

# Desain Formula Perintah MIDI Untuk Aplikasi Musik Berbasis *Chord*

Studi Kasus: *Chord* Mayor dan Minor

Hadi Prasetyo Utomo

Program Studi Informatika Fakultas Teknik  
Universitas Langlangbuana  
Bandung, Indonesia  
hadi@informatika.unla.ac.id

**Abstrak**— Secara umum, permainan musik bernada dapat dilakukan dengan dua cara, yaitu secara melodi dan *chord*. *Chord* adalah harmonisasi tiga nada atau lebih yang dibunyikan secara bersamaan. Pada MIDI, satu perintah dikhususkan hanya untuk membunyikan satu nada saja. Sehingga untuk *chord*, dibutuhkan minimal tiga perintah MIDI yang dijalankan bersamaan. Jika diimplementasikan pada sebuah aplikasi, diperlukan formulasi agar perintah MIDI dapat di-generate oleh program tanpa harus menuliskan perintah MIDI satu per satu secara manual. Hasil formulasi kemudian diuji menggunakan bahasa pemrograman Java. Berdasarkan hasil pengujian, formula sudah sesuai dengan teori musik *chord* dan dapat diimplementasikan pada aplikasi sejenis.

**Kata kunci**— *formula, perintah, MIDI, musik, chord*

## I. PENDAHULUAN

Musik merupakan seni menyatukan suara-suara menjadi suatu kombinasi yang ekspresif dan dapat dinikmati. Kombinasi-kombinasi suara tersebut kemudian disusun menjadi suatu komposisi dengan struktur tertentu berdasarkan hukum harmoni melodi dan ritme [2]. Pada dasarnya, setiap otak manusia memiliki kemampuan untuk bermusik. Hal ini terbukti dengan mudahnya manusia menciptakan irama-irama tertentu dalam bentuk senandung, siulan, ketukan tangan hingga hentakan kaki [3]. Saat ini, bermain musik tidak hanya dilakukan dengan menggunakan alat musik konvensional. Kita dapat menggunakan laptop, *Tablet PC* ataupun *smartphone* sebagai medianya.

Dengan bantuan aplikasi khusus, media-media tersebut dapat menjadi alternatif dalam bermusik. Diharapkan ada aplikasi yang dapat mempermudah memainkan *chord* musik tanpa perlu menghafalkan posisi-posisi *chord*-nya. Hanya dengan menekan satu buah tombol, suara sesuai dengan nama *chord* yang tertulis pada tombol tersebutpun akan berbunyi. Agar dapat dipahami oleh *synthesizer* yang berbasis MIDI, perlu didesain formula yang dapat menghasilkan perintah MIDI untuk *chord* dan dapat dengan mudah diimplementasikan ke dalam aplikasi dengan bahasa pemrograman apapun.

## II. KAJIAN LITERATUR

Nada/not merupakan satuan dari notasi. Nada-nada ini dapat ditulis menggunakan huruf (A, B, C, D, E, F, G) atau ditulis menggunakan solmisasi (do, re, mi, fa, sol, la, si). Nada-nada solmisasi biasa disebut dengan nada-nada diatonik. *Chord* adalah kombinasi tiga nada diatonik atau lebih yang dibunyikan secara bersamaan sehingga memiliki harmoni tertentu [1]. Dalam *chord*, nada do biasanya digunakan sebagai nada dasar atau akar (*root*). Jika nada do = C, maka penamaan *chord*-nya adalah *chord* C, baik mayor, minor maupun yang lainnya.

*Chord* mayor terdiri dari tiga nada, yaitu do, mi dan sol. *Chord* minor juga terdiri dari tiga nada, yaitu do, re# dan sol. Untuk lebih jelasnya dapat dilihat pada Tabel 1 dan Tabel 2.

TABEL 1. *CHORD* MAYOR

Nama <i>chord</i>	Root/ Nada ke-1	Nada ke-2	Nada ke-3
C	C	E	G
C# (Db)	C# (Db)	F	G# (Ab)
D	D	F# (Gb)	A
D# (Eb)	D# (Eb)	G	A# (Bb)
E	E	G# (Ab)	B
F	F	A	C
F# (Gb)	F# (Gb)	A# (Bb)	C# (Db)
G	G	B	D
G# (Ab)	G# (Ab)	C	D# (Eb)
A	A	C# (Db)	E
A# (Bb)	A# (Bb)	D	F
B	B	D# (Eb)	F# (Gb)

TABEL 2. *CHORD* MINOR

Nama <i>chord</i>	Root/ Nada ke-1	Nada ke-2	Nada ke-3
Cm	C	D# (Eb)	G
C#m (Dbm)	C# (Db)	E	G# (Ab)
Dm	D	F	A
D#m (Ebm)	D# (Eb)	F# (Gb)	A# (Bb)
Em	E	G	B
Fm	F	G# (Ab)	C
F#m (Gbm)	F# (Gb)	A	C# (Db)

Nama chord	Root/ Nada ke-1	Nada ke-2	Nada ke-3
Gm	G	A# (Bb)	D
G#m (Abm)	G# (Ab)	B	D# (Eb)
Am	A	C	E
A#m (Bbm)	A# (Bb)	C# (Db)	F
Bm	B	D	F# (Gb)

MIDI (*Musical Instrument Digital Interface*) merupakan spesifikasi perangkat keras dan perangkat lunak yang digunakan oleh instrumen-instrumen musik dan perangkat lain untuk melakukan komunikasi antara satu sama lain [4]. MIDI menyediakan protokol komunikasi yang dapat digunakan untuk menyampaikan data kepada *synthesizer* yang menghasilkan bunyi, dan dapat juga digunakan untuk mengendalikan perangkat-perangkat non musik misalnya perangkat pencahayaan. Data MIDI sama sekali tidak mengandung data audio. Audio dihasilkan oleh *synthesizer* berdasarkan pesan-pesan MIDI tertentu.

Standar MIDI berbasiskan 16 kanal yang menjadi media pengiriman dan penerimaan data. Pesan yang dikirim pada suatu kanal tidak bergantung pada pesan yang dikirimkan pada kanal-kanal yang lain. Cara penggunaan kanal adalah dengan menyertakan nomor kanal pada setiap pesan MIDI yang dikirimkan sehingga masing-masing pesan tidak bertukar antara kanal satu dengan yang lain.

Komunikasi MIDI dilakukan dengan mengirimkan pesan-pesan yang terdiri dari *byte* status dan diikuti oleh nol atau lebih *byte* data. *Byte* status mendefinisikan jenis pesan dan kanal yang digunakan. Sedangkan *byte* data memberikan parameter terhadap pesan tersebut. Jenis pesan yang paling sering dipakai di dalam komunikasi MIDI adalah pesan yang secara langsung memberikan perintah terhadap suatu *synthesizer* untuk menghasilkan suara. Oleh karena itu, pesan tersebut dinamakan pesan kanal suara. Jenis-jenis pesan kanal suara dapat dilihat pada Tabel 3.

TABEL 3. PESAN KANAL SUARA

Pesan	Nilai Byte Status	Byte Data I	Byte Data II
<i>Note Off</i>	0x80 – 0x8F	Nada	Kecepatan
<i>Note On</i>	0x90 – 0x9F	Nada	Kecepatan
<i>Poly Key Pressure</i>	0xA0 – 0xA9	Nada	Tekanan
<i>Controller Change</i>	0xB0 – 0xBF	Nomor Kendali	Nilai
<i>Program Change</i>	0xC0 – 0xCF	Nomor program	-
<i>Channel Pressure</i>	0xD0 – 0xDD	Tekanan	-
<i>Pitch Bend</i>	0xF0 – 0xF9	Nilai Kasar	Nilai Halus

TABEL 4. PESAN NOTE ON DAN NOTE OFF

Kanal	Note On	Note Off	Nomor nada MIDI	Kecepatan
1	144	128	0 - 127	0 - 127
2	145	129	0 - 127	0 - 127
3	146	130	0 - 127	0 - 127
4	147	131	0 - 127	0 - 127
5	148	130	0 - 127	0 - 127
6	149	133	0 - 127	0 - 127

Kanal	Note On	Note Off	Nomor nada MIDI	Kecepatan
7	150	134	0 - 127	0 - 127
8	151	135	0 - 127	0 - 127
9	152	136	0 - 127	0 - 127
10	153	137	0 - 127	0 - 127
11	154	138	0 - 127	0 - 127
12	155	139	0 - 127	0 - 127
13	156	140	0 - 127	0 - 127
14	157	141	0 - 127	0 - 127
15	158	142	0 - 127	0 - 127
16	159	143	0 - 127	0 - 127

### III. METODE PENELITIAN

Dalam pelaksanaan penelitian ini penulis menggunakan metode pengumpulan data sebagai berikut:

#### A. Observasi Lapangan

Observasi ini dilakukan untuk mendapatkan gambaran situasi dan kondisi objek penelitian saat ini aplikasi musik berbasis *chord* yaitu BeatMe. Kegiatan ini juga berfungsi untuk mengetahui data primer yang dapat digunakan sebagai acuan awal penelitian.

#### B. Wawancara

Wawancara dilakukan terhadap pihak-pihak yang terkait dengan aplikasi dan disiplin ilmu yang berkaitan dengan *chord* musik.

#### C. Studi Kepustakaan

Studi kepustakaan ini dimaksudkan untuk pengumpulan dan memperoleh data sekunder dengan cara mempelajari, membaca dan mencatat literatur dari beberapa buku dan sumber yang berkaitan dengan permasalahan di atas.

### IV. HASIL DAN PEMBAHASAN

MIDI memiliki 128 nada dengan dengan jangkauan nilai antara 0 hingga 127. Nilai 60 merupakan nada medium C [4]. Setelah diketahui pemetaan pada masing-masing nada MIDI, selanjutnya dapat dipetakan nada-nada apa saja yang harus disiapkan untuk masing-masing *chord*, baik mayor maupun minor. Nada-nada inilah yang akan dijadikan sebagai rujukan untuk mendesain perintah MIDI sesuai masing-masing *chord*.

TABEL 5. PEMETAAN NADA MIDI UNTUK CHORD MAYOR

Nama chord	Nada-nada chord	Nada-nada MIDI (basis C=60)
C	C, E, G	60, 64, 67
C# (Db)	C# (Db), F, G# (Ab)	61, 65, 68
D	D, F# (Gb), A	62, 66, 69
D# (Eb)	D# (Eb), G, A# (Bb)	63, 67, 70
E	E, G# (Ab), B	64, 68, 71
F	F, A, C	65, 69, 72
F# (Gb)	F# (Gb), A# (Bb), C# (Db)	66, 70, 73
G	G, B, D	67, 71, 74

Nama chord	Nada-nada chord	Nada-nada MIDI (basis C=60)
G# (Ab)	G# (Ab), C, D# (Eb)	68, 72, 75
A	A, C# (Db), E	69, 73, 76
A# (Bb)	A# (Bb), D, F	70, 74, 77
B	B, D# (Eb), F# (Gb)	71, 75, 78

TABEL 6. PEMETAAN NADA MIDI UNTUK CHORD MINOR

Nama chord	Nada-nada chord	Nada-nada MIDI (basis C=60)
Cm	C, D# (Eb), G	60, 63, 67
C#m (Dbm)	C# (Db), E, G# (Ab)	61, 64, 68
Dm	D, F, A	62, 65, 69
D#m (Ebm)	D# (Eb), F# (Gb), A# (Bb)	63, 66, 70
Em	E, G, B	64, 67, 71
Fm	F, G# (Ab), C	65, 68, 72
F#m (Gbm)	F# (Gb), A, C# (Db)	66, 69, 73
Gm	G, A# (Bb), D	67, 70, 74
G#m (Abm)	G# (Ab), B, D# (Eb)	68, 71, 75
Am	A, C, E	69, 72, 76
A#m (Bbm)	A# (Bb), C# (Db), F	70, 73, 77
Bm	B, D, F# (Gb)	71, 74, 78

Berdasarkan pemetaan-pemetaan di atas, dapat didesain perintah-perintah MIDI yang dibutuhkan untuk bermain musik secara *chord* dengan menggunakan 16 kanal yang tersedia. Tipe pesan yang digunakan hanya dua, yaitu *Note On* dan *Note Off*. Pesan inilah yang akan dikirimkan untuk membunyikan atau pun mematikan nada MIDI pada *synthesizer*. Untuk *chord* mayor dan minor, ada tiga perintah MIDI yang dikirimkan sekaligus, baik untuk *Note On* maupun *Note Off*.

Setelah diamati, perintah-perintah tersebut ternyata membentuk sebuah pola yang dapat dibuat model matematisnya. Formula atau rumus yang tercipta dapat mempermudah penerapan perintah MIDI untuk *chord* ke dalam bahasa pemrograman. Algoritmanya dapat dijabarkan sebagai berikut.

- Masukkan perintah *Note On* dan *Note Off* setiap kanal secara terurut ke dalam *array*.
- Pilih kanal, *Note On* dan *Note Off* ditentukan sesuai kanal yang dipilih.
- Misal kanal yang dipilih adalah 1, maka *Note On* dan *Note Off* yang digunakan adalah indeks *array* yang ke-0 atau nomor kanal dikurangi 1.
- Tentukan basis awal, misal C=24.
- Penambahan dan pengurangan basis dapat dilakukan dengan menggunakan angka 12.
- Setiap naik atau turun setengah nada, nada MIDI bertambah atau berkurang 1, misal C=12 maka C#=13, D=14 dan seterusnya.

- Untuk *chord* mayor, formulanya adalah basis, basis+4, basis+7. Untuk *chord* C maka nada MIDI-nya adalah 24, 24+4, 24+7 hasilnya 24, 28, 31. Untuk *chord* C#, tambahkan basis dengan angka 1, lalu gunakan formula yang sama. Untuk *chord* D, tambahkan basis dengan angka 2, D# tambahkan basis dengan angka 3 dan begitu seterusnya sampai *chord* B.
- Untuk *chord* minor, formulanya adalah basis, basis+3, basis+7. Untuk *chord* Cm maka nada MIDI-nya adalah 24, 24+3, 24+7 hasilnya 24, 27, 31. Untuk *chord* C#m, tambahkan basis dengan angka 1, lalu gunakan formula yang sama. Untuk *chord* Dm, tambahkan basis dengan angka 2, D#m tambahkan basis dengan angka 3 dan begitu seterusnya sampai *chord* Bm.
- Tentukan kecepatannya, misal 120.
- Formula perintah MIDI untuk membunyikan *chord* mayor adalah *Note On*, basis, kecepatan | *Note On*, basis+4, kecepatan | *Note On*, basis+7, kecepatan.
- Formula perintah MIDI untuk membunyikan *chord* minor adalah *Note Off*, basis, 0 | *Note Off*, basis+3, 0 | *Note Off*, basis+7, 0.

Java adalah bahasa pemrograman yang digunakan dalam pengimplementasi-an. Formula perintah MIDI yang sudah didesain, diubah menjadi sintaks-sintaks yang sesuai dengan bahasa pemrograman Java. Hal yang harus dilakukan pertama kali adalah memasukkan pesan *Note On* dan *Note Off* semua kanal ke dalam *array*.

```
int arrOff[] = {128,129,130,131,132,133,134,135,136,137,138,139,140,141,142, 143};
```

```
int arrOn[] = {144,145,146,147,148,149,150,151,152,153,154,155,156,157,158, 159};
```

Selanjutnya adalah penentuan kanal yang akan digunakan oleh *user*. Setelah *user* menentukan kanal mana yang akan digunakannya, pesan *Note On* dan *Note Off* yang harus digunakan pun dapat ditentukan.

```
channel = 3;
```

```
noteOn = arrOn[channel-1]; noteOff = arrOff[channel-1];
```

Setelah kanal serta pesan *Note On* dan *Note Off* dapat ditentukan, selanjutnya adalah penentuan basis dan kecepatan yang akan digunakan.

```
initbase = 60; v = 100;
```

Setelah basis dan kecepatan diketahui, implementasi formula perintah MIDI untuk *chord* siap untuk dilakukan.

Pesan *Note On* akan dikirim jika *user* menekan *chord*. Sedangkan pesan *Note Off* akan dikirim jika *user* melepaskan *chord*.

Setelah implementasi formula perintah MIDI ini selesai, perintah MIDI untuk *chord* pun siap untuk diuji. Pengujian dilakukan untuk mengetahui apakah setiap formula yang telah diimplementasikan ke dalam bahasa pemrograman menghasilkan perintah MIDI sesuai dengan *chord* yang diharapkan. Pengujian dilakukan dengan cara menangkap *log* setiap perintah MIDI yang dihasilkan pada saat *user* menekan salah satu *chord*. Cara penangkapan *log* pada Eclipse dapat dilihat pada Gambar 1.

Application	Tag	Text
beat.me.chord	MIDI 1:	144,60,127
beat.me.chord	MIDI 2:	144,64,127
beat.me.chord	MIDI 3:	144,67,127
beat.me.chord	MIDI 1:	128,60,0
beat.me.chord	MIDI 2:	128,64,0
beat.me.chord	MIDI 3:	128,67,0

Gambar 1. Contoh hasil *screenshot log* pada Eclipse

Beberapa skenario dan hasil pengujiannya dapat dilihat pada Tabel 7.

TABEL 7. SKENARIO DAN HASIL PENGUJIAN PERINTAH MIDI

Skenario (kanal=1, basis=60, v=127)	Hasil Log	Kesimpulan
User menekan <i>chord C</i>	144,60,127 144,64,127 144,67,127	Sesuai
User melepas <i>chord C</i>	128,60,0 128,64,0 128,67,0	Sesuai
User menekan <i>chord C#</i>	144,61,127 144,65,127 144,68,127	Sesuai
User melepas <i>chord C#</i>	128,61,0 128,65,0 128,68,0	Sesuai
User menekan <i>chord Fm</i>	144,65,127 144,68,127 144,72,127	Sesuai
User melepas <i>chord Fm</i>	128,65,0 128,68,0 128,72,0	Sesuai
User menekan <i>chord F#m</i>	144,66,127 144,69,127 144,73,127	Sesuai

Skenario (kanal=1, basis=60, v=127)	Hasil Log	Kesimpulan
User melepas <i>chord F#m</i>	128,66,0 128,69,0 128,73,0	Sesuai

## V. SIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa formula yang didesain telah sesuai dengan perintah MIDI untuk *chord*. Hal ini dapat terlihat pada *log* yang dihasilkan, sehingga aplikasi musik berbasis *chord* sudah dapat mengimplementasikan formula yang telah didesain.

## REFERENSI

- [1] Benward, B., Saker, M., "Music in Theory and Practice", 8th ed., McGraw-Hill, New York, NY, 2009.
- [2] Bode, H., "Sound Synthesizer Creates New Musical Effects", Electronics, 34, 33-37, December 1961.
- [3] Ingham, R., "The Impact of Digital Music on Composition, Performance and Listening", Ends and Means, 4, no. 2, 2000. <http://www.abdn.ac.uk/philosophy/endsandmeans/vol4no2/ingham.shtml>. Diakses tanggal 12 Mei 2015, 15.00 WIB.
- [4] \_\_\_\_\_, "Complete MIDI 1.0 Detailed Specification", Standard, Version 96.1, 2nd Edition, MIDI Manufacturers Association, Los Angeles, CA, 2001.