukan rangkaian *hand-shaking* yang berfungsi agar bagian pengiriman data tidak mengirimkan data terlalu cepat yang dapat mengakibatkan bagian penerimaan data tidak sempat untuk menerima data.

2.2.2 Komunikasi Data UART

IC UART (*Universal Asynchronous Receiver Transmitter*) dibuat khusus untuk mengubah data parallel menjadi data serial dan menerima data serial yang kemudian dirubah lagi menjadi data parallel. Pada UART, kecepatan pengiriman data (*Baud Rate*) dan *fase clock* pada sisi *transmitter* dan sisi *receiver* harus sinkron, untuk itu diperlukan sinkronisasi antara *transmitter* dan *receiver*. Hal ini dilakukan oleh bit "*Start*" dan bit "*Stop*". Ketika saluran transmisi dalam keadaan *idle*, *output* UART adalah dalam keadaan logika "1" [5].



Gambar 2.4 Pengiriman huruf "A" tanpa bit paritas.

Ketika *transmitter* ingin mengirimkan data, *output* UART akan diset dulu ke logika "0" untuk waktu satu bit. Sinyal ini pada *receiver* akan dikenali sebagai sinyal "*Start*" yang digunakan untuk menyinkronkan *fase clocknya* sehingga

sinkron dengan *fase clock transmitter*. Selanjutnya data akan dikirimkan secara serial dari bit yang paling rendah (bit 0) sampai bit tertinggi. Selanjutnya akan dikirimkan sinyal "*Stop*" sebagai akhir dari pengiriman data serial. Sebagai contoh misalnya akan dikirimkan data huruf "A" dalam *format ASCII* (atau sama dengan 41 heksa atau 0100 0001).

Kecepatan transmisi dapat dipilih bebas dalam rentang tertentu. *Baud rate* yang umum dipakai adalah 110, 135, 150, 300, 600, 1200, 2400, dan 9600 (bit/per detik). Dalam komunikasi data serial, *baud rate* dari kedua alat yang berhubungan harus diatur pada kecepatan yang sama. Selanjutnya harus ditentukan panjang data (6,7 atau 8 bit), paritas (genap, ganjil, atau tanpa paritas), dan jumlah bit "*Stop*" (1, 1 1 2 , atau 2 bit). Untuk dapat menggunakan *port* serial harus diketahui dahulu alamat dari *port* serial tersebut. Biasanya tersedia dua *port* serial pada CPU, yaitu COM1 dan COM2.

2.3 MIDI (*Musical Instrument Digital Interface*)

Musical Instrument Digital Interface (MIDI) adalah sebuah standar hardware dan software internasional untuk saling bertukar data (seperti kode musik) di antara perangkat musik elektronik dan komputer dari merek yang berbeda. Standar MIDI ditetapkan pada tahun 1982 yang memungkinkan alat-alat musik elektronik seperti keyboard, komputer, dan peralatan elektronik lainnya untuk saling berkomunikasi, melakukan kontrol, serta sinkronisasi dengan peralatan music lain [6].

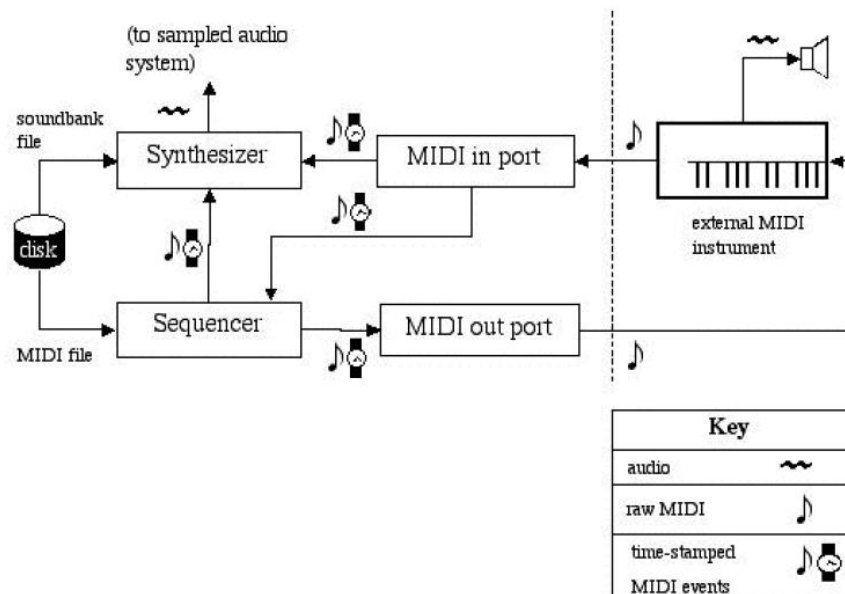
Beberapa definisi MIDI adalah:

- Media untuk menghubungkan antara perangkat musik, komputer, dan perangkat lainnya.
- Media untuk memainkan suara yang telah diolah.
- Media untuk standarisasi protokol komunikasi.
- Berguna sebagai media untuk mengendalikan perangkat non-musical, seperti sistem pencahayaan pada sebuah pertunjukkan teater, tape rekorder, dan lain-lain.

- Sebuah kegiatan yang berhubungan erat dengan proses tetapi tidak dengan audio.
- Bentuk nyata dari proses pembuatan musik.
- Media yang dapat diubah-ubah dengan mudah untuk mengganti not, waktu, dan bahkan bunyi yang dihasilkan oleh instrument.

Pada dasarnya, MIDI tidak lebih dari protokol komunikasi. MIDI digunakan untuk mengirim informasi mengenai proses kinerja musical. MIDI tidak mengenal bunyi dari not yang dihasilkan, akan tetapi MIDI dapat mengenali not apa yang dimainkan menurut waktu dan cara not tersebut dihasilkan. Sumber bunyi, seperti *synthesizer*, menerima data MIDI dan menghasilkan suara sebagai respon. Jika *synthesizer* diatur untuk membunyikan suara terompet, maka MIDI *event* membuat itu untuk menjadi suara yang menyerupai terompet. Dalam hal ini, MIDI seperti partiture music, bukan berupa suara. Namun tidak sepenuhnya seperti partiture music. MIDI *messages* dapat mengalir melalui kabel dan menyebabkan *synthesizer* menghasilkan suara seperti yang diinginkan [7].

2.3.1 Konfigurasi MIDI



Gambar 2.5 Konfigurasi MIDI
Sumber: (java.sun.com)

Dalam contoh ini, aplikasi program pada Gambar 2.5 memainkan musik dengan memuat *musical score* yang ada pada *disk*. *Standart MIDI file* memiliki *track* yang juga merupakan bagian dari *MIDI events*. Sebagian *events* memainkan not baik yang berisi melodi maupun ritme. *MIDI file* yang dibaca dan dimainkan oleh piranti lunak berbentuk *sequencer*.

Sequencer adalah sebuah perangkat untuk merekam dan memainkan sebuah *sequence* *MIDI event*. *Sequencer* tersebut memainkan musiknya dengan mengirimkan *MIDI messages* kepada perangkat lainnya seperti internal atau eksternal kepada perangkat lainnya seperti internal atau eksternal *synthesizer*.

Synthesizer adalah alat untuk menghasilkan suara. Suatu *synthesizer* dapat menampung serangkaian objek dari *MIDI Channel* khususnya 16 *channel* *MIDI* yang ada. *Synthesizer* itu sendiri akan membaca data berupa kumpulan suara termasuk intruksi untuk mengemulasikan suara dari jenis instrument music tertentu. Jika tidak demikian maka *synthesizer* akan memainkan not yang tersimpan di *MIDI file* dengan menggunakan suara yang telah ada pada *synthesizer* tersebut [7].

2.4 Protokol pada MIDI

MIDI adalah protocol berbentuk serial dimana setiap informasi dikemas dalam bentuk bit dan dikirimkan melalui kabel dan bit berikutnya mengikuti, dan seterusnya. Setiap *byte* yang dikirim pada 31.250 *baud* yang didalamnya terdapat 10 bit, terdiri atas *start* bit, 8 data bit, dan *stop* bit. Jadi 31.250 *byte* dapat dikirimkan setiap detik atau setiap *byte* memerlukan waktu 320 *microsecond* untuk dikirim [7].

2.4.1 MIDI Messages

MIDI message adalah kumpulan data yang memiliki spesifikasi secara penuh dalam sebuah *event*. Sebagian besar *messages* memiliki Panjang satu, dua, atau tiga *byte* atau lebih dari itu. Setiap tiga-*byte messages* dihasilkan sebuah *NoteOn*. Contohnya, memerlukan 96 *microsecond*, hampir mendekati satu *millisecond* untuk mengirim. Pada intinya, lebih dari sekitar 1000 *not* dapat dikirim dalam waktu satu detik [7].

Octave	Note Numbers											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127				

Gambar 2.6 Nomor Note MIDI

(Sumber : <https://www.instructables.com/id/What-is-MIDI/>)

Selain itu, *MIDI messages* juga memiliki satu *status byte* dan *note* atau lebih data *byte*. Nilai numerik dari satu *status byte* selalu terpaut antara 128 dan 255 sedangkan semua data *byte* terpaut 0 sampai dengan 127. Perbedaan ini memudahkan proses identifikasi antara *status* dan *data*, tetapi batas dari nilai satu data berisi atas tujuh bit.

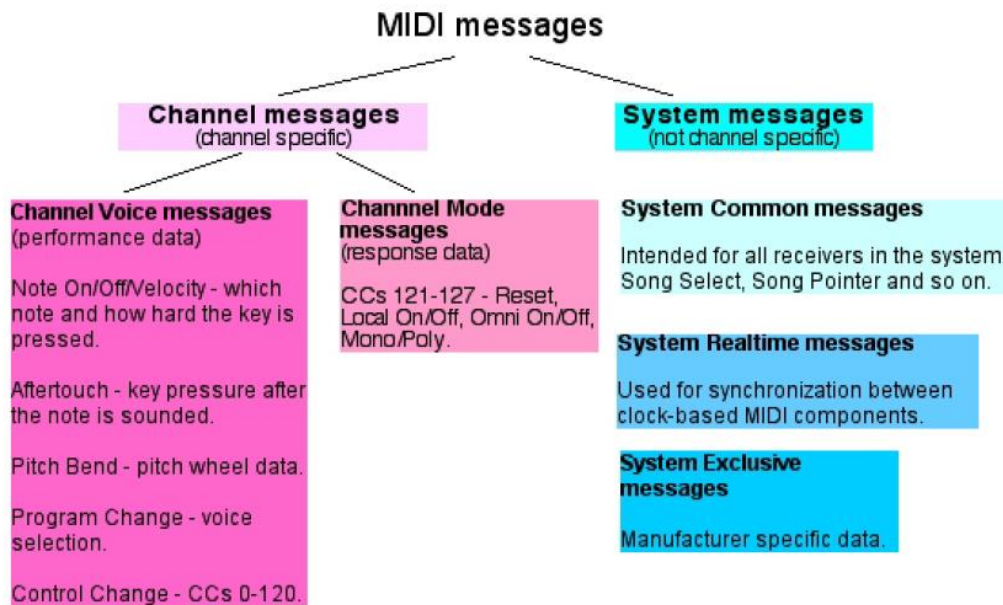
Pada dasarnya MIDI messages terdapat dua jenis yaitu:

a. *Channel Messages*

Channel messages lebih mengarah kepada hal-hal yang bersifat detail dan dibagi atas *Channel Voice messages* (untuk mengidentifikasi not) dan *Channel Mode messages* (untuk mengidentifikasi penerima).

b. *System Messages*

System messages sendiri terbagi tiga yaitu, *System Common messages* (berisi data untuk mengenali perangkat yang ada dalam sebuah sistem). *System Real Time messages* (menangani waktu dalam satu *event* yang berhubungan, (*System Exclusive message (messages)* untuk menangani semua yang belum terpikirkan oleh pembuat MIDI).



Gambar 2.7 Pembagian MIDI Messages

Sumber: (www.musicplus.com)

2.4.2 Channel Voice Messages

Sejauh ini sebagian besar cara pengiriman data yang menggunakan MIDI adalah *Channel Voice messages* atau sering disebut *channel messages*. Messages dapat diarahkan ke dalam perangkat detail yang diinginkan (dengan cara mengatur *channel* yang ada). Semua *Channel messages* mengidentifikasi informasi dari permainan seperti not yang dimainkan dan bagaimana jenis suara yang dihasilkan oleh not tersebut. Contohnya sebuah *Note On Messages* dapat mengenali not apa yang dimainkan serta bagaimana cara not tersebut dimainkan, dan *Pitch Bend messages* akan mempengaruhi *pitch* dari not yang berbunyi.

Empat bit *status byte* dari *Channel messages* menentukan *channel* dari masing-masing not tersebut sedangkan empat bit yang lain mengindikasikan seperti apa *channel messages* yang ada. Untuk meminimalkan jumlah bit yang dibutuhkan channel satu sampai dengan enam belas ditetapkan sebagai nol sampai dengan lima belas (0x00 – 0x0F) dalam empat bit yang ada pada awal *status byte*. Pada Tabel 2.1 di bawah ini terlampir spesifikasi MIDI *messages* dalam 8 data bit.

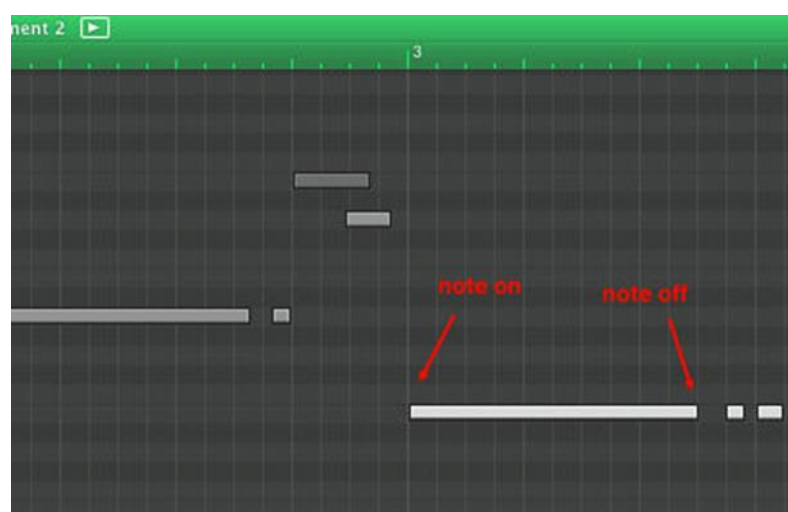
Tabel 2.2 Channel Voice Messages

No.	Status byte (Biner)	Data Byte	Message
1.	10000000	Note, Velocity	Note Off
2.	10010000	Note, Velocity	Note On
3.	10100000	Note, Pressure	Key Aftertouch
4.	10110000	Control, Value	Control Change
5.	11000000	Prog	Program Change
6.	11010000	Pressure	Channel Aftertouch
7.	11100000	LSB,MSB	Pitch Bend

Sumber: (Maximum MIDI)

2.4.2.1 Note On/Off

Dua messages ini adalah sumber suara dalam MIDI. Masing – masing memiliki pitch dari setiap note dan velocity. Pada umumnya nada C pada tengah – tengah keyboard mempunyai nilai not MIDI 60. Beberapa pitch diluar jangkauan dengar yang terdeteksi bisa dimainkan dengan MIDI, contohnya seperti canine (frekuensi suara yang hanya bisa ditangkap oleh hewan seperti anjing atau kelelawar) atau bunyi gerak aktivitas dalam bumi. *Velocity* memiliki pengertian secara umum sebagai sesuatu yang berhubungan terhadap keras-lembutnya bunyi: 0 untuk diam, 64 untuk *mezzopiano (mp)*, dan 127 untuk *fortississimo (fff)*.



Gambar 2.8 Keterangan Note ON dan Note OFF
(Sumber : <https://www.instructables.com/id/What-is-MIDI/>)

Note On akan membunyikan not pada *pitch* yang ditentukan. Sekali dimulai, sebuah not harus dihentikan, kalau tidak akan menjadi not yang terperangkap (bunyi terus menerus). MIDI memiliki dua cara untuk menghentikan sebuah not dari bunyi: *Note Off message* dan *Note On message* dengan *velocity* 0 (nol). *Note Off message* sama dengan *Note On message*. Hal tersebut mendeteksi not-not yang dituju dan *velocity*. Bagaimana pun, *velocity* dapat mendeteksi bagaimana sebuah not bisa dibunyikan pada saat ditekan atau berhenti pada saat dilepas. *Note On* pada *velocity* 0 (nol) dapat bekerja dengan baik. Apabila *velocity* berhubungan dengan keras-lembutnya suara, maka akan mungkin pada *velocity* 0 (nol) akan menghasilkan diam. Mengakhiri sebuah not dengan menggunakan *Note On message* pada *velocity* 0 (nol) merupakan hal yang sudah umum digunakan. *Note On* dan *Note Off*, keduanya memiliki status nilai yang sama. Hal ini dapat terjadi pada *running status* yang ditujukan mengurangi sejumlah *bytes* yang dibutuhkan untuk memulai dan mengakhiri setiap not. Durasi dari sebuah not adalah waktu yang terkait antara *Note On* dan *Note Off* yang sama.

2.4.2.2 Aftertouch

Aftertouch messages dikirimkan oleh beberapa *keyboard* setelah *tuts* pada *keyboard* tersebut dilepas. Terdapat beberapa tekanan horisontal dan vertikal pada sebuah not selama not itu sendiri berbunyi dan menghasilkan beberapa *aftertouch messages*.

2.4.3 Channel Mode Messages

Channel mode messages dapat mengganti kebiasaan penerima (*receiver*) dalam menerima *messages*. Sebagian dari *messages* tersebut dapat mengubah penerima menjadi *state* khusus sementara yang lain menerima lanjutan dari *channel voice messages* [7].

Seperti yang telah dibicarakan bahwa *channel mode messages* adalah bagian dari *control change messages* dari *control change messages* dengan lanjutan nomor 120 – 127 (Tabel 2.3).

Tabel 2.3 Channel Mode Messages

No	Mode Messages (decimal)	Status/Data Bytes	Usage
1	All Sound Off (120)	0xB0 0x78 0x00	
2	Reset All Controller (121)	0xB0 0x79 0x00	
3	Local Control (122)	0xB0 0x7A VAL	VAL = 0 for off, VAL = 127 for on
4	All Notes Off (123)	0xB0 0x7B 0x00	
5	Omni Off (124)	0xB0 0x7C 0x00	
6	Omni On (125)	0xB0 0x7D 0x00	
7	Mono On (Poly off) (126)	0xB0 0x7E VAL	VAL = number of channel, or 0 to set the number of channels to number of voices available
8	Poly On (Mono Off) (127)	0xB0 0x7F 0x00	

(Sumber : Maximum MIDI)

Channel mode messages akan dikirimkan ke *basic channel* dari masing – masing instrumen penerima. *Basic channel* (atau sering disebut dengan *global channel*) adalah *channel* tertentu dari sebuah perangkat yang sudah ditentukan oleh perangkat tersebut (secara permanen) atau yang dapat diatur oleh pengguna pada panel depan dari perangkat tersebut. Dengan mengatur *basic channel* tersebut maka *Channel Mode messages* yang dikirimkan pada *channel* lain akan diabaikan.

Perangkat penerima (*receiving device*) dapat mempunyai satu dari empat jenis *mode*, yaitu *omni on*, *omni off*, *mono on*, *poly on*. Seperti pada Tabel 2.4 di bawah ini :

Tabel 2.4 Tabel Penerimaan

	Poly	Mono
Omni On	Mode 1 Omny On Poly	Mode 2 Omny On Mono
Omni Off	Mode 3 Omny Off Poly	Mode 1 Omny Off Mono

(Sumber : Maximum MIDI)

Contohnya untuk memilih *Mode 1* pada penerima, kirim *Omni On message* diikuti oleh *poly on messages*.

Mengaktifkan *omni on* memberikan penerima untuk merespon *messages* yang ada pada semua *channel*. Sedangkan *omni off* akan memberikan *messages* untuk penerima hanya pada *basic channel*, dan akan mengabaikan semua *messages* yang yang dikirimkan ke *channel* lainnya.

Poly adalah kependekan dari *polyphonic*. Instrumen yang mempunyai karakteristik *polyphonic* dapat membunyikan lebih dari satu nada dalam satu waktu. Sedangkan memilih *mono* akan membatasi instrumen untuk membunyikan satu not pada satu waktu. Penggunaan *poly* atau *mono* biasanya bergantung pada tipe dan karakter suatu instrumen tersebut dimainkan. Piano adalah instrumen *polyphonic*, sedangkan memainkan *chord* di tuba adalah hal yang paling sulit (karena tuba memiliki sifat *mono*).

2.4.4 System (Common) Messages

Tabel 2.5 System Common Messages

<i>No.</i>	<i>Status Byte (Biner)</i>	<i>Data Byte(s)</i>	<i>Message</i>
1.	11110001	<i>VAL</i>	<i>MIDI Time Code Quarter Frame</i>
2.	11110010	<i>LSB, MSB</i>	<i>Song Position Pointer</i>
3.	11110011	<i>SONG#</i>	<i>Song Select</i>
4.	11110100	-	<i>Undefined</i>
5.	11110101	-	<i>Undefined</i>
6.	11110110	-	<i>Tune Request</i>
7.	11110111	-	<i>EOX (End of Exclusive)</i>

(Sumber : Maximum MIDI)

System Common messages digunakan untuk mengkoordinasi kegiatan yang berhubungan dengan semua perangkat yang terhubung oleh sistem yang diberikan. Di antara dua dari tujuh *System Common messages* yang ada, adalah tidak terdefinisi. *EOX(End Of Exclusive) messages* dan *MIDI Time Code Quarter Frame messages* akan dijelaskan berikutnya [7].

The Song Position Pointer (SPP) bertugas memberitahu *sequencer* atau *drum machine* dimana posisi *playback* ditempatkan. Posisi yang mempunyai nilai sebanyak 14-bit adalah perhitungan waktu dari permulaan urutan. Masing – masing dari unit waktu tersebut sama dengan enam *MIDI clock*. Sejak ada 24 *MIDI clock* dari seperempat not, satu waktu unit *SPP* terdiri atas enam belas not. Karena itu *SPP messages* menentukan posisi dari enam belas not berurutan. *Messages* ini telah dikirim sebelum *System Real-Time messages* seperti *Start* atau *Continue*. Tetapi *messages* ini tidak mempunyai arti jika sistem *MIDI* tidak bersinkronisasi dengan *MIDI sync*.

Tune Request adalah *messages* yang tidak lagi dimiliki oleh *synthesizer analog*. Instrumen ini sangat sensitif dengan frekuensi *out of tune*, dan *Tune Request* akan membuat frekuensi tersebut untuk kembali memusat pada *oscillator*. *Messages* ini sering digunakan pada masa ini terhadap sebagian besar *synthesizer* dan modul bunyinya akan sedikit lebih baik jika dibandingkan dengan komputer dengan *output audio*.

2.4.5 System Real-Time Messages

System Real Time Messages bertugas memberi indikasi bahwa suatu proses sedang terjadi pada saat ini. *Messages* ini berbentuk *single-byte*, mempunyai sifat yang unik untuk muncul dimana saja bahkan antara *status byte* dan *data byte* dari *MIDI messages* yang ada [7].

MIDI Timing Clock messages, atau sering disebut dengan *MIDI Clock messages*, dikirimkan ketika sistem sedang mensinkronisasi waktu dengan perangkat *MIDI* lainnya. Biasanya, jenis dari *MIDI sync* ini digunakan dengan *drum machine* dan dengan beberapa tipe dari *tape synchronization*. Dua puluh

empat MIDI *Clock* dikirimkan untuk setiap seperempat not. Tingkat pengiriman *messages* ini tergantung dari kecepatan tempo. Faktor lain juga terdapat pada MIDI *sync* yang berguna untuk mengurangi kerja MIDI *Clock* yang berjalan. Dengan hanya 24 *clocks* per ketukan akan banyak sekali kemungkinan untuk terjadi *Error*.

Start (0xFA), Continue (0xFB) dan Stop (0xFC) adalah *real-time messages* yang mengatur sinkronisasi saat menggunakan MIDI *sync*. *Start messages* mengindikasikan waktu mulai pada awal dari sebuah *sequence* yang sedang terjadi. *Stop messages* akan segera memberhentikan *playback* atau *record*. Walaupun MIDI *clocks* terus – menerus menerima *messages*, penerima (receiver) akan menahan posisi tersebut sekarang dan mengabaikan MIDI *Clocks* sampai pesan *start* atau *continue* diterima. *Continue messages* akan menyebabkan *timing* yang sedang berjalan melanjutkan dari posisi yang sudah ada ke perintah MIDI *Clocks* yang selanjutnya. *Messages* ini memberikan *sequence* untuk dihentikan sementara (dengan menggunakan *stop messages*) dan melanjutkannya kembali. Mengirimkan *Start messages* setelah *Stop messages* akan menyebabkan siklus *sequence* berhenti dan memulai lagi dari paling awal.

System reset messages adalah pesan yang juga sering digunakan oleh MIDI *messages*. Jika diterima dan dikenali oleh sebuah perangkat, *System reset messages* ini bertugas untuk membuat sebuah perangkat kembali seperti posisi *power* dinyalakan. Tidak ada status yang berjalan, mematikan semua not yang berbunyi, menyalakan kendali lokal, dan membuat semua bagian – bagian yang ada pada waktu kondisi sebuah perangkat dinyalakan.

2.4.6 System Exclusive Messages

Sistem ini adalah kambing hitam dari MIDI *protocol*. Pesan – pesan ini adalah pesan baru yang dapat digunakan untuk tujuan yang akan datang dan dapat dibayangkan. Pada awalnya, sistem ini telah didesain sebagai suatu *method* untuk mentransfer semua pengaturan penyimpanan dari satu instrumen ke instrumen lain. Semua *sysex messages* mempunyai struktur yang sederhana. Dimulai dengan *System Exclusive status byte*, 0xF0. Satu sampai tiga *byte Manufacturer ID* (kode

pabrik) untuk mengidentifikasi instrumen mana yang dituju. Jika *bit* pertama adalah nol maka diikuti oleh bentuk *bit* kedua yaitu ID dengan 16 *bit*.

Semua nomor pada *data bytes* dapat dimasukkan (Semua 7 *bit*, dengan *bit* ke delapan dibuat menjadi nol). Biasanya, Satu atau lebih *data bytes* awal akan menentukan jenis instrumen dan jenis dari *sysex messages* [7].

2.5 Perangkat Keras

Perangkat keras atau *Hardware* merupakan gabungan dari beberapa komponen kecil disatukan atau diintegrasikan untuk menjadi suatu perangkat utuh untuk keperluan tertentu dan mengambil data dengan variabel atau parameter yang berbeda, maka dari itu pada sub bab ini akan dibahas komponen atau part-part pendukung untuk membangun Kontrol Alat Musik Angklung.

2.5.1 Arduino Mega 2560

Arduino Mega 2560 adalah sebuah Board Arduino yang menggunakan IC Mikrokontroler ATmega 2560. *Board* ini memiliki Pin I/O yang relatif banyak, 54 *digital Input / Output*, 15 buah di antaranya dapat digunakan sebagai *output PWM*, 16 buah *analog input*, 4 UART (*port serial*). Arduino Mega 2560 di lengkapi kristal 16 Mhz, koneksi USB, adaptor listrik, header ICSP, dan tombol reset [8].



Gambar 2.9 Tampilan Arduino MEGA 2560
(Sumber: <https://store.arduino.cc/usa/mega-2560-r3>)

Tabel 2.6 Ringkasan Spesifikasi Arduino MEGA 2560

No	Nama	Keterangan
1.	Mikrokontroler	ATMega2560
2.	Tegangan Operasi	5V
3.	Tegangan <i>Input</i>	7-12V
4.	Batas Tegangan <i>Input</i>	6-20V
5.	Pin <i>Digital I/O</i>	54 (<i>15 pin output PWM</i>)
6.	Pin <i>Analog Input</i>	16 pin
7.	Arus DC per pin I/O	40 mA
8.	Arus DC pin 3.3V	50 mA
9.	<i>Flash Memory</i>	256 KB (<i>8 KB Bootloader</i>)
10.	SRAM	8 KB
11.	EEPROM	4 KB
12.	<i>16 Clock Speed</i>	17 MHz

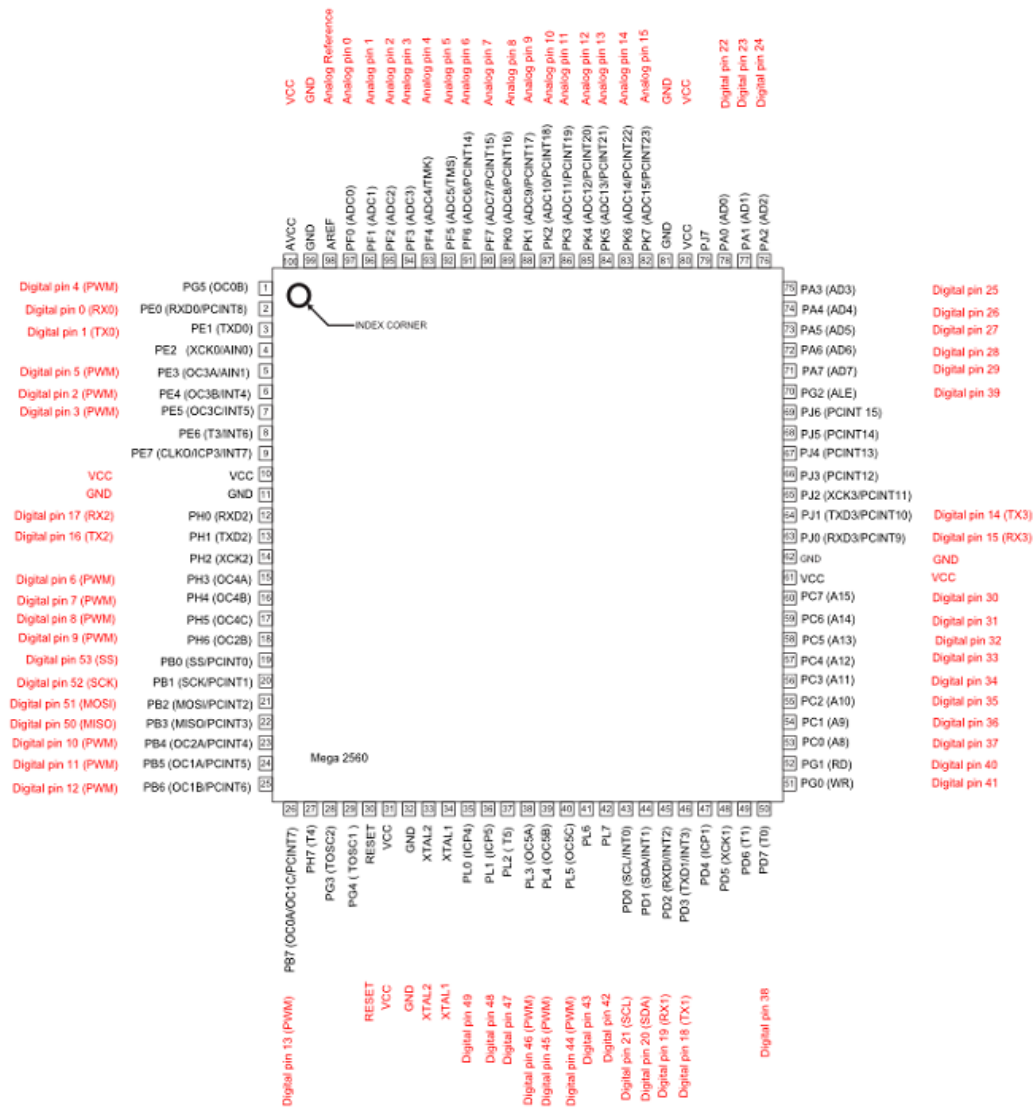
Arduino Mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Sumber daya eksternal (*non-USB*) dapat berasal baik dari *adaptor AC-DC* atau baterai. *Adaptor* dapat dihubungkan dengan mencolokkan steker 2,1 mm yang bagian tengahnya terminal positif ke *jack* sumber tegangan pada papan. Jika tegangan berasal dari baterai dapat langsung dihubungkan melalui header pin Gnd dan pin Vin dari konektor POWER [9].

Papan Arduino ATmega2560 dapat beroperasi dengan pasokan daya eksternal 6 Volt sampai 20 Volt. Jika diberi tegangan kurang dari 7 Volt, maka pin 5 Volt mungkin akan menghasilkan tegangan kurang dari 5 Volt dan ini akan membuat papan menjadi tidak stabil. Jika sumber tegangan menggunakan lebih dari 12 Volt, regulator tegangan akan mengalami panas berlebihan dan bisa merusak papan. Rentang sumber tegangan yang dianjurkan adalah 7 Volt sampai 12 Volt.

Pin tegangan yang tersedia pada papan Arduino adalah sebagai berikut:

- **VIN** : *Input* tegangan untuk papan Arduino ketika menggunakan sumber daya eksternal (sebagai 'saingan' tegangan 5 Volt dari koneksi USB atau sumber daya ter-regulator lainnya).
- **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.
- **3.3V** : 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.
- **GND** : Pin *Ground*.
- **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.

Arduino ATmega2560 memiliki 256 KB *flash memory* untuk menyimpan kode (yang 8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).



Gambar 2.10 Digital Pin Arduino MEGA 2560

(Sumber: <https://www.theengineeringprojects.com/wp-content/uploads/2018/06/introduction-to-arduino-mega-5.png>)

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:

- **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*.
- **Eksternal Interrupt** : 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 perubahan nilai.
- **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.
- **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, dan ketika pin diset bernilai *LOW*, maka LED padam.
- **TWI** : Pin 20 (*SDA*) dan pin 21 (*SCL*). Yang mendukung komunikasi *TWI* menggunakan perpustakaan *wire*. Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin *TWI* pada Arduino Duemilanove atau Arduino Diecimila.

Arduino Mega2560 memiliki 16 pin sebagai *analog input*, yang masing-masing menyediakan resolusi 10 *bit* (yaitu 1024 nilai yang berbeda). Secara *default* pin ini dapat diukur/diatur dari mulai *Ground* sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin *AREF* dan fungsi *analogReference()*.

Ada beberapa pin lainnya yang tersedia, antara lain:

- **AREF** : Referensi tegangan untuk input analog. Digunakan dengan fungsi *analogReference()*.

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

Arduino Mega2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan Arduino lain, atau dengan mikrokontroler lainnya. Arduino ATmega328 menyediakan 4 *hardware* komunikasi *serial UART TTL* (5 Volt). Sebuah *chip* ATmega16U2 (ATmega8U2 pada papan Revisi 1 dan Revisi 2) yang terdapat pada papan digunakan sebagai media komunikasi *serial* melalui USB dan muncul sebagai *COM Port Virtual* (pada *device* komputer) untuk berkomunikasi dengan perangkat lunak pada komputer, untuk sistem operasi Windows masih tetap memerlukan file inf, tetapi untuk sistem operasi OS X dan Linux akan mengenali papan sebagai *port COM* secara otomatis. Perangkat lunak Arduino termasuk didalamnya *serial monitor* memungkinkan *data* tekstual sederhana dikirim ke dan dari papan Arduino. *LED RX* dan *TX* yang tersedia pada papan akan berkedip ketika data sedang dikirim atau diterima melalui *chip USB-to-serial* yang terhubung melalui USB komputer (tetapi tidak untuk komunikasi *serial* seperti pada pin 0 dan 1).

2.5.2 MIDI Controller

MIDI controller terdiri dari dua kata yaitu MIDI adalah singkatan dari *Musical Instrument Digital Interface* dan *Controller* adalah alat pengendali, sehingga MIDI controller adalah sebuah piranti pengendali MIDI yang berperan dalam sebuah sistem pengoperasian instrumen virtual di komputer. Untuk jenis MIDI Controller yang akan dipakai pada penelitian ini adalah jenis *Worldde Panda Mini keyboard piano* dengan tuts yang berjumlah 25 [10].

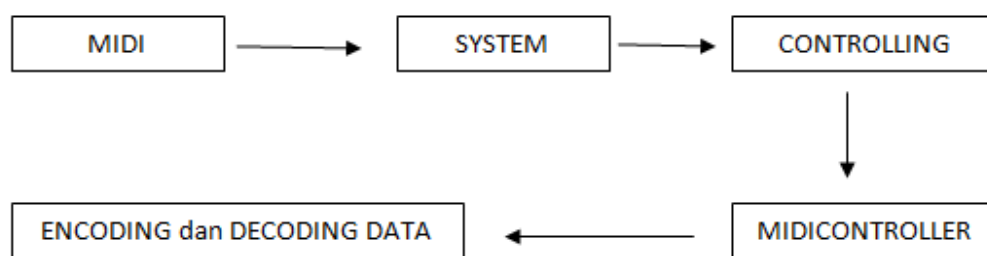


Gambar 2.11 Worldde Panda Mini Keyboard MIDI 25-key
(Sumber : <https://id.aliexpress.com/item/32850531765.html>)

Tabel 2.7 Spesifikasi Worlde Panda Mini Keyboard MIDI 25-key

No.	Spesifikasi
1.	8-pad dengan dinamika dan dapat ditetapkan sebagai pengontrol
2.	25-tuts dengan dinamika dan dapat ditetapkan sebagai pengontrol
3.	Mempunyai 4 tombol pengontrol
4.	Mempunyai 4 <i>knob</i> penggeser
5.	Mempunyai 4 <i>knob</i> potensio
6.	Kompatibel dengan Win10 / Win8 / Win7 / WinXP / WinVista dan Mac OSX

MIDI *controller* juga dapat dikatakan sebagai piranti esensial dalam sebuah *system* [8]. Pada prinsipnya, MIDI *controller* melakukan *encoding* dan *decoding* data *midi*. *Encoding* dan *decoding* data dapat mencapai 16.000 *steps* secara simultan. Berikut adalah rangkaian *system* kerja MIDI *controller*.



Gambar 2.12 Sistem Kerja MIDI

(Sumber : imajiner07.blogspot.com/2013/01/mengenal-midicontroller-majalah.html?m=1)

Dari rangkaian di atas dapat dijelaskan sebagai berikut: kolom MIDI merupakan kumpulan *file-file* yang ditampung dalam sebuah *soundbank* yang kemudian di *register* oleh *system* dalam hal ini adalah *software* DAW, kemudian pengendalian *software* dialihkan ke MIDI *controller* yang nantinya semua *file* MIDI yang ada pada *software* akan dipanggil oleh MIDI *controller* untuk dilakukan proses *encoding* dan *decoding* data. MIDI *controller* tidak memiliki *sound bank* atau bunyi-bunyian yang terprogram secara internal di dalam *hardware*, melainkan MIDI *controller* digunakan untuk mengoperasikan berbagai instrument *virtual* dengan berbagai jenis suara yang tersedia di dalam komputer.

Dari sumber di atas dapat diambil kesimpulan bahwa *MIDI controller* merupakan piranti penghubung yang berguna dalam pengoperasian instrumen *virtual* di komputer. *MIDI controller* tidak berfungsi tanpa adanya komputer yang dilengkapi *software* DAW. Untuk menghubungkan *MIDI controller* dengan komputer agar dapat beroperasi diperlukan alat tambahan berupa kabel USB (*Universal Serial Bus*) yang menghubungkan antara *MIDI controller* dengan komputer.

Konektor *MIDI controller* terdiri dari dua jenis yaitu USB dan *MIDI ports*..

a. USB

USB yaitu singkatan dari *Universal Serial Bus* yang banyak digunakan pada komputer saat ini. Hal ini dikarenakan banyaknya produsen komputer yang mengganti port MIDI dengan USB sehingga banyak *MIDI controller* yang dahulu menggunakan MIDI port, kemudian beralih menggunakan USB yang lebih mudah untuk digunakan dalam hubungan antar muka dengan computer.

USB memiliki 2 tipe yaitu tipe A dan tipe B. USB tipe A yaitu kabel USB yang memiliki 2 konektor yang sejenis yaitu tipe A *male* dan tipe A *female*. Seperti yang terlihat pada gambar.



Gambar 2.13 USB Type A male
(Sumber : www.weiku.com/products/10383706/High_Speed_USB_2_0_Connecting_Cable_Type_A_Male_To_Type_A_Male.html)



Gambar 2.14 USB Type A female
 (Sumber : www.google.com/imghp?hl=id&tab=wi&ei=ZAeyVPz5F8Tk8AWxpIG4AQ&ved=0CAQQqi4oAg)

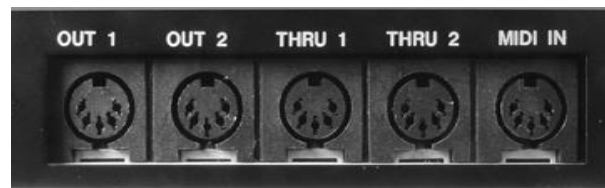
USB tipe B yaitu kabel USB yang memiliki 2 konektor yang berbeda jenis. Biasanya kabel USB tipe B memiliki konektor tipe A *male* dan tipe B *male* dalam satu kabel.



Gambar 2.15 USB Type B
 (Sumber : <https://www.perfectcircuit.com/lmntl-type-b-usb-cable-1.html>)

b. MIDI Ports

Kegunaan MIDI *port* sama seperti USB yaitu sebagai pengirim dan penerima data. Akan tetapi, MIDI *port* dan USB memiliki perbedaan bentuk terutama pada kabel jack. MIDI *port* masih menggunakan serial komunikasi RS232 sehingga masih membutuhkan dukungan dari *serial to PC (Personal Computer) port*. Seperti terlihat pada gambar.



Gambar 2.16 MIDI Ports
(Sumber : pixgood.com/midi-port.html)



Gambar 2.17 Kabel jack serial to PC
(Sumber : pixgood.com/jack.html)

2.5.3 USB Host Shield

USB Host Shield merupakan sebuah board tambahan yang berfungsi untuk komunikasi perangkat antarmuka USB seperti *mouse*, *keyboard*, *joystick*, *MIDI controller*, *digital camera*, *bluetooth* dan sebagainya. Untuk jenis *USB host shield* yang akan dipakai pada penelitian ini adalah jenis yang *full size*, dimana cara penggunaan *shield* ini langsung mengikuti desain pin pada Arduino. *Shield* ini bisa digunakan pada jenis *Uno*, *Mega* dan *Duemilanove* yang tidak perlu penambahan kabel untuk menghubungkan ke *SPI*. Di bawah ini merupakan tampilan dari *USB host shield* [11].



Gambar 2.18 Board USB Host Shield
(Sumber: <https://robu.in/wp-content/uploads/2015/01/usb-host-shield-800x800.jpg>)

Tabel 2.8 Spesifikasi USB Host Shield

No.	Nama	Keterangan
1.	<i>Operating Voltage</i>	5V
2.	<i>USB Controller</i>	MAX3421E
3.	<i>Max Current</i>	500mA (ketika Arduino ditenagai oleh catu daya yang cocok yang terhubung ke colokan listrik Arduino)
4.	<i>Max Current</i>	400mA (saat Arduino ditenagai oleh port USB-nya)

2.5.4 Motor DC

Motor Listrik DC adalah suatu perangkat yang mengubah energi listrik menjadi energi kinetik. Motor DC ini juga dapat disebut sebagai Motor Arus Searah. Seperti namanya, DC Motor memiliki dua terminal dan memerlukan tegangan arus searah atau DC (*Direct Current*) untuk dapat menggerakannya. Motor Listrik DC ini biasanya digunakan pada perangkat-perangkat Elektronik dan listrik yang menggunakan sumber listrik DC seperti Vibrator Ponsel, Kipas DC dan Bor Listrik DC [12].



Gambar 2.19 Motor DC gearbox dual shaft

(Sumber: <http://qqtrading.com.my/dc-geared-motor-130-dual-shaft-w-yellow-gear-box-3-6vdc>)

Tabel 2.9 Spesifikasi Motor DC gearbox dual shaft

No.	Nama	Keterangan
1.	<i>Rated Voltage</i>	5.0V DC
2.	<i>Working Voltage</i>	3.0V – 6.
3.	<i>Current (No load)</i>	3V-150mA , 6V-200mA
4.	<i>Speed (No load)</i>	3V-90rpm±10% , 6V-200rpm±10%
5.	<i>Torque (tors)</i>	Max. 0.8 kg/cm
6.	<i>Reduction ratios</i>	1:48
7.	<i>Brush-type</i>	Brush
8.	<i>Body Size</i>	70mm x 37mm x 23mm
9.	<i>Shaft Size</i>	8mm x 2mm diameter
10.	<i>Weight</i>	17.5 grams

Pada motor DC terdapat jangkar dengan satu atau lebih kumparan terpisah. Tiap kumparan berujung pada cincin belah (komutator). Dengan adanya insulator antara komutator, cincin belah dapat berperan sebagai saklar kutub ganda (*double pole, double throw switch*). Motor DC bekerja berdasarkan prinsip *Gaya Lorentz*, yang menyatakan ketika sebuah konduktor beraliran arus diletakkan dalam medan magnet, maka sebuah gaya (yang dikenal dengan *Gaya Lorentz*) akan tercipta secara *ortogonal* diantara arah medan magnet dan arah aliran arus [4]. Kecepatan putar Motor DC (N) dirumuskan dengan Persamaan berikut.

$$N = \frac{V_{TM} - I_A R_A}{K\phi} \quad \dots(2.1)$$

Keterangan:

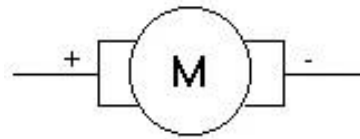
V_{TM} : Tegangan terminal

I_A : Arus jangkar motor

R_A : Hambatan jangkar motor

K : Konstanta motor

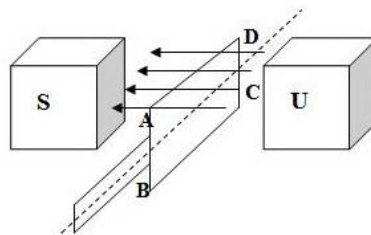
ϕ : Fluk magnet yang terbentuk pada motor.



Gambar 2.20 Simbol Motor DC

(Sumber : <http://elektronika-dasar.web.id/wp-content/uploads/2012/05/Simbol-Motor-DC.jpg>)

Motor DC tersusun dari dua bagian yaitu bagian diam (*stator*) dan bagian bergerak (*rotor*). *Stator* motor arus searah adalah badan motor atau kutub magnet (sikat-sikat), sedangkan yang termasuk *rotor* adalah jangkar lilitanya. Pada motor, kawat penghantar listrik yang bergerak tersebut pada dasarnya merupakan lilitan yang berbentuk persegi panjang yang disebut kumparan [4].



Gambar 2.21 Prinsip Kerja Motor DC

(Sumber : <http://elektronika-dasar.web.id/wp-content/uploads/2012/05/Prinsip-Kerja-Motor-DC.jpg>)

Kumparan ABCD terletak dalam medan magnet serba sama dengan kedudukan sisi aktif AD dan CB yang terletak tepat lurus arah *fluks* magnet. Sedangkan sisi AB dan DC ditahan pada bagian tengahnya, sehingga apabila sisi AD dan CB berputar karena adanya gaya lorentz, maka kumparan ABCD akan berputar.

Hasil perkalian gaya dengan jarak pada suatu titik tertentu disebut momen, sisi aktif AD dan CB akan berputar pada porosnya karena pengaruh momen putar (T). Setiap sisi kumparan aktif AD dan CB pada gambar diatas akan mengalami momen putar sebesar :

$$T = F.r \quad \dots(2.2)$$

Dimana :

T = momen putar (Nm)

F = gaya tolak (newton)

r = jarak sisi kumparan pada sumbu putar (meter)

Pada daerah dibawah kutub-kutub magnet besarnya momen putar tetap karena besarnya *Gaya Lorentz*. Hal ini berarti bahwa kedudukan garis netral sisi-sisi kumparan akan berhenti berputar. Supaya motor dapat berputar terus dengan baik, maka perlu ditambah jumlah kumparan yang digunakan. Kumparan-kumparan harus diletakkan sedemikian rupa sehingga momen putar yang dialami setiap sisi kumparan akan saling membantu dan menghasilkan putaran yang baik. Dengan pertimbangan teknis, maka kumparan-kumparan yang berputar tersebut dililitkan pada suatu alat yang disebut jangkar, sehingga lilitan kumparan itupun disebut lilitan jangkar [4].

Hubungan antara kecepatan, flux medan dan tegangan ditunjukkan dalam persamaan berikut:

Gaya Elektromagnetik (E):

$$E = K\Phi N \quad \dots(2.3)$$

Torque (T) :

$$T = K\Phi I_a \quad \dots(2.4)$$

Kecepatan Motor Tak Berbeban ($N_{\text{tidak berbeban (rpm)}}$) :

$$N_{\text{tidak berbeban (rpm)}} = T \times (N_{\text{Max/km}}) \quad \dots(2.5)$$

Kecepatan Motor Berbeban ($N_{\text{berbeban (rpm)}}$)

$$N_{\text{Max}} - N_{\text{tidak berbeban}} = N_{\text{berbeban (rpm)}} \quad \dots(2.6)$$

Arus :

$$I \text{ (A)} = T \times K_m/A \quad \dots(2.7)$$

Daya Masukan :

$$T \times N_{\text{berbeban}} \times \text{Konfersi Faktor antara T dan N} = \text{Daya Masukan (Win)} \dots(2.8)$$

Daya Keluaran :

$$V \times A = \text{Daya Keluaran (Wout)} \quad \dots(2.9)$$

Efisiensi :

$$\text{Efisiensi (\%)} = W_{\text{out}} / W_{\text{in}} \quad \dots(2.10)$$

Dimana:

E = gaya elektromagnetik yang dikembangkan pada terminal dinamo (volt)

Φ = flux medan yang berbanding lurus dengan arus medan

N = kecepatan dalam RPM (putaran per menit)

T = torque elektromagnetik (g/cm)

I_a = arus dinamo (A)

K = konstanta persamaan

V = Tegangan (Volt)

W_{out} = Daya Keluaran (Watt)

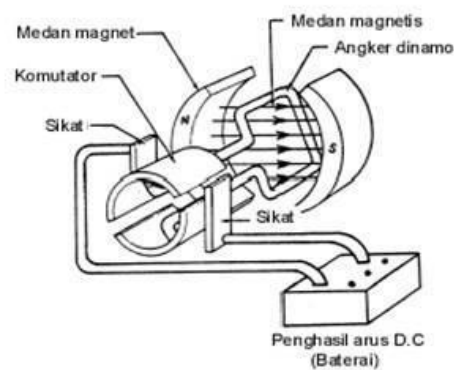
W_{in} = Daya Masukan (Watt)

Motor DC memiliki 3 bagian atau komponen utama untuk dapat berputar sebagai berikut.

- **Kutub medan.** Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi ruang terbuka diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih elektromagnet.
- **Kumparan Motor DC.** Bila arus masuk menuju kumparan motor DC, maka arus ini akan menjadi elektromagnet. kumparan motor DC yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan

beban. Untuk kasus motor DC yang kecil, kumparan motor DC berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan kumparan motor DC.

- **Komutator Motor DC** . Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikan arah arus listrik dalam kumparan motor DC dan juga membantu dalam transmisi arus antara kumparan motor DC dan sumber daya [4].

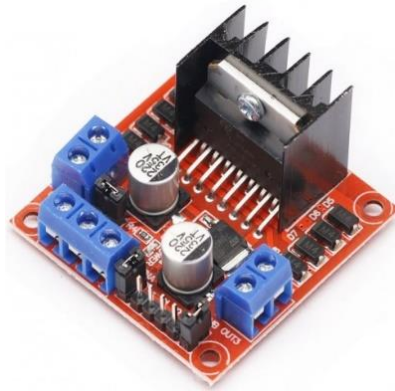


Gambar 2.22 Bagian-Bagian Motor DC

(Sumber : <http://elektronika-dasar.web.id/wp-content/uploads/2012/05/Bagian-Bagian-Motor-DC.jpg>)

2.5.5 Module L298N Dual H-Bridge

Module L298N Dual H-Bridge Driver Motor ini berfungsi untuk "mendrive" atau menyetir atau dengan kata lain mempermudah kita dalam urusan mengontrol motor DC menggunakan Mikrokontroler. Kita tau bahwa logic level output dari mikrokontroler yaitu 3.3V dan 5V dengan arus yang sangat terbatas, sehingga kita tidak bisa mengendalikan motor secara langsung apalagi motor tersebut membutuhkan level tegangan dan arus yang lebih besar, jika motor DC kecil sih bisa sajah tpi itu juga beresiko. Oleh sebab itu dalam mengendalikan motor menggunakan mikrokontroler maka diperlukan sebuah Driver [13].



Gambar 2.23 Driver Motor L298N H-Bridge

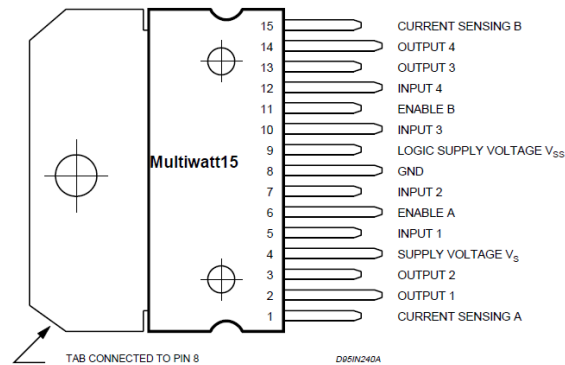
(Sumber : <https://www.dnatechindia.com/l298-motor-driver-module-board.html>)

Tabel 2.10 Spesifikasi Driver Motor L298N H-Bridge

No.	Nama	Keterangan
1.	<i>Type</i>	Dual H-Bridge
2.	<i>Chip control</i>	ST L298N
3.	<i>Logic voltage</i>	5V DC
4.	<i>Drive voltage</i>	5-35V DC
5.	<i>Logical current</i>	0mA-36mA
6.	<i>Driving current</i>	2A (MAX single bridge)
7.	<i>Temperature</i>	-20 C – 135 °C
8.	<i>Maximum Power</i>	25 W
9.	<i>Weight</i>	30gr
10.	<i>Size</i>	43x43x27mm

L298N adalah IC yang digunakan sebagai *driver* motor DC pada penelitian ini. IC ini menggunakan prinsip kerja *H-Bridge*. Tiap *H-Bridge* dikontrol menggunakan level tegangan TTL yang berasal dari *output* mikrokontroler. Tegangan yang dapat digunakan untuk mengendalikan robot bisa mencapai tegangan 46 VDC dan arus 2A untuk setiap kanalnya. L298 dapat mengontrol 4 buah motor DC, karena di dalam satu komponen L 298 N terdapat dua rangkaian

H-Bridge. Berikut Gambar 2.9 adalah bentuk IC L298 yang digunakan sebagai motor *driver*.



Gambar 2.24 Konfigurasi Pin L298N
(Sumber : <https://elib.unikom.ac.id/download.php?id=223664>)

Pengaturan kecepatan kedua motor dilakukan dengan cara pengontrolan lama pulsa aktif (mode PWM – *Pulse width Modulation*) yang dikirimkan ke rangkaian *driver* motor oleh pengendali (mikrokontroler). *Duty cycle* PWM yang dikirimkan menentukan kecepatan putar motor DC.

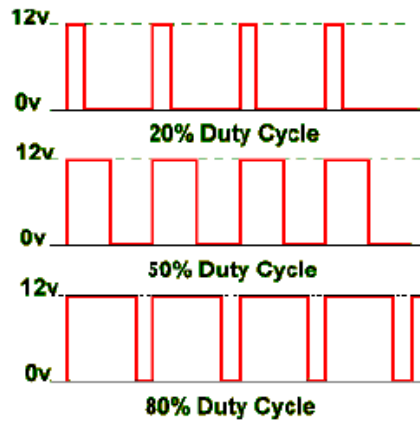
PWM (*Pulse width Modulation*), adalah sebuah metode untuk pengaturan kecepatan perputaran, dalam hal ini adalah motor DC untuk gerak robot. PWM dapat dihasilkan oleh empat metode, sebagai berikut.

1. Metode analog
2. Metode digital
3. IC diskrit
4. Mikrokontroler

Pada penelitian ini, metode PWM dikerjakan oleh mikrokontroler. Metode PWM ini akan mengatur lebar atau sempitnya periode pulsa aktif yang dikirimkan oleh mikrokontroler ke driver motor. Pada pengaturan kecepatan robot, nilai PWM mulai dari 0-255. Secara *analog* besaran PWM dihitung dalam prosentase, nilai ini didapat dari perbandingan:

$$T_{high} / (T_{high} + T_{low}) * 100\%$$

Dimana T adalah periode atau waktu tempuh untuk sebuah pulsa, yang terbagi menjadi bagian puncak positif (T_{high}) dan puncak negatif (T_{low}).



Gambar 2.25 Ilustrasi Pulse Width Modulation
(Sumber : <https://elib.unikom.ac.id/download.php?id=223664>)

Semakin rapat periode antar pulsa, maka frekuensi yang dihasilkan akan semakin tinggi, ini berarti kecepatan akan bertambah. Semakin lebar jarak antar pulsa, maka frekuensi semakin rendah ini berarti kecepatan berkurang atau menurun. Kondisi pemberian kecepatan harus disesuaikan dengan kondisi *track* yang akan dilewati oleh robot, misal pada saat jalan lurus, naik atau turun harus mendapatkan nilai PWM yang tepat.

2.5.6 LM2596 Adjustable DC-DC Step Down

LM2596 merupakan module IC yang berfungsi menurunkan power DC sehingga dapat sesuai dengan perangkat penerimanya. Alat jauh lebih praktis dan mudah ketimbang mengandalkan resistor.



Gambar 2.26 LM2596 Adjustable DC-DC Step Down
(Sumber : <http://www.instructables.com/id/The-Introduction-of-LM2596-Step-Down-Power-Module/>)

Selain digunakan untuk penurun tegangan output power adapter, bisa digunakan untuk barang elektronik lainnya, seperti *power bank*, *power supply LED*, *lightning*, dan sebagainya . Misalnya untuk *power supply* yang membutuhkan *short circuit protection*, LED yang membutuhkan *constant current*, serta untuk *power bank* yang memiliki output tak sesuai dengan perangkat penerimanya, misalnya *USB Lamp*. Module step down ini juga dilengkapi layar *digital* yang berfungsi menginformasi *voltase input* maupun *output* secara akurat. Berikut spesifikasi dari alat penurun tegangan ini:

Tabel 2.11 Spesifikasi LM2596 Adjustable DC-DC Step Down

No.	Nama	Keterangan
1.	<i>The input voltage</i>	3.2 V ~ 40 V
2.	<i>The output voltage</i>	1.25 V ~ 35 V
3.	<i>Conversion efficiency</i>	92% (the highest)
4.	<i>The output ripple</i>	< 30 mv
5.	<i>witching frequency</i>	65 KHZ
6.	<i>Working temperature</i>	- 45 ~ + 85
7.	<i>The size (length x width x height)</i>	43 mm x 21 mm x 14 mm

2.5.7 Catu Daya

Catu daya memegang peranan yang sangat penting dalam hal perancangan sebuah alat. Penentuan sistem catu daya yang akan digunakan ditentukan oleh dua faktor, diantaranya : [14]

1. Tegangan

Setiap aktuator tidak memiliki tegangan yang sama. Hal ini akan berpengaruh terhadap desain catu daya. Tegangan tertinggi dari salah satu aktuator akan menentukan nilai tegangan catu daya.

2. Arus

Arus memiliki satuan Ah (Ampere hour). Semakin besar Ah, semakin lama daya tahannya bila digunakan pada beban yang sama.

Tanpa adanya masukan daya maka perangkat tidak dapat berfungsi. Begitu juga apabila pemilihan catu daya tidak tepat, maka perangkat tidak dapat bekerja dengan baik [14].

Pada perancangan alat ini membutuhkan catu daya sebesar 24V DC (*Power Supply 24V DC*) yang dihubungkan dengan tegangan AC (*Alternating Current* atau arus bolak-balik) biasa kita sebut tegangan PLN. Catu daya ini dapat mengeluarkan tegangan DC (*Direct Current* atau arus listrik searah) yang bisa digunakan pada setiap komponen elektronika, tegangan catu daya ini sebesar 24 VDC gunanya untuk menggerakkan setiap motor dc yang mempunyai beban yang berbeda-beda.



Gambar 2.27 Bentuk Catu Daya 24V DC

(Sumber : www.electron.com/24v-dc-switching-power-supply-mean-well-rs-150-24-p56260/)

Tabel 2.12 Spesifikasi Catu Daya 24V DC

No.	Nama	Keterangan
1.	<i>Input</i>	AC 200-240V
2.	<i>Output</i>	3 x DC output, 24V DC 10A atau 240Watt
3.	<i>Setelan Voltase Output</i>	+/-5%
4.	<i>SLA</i>	10%
5.	<i>Efficiency</i>	85%
6.	<i>Casing</i>	Non-waterproof metal casing
7.	<i>Ukuran</i>	20cm x 11cm x 5cm

2.6 Perangkat Lunak

2.6.1 Arduino IDE

Software Arduino yang akan digunakan adalah *driver* dan IDE. IDE diciptakan untuk para pemula bahkan yang tidak memiliki *basic* bahasa pemrograman sama sekali karena menggunakan bahasa C++ yang telah dipermudah melalui *library*. IDE Arduino adalah *software* canggih yang ditulis dengan menggunakan bahasa Java. Software IDE arduino terdiri dari 3 (tiga) bagian:

1. Editor program, untuk menulis dan mengedit program dalam bahasa *processing*. Listing program pada arduino disebut *sketch*.
2. *Compiler*, sebuah modul yang berfungsi mengubah bahasa *processing* (kode program) menjadi kode biner karena kode biner merupakan bahasa program yang dipahami oleh mikrokontroler.
3. *Uploader*, sebuah modul yang berfungsi memasukkan kode biner kedalam memori mikrokontroler.

Tampilan ketika Arduino IDE pertama kali dijalankan, maka akan menampilkan tampilan seperti berikut.



```

midipc_midipiano_Rev1 | Arduino 1.8.8
File Edit Sketch Tools Help
midipc_midipiano_Rev1
1 #include <usbh_midi.h>
2 #include <usbhub.h>
3
4 // Satisfy the IDE, which needs to see the include statement in the ino too.
5 #ifndef dobogusinclude
6 #include <spi4teensy3.h>
7 #endif
8 #include <SPI.h>
9
10 USB Usb;
11 USBH_MIDI Midi(&Usb);
12 void MIDI_poll();
13 void checkMIDI();
14
15 //Receive MIDI
16 byte commandByte;
17 byte noteByte;
18

```

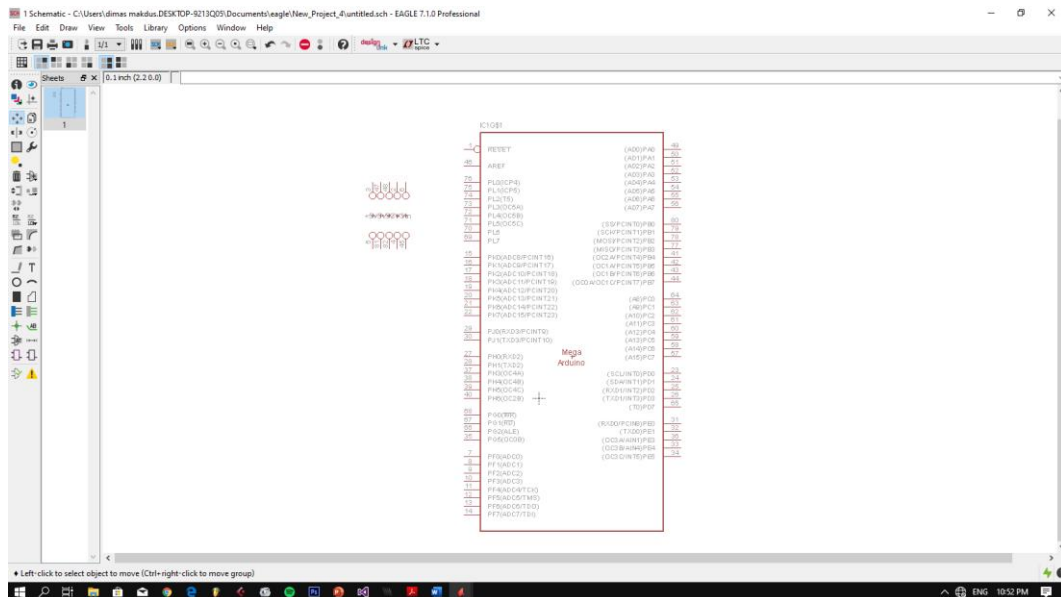
Gambar 2.28 Tampilan editor Arduino IDE

2.6.2 EAGLE

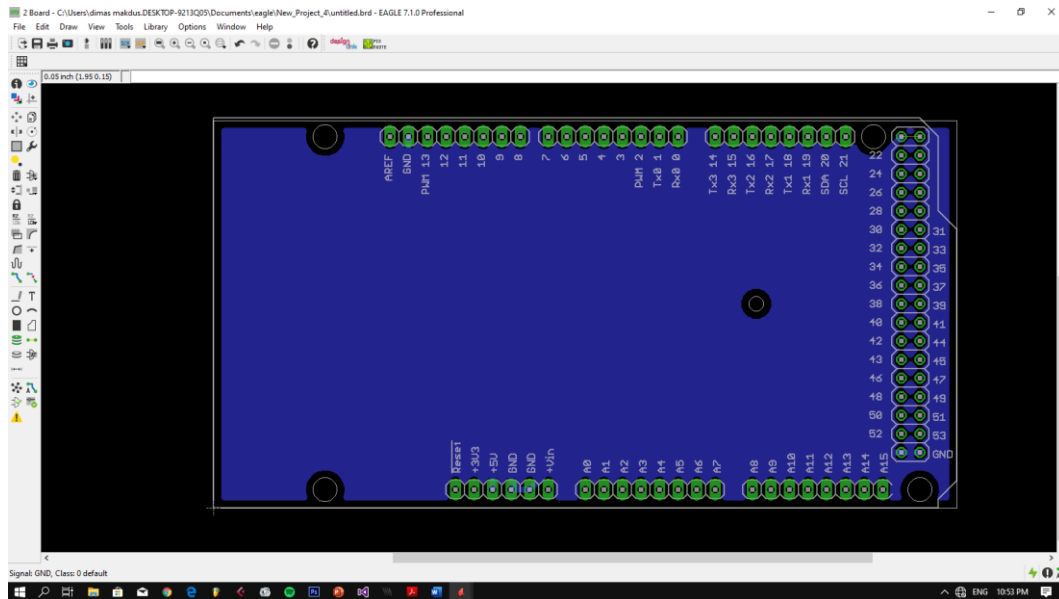
EAGLE (*Easily Applicable Graphical Layout Editor*) adalah perangkat lunak untuk mendesain skematik elektronika maupun PCB (*Printed Circuit Board*) agar desain skematik yang dibuat lebih tertata dengan rapi dan mudah dipahami setiap rangkaian maupun komponennya dapat dibuat secara *custom* sesuai kebutuhan. Dengan 3 modul dan antarmuka yang cukup umum dapat dilihat pada Gambar 2.30 dan Gambar 2.31. EAGLE menawarkan berbagai kombinasi produk dan memungkinkan setiap pengguna untuk memilih konfigurasi yang dapat memenuhi kebutuhan masing-masing pengguna.



Gambar 2.29 Eagle V 7.3.0.



Gambar 2.30 Tampilan editor desain skematik.



Gambar 2.31 Tampilan editor desain PCB (Printed Circui Board).

2.6.3 SketchUp

SketchUp adalah sebuah program 3 grafis 3 dimensi (3D) paling banyak penggunaannya saat ini. Tercatat lebih dari 30 juta pengguna SketchUp saat dan terus bertambah.













Gambar 2.32 Logo Skethcup

(Sumber : <https://seeklogo.com/vector-logo/282937/sketchup>)

SketchUp dibuat pada tahun 1999 oleh sebuah perusahaan bernama @Last Software, kemudian pada tahun 2006 dibeli oleh raksasa mesin pencari Google dengan tujuan untuk di-integrasikan dengan proyek ambisius mereka Google Earth. Di bawah Google menggratiskan penggunaan software keren ini. Fungsi dari tool tersebut sebagai berikut:

Tabel 2.13 Fungsi-Fungsi Tools Panel Google Sketchup

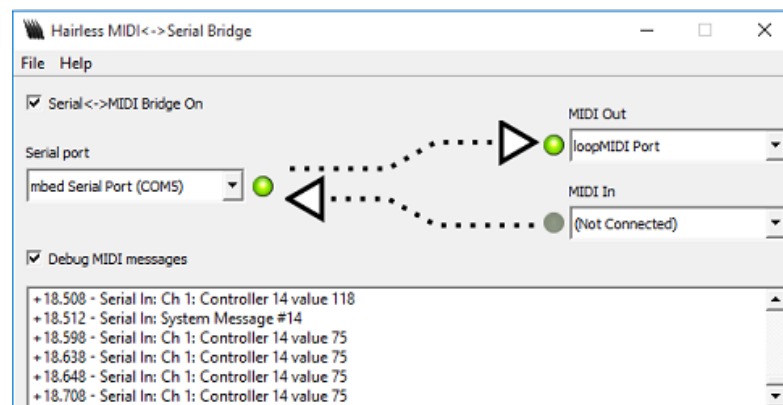
No	Gambar	Tools	Penjelasan
1		Select	Berfungsi untuk menyeleksi obyek-obyek yang ada di area
2		Line	Berfungsi untuk membuat obyek garis lurus
3		Rectangle	Berfungsi untuk membuat obyek berbentuk persegi empat
4		Circle	Sama halnya dengan line, namun bedanya tools hanya untuk membuat obyek berbentuk lingkaran
5		Arc	Sama halnya dengan line, namun tools ini digunakan untuk membuat obyek garis lengkung
6		Make component	Berfungsi membuat obyek-obyek yang terseleksi menjadi satu bagian/ komponen
7		Eraser	Bedanya dengan eraser yang ada pada software 2D, eraser ini hanya untuk menghapus line.
8		Tape measure	Berfungsi untuk membuat sebuah garis putus-putus
9		Paint bucket	Berfungsi untuk menambahkan material / warna / texture pada sisi obyek
10		Push / pull	Berfungsi untuk meng-extrude sisi obyek yang terseleksi

Tabel 2.13 Fungsi-Fungsi Tools Panel Google Sketchup (Lanjutan)

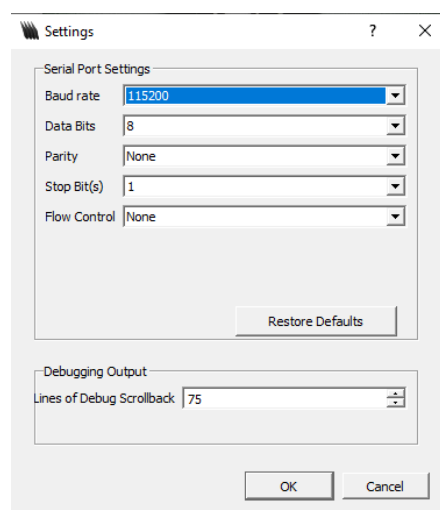
No	Gambar	Tools	Penjelasan
11		Move	Berfungsi untuk pindah obyek
12		Rotate	Rotasi
13		Offset	Membuat kloningan dari grid sisi obyek yang terseleksi
14		Orbit	Rotate untuk merotasi obyek yang terseleksi, sedangkan orbit digunakan untuk merotasi view terhadap obyek
15		Pan	Hampir sama dengan fungsi dari move, namun bedanya tools ini digunakan untuk memindahkan posisi view terhadap obyek
16		Zoom	Berfungsi untuk menzoom out / zoom in view terhadap obyek
17		Zoom extens	Hampir sama dengan zoom, bedanya tools ini membuat kita menzoom out view kita terhadap seluruh obyek yang ada di area

2.6.4 Hairless MIDI Serial Bridge

Hairless MIDI Serial Bridge adalah aplikasi *virtual* untuk menghubungkan perangkat serial seperti Arduino untuk mengirim dan menerima pesan MIDI dari aplikasi produksi musik seperti *FruityLoop (FL Studio)*, *Cubase*, dan lainnya [7].



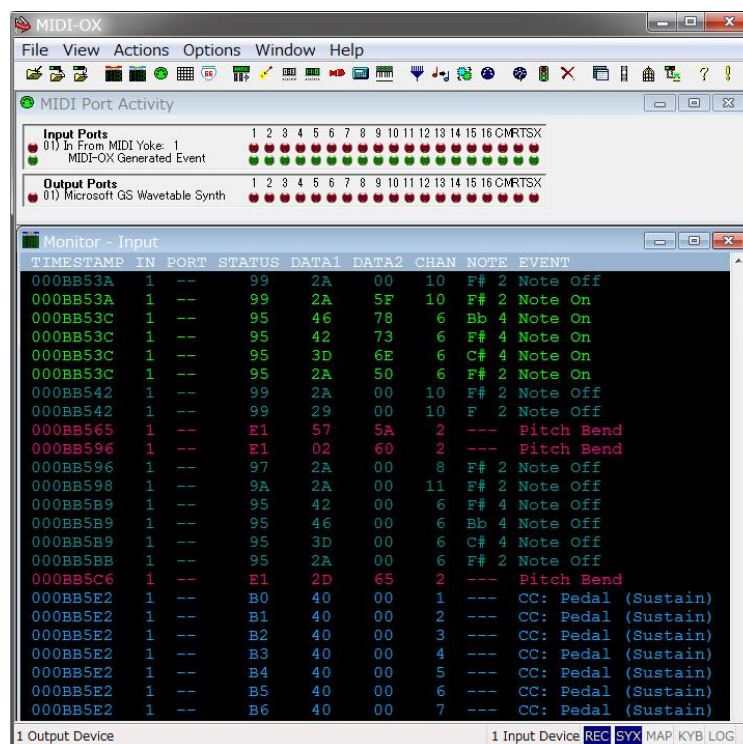
Gambar 2.33 Aplikasi Hairless MIDI Serial Bridge
(Sumber : <https://projectgus.github.io/hairless-midiserial/>)



Gambar 2.34 Pengaturan Serial Port Hairless MIDI Serial Bridge
(Sumber : <https://projectgus.github.io/hairless-midiserial/>)

2.6.5 MIDI-OX

MIDI-OX adalah alat diagnostik dan pustakawan eksklusif sistem. Aplikasi ini dapat melakukan penyaringan dan pemetaan aliran data MIDI serta menampilkan aliran MIDI yang masuk, dan meneruskan data ke driver output MIDI atau MIDI Mapper. Aplikasi ini dapat menghasilkan data MIDI menggunakan keyboard komputer atau panel kontrol internal bahkan dapat merekam dan mencatat data MIDI dan kemudian mengonversinya menjadi File MIDI Standar untuk pemutaran oleh sequencer.



Gambar 2.35 Tampilan Aplikasi MIDI-OX
(Sumber : <https://projectgus.github.io/hairless-midiserial/>)

2.6.6 FL Studio

FL Studio (Fruity Loops) merupakan sebuah aplikasi yang digunakan untuk merekam, mengubah dan membuat audio (lagu) yang diterbitkan oleh *Image-Line* yang dimana aplikasi ini ditujukan untuk musisi-musisi yang umumnya berkecimpung di dunia musik EDM (*Electronic Dance Music*). Banyak musisi dunia yang menggunakan aplikasi ini dalam bermusik seperti Martin Garrix, Avichi, Alan Walker, David Guetta dan lainnya. Namun, bukan berarti aplikasi ini hanya untuk mereka yang merupakan musisi EDM. Aplikasi ini juga bisa digunakan untuk membuat musik-musik *pop*, *rock*, dangdut bahkan musik bernuansa etnik / tradisional seperti karawitan sunda, karawitan jawa, karawitan bali dan musik etnik lainnya. Dengan aplikasi ini, kalian juga bisa mengkolaborasikan musik modern dengan musik tradisional menjadi satu bagian yang padu [10].

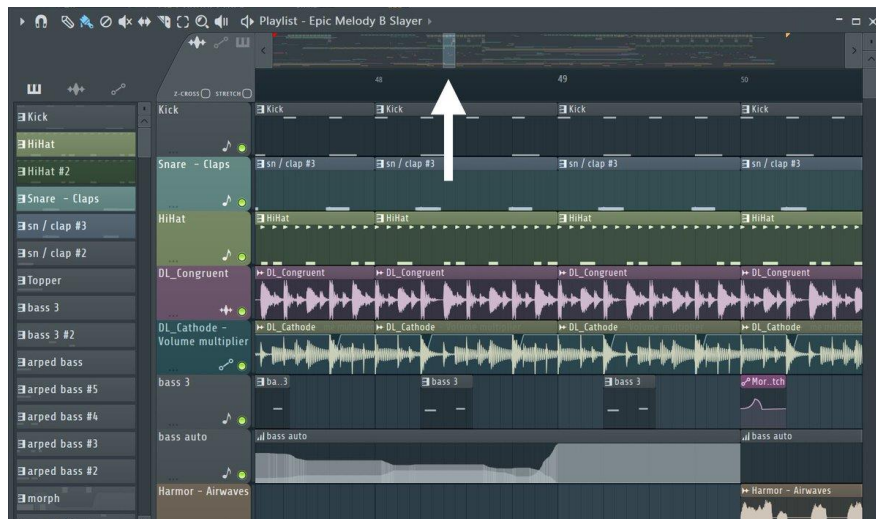


Gambar 2.36 Logo FL Studio
(Sumber : <https://www.image-line.com/flstudio/>)

Awal ditemukan pada tanggal 18 Desember 1997 oleh Didier Dambrin untuk *Image-Line*. Peluncuran resminya terjadi di tahun 1998, ketika itu masih empat channel mesin drum MIDI. Dambrin menjadi *Chief Software Architect* untuk program ini, dan dengan cepat menjalani serangkaian upgrade besar yang berhasil masuk ke *workstation audio digital* yang populer dan kompleks.

2.6.6.1 Playlist Track

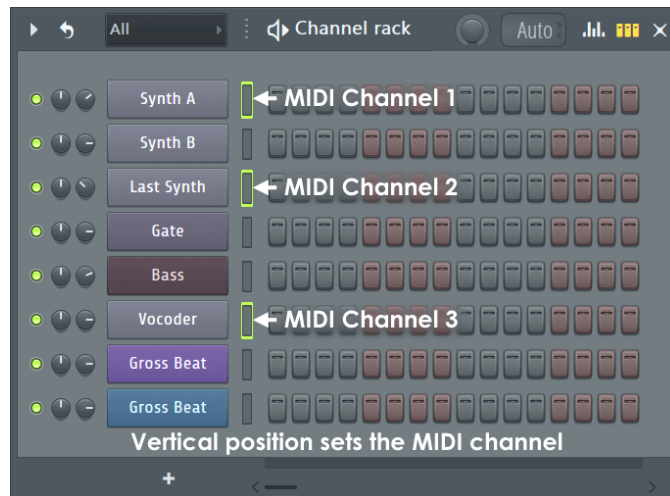
Playlist track adalah tempat disatukannya berbagai pattern menjadi satu track, seperti yang diperlihatkan pada Gambar 2.37.



Gambar 2.37 Playlist Track
(Sumber : <https://pbs.twimg.com/media/DNC3PRIVoAAlwp1.jpg>)

2.6.6.2 Sequencer

Sequencer berfungsi untuk menempatkan berbagai sample instrument, seperti midi, *synth*, *drum* dan gitar yang diperlihatkan pada Gambar 2.38.

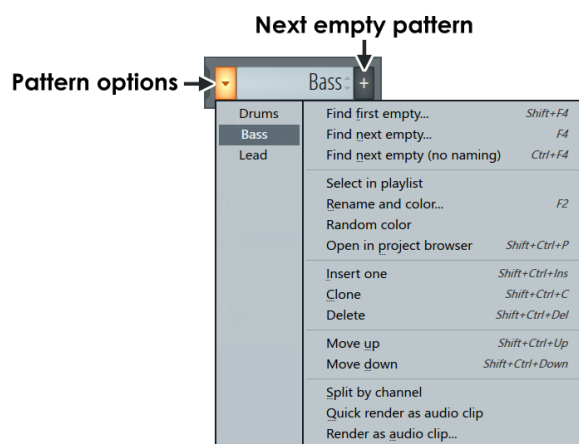


Gambar 2.38 Sequencer
(Sumber : [https://www.image-](https://www.image-line.com/support/flstudio_online_manual/html/img_shot/chanwindow_MIDI.png)

[line.com/support/flstudio_online_manual/html/img_shot/chanwindow_MIDI.png](https://www.image-line.com/support/flstudio_online_manual/html/img_shot/chanwindow_MIDI.png))

2.6.6.3 Pattern Selector

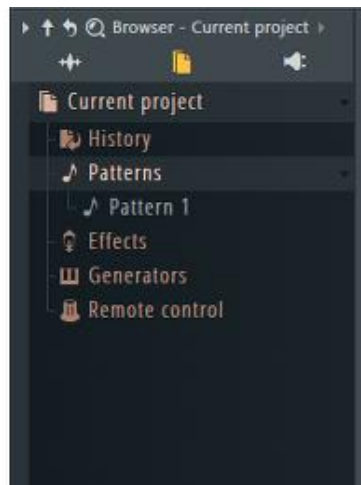
Pattern Selector berfungsi untuk menambahkan jumlah pattern pada aplikasi FL Studio jika ingin menambah instrument, yang diperlihatkan pada Gambar 2.39.



Gambar 2.39 Pattern Selector
(Sumber : <https://www.image-line.com/flstudio/>)

2.6.6.4 Browser View

Browser View adalah kumpulan data-data penunjang untuk FL Studio (salah satunya packs instrument), seperti yang diperlihatkan pada Gambar 2.40.



Gambar 2.40 Browser View

(Sumber : <https://www.image-line.com/flstudio/>)

2.6.6.5 Mixer

Mixer berfungsi untuk mengedit besar-kecilnya suara instrument (equalizer). Seperti yang diperlihatkan pada Gambar 2.41.



Gambar 2.41 Mixer

(Sumber : <https://www.image-line.com/flstudio/>)

2.6.6.6 Tempo

Tempo berfungsi untuk Mengubah tempo (dalam bentuk BPM / Beat per Minute). Seperti yang diperlihatkan pada Gambar 2.42.



Gambar 2.42 Tempo

(Sumber : <https://www.image-line.com/flstudio/>)