BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Akord atau *chord* merupakan harmonisasi dan kombinasi nada-nada yang dihasilkan ketika dimainkan secara bersamaan. Otomatis *chord* dalam gitar adalah kombinasi nada-nada dari alat musik gitar yang menimbulkan harmonisasi sehingga enak untuk didengarkan [1]. Salah satu permasalahan dalam bidang musik adalah bagaimana membuat beberapa komposisi akord untuk sebuah lagu. Dengan banyaknya variasi komposisi akord sebuah lagu dapat memberikan warna baru dalam melakukan aransemen pada musik dan memberikan karakter tersendiri bagi lagu tersebut.

Banyak penelitian mengenai pembangkitan akord untuk mendapatkan beberapa komposisi akord sebuah lagu baik itu dari suara musik atau sinyal audio maupun dari not-not angka sebuah lagu. Pada penelitian sebelumnya pembentukan akord dengan not-not angka yang tersedia, dilakukan dengan metode algoritma *direct product*. Komposisi akord lagu yang terbentuk dengan algoritma ini merupakan hasil perkalian langsung beberapa komposisi akord tiap bar pada lagu. Pada masing-masing bar akan muncul beberapa komposisi akord, dimana komposisi akord tersebut mungkin akan sama antara bar yang satu dengan bar yang lainnya sehingga memungkinkan proses penyimpanan komposisi akord bar yang sama. Hal ini dapat menyebabkan pemakaian alokasi memori yang cukup besar untuk menyimpan komposisi akord bar. Oleh karena permasalahan memori tersebut, metode ini memiliki keterbatasan dalam menghasilkan variasi susunan akord yang dihasilkan [4].

Penggunaan *resource* memori dapat dialokasikan secara efisien sehingga penyimpanan data yang sama tidak perlu disimpan ke dalam *resource* yang baru. Saat proses penyimpanan ke dalam alokasi memori, diperlukan sebuah fungsi atau metode pengecekan nilai untuk mengecek kesamaan nilai atau string antara nilai yang terbentuk dengan nilai yang ada sebelumnya sehingga tidak terjadi

penyimpanan untuk nilai yang sama. Pada metode pembangkitan akord, seharusnya komposisi akord bar yang terbentuk tidak langsung disimpan ke dalam alokasi atau *resource* memori yang baru namun dilakukan pengecekan komposisi akord terlebih dahulu dengan beberapa komposisi akord bar sebelumnya. Jika tidak terdapat komposisi akord yang sama maka komposisi akord yang baru akan disimpan pada *resource* memori yang baru. Salah satu metode untuk membangkitkan akord birama lagu adalah Metode Model NFA sedangkan metode yang digunakan untuk mengecek komposisi akord dapat menggunakan metode algoritma *levenshtein*.

Metode NFA merupakan bagian dari FA (*Finite Automata*) yang merupakan mesin abstrak berupa sistem model matematika dengan masukan dan keluaran diskrit yang dapat mengenali bahasa paling sederhana dan dapat diimplementasi secara nyata. Sedangakan NFA (*Nondeterministik Finite Automata*) sendiri yaitu *finite automata* dengan aturan yang memungkinkan satu simbol menimbulkan transisi ke lebih dari satu state. Model NFA ini digunakan untuk memunculkan komposisi akord-akord yang muncul dari susunan not-not angka.

Sedangkan algoritma *levenshtein* adalah algoritma pengecekan string yang berguna untuk menentukan apakah string yang kita masukan sama dengan pola string yang ada [7]. Pengecekan ini akan dibuatkan secara matriks 2 dimensi, kemudian dilakukan pengecekan satu persatu karakter pada dua string yang dibandingkan. Setelah itu kita akan mendapatkan suatu nilai *edit distance* berupa suatu pengukuran (*metrik*) yang dihasilkan melalui perhitungan jumlah perbedaan yang terdapat pada dua string[5]. Algoritma ini akan digunakan untuk mengecek apakah komposisi akord yang terbentuk pada akord selanjutnya telah ada pada komposisi akord sebelumnya. Sehingga selama pemakaian *resource* memori dapat dialokasikan secara efisien saat proses penyimpanan komposisi akord.

Berdasarkan latar belakang di atas maka pada penelitian ini akan mengimplementasikan model *Nondeterministic State Automata* (NFA) dan algoritma *levenstein* untuk mengefisienkan penggunaan memori dalam membangkitkan komposisi akord bar dari susunan not angka sebuah lagu.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan di atas maka rumusan masalah dalam penelitian ini adalah bagaimana mengefisiensikan penggunaan memori dengan mengimplementasikan model *Nondeterministic State Automata* (NFA) dan algoritma *levenstein* dalam membangkitkan komposisi akord bar dari susunan not angka sebuah lagu.

1.3 Maksud dan Tujuan

Berdasarkan permasalahan yang dipaparkan dalam latar belakang, maka maksud dari penelitian ini adalah mengimplementasikan model *Nondeterministic State Automata* (NFA) dan algoritma *levenshtein* dalam membangkitkan komposisi akord dari not-not angka sebuah lagu.

Adapun tujuan penelitian ini, yaitu untuk mengetahui efisiensi penggunaan memori dari implementasi model *Nondeterministic State Automata* (NFA) dan algoritma *levenshtein* dalam membangkitkan komposisi akord bar sebuah lagu.

1.4 Batasan Masalah

Berdasarkan latar belakang yang telah dideskripsikan di atas maka dibatasi masalah sebagai berikut :

1. Notasi lagu yang dijadikan objek penelitian hanya lagu-lagu daerah yang bergenre musik mayor.
2. Nada dasar Do setiap lagu diasumsikan adalah Do = C.
3. Birama lagu menggunakan birama 2/4, 3/4 atau 4/4.
4. Nilai tiap not adalah 1/4 ketuk.
5. Data masukan berupa teks not-not angka dan notasi musik yang tersimpan pada file berformat *txt.
6. Model *Nondeterministic Finite Automata* (NFA) digunakan untuk memunculkan akord yang mungkin.
7. Algoritma *levenshtein* digunakan dalam pengecekan komposisi akord tiap birama.
8. Akord yang dimunculkan merupakan akord *triad* (akord yang terbentuk maksimal terdiri dari 3 susunan not angka).

9. Aplikasi dibangun dengan bahasa pemrograman desktop.

1.5 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penelitian tugas akhir ini adalah metodologi penelitian deskriptif yaitu metode pemecahan masalah yang diselidiki dengan menggambarkan keadaan subjek, objek, suatu set kondisi pada masa sekarang sehingga penelitian dapat dipusatkan pada masalah aktual [9].

1.5.1 Metode Pengumpulan data

1. Pengumpulan data
   a. Studi Literatur.

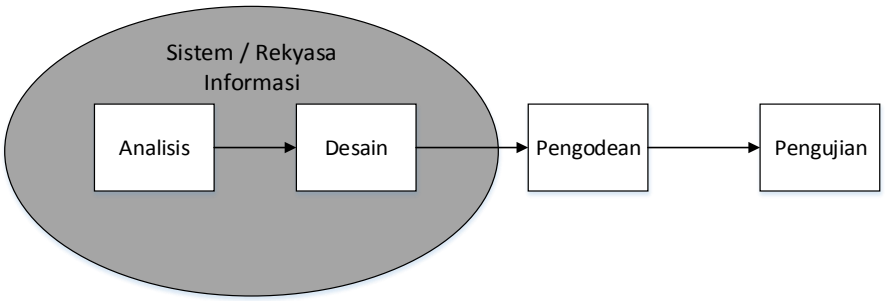
      Teknik pengumpulan data dengan mengkaji dan memahami berbagai literatur yang bersumber baik dari buku, jurnal, situs internet maupun literatur lainnya yang berkenaan dengan teori musik, teori TBA (Teori Bahasa Automata) yang mencangkup *lexical analizer* dan *parser* model NFA, algoritma levenshtein, bahasa pemrograman desktop (dalam penelitian ini menggunakan bahasa pemrograman C#), teori pembuatan *flowchart* dan UML (*Unified Modelling Language*) dan teori perancangan antarmuka.

   b. Wawancara.

      teknik pengumpulan data yang dilakukan dengan melakukan wawancara langsung kepada orang yang mampu memberikan pengetahuan berkaitan dengan objek penelitian. Dalam hal ini peneliti mencoba melakukan wawancara kepada salah seorang pengajar musik, yaitu Rian Victor S.Sn., M. Pd. Beliau berperan sebagai narasumber yang menerangkan bagaimana cara membuat komposisi akord dari not-not angka.

1.5.2 Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah metode *waterfall*. Secara garis besar metode *waterfall* meliputi beberapa tahapan yang digambarkan oleh gambar 1.1.



Gambar 1.1 Model Pengembangan Perangkat Lunak *Waterfall* [10]

Berikut tahapan proses yang akan dikerjakan :

1. Analisis

   Pada tahap ini akan dilakukan analisis pembangkitan akord dari not-not angka, analisis *lexical analyzer*, analisis *semantic analizer* (*parser*), analisis model NFA dan algoritma *levenshtein*. Kemudian menganalisis penggabungan antara model NFA dan algoritma *levenshtein* dalam membuat komposisi akord.

2. Desain

   Pada tahap ini akan dilakukan proses multi lankgah yang berfokus pada beberapa atribut atau hal yang berbeda, yaitu perancangan aplikasi mengenai penelitian ini berupa perancangan *flowchart* (prosedural atau algoritmik), perancangan perangkat lunak berupa perancangan UML (*Unified Modelling Language*) dan perancangan antarmuka (*interface*).

3. Pengodean

   Pada tahap ini akan dilakukan penerjemahan dari perancangan yang sebelumnya telah dibuat pada proses *deisgn* (perancangan) ke dalam bahasa pemrograman desktop (bahasa pemrograman yang dipakai C#).

4. Pengujian

Pada tahap ini akan dilakukan pengujian terhadap aplikasi sebagai implementasi metode NFA dan algoritma levenshtein yang bertujuan untuk mengecek apakah metode tersebut dapat menangani kesalahan. Tahapan pengujian yang dilakukan adalah tahapan *profiling*. Tahapan *profiling* yaitu untuk mengukur penggunaan memori program yang digunakan dengan melihat berapa indeks array komposisi akord bar yang terbentuk dengan metode *Nondeterministic Finite Automata* (NFA) dan algoritma *levenshtein*.

1.6 Sistematika Penulisan

Sistematika penulisan laporan penelitian ini disusun untuk memberikan gambaran umum mengenai penelitian yang dilakukan. Sistematika penelitian ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini menjelaskan secara umum mengenai latar belakang masalah, rumusan masalah, maksud dan tujuan, batasan masalah, metodologi penelitian serta sistematika penulisan yang digunakan dalam penyusunan laporan penelitian.

BAB II LANDASAN TEORI

Bab ini berisi mengenai teori-teori yang berkaitan dengan musik yang meliputi pengertian musik, pengertian not musik dan not angka, pengertian tangga nada, pengertian akord. Selain teori mengenai Teori Bahasa Otomata (TBA) yang meliputi *Finite State Automata* (FSA), model NFA (*Nondeterministic Finite Automata*), *lexical analyzer* dan *semantic analyzer*, algoritma *levenshtein*, teori mengenai *object oriented programming* (OOP), teori perancangan perangkat lunak UML (*Unified Modelling Language*) serta bahasa pemrograman C# yang digunakan berikut konsep dasar dan teori-teori yang beruhubungan dengan topik penelitian.

BAB III ANALISIS DAN PERANCANGAN

Bab ini berisi analisis terhadap akord yang dibangkitkan dari not angka, analisis terhadap masukan not-not angka dengan *lexical analyzer* yang selanjutnya akan membentuk token-token, analisis terhadap kebenaran dan urutan

kemunculan token dengan *parser analyzer*, analisis pembangkitan akord dari not angka dengan model NFA dan algoritma levenstein, perancangan flowchart atau prosedural algoritmik, perancangan struktur data, perancangan perangkat lunak untuk membuat simulasi berdasarkan hasil analisis terhadap musik yang digambarkan dengan model UML (*Unified Modelling Language*) dan perancangan antarmuka (*interface*).

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi mengenai implementasi atau menerjemahkan hasil perancangan perangkat lunak berupa penulisan kode program dan pengujian terhadap aplikasi yang bertujuan untuk mengetahui seberapa baik implementasi model *Nondeterministic State Automata* (NFA) dan algoritma *levenshtein* untuk mengefisienkan penggunaan memori dalam membangkitkan komposisi akord bar sebuah lagu.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan berupa hasil penelitian yang dilakukan dan saran yang diharapkan mengenai implementasi algoritma pembangkitan akord agar dapat bermanfaat untuk pengembangan selanjutnya.