

PEMBUATAN TRANSKRIP AKORD INSTRUMEN TUNGGAL MENGGUNAKAN METODE *ENHANCED PITCH CLASS PROFILE*

Nabila Azzahra Syahbani¹, Miftahul Huda², Setiawardhana²

¹Mahasiswa ²Dosen

Jurusan Telekomunikasi – Politeknik Elektronika Negeri Surabaya

Institut Teknologi Sepuluh Nopember (ITS) Surabaya

Kampus PENS – ITS, Keputih, Sukolilo, Surabaya

Telp : +62+031+5947280, Fax +62+031+5946011

e-mail : nabila_budayana@yahoo.co.id

nabila@student.eepis-its.edu

Abstrak - Untuk dapat memainkan musik dengan baik, seorang pemusik tentu harus mengenali nama dan jenis akord pada piano. Namun dalam melakukan proses pembelajaran akord, dibutuhkan proses pengenalan dan penghapalan yang harus dilakukan. Hal tersebut kadang menjadi hambatan banyak orang dalam mempelajari musik, terutama bagi pemula.

Pada proyek akhir ini, dibuat perangkat lunak untuk menampilkan nama akord dari not yang dimainkan pada *keyboard*. Dimana dalam hal ini dilakukan dengan metode *EPCP (Enhanced Pitch Class Profile)* sebagai metode pengkelasan *pitch*. *EPCP* memungkinkan akord terdeteksi tanpa memperdulikan pada *oktaf* mana akord tersebut dimainkan. Selanjutnya akan dilakukan proses pemadanan dengan data yang telah dimasukkan pada *EPCP reference*.

Kata Kunci : *EPCP (Enhanced Pitch Class Profile)*, akord, *oktaf*

1. PENDAHULUAN

1.1 Latar Belakang

Saat ini musik telah menjadi suatu hal yang umum pada masyarakat. Sehingga, menarik minat banyak orang untuk mempelajarinya. Seiring dengan hal tersebut, dibutuhkan kemudahan-kemudahan dalam melakukannya, salah satunya adalah dengan mempermudah pembelajaran akord yang merupakan salah satu unsur penting dalam mempelajari musik.

Diketahui bahwa selama ini akord dikenali sebagai beberapa not yang dimainkan yang membentuk nada tertentu untuk mengiringi not tunggal. Jika dilakukan pengamatan, diketahui bahwa akord memiliki banyak jenis dan karakteristik pembentukannya dengan aturannya harus dipelajari dan dipahami agar dapat dimainkan dengan baik. Untuk itu, peminat musik biasanya masih harus menggantungkan pembelajaran pada seorang guru. Sehingga, terkadang hal tersebut menghambat para peminat untuk mempelajari musik.

Untuk mengatasi masalah tersebut diperlukan suatu perangkat lunak yang mempermudah pengguna dalam mempelajari jenis akord. Oleh karena itu, dirancang sebuah perangkat lunak yang dapat menampilkan nama

akord secara otomatis untuk memudahkan seseorang dalam mempelajari musik.

1.2 Permasalahan

Permasalahan yang dijadikan objek pengembangan proyek akhir ini yaitu :

1. Bagaimana membuat suatu perangkat lunak transkrip akord yang mudah digunakan
2. Bagaimana membuat keluaran akord dengan akurat (*error* kesalahan seminimal mungkin)
3. Bagaimana membuat algoritma yang tepat agar perangkat lunak dapat menghasilkan keluaran sesuai keinginan
4. Bagaimana menentukan metode *matching* yang tepat agar masukan dapat dideteksi berdasarkan database

2. PERENCANAAN DAN TEORI PENUNJANG

Dalam proyek akhir ini, dasar teori yang digunakan untuk mendukung perencanaan sistem yaitu, materi tentang akord, *EPCP* dan *Sum Square Error*.



Gambar 1 Rancangan Sistem

Awal mula pengerjaan dilakukan pengambilan data dengan memainkan akord (yang terdiri dari

beberapa not/bin) pada *keyboard* yang telah terhubung dengan *soundcard* laptop. Data yang diambil berupa file audio .wav. Setelah itu dilakukan pemrosesan sinyal pertama-tama dengan melakukan *powering* pada sinyal audio yang didapat. Dilanjutkan dengan melakukan *threshold* pada *peak* sinyal-sinyal yang dominan pada sinyal hasil *powering*. Hasil dari proses tersebut akan dilanjutkan dengan melakukan *frame blocking* untuk menentukan bagian sinyal yang akan dilanjutkan untuk diproses.

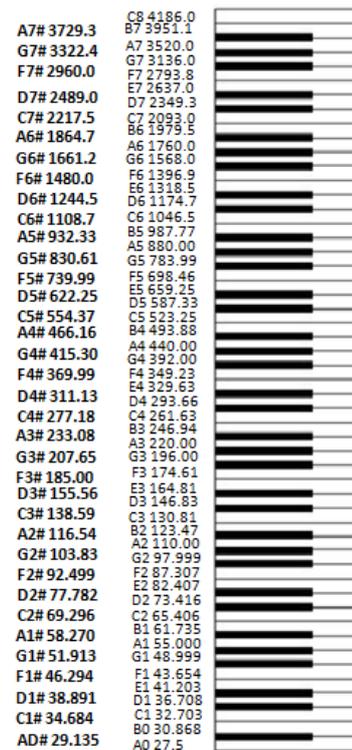
Setelah itu, dilakukan proses *windowing* untuk setiap *frame* sinyal dengan meng-nol-kan pada sinyal yang tidak diinginkan. Hal tersebut berguna untuk membentuk sinyal sebelum dilakukan FFT. Kemudian dilakukan perubahan domain sinyal dari domain waktu ke frekuensi dengan FFT. Memasuki proses EPCP, untuk mengurangi *error* pada pengambilan data, maka dilakukan rata-rata dari hasil PCP beberapa sinyal. Hal tersebut menyebabkan metode EPCP (*Enhanced Pitch Class Profile*) lebih unggul dari PCP (*Pitch Class Profile*). Hal ini akan menyebabkan secara langsung perubahan pada keluaran yang sinyal, sehingga didapatkan hasil PCP yang lebih akurat.

Proses dilanjutkan dengan melakukan *matching* data dengan metode *Sum Square Error*, dimana parameter yang digunakan adalah *error* minimum dari sinyal. Data tersebut dilakukan pemadanan dengan data yang terdapat pada EPCP Reference sebagai sistem penyimpanan data yang telah dilakukan penyimpanan 24 akord mayor dan minor di dalamnya.

Metode *matching* menggunakan metode *Sum Square Error*, dimana data *train* yang berupa nilai pengkelasan 24 akord mayor dan minor yang terdapat pada *database* dipadankan dengan data masukan berupa deretan akord (lagu) yang di-*test*. Metode pemadannya dengan mencari 3 *error* terkecil dari 12 *class* yang didapatkan dari masing-masing akord. *Error* didapatkan dengan membandingkan data *test* dan *train*. *Software Development* yang digunakan adalah Borland Delphi 7.0.

2.1 AKORD/CHORD

Akord merupakan gabungan dari beberapa not yang dimainkan secara bersamaan membentuk harmoni sebagai pengiring melodi yang dimainkan. Akord dapat dimainkan dengan iringan yang bervariasi sesuai dengan kesesuaian dengan jenis dan tempo dari lagu yang dimainkan. Jenis akord bermacam-macam, antara lain akord *diminished*, akord *augmented*, akord minor 6, akord mayor 7, akord *suspended* dan lain-lain. Masing-masing jenis akord tersebut memiliki karakteristik dan aturan pembentukannya sendiri.



Gambar 2 Frekuensi Pada Tiap Bin Piano

Setiap nada memiliki frekuensi masing-masing. Secara internasional, nada A4(A pada oktaf ke-4) memiliki frekuensi 440 Hz. Dan untuk mengetahui berapa frekuensi nada-nada lainnya dapat menggunakan persamaan :

$$f_n = f_0 * a^n$$

dimana:

f_n adalah frekuensi nada dengan jarak-n yang dicari

f_0 adalah frekuensi yang diketahui (A4 =440 Hz)

a^n adalah $(2)^{1/12}$

2.1.1 AKORD MAYOR

Akord Mayor biasanya dituliskan hanya berupa huruf kapital seperti C, D, E, F, G, A, B. Untuk mencari akord mayor maka anda dapat menggunakan nada ke 1 - 3 - 5 dari (C) *Mayor Scales*, contoh akord mayor:

- Cb (Cb-Eb-Gb) = B
- C (C-E-G)
- C# (C#-E#-G#) = Db (Db-F-Ab)
- D (D-F#-A)
- D# (D#-G-A#) = Eb (Eb-G-Bb)
- E (E-G#-B) = Fb (Fb-Ab-Cb)
- E# (E#-A-B#) = F (F-A-C)
- F (F-A-C)

- F# (F#-A#-C#) = Gb (Gb-Bb-Db)
- G (G-B-D)
- G# (G#-B#-D#) = As (Ab-C-Eb)
- A (A-C#-E)
- A# (A#-D-E#) = Bb (Bb-D-F)
- B (B-D#-F#) = Cb
- B# (B#-E-G) = C

Akord yang memiliki nama berbeda namun bila dimainkan bersuara sama disebut Akord Enharmonis

2.1.2 AKORD MINOR

Akord Minor biasanya dituliskan dengan penambahan karakter 'm' setelah huruf Kapital seperti Cm, Dm, Em, Fm, Gm, Am, Bm. Biasanya penulisan akord minor dilakukan dengan huruf kecil seperti c, d, e, f, g, a, b. Apabila pengguna sudah mengetahui suatu akord mayor misalnya; C mayor maka anda bisa mengetahui pula akord minornya (C minor) yaitu dengan cara menurunkan nada yang ada ditengah sebanyak setengah interval. Sehingga didapat akord C minor adalah C-Es(E diturunkan setengah menjadi Es)-G. Dapat juga dapat dicari dengan menggunakan nada ke 1 – 3b (nada ke 3 diturunkan setengah interfal) – 5 dari (C) *Mayor Scales*, sebagai contoh :Akord Cm = Nada C - D#/Eb – G.

Apabila ditemukan nada maupun akord seperti D#/Eb, hal tersebut berarti bunyi dari D# terdengar sama dengan bunyi Eb. Akord inilah yang disebut sebagai akord atau nada Enharmonis. Karena akord atau nada seperti ini memiliki nama berbeda namun bila dimainkan menghasilkan frekuensi dan suara yang sama

2.2 EPCP (ENHANCED PITCH CLASS PROFILE)

EPCP merupakan singkatan dari *Enhanced Pitch Class Profile*. Masing-masing nada dalam musik memiliki pitch. Metode EPCP merupakan perkembangan dari metode PCP (*Pitch Class Profile*). Pada metode ini, pitch dari nada-nada yang ada dikelompokkan berdasarkan 12 *class* dari *pitch*, yaitu C, C#, D, D#, E, F, F#, G, G#, A, A#, B. Yang membedakan metode EPCP dengan yang lain adalah dalam metode ini dilakukan proses rata-rata pada hasil PCP, sehingga EPCP menghasilkan *error* yang lebih kecil. Pada aplikasinya, dengan menggunakan metode EPCP untuk *Chord Recognition* maka pada oktaf mana pun akord dimainkan, tidak akan mempengaruhi keluaran dari perangkat lunak.

2.3 SUM SQUARE ERROR

Proses *matching* merupakan proses pemadanan data antara database dengan data masukan, agar dapat menghasilkan keluaran yang sesuai. Metode *matching*

yang digunakan adalah *Sum Square Error*. Metode ini mencari *error* minimum dari perbandingan data *test* dan *train*. *Error* terkecil akan dianggap sebagai hasil.

$$SSE(X) = \sum_{k=1}^m (X_k - c(X))^2 \quad \dots(1)$$

Keterangan:

SSE(X) = Jumlah error input ke-x

X_k = input PCP nada ke-k

c(X) = nilai PCP pada *database* ke-X

2.4 FRAME BLOCK

Frame Blocking adalah pembagian sinyal audio menjadi beberapa *frame* yang nantinya dapat memudahkan dalam perhitungan dan analisa sinyal, satu *frame* terdiri dari beberapa sampel tergantung tiap berapa detik suara akan disampel dan berapa besar frekuensi *samplingnya*. Pada proses ini dilakukan pemotongan sinyal dalam slot-slot tertentu agar memenuhi dua syarat yaitu linear dan *time invariant*.

2.5 THRESHOLDING

Pada proses ini sinyal input diproses dengan mengambil *peak-peak* dominan. Sebelumnya, sinyal akan mengalami *powering* terlebih dahulu dengan melakukan pengkuadratan pada sinyal, dengan persamaan

$$p = \sum_{i=0}^{i=N} \sqrt{X_i^2} \quad \dots(2)$$

Dimana:

P = *power* sinyal (dB)

X_i = *Frame* ke-i (dB)

N = Jumlah sampel per *frame*

I = Sampel ke-i

Powering bertujuan untuk mengetahui letak awal dan akhir dari suatu sinyal yang berada di suatu *frame*. Dari nilai *power* didapatkan suatu nilai rata-rata, dengan menambahkannya dengan standar deviasi, maka didapatkan nilai awal dan akhir dari suatu *frame*.

Standar deviasi merupakan penyimpangan data standar dari data yang ada. Jika nilai data mendekati nilai rata-rata, maka standar deviasinya lebih kecil (mendekati nol) dan sebaliknya. Ditunjukkan dengan persamaan:

$$\text{Standar Deviasi} = \sum_{i=0}^n \frac{x_i^2}{n} \quad \dots(3)$$

$$\text{Rata - rata} = \sum_{i=0}^n \frac{x_i}{n} \quad \dots(4)$$

Sinyal dengan nilai power diatas standart deviasi diambil sebagai awal dan akhir sinyal serta dianggap sebagai suatu voice.

2.6 WINDOWING

Dalam melakukan pemrosesan sinyal, maka dari input yang dimasukkan akan terbentuk sinyal yang magnitudenya bervariasi pada awal maupun akhir *frame*. Hal tersebut dapat menghambat pemrosesan sinyal dan menghasilkan keluaran yang kurang akurat. Untuk itu perlu diaplikasikan suatu *window* penghalus pada setiap *frame* dengan melakukan *overlapping* antara satu *frame* dengan *frame* yang lain, sehingga dapat dibangkitkan suatu *feature* yang lebih halus sepanjang durasi waktu tersebut. *Windowing* terdiri dari beberapa macam yaitu *Hamming*, *Hanning*, *Bartlet*, *Rectangular* dan *Blackman*. Dalam proyek ini akan digunakan Metode *Hamming*. Digunakan *Hamming window* karena *Hamming window* memiliki *side lobe* yang paling kecil dan *Main lobe* yang paling besar sehingga hasil *windowing* akan lebih halus dalam menghilangkan efek diskontinuitas dengan persamaan:

$$w(n) = 0.54 - 0.46 \cdot \cos\left(\frac{2 \cdot \pi \cdot n}{N-1}\right) \dots(5)$$

Dimana:

$w(n)$ adalah *windowing*

N merupakan jumlah data dari sinyal

n merupakan waktu diskrit ke

3. PEMBUATAN DAN PENGUJIAN

3.1 PEREKAMAN

Proses pengambilan data dilakukan dengan melakukan perekaman terhadap akord yang dimainkan dari *keyboard*. Setiap akord direkam sebanyak 5 kali. Dengan *sampling rate* sebesar 44100 Hz, *mono channel*. Perekaman dilakukan dengan menggunakan perangkat lunak Steinberg Nuendo.

3.2 PENGOLAHAN SINYAL

3.2.1 2 SAMPLING

Proses *sampling* merupakan proses dimana data wav diambil *sample* data untuk menentukan data mana yang akan diolah dalam proses selanjutnya. *Sampling Rate* yang digunakan adalah sebesar 44100 Hz, *mono channel*. Data yang sudah didapatkan disimpan dalam array.

3.2.1.3 THRESHOLDING dan FRAME BLOCKING

Pada proses ini dilakukan pengambilan *peak* dominan dari sinyal dengan terlebih dahulu melakukan *powering* dan penentuan standar deviasi yang berguna sebagai pendeteksi awal dan akhir sinyal dalam *frame*. Setelah itu, sinyal dibagi menjadi *frame-frame* untuk mempermudah pemrosesan sinyal. Pengambilan *sample* setiap *frame* diambil dalam waktu milisecond (ms). Digunakan waktu 62 ms sehingga nilai *sample* per *framenya* sebesar 2732.

3.2.1.4 WINDOWING dan FFT

Proses *windowing* bertujuan menghilangkan sinyal-sinyal yang tak dibutuhkan. Dengan adanya *windowing*, maka puncak-puncak dominan dapat lebih terlihat dan menghasilkan bentuk sinyal yang lebih halus, sehingga mempermudah proses FFT. Jenis *windowing* yang digunakan adalah *Hamming*, karena memiliki *main lobe* paling besar dan *side lobe* yang paling kecil, sehingga hasil *windowing* akan lebih halus dalam menghilangkan efek diskontinuitas

Proses FFT mengubah domain waktu sinyal ke domain frekuensi. Untuk mengaplikasikannya, cukup menggunakan komponen FFT dari DSP Lab. Nilai awal yang sudah didapatkan melalui proses *thresholding* menjadi parameter pada fungsi FFT. Dalam pemrosesannya, mulai dari titik awal sinyal akan diambil *sample* sebanyak *buffer size* dari FFT yang digunakan untuk dimasukkan ke dalam variabel array *RealIn* yang akan dihitung magnitudenya.

3.2.1.5 EPCP

Dari hasil FFT yang telah didapatkan dideteksi puncak-puncaknya. Parameter frekuensi dan amplitudo puncak sinyal menentukan pengkelasan dari *pitch* tersebut. Dengan didapatkan *range* frekuensi masing-masing *class* maka puncak dapat dikelaskan menurut besar frekuensinya. Dari grafik dapat diketahui jumlah puncak yang terdeteksi, frekuensinya serta nilai PCP dari masing-masing kelas. Masing-masing *frame* sinyal akan memiliki jumlah puncak, frekuensi dan nilai PCP yang berbeda. Oleh karena itu digunakan pengambilan data *mean* PCP (EPCP) sebagai rata-rata nilai PCP agar memiliki nilai yang lebih akurat. Dari hasil deteksi puncak, maka dapat dicari pengelompokan *range* frekuensi nada dengan menghitung jarak awal dan akhir setiap nada. Berikut penjelasan perhitungan :

Awal nada ke-i = (frekuensi nada ke-(i-1) + frekuensi nada ke - i)/2

Akhir nada ke-i = (frekuensi nada ke-i + frekuensi nada ke - (i+1))/2

Sebagai contoh untuk nada A :

$$\begin{aligned} \text{Awal} &= (\text{frek. nada G2\#} + \text{frek. nada A2})/2 \\ &= (103.83 + 110.00)/2 \\ &= 213.83/2 \\ &= 106.91 \end{aligned}$$

$$\begin{aligned} \text{Akhir} &= (\text{frek. nada A2} + \text{frek. nada A2\#})/2 \\ &= (110.00 + 116.54)/2 \\ &= 226.54/2 \\ &= 113.27 \end{aligned}$$

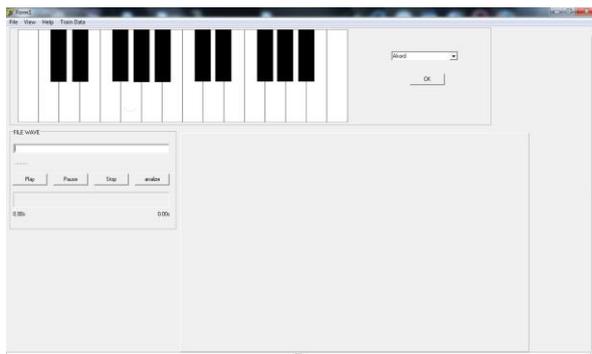
3.2.1.5 MATCHING

Proses *matching* menggunakan metode euclidean dimana dilakukan pengurangan untuk mendapatkan nilai *error* antara data *test* dan data *train (database)* dengan membandingkan index dari string grid *database* dengan index dari *string grid train* data. Setelah itu nilai dengan 3 *error* terkecil akan dianggap sebagai hasil dari pemadanan data. Hasil dari proses *matching* adalah baris akord yang mendeteksi data akord masukan. Berikut adalah flowchart dan *script* dan hasil dari *matching*

3.3 PENGUJIAN

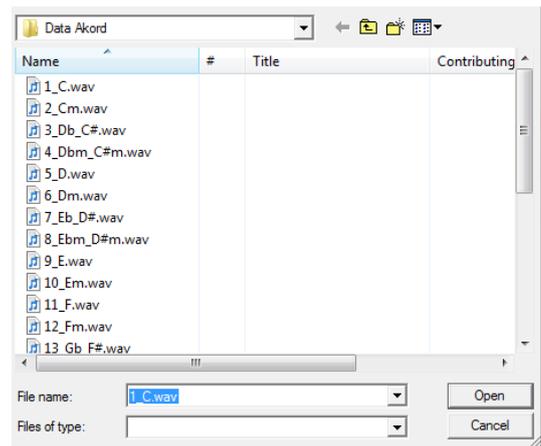
Berikut langkah-langkah dalam pengoperasian perangkat lunak.

1. Tampilan awal program



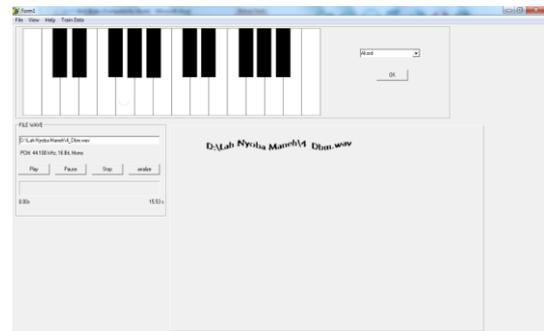
Gambar 3.18 Tampilan awal *interface*

2. Memasukkan file audio wav dengan memilih open file wave maka akan muncul tampilan seperti berikut :



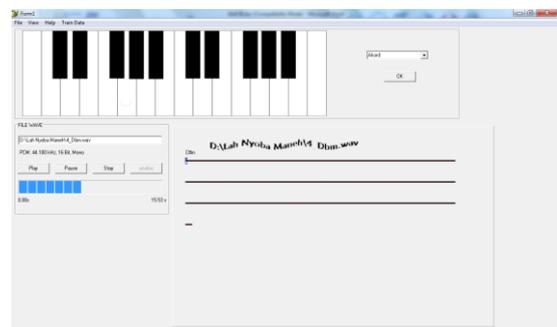
Gambar 3.19 Tampilan *Open Wave*

3. Tampilan File yang akan diproses.



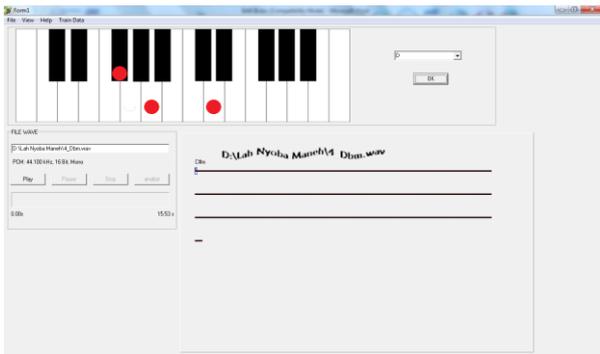
Gambar 3.20 Tampilan Proses Siap

4. Tampilan hasil baris akord dari file masukan



Gambar 3.21 Tampilan Hasil Akhir

5. Penggunaan *interface* akord piano



Gambar 3.22 Tampilan *Interface* Akord

4. PENUTUP

4.1 KESIMPULAN

Berdasarkan pada hasil pengujian sistem dan analisa hasil, didapatkan kesimpulan , yaitu:

1. Perangkat lunak ini memudahkan pengguna untuk mencari jenis akord apa saja yang dimainkan.
2. Pendeteksian akord dikhususkan hanya pada akord yang dimainkan dengan instrument tunggal.
3. Dari hasil pencocokan, diketahui bahwa akord yang terdeteksi sesuai dengan akord yang dimainkan.
4. Perangkat lunak bersifat praktis karena menggunakan format txt sebagai penyimpanan data, sehingga pengguna langsung dapat mengoperasikan perangkat lunak tanpa harus menginstall apapun.

5. REFERENSI

- [1] Kyogu Lee, "Automatic Chord Recogniton from Audio using Enhanced Pitch Class Profile", Sunnyvale, 2001
- [2] Febrianzah Junaidy Permana, "Pembuatan Database Software Digital Musik Mentor", PENS-ITS, Surabaya, Agustus 2009
- [3] Fandy Akbar, "Pembuatan Software Digital Musik Mentor", PENS-ITS, Surabaya, Agustus 2009
- [4] <http://journal.uii.ac.id/index.php/Snati/article/viewFile/1287/1097>
- [5] Ir. Rinaldi Munir, M.T. "Bahan Kuliah ke-3 IF5054 Kriptografi Teori Bilangan (Number Theory) Departemen Teknik Informatika Institut Teknologi Bandung, 2004
- [6] Thiang, "Implementasi Sistem Pengenalan Kata pada Mikrokontroler Keluarga MCS51" Jurusan Teknik Elektro, Universitas Kristen Petra

- [7] Agustina trifena Dame Saragih, Achmad Rizal, Rita Magdalena, "Penentuan akor gitar dengan menggunakan algoritma Short time Fourier transform" Departemen Teknik Elektro-Institut Teknologi Telkom
- [8] Steven W Smith, "The Scientist and Engineer's Guide to Digital Signal Processing", San Diego, California, 1997-1999

