

# VOICE RECOGNITION: PENGENALAN CHORD UKULELE 4 SENAR DENGAN MENGGUNAKAN METODE BACK PROPAGATION NEURAL NETWORK

Septian Andi Setiawan<sup>1</sup>, Yuri Ariyanto<sup>2</sup>

Teknik Informatika, Teknologi Informasi, Politeknik Negeri Malang

<sup>1</sup>[septianandi.s@gmail.com](mailto:septianandi.s@gmail.com) || <sup>2</sup>[yuri@polinema.ac.id](mailto:yuri@polinema.ac.id)

---

## Abstrak

Ukulele merupakan salah satu alat musik petik yang umum dimainkan oleh berbagai kalangan usia dari tua maupun muda. Salah satu jenis musik yaitu musik keroncong menggunakan ukulele sebagai instrumen utamanya. Untuk menghasilkan suara-suara yang harmonis dibutuhkan penguasaan *chord* atau kunci ukulele. Sebuah *chord* merupakan satuan nada-nada yang dibunyikan secara bersamaan yang berfungsi sebagai pengiring dalam sebuah lagu. Teknologi pengenalan suara (*voice recognition*) adalah teknologi yang dapat mengubah sinyal suara menjadi sebuah informasi berupa teks. Dengan menerapkan teknologi pengenalan suara (*voice recognition*) dapat dijadikan sebagai solusi dari masalah tersebut, dengan cara membuat sebuah penelitian yang bertujuan untuk mengidentifikasi *chord-chord* pada ukulele dengan tingkat akurasi yang tinggi. Oleh karena itu penulis membuat sistem "*Voice Recognition : Pengenalan Chord ukulele (4 senar) dengan menggunakan metode Back Propagation Neural Network*". Hasil yang diperoleh berupa informasi mengenai *chord* yang dimainkan. Berdasarkan pengujian 7 sampel yang dilakukan menunjukkan bahwa aplikasi menghasilkan akurasi secara maksimal.

**Kata kunci :** *Voice recognition, Chord, Ukulele, Jaringan Syaraf Tiruan, Backpropagation*

---

## 1. Pendahuluan

Ukulele merupakan salah satu alat musik petik yang umum dimainkan oleh berbagai kalangan usia dari tua maupun muda. Salah satu jenis musik yaitu musik keroncong menggunakan ukulele sebagai instrumen utamanya. Untuk menghasilkan suara-suara yang harmonis dibutuhkan penguasaan *chord* atau kunci ukulele. Sebuah *chord* merupakan satuan nada-nada yang dibunyikan secara bersamaan yang berfungsi sebagai pengiring dalam sebuah lagu. Teknologi pengenalan suara (*voice recognition*) adalah teknologi yang dapat mengubah sinyal suara menjadi sebuah informasi berupa teks.

Dengan menerapkan teknologi pengenalan suara (*voice recognition*) dapat dijadikan sebagai solusi dari masalah tersebut, dengan cara membuat sebuah penelitian yang bertujuan untuk mengidentifikasi *chord-chord* pada ukulele dengan tingkat akurasi yang tinggi.

Tujuan dari penelitian ini adalah untuk memperkenalkan *chord-chord* yang ada pada gitar ukulele serta untuk mengetahui berapa frekuensi suara pada setiap *chord*-nya. Sehingga pengguna yang belum pernah bermain gitar ukulele akan mengetahui *chord-chord* yang umum digunakan dan pengguna bisa mengetahui *chord* yang dimainkan salah atau tidak.

Permasalahannya adalah Bagaimana cara mengenalkan *chord* ukulele pada orang awam serta

Bagaimana cara mengidentifikasi *chord-chord* ukulele dengan tingkat akurasi tinggi.

## 2. Voice Recognition

*Voice recognition* merupakan proses pengenalan secara otomatis suatu sinyal suara dengan membandingkan pola karakteristiknya dengan sinyal suara yang menjadi referensi atau acuan. Fungsi dari *voice recognition* adalah dapat mengidentifikasi seseorang melalui suaranya. *Voice Recognition* dibagi menjadi 2 bagian antara lain *speech recognition* dan *speaker recognition*. Jika *voice recognition* dapat mengidentifikasi seseorang melalui suaranya maka *speech recognition* dapat mengidentifikasi apa yang diucapkan oleh seseorang. Sedangkan *speaker recognition* adalah sistem pengenalan identitas yang diklaim oleh seseorang dari suaranya atau berdasarkan orang yang berbicara.

## 3. Android

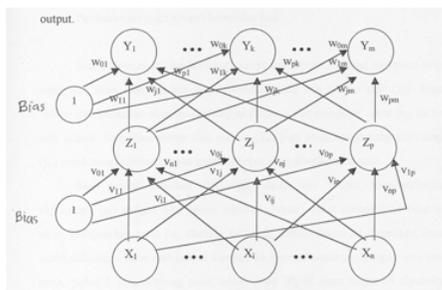
*Android* adalah sebuah sistem operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. *Android* merupakan generasi baru *platform mobile* yang memberikan kesempatan kepada pengembang untuk melakukan pengembangan sesuai dengan yang diharapkan.

Sistem operasi yang mendasari *Android* merupakan lisensi di bawah naungan GNU, *General Public License Versi 2 (GPLv2)*, yang biasa dikenal dengan istilah *copyleft*. Istilah *copyleft* ini merupakan lisensi yang setiap perbaikan oleh pihak ketiga harus terus jatuh di bawah terms.

Distribusi *Android* berada di bawah lisensi *ApacheSoftware* (ASL/Apache2), yang memungkinkan untuk distribusi kedua atau seterusnya. Pengembang aplikasi *Android* diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apapun yang mereka inginkan. Pengembang memiliki beberapa pilihan dalam membuat aplikasi yang berbasis *Android*. Namun kebanyakan pengembang menggunakan *Eclipse* sebagai IDE untuk merancang aplikasi mereka. Hal ini dikarenakan *Eclipse* mendapat dukungan langsung dari *Google* untuk menjadi IDE pengembangan aplikasi *Android*.

**4. Backpropagation Neural Network**

Algoritma Backpropagation meliputi 3 (tiga) fase. Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran. Fase 18 ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi.



Gambar 1. Jaringan multilayer pada backpropagation

Berikut adalah penjelasan Algoritma backpropagation (Fauset, 1993:294; (Widodo, 2013)):

Langkah 0 :Inisialisasi bobot dari  $X_i$  ke  $Z_j$  dan dari  $Z_j$  ke  $Y_k$  dengan nilai acak serendah mungkin, set harga error minimal.

Langkah 1 :Selama kondisi stop belum dilalui maka lakukan langkah 2-8.

Langkah 2 :Untuk setiap pasang data masukan lakukan langkah 3-8.

**4.1 Umpan Maju (feed forward)**

Langkah 3 :Setiap unit sel masukan ( $X_i, i=1, \dots, n$ ) menerima sinyal  $x_i$  dan menyebarkan sinyal ke seluruh lapis tersembunyi (hidden layer).

Langkah 4 :Setiap unit sel tersembunyi ( $Z_j, j=1, \dots, p$ ) menjumlahkan setiap sinyal masukan yang telah diberi bobot ( $v_{ij}$ ) dan bias ( $v_{oj}$ ), sinyal keluaran dihitung dengan fungsi aktivasi, dan mengirimkan hasilnya ke setiap unit sel di lapisan di atasnya (unit-unit keluaran).

Langkah 5 :Setiap unit sel keluaran ( $Y_k, k=1, \dots, m$ ) menjumlahkan setiap sinyal yang telah diberi bobot ( $w_{jk}$ ) dan bias ( $w_{ok}$ ), dengan fungsi aktivasi sinyal keluaran dihitung dengan  $Y_k = f(y_{in_k})$

**4.2 Perambatan balik nilai kesalahan (backpropagation of error).**

Langkah 6 :Setiap unit sel keluaran ( $Y_k, k=1, \dots, m$ ) menerima pola target ( $t_k$ ) yang sesuai dengan pola masukan pelatihan, hitung factor, hitung koreksi bobot, hitung koreksi bias, selanjutnya nilai digunakan di lapisan dibawah.

Langkah 7 :Setiap unit sel tersembunyi ( $Z_j, j=1, \dots, k$ ) menjumlahkan nilai  $\delta$  yang telah diberi bobot dari lapisan di atasnya,

**4.3 Menghitung bobot baru**

Langkah 8 :Setiap unit sel keluaran ( $Y_k, k=1, \dots, m$ ) memperbaharui bobot bias dan Setiap unit sel tersembunyi ( $Z_j, j=1, \dots, p$ ) memperbaharui bobot bias

Langkah 9 :Test kondisi stop. Kondisi stop yang digunakan adalah nilai Mean Squire Error < toleransi, maksimum iterasi 1500, nilai toleransi 0,001.

Dimana:

$x$  = masukkan (input)

$j = 1 \text{ s/d } n \text{ (} n = 10 \text{)}$

$v_{ij}$  = bobot pada lapisan tersembunyi

$w_{ij}$  = bobot pada lapisan keluaran

$n$  = jumlah unit pengolah pada lapisan tersembunyi

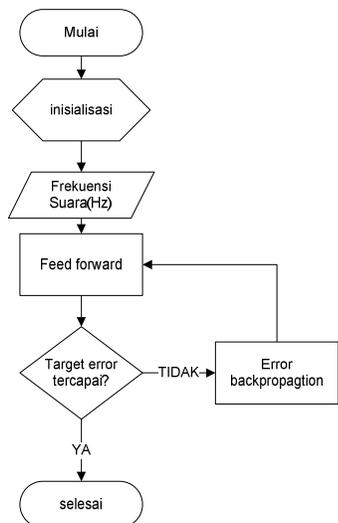
$b$  = bias pada lapisan tersembunyi dan lapisan keluaran

$k$  = jumlah unit pengolah pada lapisan keluaran

$Y$  = output

**5. Proses Alur dari Algoritma Backpropagation**

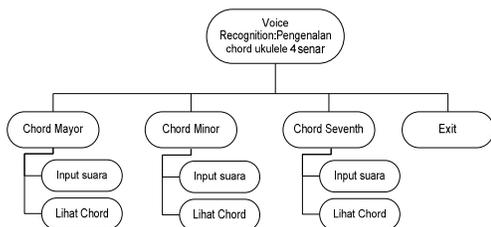
Gambar dibawah ini adalah alur dari sistem dengan menggunakan metode *Backpropagation*.



Gambar 2. Flowchart algoritma *backpropagation*

## 6. WBS(Work Breakdown Structure)

Berikut ini adalah perancangan sistem WBS sebagai gambaran dalam menjelaskan proses berjalannya sistem :



Gambar 3. WBS Voice Recognition

*WBS (Work Breakdown Structure)* merupakan gambaran struktur pembuatan system. WBS terdiri dari 3 (tiga) bagian yaitu chord Mayor, chord Minor, chord seventh.

## 7. Tampilan Antarmuka

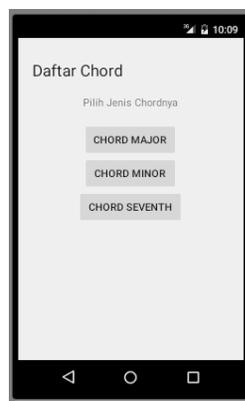
Berikut adalah contoh gambar dari aplikasi pengenalan chord ukulele:

### 7.1 Tampilan Utama



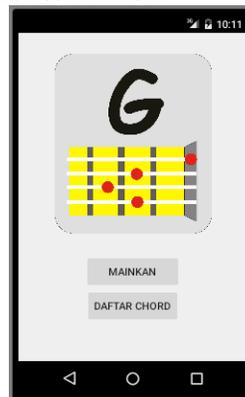
Gambar 4. Tampilan utama Aplikasi

### 7.2 Tampilan Pilih Chord



Gambar 5. Tampilan pilih Chord

### 7.3 Tampilan cek soundchord



Gambar 6. Tampilan Chord Gmajor

## 8. Kesimpulan

Aplikasi yang telah dihasilkan mampu menampilkan informasi mengenai berapa besar frekuensi pada tiap-tiap senar ukulele serta menampilkan *chord-chord* yang dimainkan oleh pengguna yang merupakan hasil dari gabungan senar-senar ukulele tersebut.

## 9. Saran

Penelitian ini masih dapat dikembangkan menjadi sistem yang lebih kompleks, tidak hanya terfokus pada pencatatan penjualan dan peramalan penjualan saja. Berikut merupakan beberapa hal yang disarankan untuk pengembangan aplikasi ini, yaitu:

- Dengan menambahkan beberapa tipe *chord* yang belum dimuat pada penelitian ini
- Untuk tampilan dari aplikasi ini bisa dibuat lebih menarik lagi.
- Kualitas Gitar ukulele yang digunakan untuk proses sampling data perlu ditingkatkan.
- Penulis berharap adanya penelitian yang sama namun berbeda pada metodenya.

#### Daftar Pustaka

- Gaffar, Imam, 2012. *Aplikasi pengkonversian nada-nada instrument tunggal menjadi chord menggunakan metode pitch class profile*. Jurusan Teknik Elektro, Universitas Diponegoro. Semarang  
<http://www.vibrasindo.com/blogvibrasi/detail/127/mengenal-fast-fourier-transform-fft> (di akses pada 26 januari 2015)
- Kadir, Abdul, 2013 “*from zero to a pro pemrograman Aplikasi Android*”. Yogyakarta; Andi Offset
- Khasanah, Amalia, 2013. *Perancangan aplikasi al qur'an menggunakan voice recognition sebagai media pencarian terjemahan al qur'an berbasis android*. Sekolah tinggi manajemen informatika dan computer amikom .Yogyakarta.