

2. TEORI PENUNJANG

Pada bab ini akan dibahas mengenai teori yang menunjang dalam pembuatan perangkat lunak ini, yaitu teori tentang *file* midi dan notasi balok.

2.1. Teori Mengenai Midi

Pada akhir tahun 1970-an, popularitas musik elektronik meningkat dan menjadi umum. Bagaimanapun juga peralatan dari berbagai merk tidak cocok satu dengan yang lain dan tidak dapat dihubungkan. Beberapa contoh model yaitu:

- *Analog control voltages* memiliki beberapa standard seperti 1 volt per oktaf atau logaritma hertz per volt.
- *Analog clock, trigger dan "gate" signal* yang menggunakan variasi positif negatif listrik antara -15V hingga +15V.
- *Proprietary digital interfaces* seperti Roland Corporation's *digital control bus* (DCB) dan Yamaha "*keycode*" system.

Untuk mencari jalan keluar dari permasalahan ini, pada tahun 1981 seorang *audio engineer* yang bernama Dave Smith mengajukan sebuah gagasan mengenai midi standard kepada kalangan *audio engineer*. Gagasan tersebut ditanggapi dengan antusias oleh kalangan industri. Pada bulan Agustus 1984, *Midi Specification 1.0* diterbitkan. Saat ini Dave Smith diberi gelar sebagai "*Father of Midi*" dan teknologi midi telah distandardisasi dan ditangani oleh *Midi Manufacturers Association* (MMA). Organisasi ini terletak di kota Los Angeles, California, USA. Di Jepang juga terdapat organisasi yang menangani tentang midi. Organisasi tersebut bernama *Committee of the Association of Musical Electronic Industry* (AMEI) yang terletak di Tokyo (*Musical Instrument*, 2007).

Definisi midi dapat dilihat dari dua sudut pandang, yaitu sudut pandang pemusik dan sudut pandang programmer.

- Dari sudut pandang pemusik

MIDI (*Musical Instrument Digital Interface*) adalah penghubung yang memungkinkan alat musik elektronik, komputer, dan peralatan lainnya untuk berkomunikasi, mengontrol, dan mensinkronisasi satu sama lain dengan waktu yang aktual.

- Dari sudut pandang programmer

MIDI (*Musical Instrument Digital Interface*) adalah protokol komunikasi. Midi menggunakan 8 bit word. Panjang pesan dalam midi biasanya lebih dari 1 *byte* dan panjangnya bervariasi. Midi tidak mengirimkan sinyal audio, tetapi hanya mengirimkan data digital (Conger, 1988).

Penyimpanan data dalam *file* midi memiliki aturan tertentu. Data dalam *file* midi dibagi menjadi dua bagian, yaitu bagian *header* dan bagian *track*. Berikut akan dijelaskan lebih lanjut mengenai dua bagian tersebut (MIDI *file*, 2007).

```
MIDI section <-----MIDI Header-----> <-----Track Header-----> <-Track data-> {Track out>
Byte number  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 [22 to x] [x+1 to x+4]
Byte data    4D 54 68 64 00 00 00 06 00 01 00 01 00 80 4D 54 72 6B 00 00 00 0A blah blah..... 00 FF 2F 00
```

2.1.1. Midi Header

Pada bagian *header* terdapat informasi mengenai lagu termasuk tipe *format* midi, jumlah *track*, dan *timing division*. Pada sebuah *file* midi hanya terdapat satu bagian *header*. Bagian *header* selalu terletak di depan. Berikut adalah tabel yang menggambarkan isi dari bagian *header*.

Offset	Panjang	Tipe	Deskripsi	Nilai
0x00	4	Char[4]	ID bagian	"MThd"(0x4D546864)
0x04	4	Dword	Ukuran bagian	6(0x00000006)
0x08	2	Word	Format tipe	0-2
0x10	2	Word	Jumlah track	1-65535
0x12	2	Word	Time division	Lihat penjelasan selanjutnya

Tabel 2.1. *Format Bagian Header Midi*

- *ID bagian midi header* dan ukuran
ID bagian midi header selalu "MThd" (0x4D546864) dan ukuran selalu 6 karena ukuran *header* midi selalu sama.
- *Format Tipe*
Menjelaskan tipe midi tersebut. Nilai tipe yaitu 0, 1, atau 2. Tipe 0 berarti jumlah track hanya terbatas 1. Tipe 1 jumlah track dapat bernilai lebih dari satu. Tipe 2 lebih kompleks lagi. Jumlah track dapat bernilai lebih dari satu, tetapi setiap *track* menggambarkan *sequence* yang berbeda, dimana tidak semuanya dimainkan secara bersamaan. Biasanya digunakan untuk menyimpan pola permainan drum.
- *Jumlah Track*
Berisi informasi mengenai jumlah *track* yang terdapat pada midi tersebut. Apabila midi bertipe 0, maka hanya dapat memiliki 1 track, sedangkan untuk tipe midi 1 atau 2 bisa berisi 1 hingga 65536 (0xFFFF).
- *Time Division*
Time Division digunakan untuk menguraikan *delta time* pada *track event* menjadi "real" time. Nilai yang terdapat pada *time division* menjelaskan jumlah *ticks per beat*. *Ticks per Beat* dapat diartikan banyaknya *ticks* pada tiap satu not seperempat.

2.1.2. Bagian *Track*

Bagian *Track* berisi informasi untuk masing-masing *track*, seperti nama *track*, dan *track event*. Berikut adalah tabel yang menggambarkan isi dari bagian *track*.

Offset	Panjang	Tipe	Deskripsi	Nilai
0x00	4	Char[4]	ID bagian	"MTrk"(0x4D54726B)
0x04	4	Dword	Ukuran	Lihat penjelasan selanjutnya
0x08	<i>Track event data</i> (Lihat penjelasan selanjutnya)			

Tabel 2.2. *Format Bagian Track Midi*

- *ID bagian track header* dan ukuran
ID bagian header selalu "MTrk" (0x4D54726B) dan ukurannya tergantung dari banyaknya *byte* yang terdapat pada *track event data*.
- *Track Event Data*
Track event data mengandung aliran *midi event* yang memberi informasi *event* apa yang terjadi pada *delta time* tertentu.

Track event digunakan untuk mendeskripsikan semua *content* musik dari *file midi*. Misalnya perubahan kondisi suatu not menjadi *on* atau *off* dan perubahan *tempo*. Setiap *Track event* selalu diawali dengan *delta-time*. Terdapat 3 jenis *track event*, yaitu : *Midi Channel Event*, *Meta Event*, dan *System Exclusive Event*.

- *Delta-time*
Delta-time merupakan variabel ukuran. Dengan *delta-time* dapat ditentukan kapan suatu *event* dijalankan. *Delta-time* bernilai 0 berarti *event* dijalankan bersamaan dengan *event* terakhir yang dijalankan. *Delta-time* dapat didefinisikan berapa lama suatu *event* harus menunggu sebelum dijalankan. *Event* yang tidak dipengaruhi oleh *delta-time* tetap memiliki *delta-time*, namun nilainya 0 dan biasanya terletak setelah *track header*. *Delta-time* tidak selalu hanya diwakili oleh 1 *byte*, namun dapat diwakili hingga 4 *bytes*. Batas nilai dari *byte* terakhir dari suatu *delta-time* yaitu dari 00 hingga 7F, dengan batasan ini maka kita akan mengetahui bahwa *byte* tersebut adalah *byte*

terakhir dari suatu *delta-time*. Jika membutuhkan *delta-time* yang melebihi nilai tersebut maka menggunakan 2 bytes yaitu 81 00 dan seterusnya. Batas maksimal *delta-time* yaitu hingga FF FF FF 7F. *Delta-time* bukan merupakan waktu yang absolut. Waktu yang aktual bisa didapatkan dari *time division* dan *tempo*. Jika *tempo* tidak didefinisikan, maka diasumsikan 120 *beats per minute*.

- *Midi Channel Event*

Informasi kontrol musik seperti dimainkannya sebuah not atau menentukan nilai *modulation* dari suatu *channel* dilakukan oleh *Midi channel event*. Setiap *Midi channel event* diawali dengan *delta-time* dan diikuti dengan 2 hingga 3 byte sesuai dengan tipe *Midi channel event*-nya. Berikut adalah bentuk *Midi channel event* secara umum.

Delta-time	Tipe Event	Channel	Parameter1	Parameter2
Variabel	4bit	4bit	1 byte	1 byte

Tabel 2.3. *Format Channel Event*

Terdapat 7 tipe *midi channel event*. Berikut adalah daftar tipe-tipe dari *midi channel event*:

Tipe Event	Nilai	Parameter 1	Parameter 2
Note Off	0x8	Nomor not	Velocity
Note On	0x9	Nomor not	Velocity
Note Aftertouch	0xA	Nomor not	Nilai aftertouch
Controller	0xB	Nomor controller	Nilai controller
Program Change	0xC	Nomor program	Tidak digunakan
Channel Aftertouch	0xD	Nilai aftertouch	Tidak digunakan
Pitch Bend	0xE	Pitch value (LSB)	Pitch value (MSB)

Tabel 2.4. Tipe – tipe *Midi Channel Event*

- *Meta Event*

Event yang tidak untuk dikirimkan atau diterima melalui *midi port* disebut dengan *meta event*. *Event* ini selalu diawali dengan *byte* 0xFF dan mempunyai variabel ukuran parameter yang tergantung tipe dari *meta event* tersebut.

Berikut adalah bentuk *meta event* secara umum.

Meta Event	Tipe	Panjang	Data
255 (0xFF)	0-255	Variabel ukuran	Spesifikasi tipe

Tabel 2.5. *Format Meta Event*

Terdapat 15 tipe *meta event*. Berikut ini adalah penjelasannya yang lebih detail.

- *Sequence Number*

Bila *event* ini terdapat pada *file* *midi* tipe 2, maka *event* ini menjelaskan pola angka. Bila terdapat pada *file* *midi* tipe 0 atau 1 maka *event* ini menjelaskan banyaknya *sequence*. *Meta event* ini selalu memiliki *delta time* bernilai 0 dan terletak sebelum *midi channel event* atau *event* lain yang memiliki *delta time* tidak sama dengan 0.

Meta Event	Tipe	Panjang	Angka(MSB)	Angka(LSB)
255 (0xFF)	0 (0x00)	2	0-255	0-255

Tabel 2.6. *Format Sequence Number Meta Event*

- *Text Event*

Meta event ini berisi *text* yang dapat digunakan untuk berbagai kebutuhan, namun biasanya digunakan untuk menuliskan komentar.

Meta Event	Tipe	Panjang	Text
255 (0xFF)	1(0x01)	Panjang string	ASCII text

Tabel 2.7. *Format Text Event Meta Event*

- *Copyright Notice*

Event ini menjelaskan informasi mengenai *copyright* (hak cipta) termasuk simbol © (0xA9), tahun, dan pencipta. *Meta event* ini selalu memiliki *delta time* bernilai 0 dan terletak sebelum *midi channel event* dan *event* lain yang memiliki *delta time* tidak sama dengan 0.

Meta Event	Tipe	Panjang	Text
255 (0xFF)	2(0x02)	Panjang string	ASCII text

Tabel 2.8. *Format Copyright Notice Meta Event*

- *Sequence/Track Name*

Pada *file* midi tipe 0 atau 2, *event* ini menjelaskan nama *sequence*. Pada *file* midi tipe 1, bila terletak pada *track* 1 berarti menjelaskan nama *sequence*. Jika tidak terletak pada *track* 1 berarti menjelaskan nama *track*.

Meta Event	Tipe	Panjang	Text
255 (0xFF)	3(0x03)	Panjang string	ASCII text

Tabel 2.9. *Format Sequence/Track Name Meta Event*

- *Instrument Name*

Event ini menjelaskan nama instrumen yang digunakan. *Event* ini dapat juga digunakan setelah *Midi Channel Prefix meta event* untuk mendefinisikan instrumen yang digunakan pada *channel* tertentu.

Meta Event	Tipe	Panjang	Text
255 (0xFF)	4(0x04)	Panjang string	ASCII text

Tabel 2.10. *Format Instrument Name Meta Event*

- *Lyrics*

Event ini digunakan untuk menuliskan lirik dari lagu.

Meta Event	Tipe	Panjang	Text
255 (0xFF)	5(0x05)	Panjang string	ASCII text

Tabel 2.11. *Format Lyrics Meta Event*

- *Marker*

Event ini digunakan untuk memberi tanda pada suatu waktu tertentu. *Event* ini sangat berguna untuk menandai awal atau akhir dari suatu *verse*(bait) atau *chorus*.

Meta Event	Tipe	Panjang	Text
255 (0xFF)	6(0x06)	Panjang string	ASCII text

Tabel 2.12. *Format Marker Meta Event*

- *Cue Point*

Event ini menandai dimulainya suatu aksi. *Event* ini biasanya digunakan untuk memberi tanda kapan aksi *playback* dimulai.

Meta Event	Tipe	Panjang	Text
255 (0xFF)	7(0x07)	Panjang string	ASCII text

Tabel 2.13. *Format Cue Point Meta Event*

- *Midi Channel Prefix*

Event ini menggabungkan *midi channel* dengan *meta event*. Efek dari *event* ini dapat dihilangkan oleh *channel prefix event* yang lain atau *non-*

meta event. *Event* ini biasanya digunakan sebelum *instrument name meta event* untuk menspesifikasi suatu nama instrumen terletak pada *channel* yang mana.

Meta Event	Tipe	Panjang	Channel
255 (0xFF)	32(0x20)	1	0-15

Tabel 2.14. *Format Midi Channel Prefix Meta Event*

○ *End of Track*

Event ini memberi informasi bahwa akhir dari *track* sudah tercapai. *Event* ini selalu terletak pada akhir dari bagian *track*.

Meta Event	Tipe	Panjang
255 (0xFF)	47(0x2F)	0

Tabel 2.15. *Format End of Track Meta Event*

○ *Set Tempo*

Event ini digunakan untuk menentukan *tempo* dalam satuan *microsecond per quarter note*. *Format*-nya dalam bentuk 3 *byte*. Bila tidak terdapat *event* ini, maka diasumsikan *tempo* = 120 *beat per minute*. Berikut adalah rumus untuk mengubah satuan *tempo* dari *microsecond per quarter-note* menjadi *beats per minute* dan sebaliknya:

$$\text{Microsecond_per_minute} = 60000000$$

$$\text{BPM} = \text{Microsecond_per_minute}/\text{MPQN}$$

$$\text{MPQN} = \text{Microsecond_per_minute}/\text{BPM}$$

Meta Event	Tipe	Panjang	Microsecond/Quarter Note
255 (0xFF)	81(0x51)	3	0-8355711

Tabel 2.16. *Format Set Tempo Meta Event*

○ *SMPTE Offset*

Event ini digunakan untuk menetapkan *SMPTE starting point* dari awal *track*. *Event* ini mendefinisikan *hours*, *minutes*, *second*, *frame*, dan *sub-frames* (selalu 100 *sub-frames per frame*). *Byte* yang digunakan untuk

menspesifikasi *hour*, juga menspesifikasi *fram rate* dengan *format* : Orrhhhhh dengan rr adalah 2 bit untuk *frame rate* dimana nilai 00=24 fps, 01=25 fps, 10=30 fps (pembulatan dari 29,97), 11=30 fps dan hhhhhh adalah untuk *hour*(0-23). *Byte* pertama *hour* selalu 0. Nilai *byte frame* yang memungkinkan tergantung pada *frame rate*. Jika 25 fps, maka nilai maksimumnya adalah 24.

Meta Event	Tipe	P	Hour	Min	Sec	Fr	SubFr
255 (0xFF)	84(0x54)	5	0-23	0-59	0-59	0-30	0-99

Tabel 2.17. *Format SMPTE Offset Meta Event*

o *Time Signature*

Event ini digunakan untuk mengatur birama(*time signature*). *Time signature* didefinisikan oleh 4 *byte* yaitu *numerator*, *denominator*, *metronome* dan jumlah not 32 per not seperempat. *Numerator* adalah angka pembilang (yang dibagi) pada birama. Nilainya sesuai dengan yang tertulis pada *byte numerator*. *Denominator* adalah angka penyebut (yang membagi) pada birama. Nilai sebenarnya didapat dengan cara 2 pangkat dari angka yang tertulis pada *byte denominator*. Misalnya *byte denominator* bernilai 0, maka nilai penyebut pada birama adalah 1 ($=2^0$), bila *byte* bernilai 1 maka penyebutnya 2 ($=2^1$), bila *byte* bernilai 2 maka penyebutnya 4 ($=2^2$) dan seterusnya. *Metronome* menjelaskan seberapa banyak ketukan, dimana standardnya 24 per note $\frac{1}{4}$. Sebagai contoh *metronome* bernilai 2 berarti ketukan terjadi setiap not $\frac{1}{4}$ (1 ketukan). *Metronome* bernilai 48 berarti ketukan terjadi tiap not $\frac{1}{2}$ (2 ketukan). *Parameter* terakhir yaitu menjelaskan banyaknya not $\frac{1}{32}$ pada tiap not $\frac{1}{4}$. Biasanya bernilai 8. Apabila tidak terdapat *event* ini berarti diasumsikan midi memiliki birama $\frac{4}{4}$, 24, 8 not $\frac{1}{32}$ pada tiap not $\frac{1}{4}$.

Meta Event	Tipe	Panjang	Numer	Denom	Metro	32nd
255 (0xFF)	88(0x58)	4	0-255	0-255	0-255	1-255

Tabel 2.18. *Format Time Signature Meta Event*

- *Key Signature*

Event ini digunakan untuk menentukan kunci (banyaknya tanda *sharp*/kres atau *flat*/mol) dan skala mayor atau minor. Nilai pada kunci positif berarti kres, sedangkan nilai negatif berarti mol. Nilai 0 pada skala berarti mayor, sedangkan 1 berarti minor.

Meta Event	Tipe	Panjang	Kunci	Skala
255 (0xFF)	89(0x59)	2	-7 hingga 7	0-1

Tabel 2.19. *Format Key Signature Meta Event*

- *Sequencer Spesific*

Event ini digunakan untuk menentukan informasi spesifik mengenai *hardware* atau *software* misalnya *ID* suatu pabrik tertentu pembuat *file* tersebut.

Meta Event	Tipe	Panjang	Data
255 (0xFF)	127(0x7F)	Panjang string	Tipe dan nilai

Tabel 2.20. *Format Sequencer Specific Meta Event*

- *System Exclusive Event*

Dikenal juga dengan nama *SysEx event*. *Event* ini digunakan untuk mengontrol midi *hardware* dan *software* yang membutuhkan data spesial tergantung spesifikasi pabrik. Setiap *Sysex event* memiliki *ID*, produk yang sesuai akan menjalankan *event*, sedangkan yang lain akan mengabaikan *event* ini. Terdapat 3 tipe *sysex event* yaitu:

- *Normal Sysex Event*

Event ini digunakan untuk menyimpan data pabrik tertentu. *Byte* pertama selalu 0xF0 dan kedua adalah panjang variabel dan diikuti oleh *sysex data*. *Sysex data* selalu diakhiri dengan 0xF7.

SysEx Event	Panjang	Data
240 (0xF0)	Panjang variabel	Data bytes, 0xF7

Tabel 2.21. *Format Normal SysEx Event*

○ *Divided SysEx Event*

Data yang besar pada *Normal SysEx* dapat menyebabkan *Midi Channel Event* yang harus dikirimkan jadi terlambat. Dengan *event* tipe yang kedua ini, masalah tersebut diatasi dengan membagi data yang besar tersebut menjadi bagian-bagian kecil. *Format*-nya hampir sama dengan *Normal SysEx Event*, hanya saja pada bagian akhir data tidak bernilai 0xF7. Hal tersebut menandakan bahwa *Event* belum berakhir dan dilanjutkan *Divided SysEx Event* yang selanjutnya. *Format* lanjutan *Divided SysEx Event* sama dengan *Normal SysEx Event*, hanya depannya berisi 0xF7 yang menandakan bahwa *event* tersebut adalah lanjutan. Jika *event* lanjutan tersebut merupakan *event* lanjutan yang terakhir, maka pada *byte* data terakhir bernilai 0xF7.

SysEx Event	Panjang	Data
240 (0xF0)	Panjang variabel	Data bytes
247 (0xF7)	Panjang variabel	Data bytes
247 (0xF7)	Panjang variabel	Data bytes,0xF7

Tabel 2.22. *Format Divided SysEx Event*

○ *Authorization SysEx Event*

Event ini dapat menyimpan catatan spesial seperti Posisi *pointer* dari lagu, *Midi Time Code*, dan pesan pilihan lagu. Nilai tipe *event* ini adalah 0xF7.

SysEx Event	Panjang	Data
247 (0xF7)	Panjang variabel	Data bytes

Tabel 2.23. *Format Authorization SysEx Event*

Berikut ini adalah contoh isi dari *file* midi tipe 0 dan tipe 1.

Type-0

MIDI Header	4D 54 68 64 00 00 00 06 00 00 00 01 00 80
Trk 1 Header	4D 54 72 6B 00 00 00 2E
Trk 1 data	00 C0 18 00 90 37 60 00 90 3C 60 81 00 3C 00 00 3E 60 81 00 3E 00 00 40 60 00 90 37 00 00 90 39 60 81 00 90 40 00 00 90 39 00
Trk 1 footer	00 FF 2F 00

File midi tipe 0, hanya dapat berisi 1 *track* artinya hanya bisa memainkan 1 jenis alat musik pada saat yang bersamaan.

Type-1

MIDI Header	4D 54 68 64 00 00 00 06 00 01 00 02 00 80
Trk 1 Header	4D 54 72 6B 00 00 00 1A
Trk 1 data	00 90 3C 60 81 00 3C 00 00 3E 60 81 00 3E 00 00 40 60 81 00 40 00
Trk 1 footer	00 FF 2F 00
Trk 2 Header	4D 54 72 6B 00 00 00 16
Trk 2 data	00 C1 18 00 91 37 60 82 00 37 00 00 39 60 81 00 39 00
Trk 2 footer	00 FF 2F 00

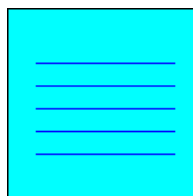
File midi tipe 1, jumlah *track* dapat berisi lebih dari 1. Artinya bisa memainkan lebih dari 1 jenis alat musik pada saat yang bersamaan.

2.2. Teori Mengenai Not Balok

Notasi musik adalah sistem penulisan karya musik. Dalam notasi musik, nada dilambangkan oleh not. Setiap nada memiliki frekuensi yang berbeda. Masing-masing nada digambarkan pada posisi yang berbeda pula. Notasi balok adalah standard yang digunakan dalam penulisan notasi musik. Notasi balok didasarkan pada paranada dengan lambang untuk tiap nada menunjukkan durasi dan ketinggian nada tersebut. Tinggi nada digambarkan secara vertikal sedangkan waktu/ritme digambarkan secara horisontal. Durasi nada ditunjukkan dalam ketukan. Penjelasan berikut direfensi dari (*Treblis's software, 2007*).

- Paranada (*Musical Staff*)

Garis Paranada menggambarkan urutan waktu dari kejadian musik. Paranada biasanya dibentuk oleh 5 buah garis yang dipisahkan oleh 4 buah spasi. Not diletakkan di bagian garis atau di bagian spasi dari paranada untuk menginformasikan nada mana yang dimainkan.



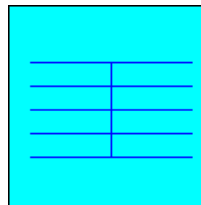
Gambar 2.1. Paranada (*Musical Staff*)

Not yang terletak di garis atau spasi lebih tinggi berarti memiliki tinggi nada yang lebih tinggi. Not pada paranada dibaca dari kiri ke kanan. Not yang terletak di sebelah kiri dimainkan sebelum not di sebelah kanan. Apabila

penulisan membutuhkan posisi not yang melebihi paranada, dapat menggunakan garis tambahan.

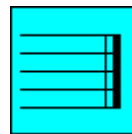
- **Garis Bar (*Bar lines*)**

Garis Bar diletakkan pada paranada untuk memisahkan bagian waktu. Waktu dibagi menjadi kumpulan ketukan sesuai dengan tanda birama. Jarak antara garis bar disebut dengan *measure* atau bar.



Gambar 2.2. Garis Bar (*Bar Lines*)

Terdapat pula tanda garis bar ganda, yang menandakan bahwa akhir dari suatu notasi sudah tercapai.



Gambar 2.3. Garis Bar Ganda (*Double Bar Line*)

- **Tanda Kunci (*Clef Symbol*)**

Tanda kunci adalah simbol yang terdapat pada bagian pertama dari paranada. Tanda kunci di sini menunjukkan rentang not dari karya musik yang dimainkan. Tanda kunci yang umum digunakan adalah tanda kunci *treble* dan tanda kunci *bass*.



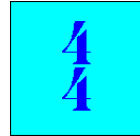
Gambar 2.4. Tanda Kunci G atau Tanda Kunci *Treble*



Gambar 2.5. Tanda Kunci F atau Tanda Kunci *Bass*

- Tanda Birama (*Time Signature*)

Angka di atas pada tanda birama menunjukkan banyaknya ketukan pada sebuah bar, sedangkan angka di bawah pada tanda birama menunjukkan nilai not yang dianggap sebagai satu ketuk. Tanda birama diletakkan setelah tanda kunci.

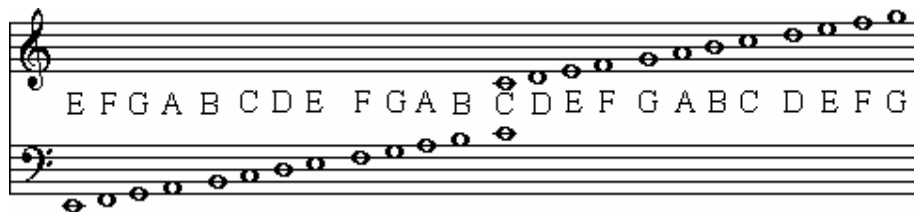


Gambar 2.6. Tanda Birama (*Time Signature*)

Terdapat bermacam-macam tanda birama, seperti *simple duple* (2/4), *simple triple* (3/4), *simple quadruple* (4/4), dan lain sebagainya.







- Not (*Note*)

Not adalah representasi dari nada di dalam notasi musik. Not memiliki informasi berupa tinggi nada dan durasinya. Berikut ini adalah posisi dari tiap-tiap not dalam tanda kunci *treble* dan *bass*.



Gambar 2.7. Posisi Not dalam Tanda Kunci *Treble* dan *Bass*

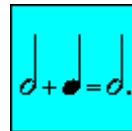
Untuk durasi dari suatu not digambarkan dengan simbol not yang berbeda. Berikut ini adalah simbol-simbol yang menggambarkan jenis nilai dari suatu not.

Not	Nilai	Not	Nilai
	Not Penuh (1)		Not Setengah (1/2)
	Not Seperempat (1/4)		Not Seperdelapan (1/8)
	Not Seperenambelas (1/16)		Not Sepertigadua (1/32)

Tabel 2.24. Jenis-jenis Nilai Not

- Not Bertitik (*Dotted Note*)

Titik diletakkan di sebelah kanan sebuah not, berarti durasi not tersebut bertambah setengah dari durasi aslinya.

Gambar 2.8. Not Bertitik (*Dotted Note*)

- Tanda Henti (*Rest*)

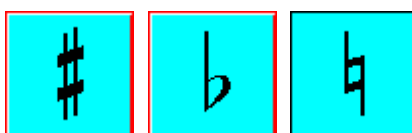
Tanda henti menunjukkan tidak adanya nada pada saat waktu tersebut. Sama halnya dengan not, tanda henti juga memiliki durasi. Namun tanda henti tidak memiliki informasi tinggi nada. Berikut ini adalah simbol yang menggambarkan nilai dari suatu tanda henti.

Henti	Nilai	Henti	Nilai
—	Istirahat Penuh (1)	▬	Istirahat Setengah (1/2)
⋈	Istirahat Seperempat (1/4)	⋏	Istirahat Seperdelapan (1/8)
⋏	Istirahat Seperenambelas (1/16)	⋏	Istirahat Sepertigadua (1/32)

Tabel 2.25. Jenis-jenis Nilai Tanda Henti

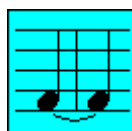
- *Accidental*

Accidental adalah simbol yang digunakan untuk menaikkan atau menurunkan tinggi nada dari suatu not. Simbol ini diletakkan di sebelah kiri suatu not. Terdapat 3 simbol *accidental*, yaitu *sharp* (menaikkan nada sebesar 1 *semitone*), *flat* (menurunkan nada sebesar 1 *semitone*), dan *natural* (membatalkan semua *accidental* pada nada tersebut). Sebagai contoh, untuk menghasilkan not yang merepresentasikan nada C#, dapat digunakan simbol *sharp* sebelum nada C. Berikut ini adalah simbol dari *accidental* yang ada.

Gambar 2.9. Tanda Kres (*Sharp*), Tanda Mol (*Flat*), Tanda Pugar (*Natural*)

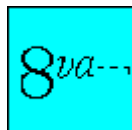
- *Tie*

Tie menghubungkan nilai suatu not dengan not yang lain. Biasanya digunakan untuk not yang nilainya melebihi nilai dari suatu bar.

Gambar 2.10. Tanda *Tie*

- *Octava*

Octava menunjukkan not-not pada paranada yang diberi garis dengan tanda ini dimainkan satu oktaf lebih tinggi dari nilai aslinya. Tanda *octava* diletakkan di atas paranada. *Octava bassa* menunjukkan bahwa not-not pada paranada yang diberi garis dengan tanda ini dimainkan satu oktaf lebih rendah. Tanda *octava bassa* diletakkan di bawah paranada.



Gambar 2.11. *Octava*