BAB III

METODOLOGI PENELITIAN

Metode yang digunakan dalam penelitian ini terdiri dari langkah-langkah sebagai berikut:

1. Studi Literatur

   Melakukan studi kepustakaan melalui hasil penelitian orang lain maupun artikel-artikel atau paper-paper lainnya yang relevan, serta mempelajari teori dan teknik yang tepat untuk diimplementasikan dalam perangkat lunak pengenal nada.

2. Analisis dan Perancangan Sistem

   Melakukan analisis terhadap masalah dan kebutuhan yang diperlukan dalam membuat perangkat lunak pengenal nada dengan fitur *Automatic Gain Control*.

3. Implementasi

   Mengimplementasikan algoritma untuk membangun aplikasi tersebut.

4. Pengujian

   Melakukan pengetesan pembandingan antara beberapa sampel audio berformat WAV dengan durasi yang berbeda dan tempo yang berbeda.

Kemudian dilakukan analisis terhadap hasil dan performa perangkat lunak.

A. Analisis Sistem

Perangkat lunak ini dibangun untuk membantu orang yang merupakan pemula dalam musik, karena kesulitan untuk bisa mengetahui nada dari alunan musik yang baru ia dengar. Perangkat lunak ini dibuat menggunakan bahasa pemrograman C# dan mengimplementasi algoritma *Fast Fourier Transform* (FFT). Seperti yang telah dipaparkan dalam bab dua, FFT berfungsi untuk mentransformasi amplitudo dalam sample WAV menjadi nada.

Untuk meningkatkan keakuratan aplikasi dalam mengenali jeda antar nada, ditambahkan fungsi *Automatic Gain Control* (AGC). Dalam bab dua telah dijelaskan bahwa AGC memiliki tujuan untuk menyesuaikan tinggi dan menurunkan level daya bila sinyal terlalu kuat dan menaikkannya bila sinyal yang diterima terlalu rendah. Konsep dasar tersebut diimplementasi dalam perangkat lunak ini untuk menyesuaikan tinggi rendah dari amplitudo file WAV.

1. Fungsionalitas Sistem

Fungsionalitas yang dimiliki oleh sistem yang dibuat antara lain:

- Membaca *file audio digital* berformat WAV.
- Ditambahkan fungsi *Automatic Gain Control* terhadap amplitudo untuk menambah akurasi dalam mencari jeda antar nada.

- Melakukan pengenalan nada dalam *file* WAV dengan algoritma *Fast Fourier Transform*.
- Fungsi *print* untuk mencetak hasil dari program.

2. Tujuan Sistem

Tujuan dari sistem yang dibuat antara lain:

1. Melakukan pengenalan nada pada *file* WAV dengan algoritma *Fast Fourier Transform.*
2. Melakukan normalisasi pada amplitudo untuk menentukan berapa kali sebuah nada dimainkan pada rentang waktu tertentu sehingga dapat menghasilkan *sheet* musik sederhana.

3. Batasan Sistem

Sistem yang dibuat memiliki batasan-batasan sebagai berikut:

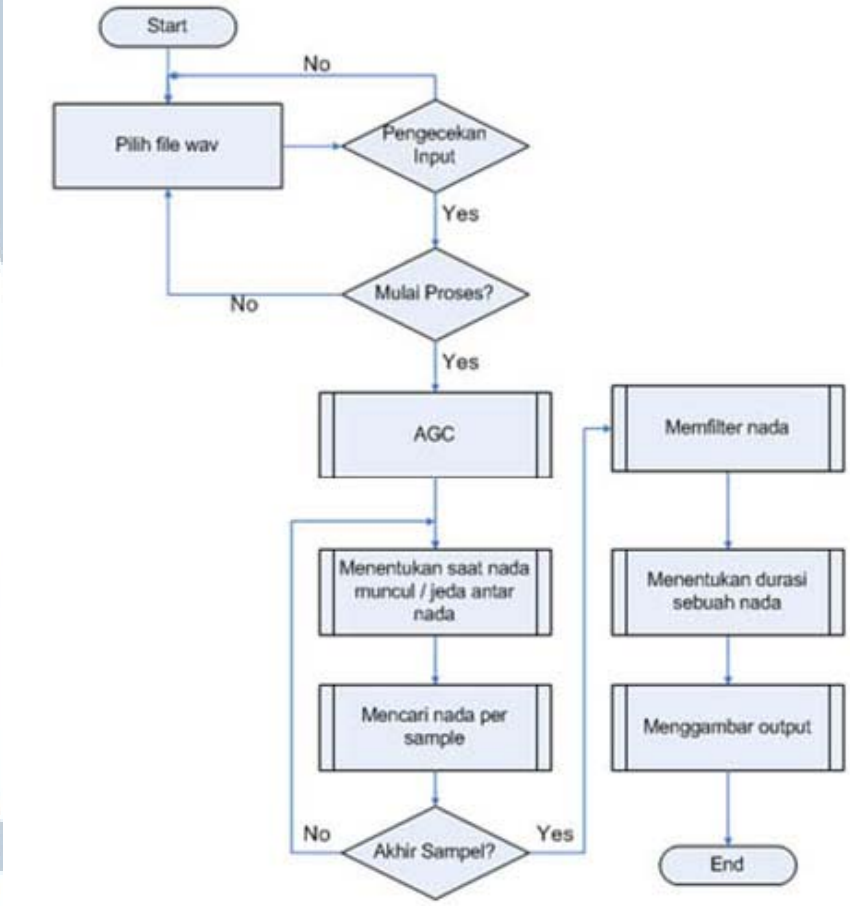
1. *File* masukan adalah sampel data audio berformat WAV dan berupa permainan solo *single instrument* gitar.
2. Menggunakan algoritma *Fast Fourier Transform* untuk mendapatkan frekuensi dari data sampel WAV kemudian dibandingkan dengan tabel frekuensi dasar dari masing-masing nada.
3. Penambahan *Automatic Gain Control* untuk memudahkan pemberian batasan untuk menentukan berapa kali sebuah nada dimainkan pada rentang waktu tertentu.
4. *Output* akan menampilkan nada dari *file* WAV per satuan waktu nada tersebut terdengar.

B. Perancangan Sistem

1. Proses Sistem

Sistem yang dibangun pada penelitian ini dapat digambarkan secara umum dengan *flowchart* seperti yang terdapat pada gambar di bawah ini.

Gambar 3.1 *Flowchart* Garis Besar Program



Dilihat dari Gambar 3.1, pada program ini terdapat 6 proses utama yaitu

1. Memilih *file* WAV yang akan diproses.

Pada proses ini akan dilakukan pengecekan apakah *file* yang dipilih merupakan *file* WAVyang benar. Program akan mengembalikan pesan error bila *file* yang dipilih adalah *file* yang salah.

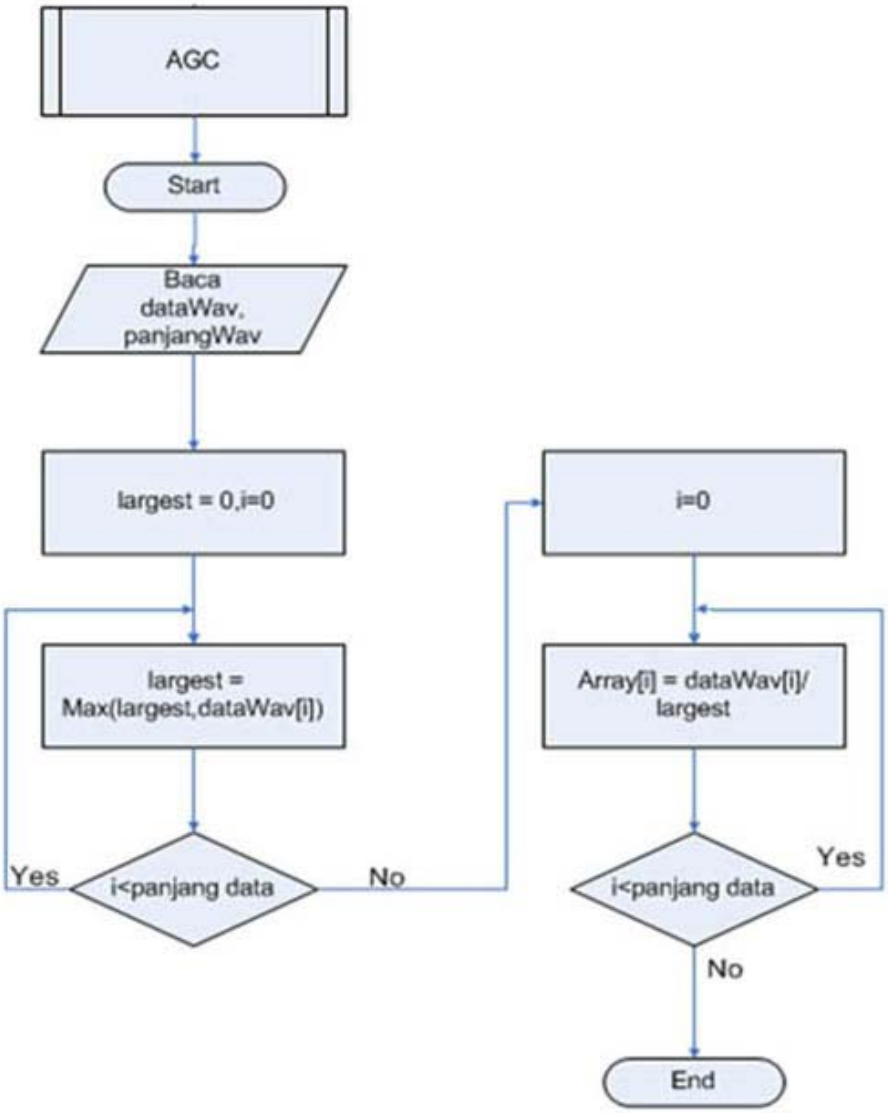
2. *Automatic Gain Control* (AGC)

Fungsi AGC bertujuan untuk menyesuaikan level dari data sample. Proses ini diawali dengan membaca dari awal sampai akhir data *sample* berformat WAV.

Kemudian dicari amplitudo terbesarnya. Setelah didapat amplitudo terbesarnya, amplitudo tersebut akan dipakai untuk penyesuaian amplitudo lainnya. Amplitudo kemudian dikonversi *range*-nya menjadi dari minus satu sampai dengan positif satu dengan membagi data pada sampel terhadap amplitudo terbesar yang telah dicari sebelumnya. Hal ini dilakukan untuk memudahkan pada saat memberi batasan. Batasan akan terpakai saat kita ingin menentukan jeda dari tiap nada yang muncul. Data yang telah dikonversi kemudian disimpan dalam array baru agar tidak lagi diperlukan pembacaan lagi pada *file* WAV. Data pada proses berikutnya dibaca dari *array* yang telah diubah amplitudonya.

Berikut adalah *flowchart* yang menjelaskan proses *automatic gain control*.

Gambar 3.2 *Flowchart* Proses AGC

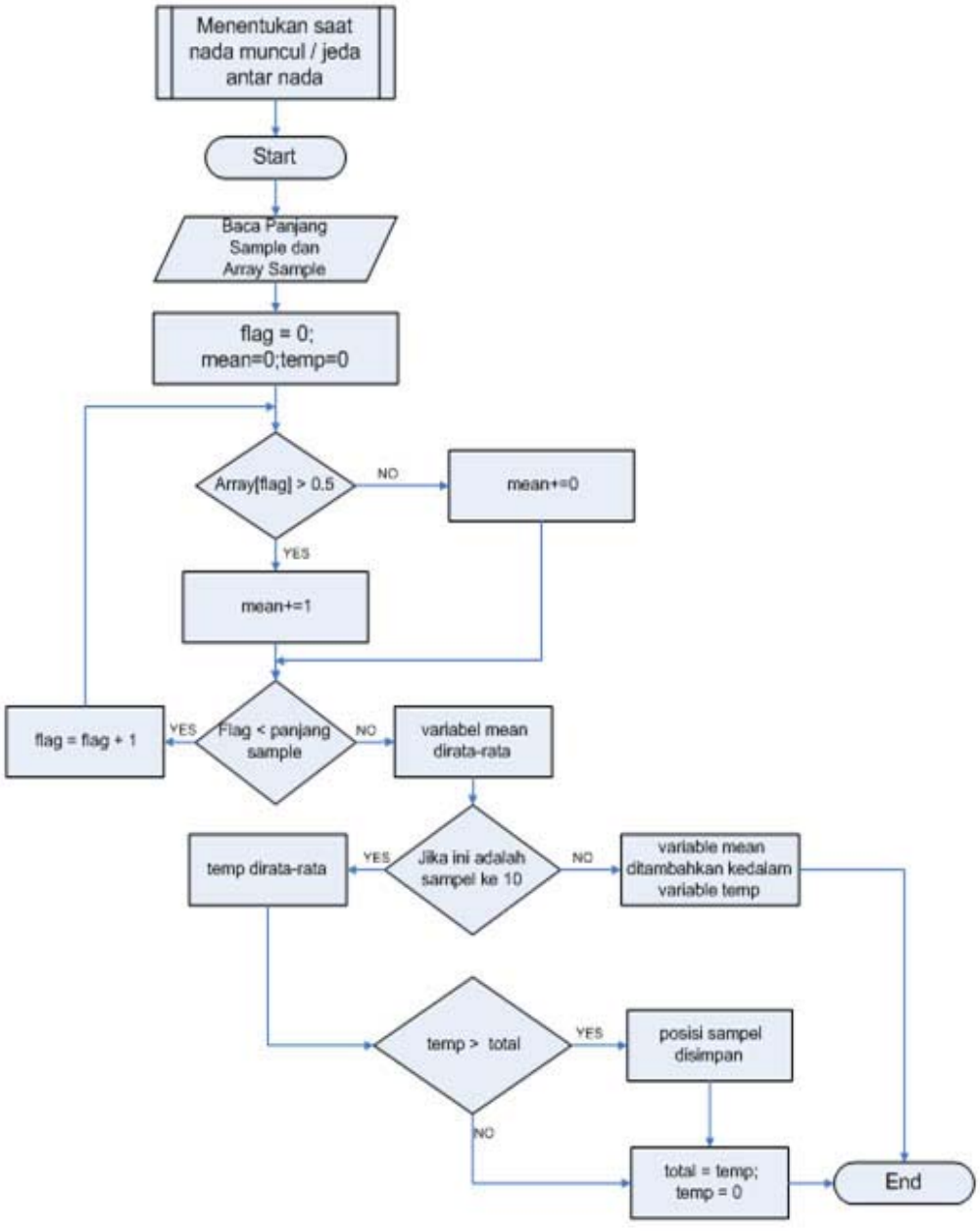


3. Menentukan jeda antar nada.

   Pada sebuah lagu sebuah nada dapat dimainkan beberapa kali pada rentang waktu tertentu. Sebagai contoh, nada C dapat dimainkan dengan panjang satu kali dalam rentang satu detik, atau bisa saja dimainkan dua kali dalam rentang waktu satu detik.

Berikut adalah *flowchart* yang menjelaskan proses penentuan saat nada muncul atau jeda antar nada.

Gambar 3.3 *Flowchart* Proses Menentukan Jeda Antar Nada



Pada proses ini, program akan mencari tahu kapan suatu nada dimainkan. Cara menentukannya adalah dengan mencari rata-

rata amplitudo tertinggi dari data sampel. Bila dalam nilai rata-rata terdapat peningkatan amplitudo yang signifikan, maka pada saat itu ada nada yang dimainkan.

4. Mencari nada per sampel

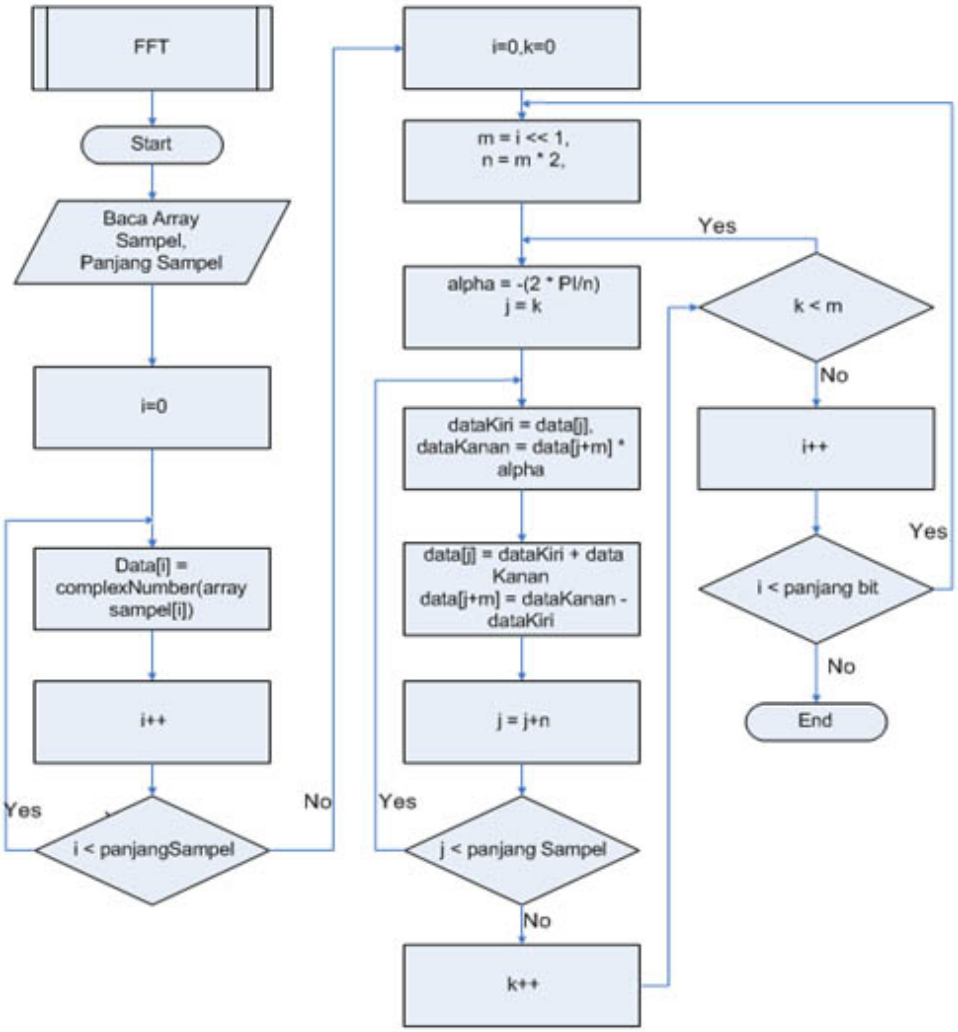
Pada proses ini dilakukan *Fast Fourier Transform* (FFT) pada *array* sampel. FFT dalam perangkat lunak ini dibuat berdasarkan pada aplikasi guitar tuner dalam website codeproject (Nomasteryet, 2010).

Dalam proses *Fast Fourier Transform*, dibutuhkan sebuah *array* yang berisi bilangan kompleks dan dilakukan *reverse bit* untuk memecah *array* sampel dalam dua bagian (Steven W. Smith, 1997). Kemudian dilakukan rekursif sampai bagian terkecil lalu ditambahkan pada bagian kanan dan bagian kirinya untuk didapat *value* frekuensinya. Fungsi perhitungan matematisnya dapat dilihat pada bab dua.

Setelah melakukan proses *Fast Fourier Transform*, akan didapatkan frekuensi dasar dari sebuah sampel yang kemudian akan diubah ke dalam nada dan menentukan pada oktaf keberapa nada tersebut.

Berikut adalah *flowchart* yang menjelaskan proses FFT.

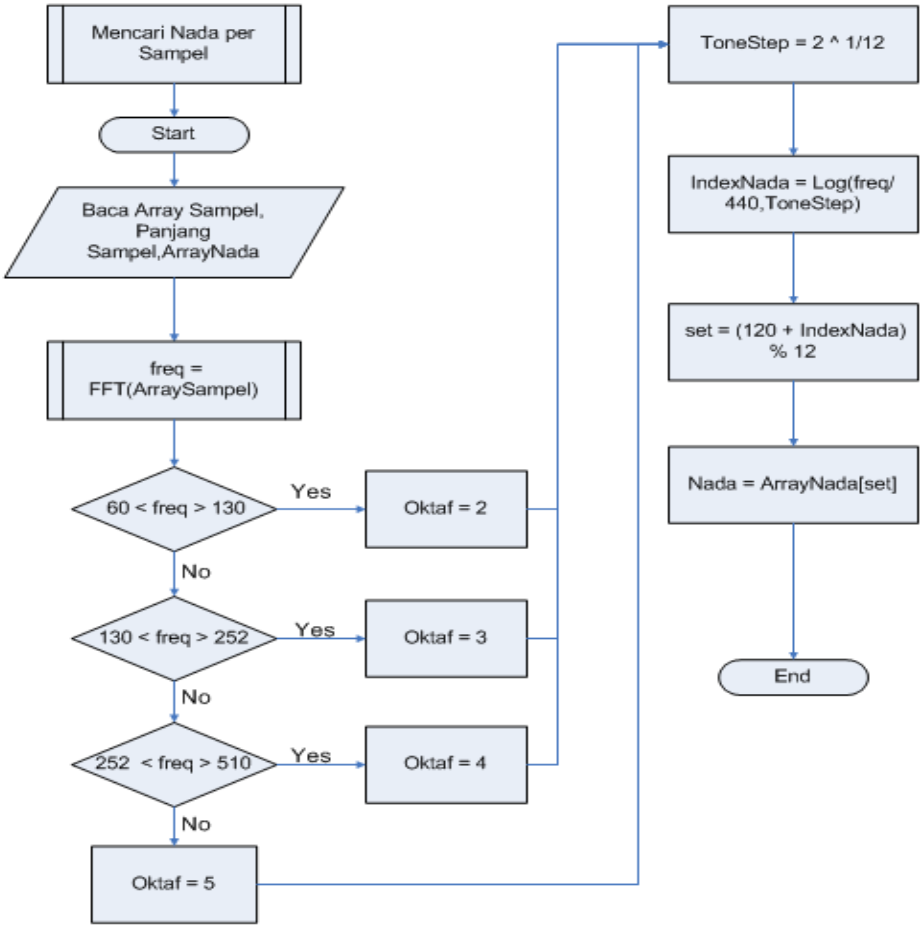
Gambar 3.4 *Flowchart* Proses FFT.



Untuk mengubah frekuensi ke dalam nada dilakukan dengan inisialisasi sebuah *array string* berisi dua belas nada dari A sampai nada G. Frekuensi yang didapat kemudian dimasukkan dalam rumus untuk menentukan nada dengan frekuensi dari nada A oktaf ke-4 yaitu 440Hz sebagai patokan.

Berikut adalah *flowchart* yang menjelaskan proses mencari nada per sampel.

Gambar 3.5 *Flowchart* Proses Mencari Nada per Sampel



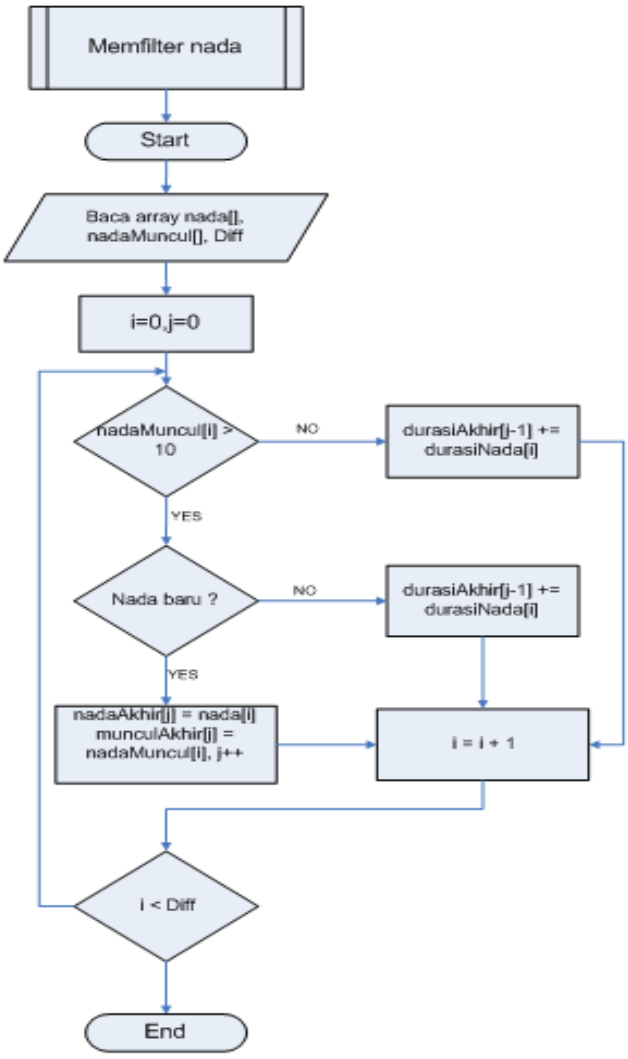
5. Memfilter Nada

Dari proses sebelumnya disimpan secara kasar nada-nada apa saja yang muncul untuk setiap sampel dan berapa kali kemunculan nada tersebut. Nada-nada yang telah didapat tersebut tidak semua adalah nada yang terdengar. Getaran dari senar gitar yang memainkan

suatu nada tidak langsung berhenti saat kita memainkan nada yang lainnya.

Berikut adalah *flowchart* yang menjelaskan proses memfilter nada.

Gambar 3.6 *Flowchart* Proses Memfilter Nada.



Ada frekuensi-frekuensi lemah yang merupakan kelanjutan dari frekuensi nada yang dimainkan. Untuk itu kita perlu proses filter nada. Frekuensi yang lemah tersebut memiliki beberapa ciri.

Frekuensi tersebut hanya tertangkap dalam jumlah yang kecil dan tidak beraturan rapat gelombangnya.

6. Menggambar *output*

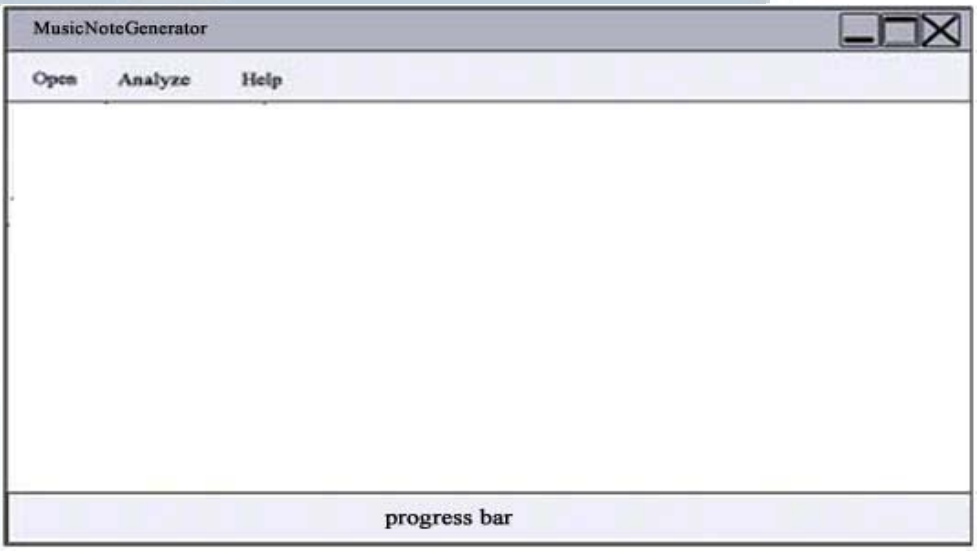
Berikut adalah *flowchart* yang menjelaskan proses menggambar *output.*

. Gambar 3.7 *Flowchar*t Proses Menggambar *Output.*

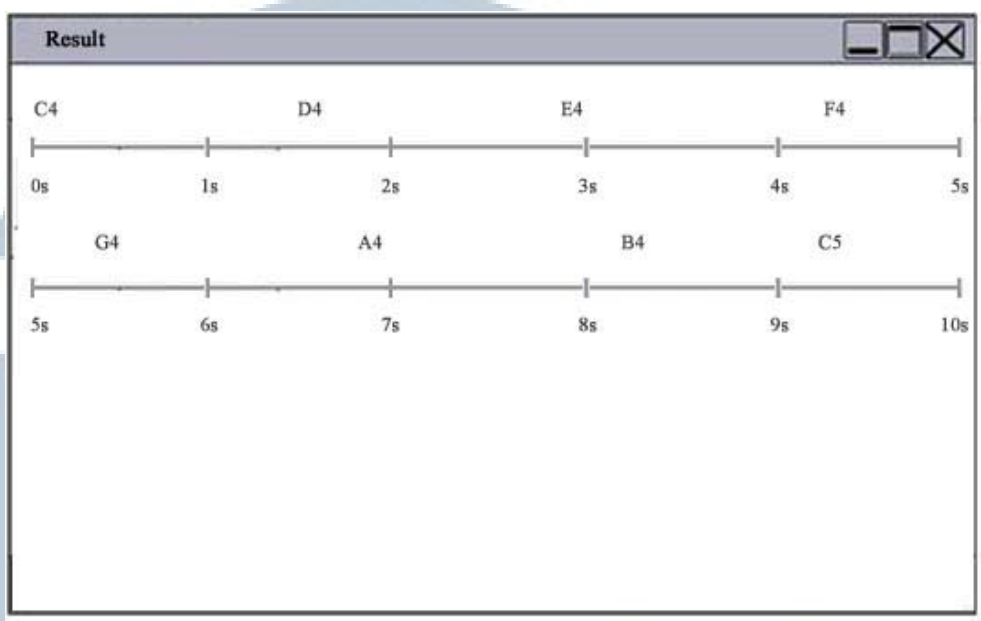
Ini adalah proses terakhir dalam program. Data yang telah didapat merupakan nada-nada yang terdapat pada *file* WAV beserta dengan durasinya. *Output* akan berupa nada yang digambar di atas *bar* waktu sesuai dengan lama nada tersebut terdengar.

2. Desain Antarmuka

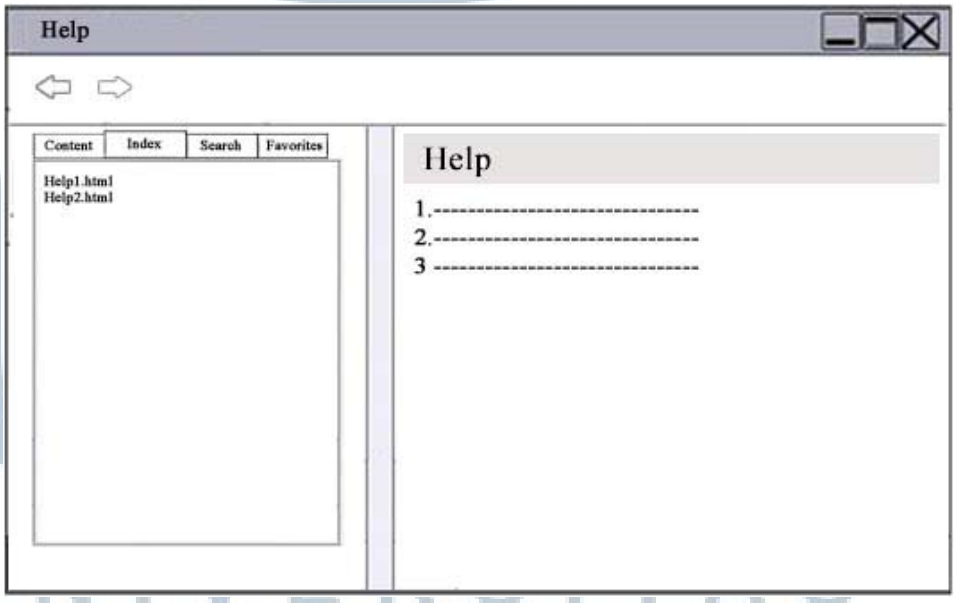
Desain tampilan antarmuka (*interface*) yang akan diimplementasikan ke dalam sistem tampak seperti gambar di bawah ini.



Gambar 3.8 Desain Tampilan Antarmuka Halaman Utama



Gambar 3.9 Desain Tampilan Antarmuka Halaman



Gambar 3.10 Desain Tampilan Antarmuka Halaman Help

C. Spesifikasi Sistem

Pada bagian ini akan dijabarkan tentang spesifikasi perangkat keras dan perangkat lunak dari komputer yang dipakai untuk proses implementasi dan pengujian.

Spesifikasi perangkat keras yang digunakan adalah

1. Laptop Acer Aspire 4920G
2. Processor Intel® Core™2Duo T7300 (2.0 GHz)
3. Memory RAM 3.0 GB
4. Sound card Realtek High Definiton Audio
5. Microphone built-in.
6. Harddisk 160 GB
7. Gitar Yamaha C370

Spesifikasi perangkat lunak yang digunakan adalah

1. *Operating System* Windows Vista™ Home Premium SP1.
2. Microsoft Visual Studio 2008.
3. Bahasa Pemrograman C#.