

BAB II STUDI AWAL

2.1 Penala Gitar

Seutas dawai yang kedua ujungnya ditarik pada tegangan tertentu dan diikat pada titik yang tetap, bila digetarkan dengan cara dipetik atau dipukul, akan menghasilkan gelombang dengan frekuensi fundamental tertentu yang dapat dihitung dengan rumus :

$$f_0 = \left(\frac{1}{2L} \right) \left(\frac{T}{\mu} \right) \quad (1)$$

f_0 = Frekuensi fundamental (Hz)

L = Panjang dawai (m)

T = Tegangan dawai (kg m/dt atau Newton)

μ = Massa dawai per satuan panjang (kg/m)

Selain menghasilkan frekuensi fundamental, dawai-dawai itu akan menghasilkan frekuensi-frekuensi harmonisanya, yaitu sebesar $2f_0$, $3f_0$, $4f_0$, $5f_0$, $6f_0$, dan seterusnya. Bila frekuensi-frekuensi tersebut berada di dalam jangkauan pendengaran manusia, yaitu 20 s/d 20000 Hz, maka akan terdengar sebagai gelombang suara. Dalam musik, gelombang suara yang mempunyai frekuensi tertentu disebut nada. Tinggi rendahnya nada tergantung dari tinggi rendahnya frekuensi gelombang suara tersebut.

Nada yang terdengar dari getaran dawai ditentukan terutama oleh frekuensi fundamentalnya, dan kualitas suaranya tergantung pada jumlah frekuensi-frekuensi harmonisnya. Dengan standar internasional, ditentukan bahwa frekuensi standar dari nada a' adalah 440 Hz. Dan ditentukan juga bahwa perbandingan frekuensi pada jarak setengah nada dari suatu nada ke nada berikutnya adalah sebesar $\sqrt[12]{2^2}$ Frekuensi nada yang ditentukan atau *frekuensi fundamentalnya*.

Dengan penentuan frekuensi standar dari sebuah nada, yaitu nada a' frekuensinya adalah 440 Hz, maka dengan perbandingan-perbandingan frekuensi yang telah dibahas sebelumnya, kita dapatkan frekuensi-frekuensi untuk setiap nada, terlihat pada Tabel 1 berikut :

Tabel 2.3 Frekuensi Nada berdasarkan standar Internasional a' = 440 Hz.

Nama nada	Nama Oktaf					
	Kontra	Besar	Kecil	Satu	Dua	Tiga
C	32,703	65,496	130,813	261,626	523,251	1046,50
D	36,708	73,416	146,832	293,665	567,330	1174,66
E	41,203	82,406	146,814	329,628	659,255	1318,51
F	43,653	87,307	174,624	349,228	698,456	1396,51
G	48,999	97,998	195,988	391,995	783,991	1567,98
A	55,000	110,000	220,000	440,000	880,000	1760,00
B	61,735	123,470	246,942	439,883	987,767	1975,53

Dalam musik, nada-nada tersebut digambarkan dalam bentuk not-not yang ditulis pada garis paranada seperti pada gambar di bawah ini:



Gambar 2.2 Representasi nada-nada pada paranada

Tabel 2.4 Frekuensi standar dari nada yang ditala

Dawai	Frekuensi (Hz)
E ₂	82,406 Hz
A ₂	110,000 Hz
D ₁	146,832 Hz
G ₁	195,988 Hz
B ₁	246,942 Hz
E ₁	329,628 Hz

Sesuai persamaan 1 tinggi rendahnya frekuensi dapat diatur dengan menambah atau mengurangi ketegangan dawai (T), bilamana panjang dawai (L) dan massa dawai per satuan panjang (μ) adalah tetap. Pekerjaan mengatur ketegangan dawai agar dawai tersebut menghasilkan frekuensi standar dari suatu nada disebut dengan menala dawai.

2.2 LabVIEW

LabVIEW (singkatan dari *Laboratory Virtual Instrumentation Engineering Workbench*) adalah perangkat lunak komputer untuk bahasa pemrograman visual dari *National Instrument*. Tujuan dari program tersebut adalah mengotomatisasi pengolahan dan penggunaan alat ukur dalam setiap setup laboratorium.

Awalnya dirilis untuk Apple Macintosh pada tahun 1986, LabVIEW umumnya digunakan untuk pengolahan dan visualisasi data dalam bidang akuisisi data, kendali instrumentasi serta automasi industri yang dapat digunakan pada berbagai platform sistem operasi termasuk Microsoft Windows, berbagai versi UNIX, Linux, dan Mac OS X.

LabVIEW terintegrasi dengan ribuan perangkat keras dan menyediakan ratusan *built-in library* untuk analisis dan visualisasi data untuk membuat instrumentasi virtual.

2.2.1 Pemrograman Dataflow

Bahasa pemrograman yang digunakan dalam LabVIEW disebut sebagai bahasa G, yang merupakan bahasa pemrograman dataflow. Eksekusi ditentukan oleh struktur blok diagram grafis (kode sumber LV) yang memungkinkan pemrogram menghubungkan node-node fungsi yang berbeda dengan menggambar garis penghubung. Garis penghubung ini berfungsi menyebarkan variable-variabel dan setiap node dapat dieksekusi segera setelah semua data inputnya tersedia. Karena situasi ini mungkin terjadi untuk beberapa node secara simultan, maka bahasa G secara inheren mampu melakukan eksekusi program secara paralel.

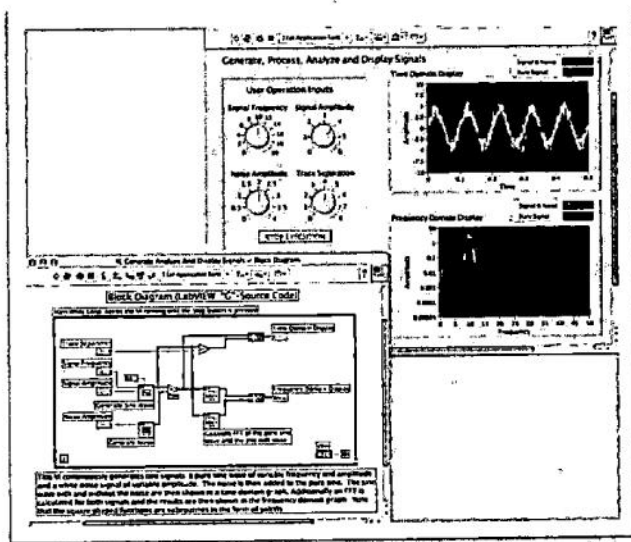
2.2.2 Pemrograman Grafis

Program (subrutin) LabVIEW disebut dengan *virtual instrument* (VI). Setiap VI memiliki tiga komponen: *block diagram*, *front panel (user interface)* dan *connector panel*. *Connector panel* digunakan untuk mewakili VI tersebut dalam *block diagram* dari VI lain yang memanggilnya.

Kontrol dan indikator pada *front panel* memungkinkan operator untuk memasukkan data ke atau mengambil data dari sebuah VI yang sedang berjalan. Namun, *front panel* juga dapat berfungsi sebagai *interface* program. Sehingga VI bisa dijalankan sebagai sebuah program, dengan *front panel* yang berfungsi sebagai *user interface*, atau, ketika dimasukkan sebagai node ke *block diagram*, *front panel* mendefinisikan input dan output untuk node tersebut melalui *connector panel*.

Ini berarti setiap VI bisa diuji dengan mudah sebelum tergabung sebagai sebuah sub-rutin dalam program yang lebih besar. Pendekatan grafis juga memungkinkan non-pemrogram untuk membangun program dengan men-*drag and drop* representasi virtual dari peralatan laboratorium yang sudah mereka kenal. Lingkungan pemrograman LabVIEW, termasuk contoh-contoh dan dokumentasi yang tersedia, memudahkan untuk membuat aplikasi kecil. Di satu sisi hal ini memberi manfaat, tetapi di sisi lain beresiko meremehkan keahlian yang diperlukan untuk pemrograman G berkualitas tinggi. Untuk algoritma yang rumit atau program berukuran besar, penting bagi pemrogram untuk memiliki pengetahuan yang luas tentang sintaks khusus LabVIEW dan topologi dari manajemen memori yang digunakan.

Pengembangan sistem LabVIEW yang paling maju menawarkan kemungkinan membangun aplikasi yang berdiri sendiri (*stand alone*). Selain itu, dimungkinkan untuk membuat aplikasi terdistribusi, yang dapat berkomunikasi dalam skema *client/server*, dan yang oleh karenanya lebih mudah diterapkan karena sifat dasar paralel dari bahasa G.



Gambar 2.2 Program sederhana LabVIEW *source code dataflow*

Gambar di atas merupakan gambaran dari sebuah program sederhana LabVIEW yang menunjukkan kode sumber *dataflow* dalam bentuk *block diagram* di jendela kiri bawah, serta variable input dan output sebagai obyek grafis di jendela kanan atas. Keduanya adalah komponen penting dari VI.

2.2.3 Interfacing

Manfaat utama dari LabVIEW dibanding lingkungan pengembangan software lainnya adalah dukungan yang luas untuk mengakses *hardware* instrumentasi. Driver dan lapisan abstraksi untuk berbagai jenis instrumen dan bus telah

disertakan, yang diwujudkan sebagai node-node grafis. Lapisan abstraksi menawarkan *software interface* standar untuk berkomunikasi dengan *hardware*. *Driver interface* yang disediakan menghemat waktu pembuatan program. Oleh karena itu, bahkan orang-orang dengan pengalaman pemrograman terbatas dapat menulis program dan menghasilkan solusi dalam waktu lebih singkat dibandingkan dengan bila menggunakan sistem lain.

2.2.4 Kompilasi Kode

LabVIEW menyertakan *compiler* yang menghasilkan kode bahasa mesin (*executable*) untuk platform CPU yang sesuai. Eksekusi berjalan dengan bantuan *runtime engine*, yaitu kumpulan prosedur yang telah dikompilasi sebelumnya, yang berfungsi melakukan tugas-tugas umum yang ditetapkan oleh bahasa G. *Runtime engine* dapat mengurangi waktu kompilasi dan juga menyediakan *interface* yang konsisten untuk berbagai sistem operasi, sistem grafis, komponen *hardware*, dan lain-lain.

2.2.5 Library

Tersedia banyak *library* dengan sejumlah fungsi untuk akuisisi data, pembangunan sinyal, matematika, statistik, pengkondisian sinyal, analisis, dan lain-lain, bersama dengan berbagai elemen *interface* grafis LabVIEW yang tersedia dalam beberapa pilihan paket.

Selain itu, LabVIEW menyertakan pula komponen pemrograman berbasis teks yang disebut MathScript dengan fungsi tambahan untuk pengolahan sinyal, analisis dan matematika. MathScript dapat diintegrasikan dengan pemrograman

grafis menggunakan *script nodes* dan menggunakan sintaks yang umumnya kompatibel dengan MATLAB.

2.2.6 Penggunaan-Ulang Kode

LabVIEW memungkinkan penggunaan-ulang (*reuse*) kode tanpa modifikasi. Selama jenis data input dan output konsisten, dua sub-VI dapat dipertukarkan.

Satu keuntungan dari lingkungan LabVIEW adalah sifat tak-bergantung platform dari bahasa G, yang membuatnya portabel untuk sistem yang berjalan di sistem operasi yang berbeda (Windows, Mac OS X dan Linux), kecuali untuk beberapa fungsi yang spesifik platform.

2.2.7 Lingkungan Runtime

LabVIEW *Profesional Development System* memungkinkan pembuatan *executable* yang berdiri sendiri (*stand alone*), yang kemudian dapat didistribusikan dalam jumlah tak terbatas. *Runtime engine* dan *runtime library*-nya dapat disertakan secara bebas bersama dengan *executable* tersebut. Namun *executable* itu tidaklah benar-benar mandiri, karena sebelum menjalankannya di sebuah komputer maka *runtime engine* harus sudah diinstal dulu di komputer tersebut.

Tidak sebagaimana *runtime library* untuk bahasa yang umum seperti C, *runtime library* yang diperlukan untuk LabVIEW tidak disediakan oleh sistem operasi dan oleh karenanya perlu diinstal secara khusus. Hal ini dapat menimbulkan masalah, yaitu ketika seorang pengguna siap menjalankan suatu aplikasi LabVIEW pada sebuah komputer tetapi ia tidak memiliki kewenangan untuk menginstal file-file tambahan pada computer tersebut.

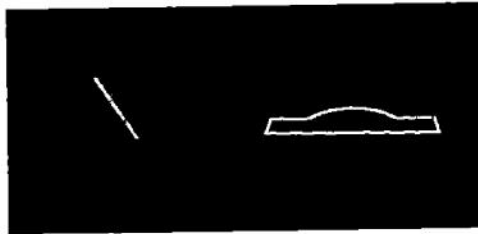
2.3 Karya Sejenis

Begitu banyak penala gitar yang dijual di pasaran dengan berbagai jenis dan merk, dari harga yang murah sampai dengan yang mahal. Seiring berkembangnya teknologi di berbagai bidang, maka penala gitar pun mengalami perkembangan yang cukup pesat.

Penala gitar analog telah berkembang sedemikian rupa menjadi penala gitar digital, beberapa gitar efek digital pun sudah menyertakan program penala gitar di dalamnya dan bahkan software penala gitar untuk PC pun sudah semakin banyak dikembangkan.

2.3.1 Penala Gitar Analog

Penala gitar analog dibuat dengan menggunakan komponen dasar rangkaian elektronis analog.



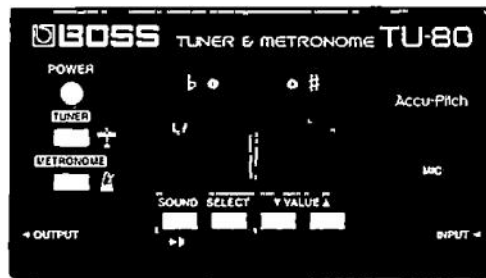
Gambar 2.3 Penala gitar analog

Gitar elektrik melalui kabel audio dihubungkan ke penala gitar analog tersebut, dan segera kita dapat melakukan penalaan. Jarum penunjuk akan menunjukkan angka 0 saat nada yang dipetik sudah ditala dengan benar (*tune*).

(<http://www.korg.com/Products.aspx?ct=46>)

2.3.2 Penala Gitar Digital

Penala gitar digital menggunakan mikroprosesor sebagai pengolah sinyal input, dan output ditampilkan dengan LCD. Penala gitar digital juga terdapat pada produk-produk efek gitar digital.



Gambar 2.4 Penala gitar digital

Cara penggunaannya sama dengan penala gitar analog, hanya saja jarum penunjuk ditampilkan secara digital dengan menggunakan LCD. Jarum penunjuk akan menunjukkan angka 0 saat nada yang dipetik sudah *tune*.

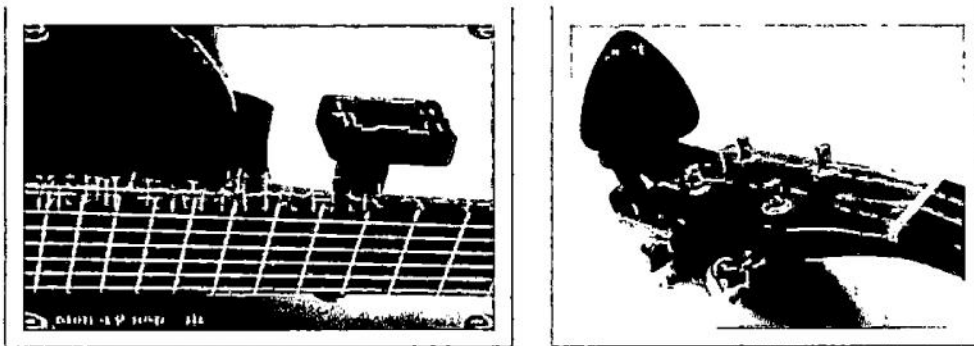
(<http://www.rolandus.com/products/productlist.php?ParentId=51>)



Gambar 2.5 Digital guitar tuner

Ada juga penala gitar digital yang cara penggunaannya dengan dijepitkan di ujung *headstock* (kepala gitar) atau *neck* (leher gitar). Dengan memanfaatkan getaran dari kayu badan gitar maka penala akan mendeteksi frekuensi getaran tersebut dan menampilkan informasinya pada LCD.

(<http://www.korg.com/Products.aspx?ct=46>)

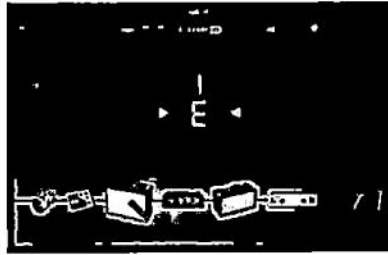


Gambar 2.6 Acoustic digital guitar tuner

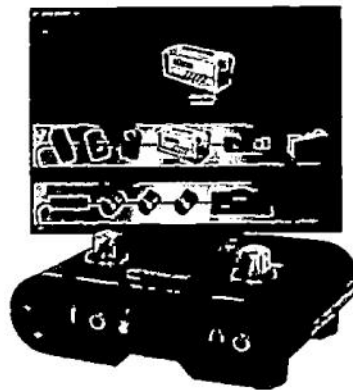
2.3.3 Software Penala Gitar

Ada beberapa contoh jenis software penala gitar untuk PC:

- Penala gitar yang terdapat di dalam software sebuah produk dan mengharuskan pengguna memasang hardware dari produk tersebut, misalnya seperti produk dari POD Line 6. Program *POD Farm Line 6* harus memasang hardware (*external soundcard*) dari produk POD yang berfungsi sebagai perangkat inputnya. (<http://line6.com/podfarm/>)



POD Farm default tuner

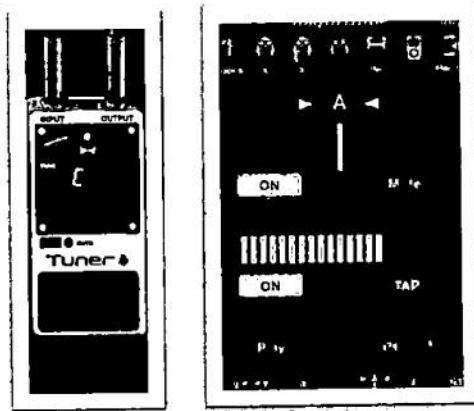


Line 6

Gambar 2.7 *POD Line 6 twin port (external device soundcard)*

- Program penala gitar yang disediakan sebagai sebuah plug-in dari software tertentu. Biasanya digunakan sebagai plug-in software recording seperti *Nuendo*, *Cubase* dan sebagainya.

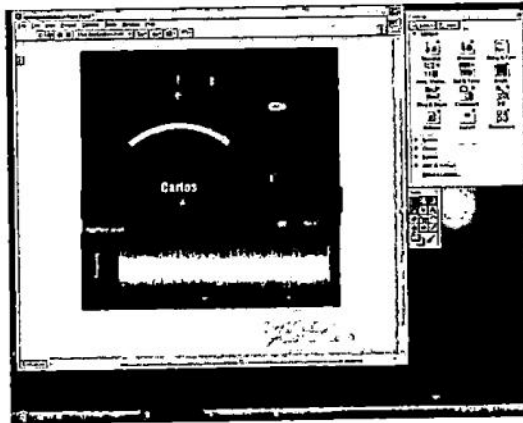
(<http://www.appletell.com/apple/comment/amplitude-review/>)



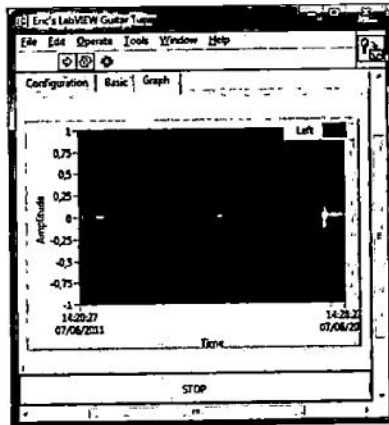
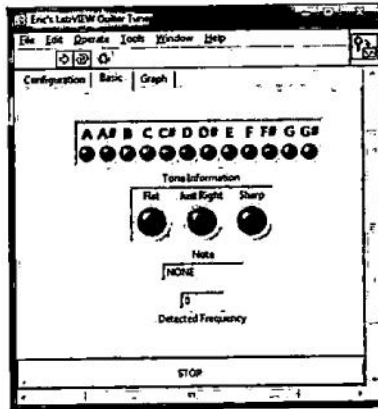
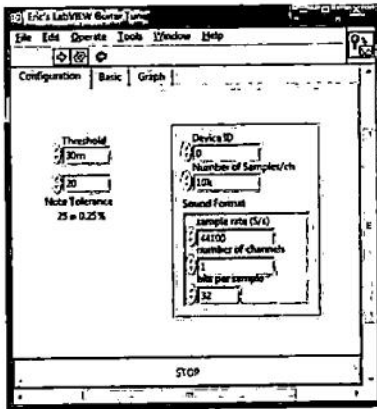
Gambar 2.8 Amplitube chromatic tuner

- Penala gitar yang dibuat dengan software tertentu, seperti Lab VIEW, yang memanfaatkan PC sebagai pengolah sinyal audio dan ditampilkan dalam bentuk *user interface* yang akan memberikan informasi tentang suatu nada.

(<http://forums.ni.com/t5/LabVIEW/Guitar-tuner/td-p/803433>)



Gambar 2.9 LabVIEW gitar tuner



Gambar 2.10 Eric's LabVIEW guitar tuner