

## BAB 2

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

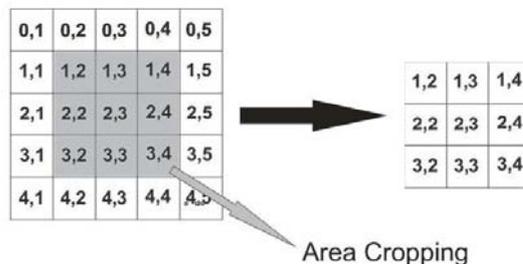
Pada tinjauan pustaka ini ada beberapa rangkuman jurnal dan skripsi yang berkaitan dengan skripsi ini. Seperti penelitian yang dilakukan oleh Adipranata, Sanjaya, dan Intan. (2010) membuat aplikasi tentang pembacaan not balok dengan metode OCR guna penyuaran piano dan violin. Hasil dari penelitiannya dinyatakan bahwa akurasi pembacaan not balok dengan metode OCR sebesar 75-80%

Untuk dapat membaca pola dari masing-masing not balok, penulis menggunakan algoritma *region* seperti Linda. (2005), ROI (*Region of Interest*) lebih sesuai jika dinyatakan sebagai upaya penurunan kualitas citra yang berada di luar area yang dipilih, sehingga berdasarkan pengamatan secara visual, area yang dipilih akan mempunyai kualitas citra yang lebih baik..

#### 2.2 Landasan Teori

##### 2.2.1 Cropping

*Cropping* merupakan proses pemotongan citra pada koordinat area tertentu dalam sebuah citra. Ada dua koordinat yang digunakan untuk melakukan pemotongan suatu citra, yaitu koordinat awal yang merupakan awal koordinat citra dan koordinat akhir yang merupakan titik koordinat akhir dari citra. Sehingga akan membentuk bangun segi empat dimana setiap *pixel* yang ada pada area koordinat tertentu akan disimpan dalam citra yang baru (Arie, 2013)



**Gambar 2.1** Proses *Cropping*

### 2.2.2 Digitizing

Menurut Gonzales dan Woods (2002) Digitizer atau *digital image acquisition system* merupakan sistem penangkap citra digital yang melakukan *scanning* citra dan mengkonversinya ke dalam bentuk numerik sebagai input bagi komputer. Hasil dari digitizer ini adalah berupa matriks yang elemen-elemennya menyatakan nilai intensitas cahaya pada suatu titik. Proses yang terjadi adalah representasi citra dari fungsi kontinu menjadi nilai-nilai diskret. Citra yang dihasilkan disebut digital image (citra digital). Sedangkan dimensi ukuran dinyatakan dengan tinggi (N) x lebar (M) atau (lebar x panjang).

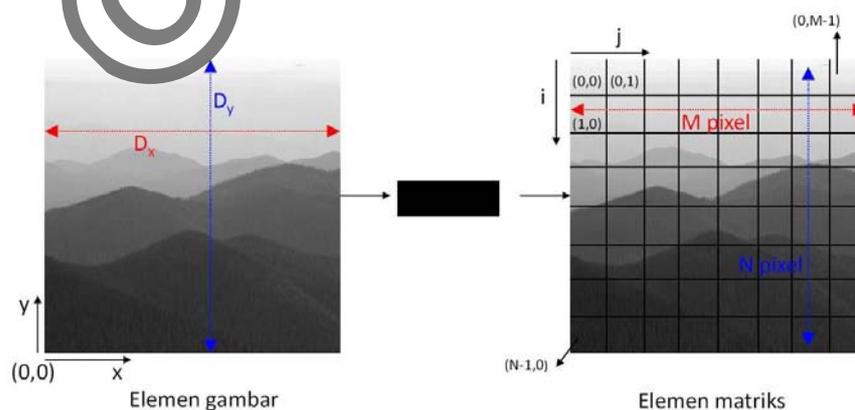
Citra digital dengan L derajat keabuan, fungsinya dapat ditulis sebagai berikut:

$$f(x,y) \begin{cases} 0 \leq x \leq M - 1 \\ 0 \leq y \leq N - 1 \\ 0 \leq f \leq L \end{cases}$$

Citra digital ukuran N (baris) x M (kolom) dinyatakan dengan matriks:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \dots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Citra kontinu di sampling pada element grid berbentuk bujur sangkar



**Gambar 2.2** Element Citra

$x$  :  $Dx / M$  increment

$y$  :  $Dy / M$  increment

$N$  : Jumlah maksimum pixel dalam satu kolom

$M$  : Jumlah maksimum pixel dalam satu baris

$Dx$  : Lebar gambar (dalam satuan panjang, misalnya inci)

$Dy$  : Tinggi gambar (dalam satuan panjang, misalnya inci)

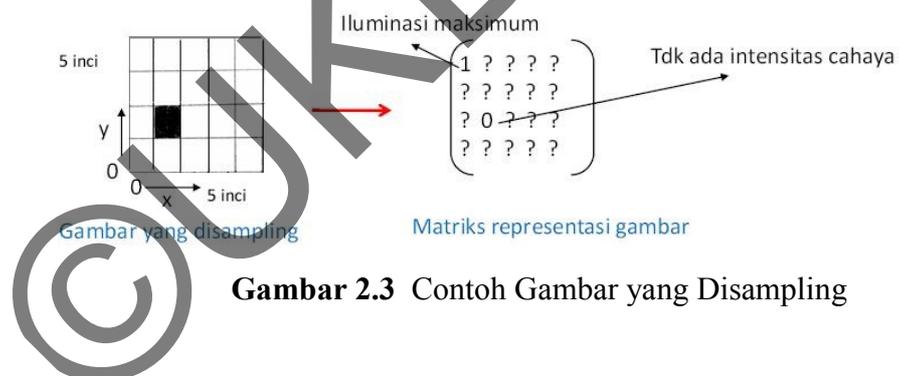
Elemen  $(i, j)$  dalam matriks menyatakan rata-rata intensitas cahaya pada area citra yang direpresentasikan oleh pixel

Contoh:

Suatu gambar yang ukurannya 5x5 inci akan dinyatakan dalam matriks ukuran 4x5 (4 baris dan 5 kolom), sebagai sebuah citra biner yang hanya mempunyai dua derajat keabuan saja:

0 (gelap)

1 (terang)



**Gambar 2.3** Contoh Gambar yang Disampling

Tiap elemen gambar lebarnya 1 inci dan tingginya 1,25 inci akan direpresentasikan dengan suatu nilai yang mewakili rata-rata intensitas cahaya pada area tersebut

Area 1x1,25 inci pada sudut kiri atas gambar direpresentasikan oleh (0,0)

Area 1x1,25 inci pada sudut kanan bawah gambar direpresentasikan oleh (3,4)

### 2.2.3 Region

Linda (2005), teknik *region-base* merupakan pengelompokan citra kedalam *region* tertentu secara langsung berdasar persamaan karakteristik suatu area citranya. Sehingga hanya area yang telah ditentukan saja yang akan diolah, sedangkan area sisa citra yang tidak diolah akan mengalami penurunan kualitas.

### 2.2.4 Thresholding

Menurut Gonzales dan Woods (2008), *Thresholding* adalah proses mengubah nilai intensitas suatu piksel dengan nilai tertentu, berdasarkan ambang yang telah ditetapkan.

$$f_B(x, y) = \begin{cases} a_1, f_A(x, y) < T \\ a_2, f_A(x, y) \geq T \end{cases}$$

atau

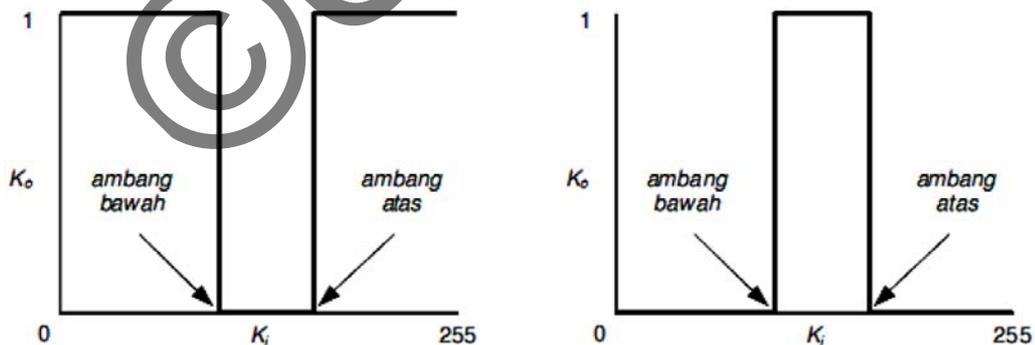
$$f_B(x, y) = \begin{cases} a_1, f_A(x, y) > T \\ a_2, f_A(x, y) \leq T \end{cases}$$

$a_1, a_2$  : Nilai intensitas baru

$f_A$  : Citra input

$f_B$  : Citra output

$T$  : Nilai ambang (*threshold*)

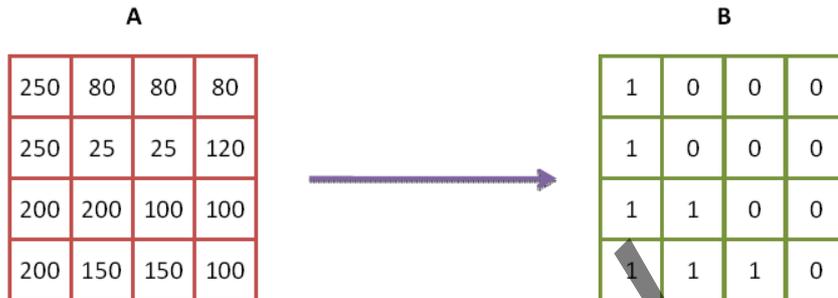


Gambar 2.4 Gambar *Threshold*

### Contoh thresholding

Didapat hasil biner dari suatu citra A dengan *thresholding* yang ditentukan sebagai berikut sehingga didapat hasil citra B

$$f_B(x, y) = \begin{cases} 0, & f_A(x, y) > 128 \\ 1, & f_A(x, y) \leq 128 \end{cases}$$



**Gambar 2.5** Contoh *Thresholding*

### 2.2.5 Notasi Balok

Ng (2003), Notasi balok merupakan standar yang digunakan dalam penulisan notasi musik. Setiap nada mempunyai frekuensi yang berbeda, sehingga penempatan posisi not pada garis paranada dilakukan berdasarkan tinggi-rendahnya nada tersebut. Nada adalah bunyi yang dihasilkan dari alat music, yang mempunyai durasi, *pitch*, intensitas, dan warna. Sebuah not balok mewakili sebuah nada, bentuk not balok tersebut menunjukkan hitungan yang terdapat pada nada yang diwakilinya. Dalam penotasian musik, dikenal 2 kondisi yaitu not dan *rest*. Not digunakan untuk menunjukkan adanya nada tertentu, sedangkan *rest* digunakan untuk menunjukkan tidak adanya nada.

Tabel berikut menunjukkan bentuk dan nilai not balok yang merepresentasikan nada (not) dan yang merepresentasikan tanda diam (*rest*).

Not	Rest	Nama (Nilai)	Not	Rest	Nama (Nilai)
		Semibreve (4)			Minims (2)
		Crotchet (1)			Quaver (1/2)
		Semiquaver (1/4)			Demisemiquaver (1/8)

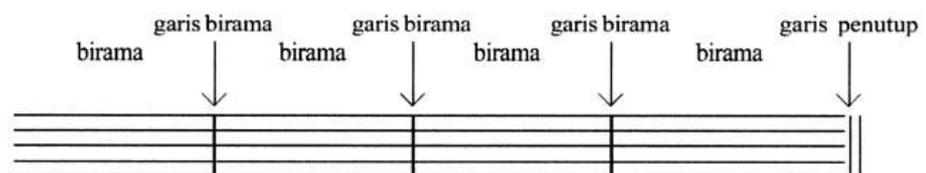
**Gambar 2.6** Bentuk dan Nilai Not Balok

Widodo (2007), Penulisan not balok juga memiliki tempat penulisan khusus yang terdiri dari lima garis. Lima garis tersebut dinamakan garis paranada. Not-not balok tersebut dapat ditulis di tengah garis paranada ataupun diantara garis paranada tersebut (biasa disebut spasi). Pembacaan garis paranada ini diurutkan mulai dari yang terbawah hingga yang teratas.



**Gambar 2.7** Gambar Garis Paranada

Birama merupakan tekanan yang tetap dan teratur, serta selalu berulang-ulang. Pada sebuah birama, terdapat bagian yang bertekanan dan bagian yang tidak bertekanan layaknya tanda baca koma dan titik dalam sebuah paragraph. Dalam penulisan musik, birama dipisahkan dengan garis tegak yang disebut garis birama.



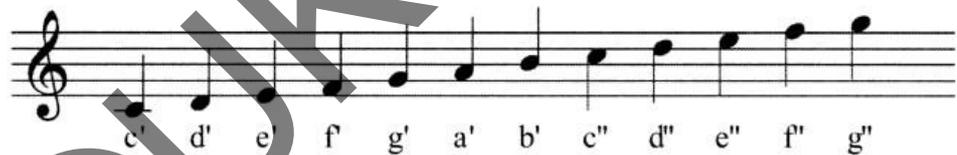
**Gambar 2.8** Gambar Birama dan Garis Birama

Ada beberapa aturan dalam penulisan not balok yang digunakan hingga saat ini. Salah satunya adalah semua not dibawah spasi paranada ke-2 menggunakan tiang yang mengarah ke atas dan berada di sebelah kanan not, sedangkan semua not diatas spasi paranada ke-3 menggunakan tiang yang mengarah ke bawah dan berada di sebelah kiri not. Pengecualian untuk not yang berada pada garis paranada ke-3, dapat menggunakan tiang ke atas maupun ke bawah tergantung dari posisi tiang not sebelumnya.



**Gambar 2.9** Contoh Penulisan Not Balok

Ada dua tanda kunci baku yang digunakan dalam penulisan notasi balok, yaitu kunci G dan kunci F. Kedua kunci ini memiliki bentuk dan posisi yang baku dalam garis paranada. Tanda kunci G menunjukkan nada g' pada garis kedua paranada. Berpangkal pada nada g', kita dapat membuat rangkaian nada sebagai berikut:



**Gambar 2.10** Contoh Nada Dalam Kunci G

## BAB 3

### RANCANGAN SISTEM

Untuk mendapatkan hasil penelitian yang optimal, dalam tugas akhir ini tentu saja memerlukan alat dan bahan yang dapat menunjang. Perancangan sistem merupakan tahap pengidentifikasian seluruh kebutuhan sistem yang akan dibangun. Kebutuhan sistem tersebut berupa data yang akan digunakan, diagram alur yang menunjukkan proses bekerjanya sistem, serta perancangan antarmuka bagi pengguna. Dalam tugas akhir ini menggunakan Teknik Pengolahan Citra Digital dan Komputer Vision dengan metode *OCR*, metode ini dipakai untuk mengenali pola citra not balok pada partitur lagu. Oleh karena itu dibutuhkan data berupa potongan partitur lagu yang dipisahkan (masing-masing 1 baris), untuk kemudian diproses dengan *cropping* kolom, *cropping* baris, yang kemudian akan dilihat berdasarkan *region* untuk mendapatkan posisi not tersebut sehingga didapatkan hasil output berupa angka posisi yang diolah menjadi barisan not angka.

#### 2.1 Spesifikasi Sistem

##### 3.1.1 Spesifikasi Perangkat Keras

Perangkat keras (*hardware*) yang digunakan untuk membangun sistem pada Tugas Akhir ini adalah sebagai berikut:

- *Processor* : Intel®Atom™ CPU @2,53GHz
- *RAM* : 2048 MB
- *Hard disk* : 320 GB
- *Mouse, keyboard, dan USBport*
- *Scanner*

### 3.1.2 Spesifikasi Perangkat Lunak

Perangkat lunak (*software*) yang digunakan untuk membangun sistem pada Tugas Akhir ini adalah sebagai berikut:

- Sistem Operasi : *Microsoft Windows 7 32-bit*
- *Tool* : *Matlab R2008b*

## 2.2 Perancangan Program

### Proses Pengolahan Citra Not Balok

Langkah 1 : Memasukan folder yang berisikan gambar 1 baris partitur not balok ke dalam sistem.

Untuk memasukan gambar diperlukan sebuah folder *load*. Untuk menampilkan folder digunakan fungsi `uigetfile('*.jpg')`. `Uigetfile` digunakan untuk menampilkan folder yang berisikan daftar file yang akan dipakai. `Uigetfile('*.jpg')` dimaksudkan file yang diambil adalah file berekstensi `.jpg`. Jika nama file tersebut valid dan jika file ada, `uigetfile` akan mengembalikan file ketika pengguna mengklik *Open*.

Langkah 2 : Membaca gambar not balok

Untuk membaca gambar digunakan fungsi `imread`. Melalui fungsi ini sistem dapat membaca gambar dari file yang ditentukan dengan memasukkan nama file.

Langkah 3 : Membagi bagian kolom not balok

Untuk memotong bagian-bagian not balok, digunakan fungsi `semgnetasi`. Melalui fungsi ini sistem dapat membagi gambar dari file yang ditentukan dengan menentukan ukuran bidang potong secara keseluruhan (1600 piksel x 130 piksel). Dilanjutkan dengan membagi panjang 1600 piksel menjadi 8 bagian (masing-masing 200 piksel x 130 piksel), kemudian

panjang citra tersebut dibagi kembali menjadi masing-masing 4 bagian yang terdiri dari 4 not (50 piksel x 130 piksel).

Langkah 4 : Membagi bagian baris not balok

Setelah dilakukan pembagian kolom sehingga diperoleh masing-masing 1 not, maka not tersebut akan kembali dibagi menjadi 12 bagian dengan proses *region* untuk menentukan posisi not tersebut dalam garis paranada.

Langkah 5 : Lakukan pengambilan nilai dari proses *region*

Setelah dilakukan pemotongan untuk membagi not menjadi 12 *regions*, akan diperoleh nilai yang menentukan posisi *region* dari not tersebut. Misalnya do pada *region* 12, re pada *region* 11, mi pada *region* 10 dan seterusnya.

Langkah 6 : Lakukan proses terjemahan posisi *region*

Setelah mendapatkan nilai untuk menentukan posisi *region* masing-masing not. Dilakukan proses penterjemahan posisi *region* ke dalam angka untuk menghasilkan not angka, seperti *region* 12 adalah do, maka akan dihasilkan output 1. *Region* 11 adalah re, maka akan dihasilkan output 2, dan seterusnya.

Langkah 7 : Tampilkan hasil output

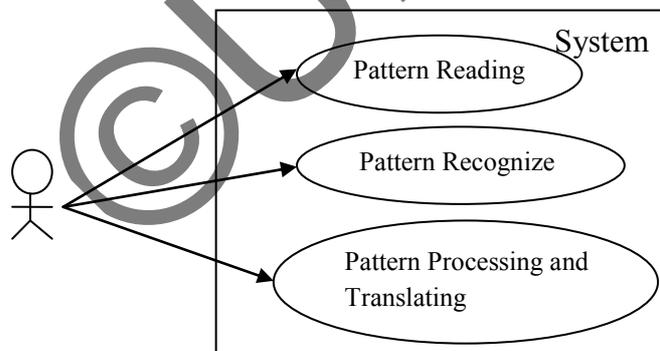
Setelah mendapatkan terjemahan dari langkah 6, tampilkan output hasil terjemahan itu kedalam kelompok-kelompok birama yang maksimum terdiri dari 4 not angka.

### 2.3 Analisis Data

Analisis data digunakan untuk mengetahui data-data apa saja yang dibutuhkan dalam membangun sistem. Data-data tersebut adalah sebagai berikut:

1. Data gambar partitur not balok per 1 baris. Data gambar yang ada nantinya dimasukan oleh pengguna. Kemudian data gambar tersebut akan di *crop* kolom kemudian baris untuk kemudian diproses secara *region*.
2. Data target untuk output, yang digunakan untuk mendapatkan hasil dari terjemahan proses *region*. Yaitu R12 untuk 1, R11 untuk 2, R10 untuk 3, dan seterusnya.
3. Data error, yaitu nilai tertentu yang digunakan sebagai error handling saat sistem tidak dapat membaca image tertentu.

*Use case* diagram digunakan untuk memodelkan dan menyatakan unit fungsi yang disediakan oleh sistem kepengguna. Dapat dikatakan bahwa *use case* menggambarkan kebutuhan sistem dari sudut pandang di luar sistem. *Use case* diagram dapat dilihat pada Gambar 3.1.



**Gambar 3.1** *Use Case Diagram*

### 3.4 Rancangan Proses

Sistem yang akan dibuat merupakan program berbasis *desktop*, sehingga hanya memanfaatkan sebuah komputer untuk menjalankannya. Rancangan

program ini menampilkan sebuah program aplikasi yang dapat mengenali pola not balok dan mengkonversinya menjadi not angka.

Sebelum sistem digunakan untuk pengenalan, terlebih dahulu user diminta untuk menginputkan sebuah partitur lagu yang ditulis dalam not balok dalam satu baris terpisah. Barulah kemudian gambar tersebut akan diolah oleh sistem untuk di *crop* kolom untuk diperoleh masing-masing satu not yang akan diproses berdasarkan *region* untuk menentukan not tersebut.

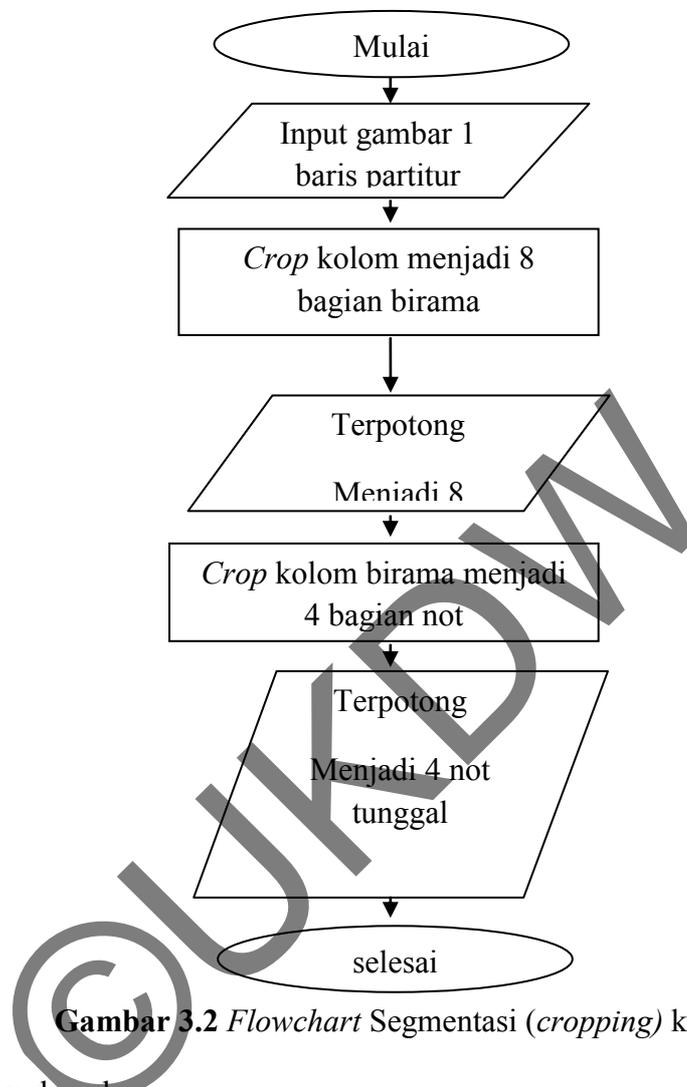
Gambar yang dimuat merupakan gambar dari foto satu baris sebuah partitur lagu atau sebuah gambar partitur lagu berformat JPG yang telah dipotong menjadi masing-masing satu baris tersendiri. Hasil potongan masing-masing satu baris partitur tersebut yang kemudian akan dimuat ke dalam komputer. Untuk proses pengenalannya, user memasukkan gambar potongan satu baris partitur tadi dengan *upload* nya melalui menu yang telah disediakan. Kemudian gambar tersebut akan diolah oleh sistem dengan pertama-tama akan memotong (*cropping*) kolom dalam satu baris partitur tersebut menjadi 8 kolom yang masing-masing terdiri dari 4 not ( $\frac{\text{Total panjang 1 baris}}{8}$ ), kemudian masing-masing 4 not tadi akan dipotong (*cropping*) kembali menjadi satuan not ( $\frac{\text{Total panjang 1 birama}}{4}$ ).

Sistem memiliki satu buah menu utama untuk memulai menjalankan sistem, dan menu konversi. Dalam menu utama hanya akan tersedia satu buah tombol untuk masuk dalam menu konversi. Saat masuk ke dalam menu konversi, akan tersedia pilihan untuk menginputkan gambar potongan satu baris partitur yang sudah disiapkan oleh user untuk kemudian diproses. Proses yang dilakukan akan dibagi menjadi tiga jenis proses:

1. Proses segmentasi (*cropping*) kolom

Pada proses *cropping* kolom, image berupa satu baris partitur yang sudah diinputkan user akan dipotong menjadi 8 birama yang masing-masing birama akan terdapat 4 not. Setelah terbagi menjadi 8 birama, masing-masing birama

tersebut akan dipotong kembali menjadi 4 bagian yang akan memisahkan masing-masing not tersebut menjadi 4 not tunggal.



**Gambar 3.2** Flowchart Segmentasi (*cropping*) kolom

Pseudocode:

Input gambar 1 baris = img;

img ← resize 1600px : 130px;

% segmentasi 8:

img = A ← segmentasi;

A = A1 lebar (1px : 200px);

```

A = A2 lebar(201px : 400px);

A = A3 lebar(401px : 600px);

A = A4 lebar(601px : 800px);

A = A5 lebar(801px : 1000px);

A = A6 lebar(1001px : 1200px);

A = A7 lebar(1201px : 1400px);

A = A8 lebar(1401px : 1600px);

print A;

A ← resize 200px : 130px;

% segmentasi 4:

B = Loop A1:A8 ← segmentasi;

B = B1 lebar(1px : 50px);

B = B2 lebar(51px : 100px);

B = B3 lebar(101px : 150px);

B = B4 lebar(151px : 200px);

Print B;

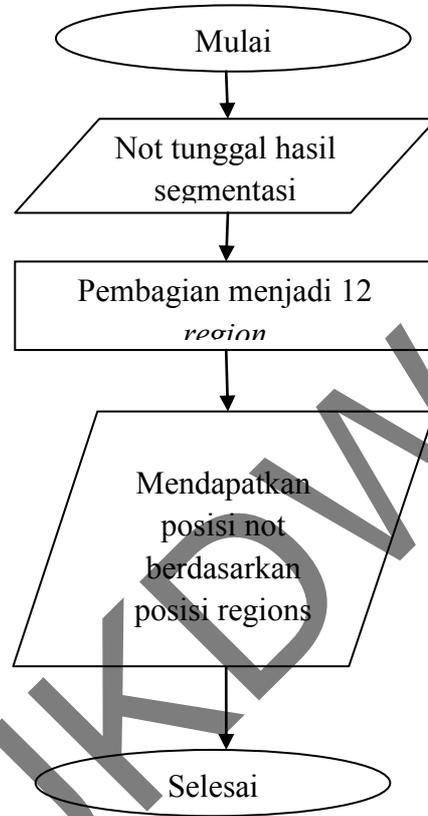
B ← resize 50px : 130px;

```

## 2. Proses *region*

Setelah didapat not tunggal dari proses *cropping* kolom, dilanjutkan dengan *region* baris pada setiap not tunggal. Pada proses *region* baris, image berupa satu not yang didapat akan dibagi menjadi 12 *regions* untuk mendapatkan posisi not

tersebut pada garis paranada. *Region* ganjil akan menentukan posisi not pada posisi diantara garis, sedangkan *region* genap akan menentukan posisi not pada posisi di dalam garis.



**Gambar 3.3** *Flowchart region*

Pseudocode:

B ← read;

% Region:

C = Loop B1:B4 ← region;

C = C1 panjang (1px : 20px);

C = C2 panjang (10px : 30px);

C = C3 panjang (20px : 40px);

C = C4 panjang (30px : 50px);

C = C5 panjang (40px : 60px);

C = C6 panjang (50px : 70px);

C = C7 panjang (60px : 80px);

C = C8 panjang (70px : 90px);

C = C9 panjang (80px : 100px);

C = C10 panjang (90px : 110px);

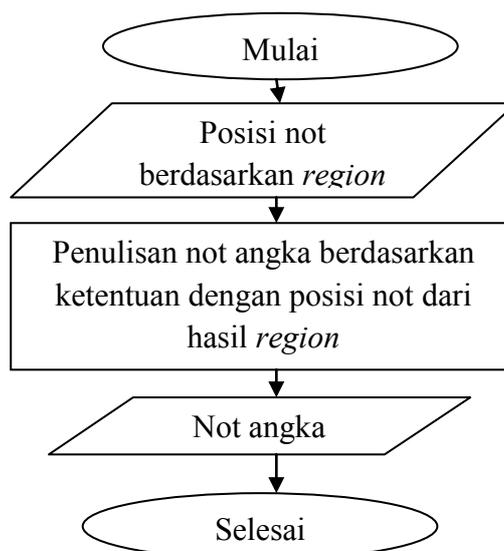
C = C11 panjang (100px : 120px);

C = C12 panjang (110px : 130px);

Print C;

### 3 Proses konversi ke not angka

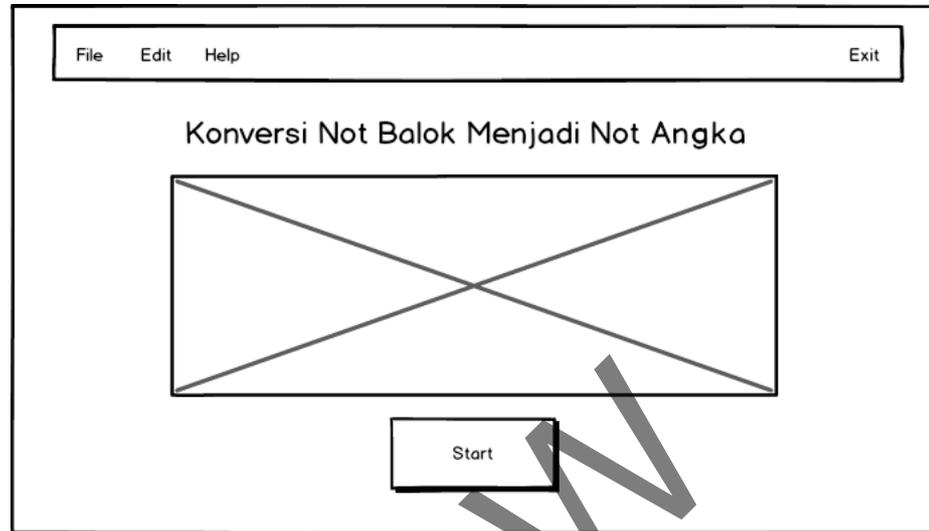
Setelah didapat posisi *region* masing-masing not balok tersebut, maka proses selanjutnya adalah menuliskan not angka berdasarkan posisi *region* yang didapat sesuai dengan ketentuan yang diatur (contohnya R12 = 1, R11 = 2, dan seterusnya)



**Gambar 3.4** Flowchart konversi not angka

### 3.5 Rancangan Antar Muka Sistem

#### 3.5.1 Perancangan Form Menu Utama



**Gambar 3.5** *Form Menu Utama*

Pada Gambar 3.5 merupakan tampilan form menu utama dari sistem yang terdiri dari menu-menu yang disediakan sistem dalam melakukan konversi not balok menjadi not angka. Menu bar terdiri dari empat menu:

1. File : Menu untuk mengakses file (seperti memulai program atau mengupload gambar, merupakan opsi tambahan disamping menggunakan tombol yang tersedia pada program)
2. Edit : Menu untuk mengganti file (seperti mengganti gambar yang di-*upload*, merupakan opsi tambahan disamping menggunakan tombol yang tersedia pada program)
3. Help : Menu untuk bantuan/ cara penggunaan sistem
4. Exit : Untuk mengakhiri program

### 3.5.2 Perancangan Form Proses Pembacaan

The diagram shows a software window with a menu bar (File, Edit, Help, Exit). The main area contains two large rectangular boxes with diagonal crosses, representing image input and output. A progress bar is located between these two boxes. To the right of the top box is a button labeled 'Browse Picture' and the text 'Insert 1 Line Partiture'. To the right of the progress bar is a button labeled 'Process Now'. Below the bottom box is the text '<-- Your Result'.

**Gambar 3.6** *Form Konversi*

Pada Gambar 3.6 merupakan form konversi yang terdiri dari dua output gambar. Pada gambar bagian atas akan menampilkan output dari gambar yang diinputkan user melalui button browse picture. Sedangkan pada gambar bagian bawah akan menampilkan output hasil yang didapatkan user setelah menekan tombol *Process Now*.

## BAB 4

### IMPLEMENTASI DAN ANALISIS SISTEM

#### 4.1 Implementasi Sistem

Pada bab ini akan dijelaskan tentang bentuk antar muka dari program, pembahasan tentang cara memasukan data / *input*, pembahasan tentang keluaran program atau *output* dan pembahasan tentang potongan dari program yang diimplementasikan kedalam metode OCR.

##### 4.1.1 Antar Muka Sistem

Menu utama seperti pada Gambar 4.1 adalah menu yang akan muncul pertama kali pada saat pengguna menjalankan aplikasi program ini. Halaman ini menampilkan menu-menu seperti menu *File* yang berisi sub menu “Mulai” untuk mulai menjalankan aplikasi, dan “Exit” untuk menutup program (Gambar 4.2), dan menu *Help* untuk bantuan penggunaan memulai aplikasi. Namun disediakan pula akses cepat tombol “Mulai” yang berfungsi sama dengan sub menu “Mulai” pada menu *File*.

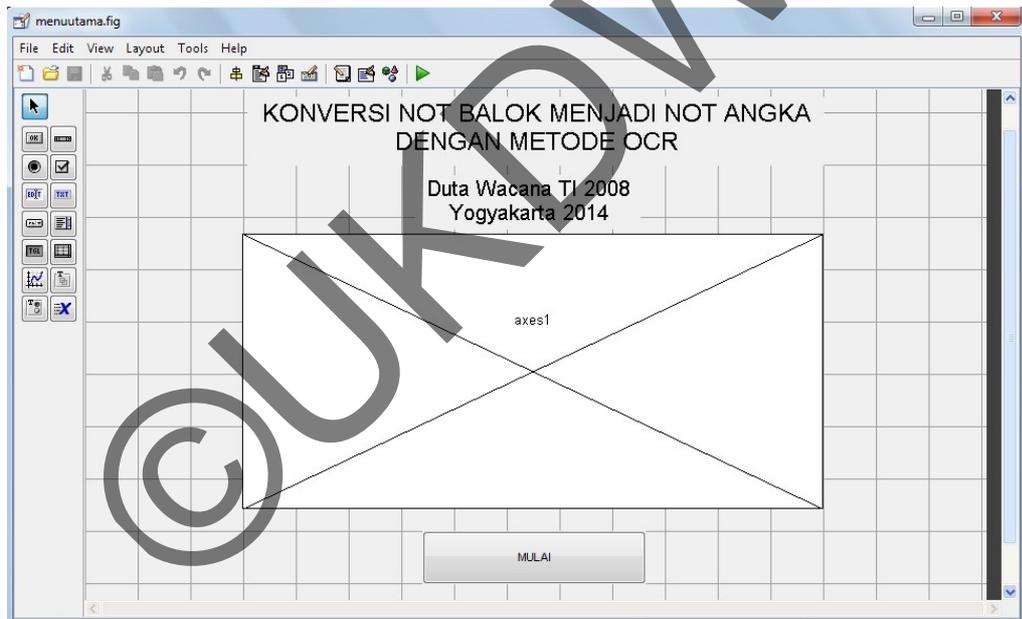


Gambar 4.1 Tampilan Antar Muka



**Gambar 4.2** Tampilan Sub Menu Antar Muka

Dengan cript program sebagai berikut:



**Gambar 4.3** Script Antar Muka

*Coding* untuk menentukan axes1 sebagai tempat untuk meletakkan gambar \*.jpg yang telah ditentukan, dimana handles.axes1 digunakan untuk menentukan axes1 sebagai tempat peletakan gambar, dan imshow untuk menampilkan gambar yang dipilih.

```

% --- Executes just before menuutama is made visible.
function menuutama_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   command line arguments to menuutama (see VARARGIN)

% Choose default command line output for menuutama
handles.output = hObject;
axes(handles.axes1);
imshow(imread('ukdw.jpg'));
grid off;
axis off;

% Update handles structure
guidata(hObject, handles);

```

**Gambar 4.4** *Coding* untuk Meletakkan Logo UKDW

*Coding* untuk menuju halaman konversi yang bisa diakses melalui button dan *menu bar*. figure(konversi) berfungsi sebagai fungsi yang mengarahkan *push button* dan *menu bar* tersebut ke halaman “konversi” saat dipilih.

```

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
figure(konversi)

% -----
function Untitled_2_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
figure(konversi)

```

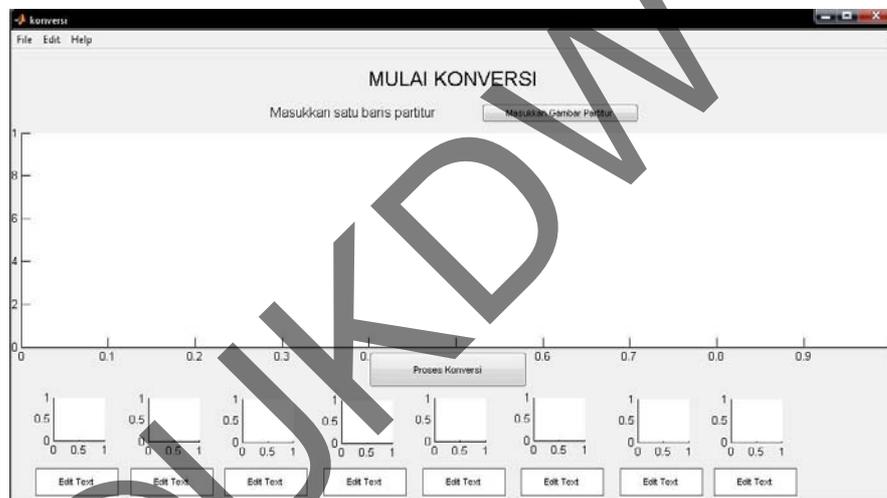
**Gambar 4.5** *Coding* untuk Menuju Halaman Konversi

#### 4.1.2 Menu Help

Menu Help, pengguna atau user bias melihat bagaimana cara penggunaan program aplikasi. Diharapkan dengan adanya menu Help ini dapat memudahkan pengguna dalam menggunakan program aplikasi.

#### 4.1.3 Halaman Konversi

Pada halaman konversi ini pengguna akan dapat memproses konversi dari gambar berupa satu baris partitur dalam not balok menjadi output berupa not angka, tampilan halaman konversi seperti pada Gambar 4.6.



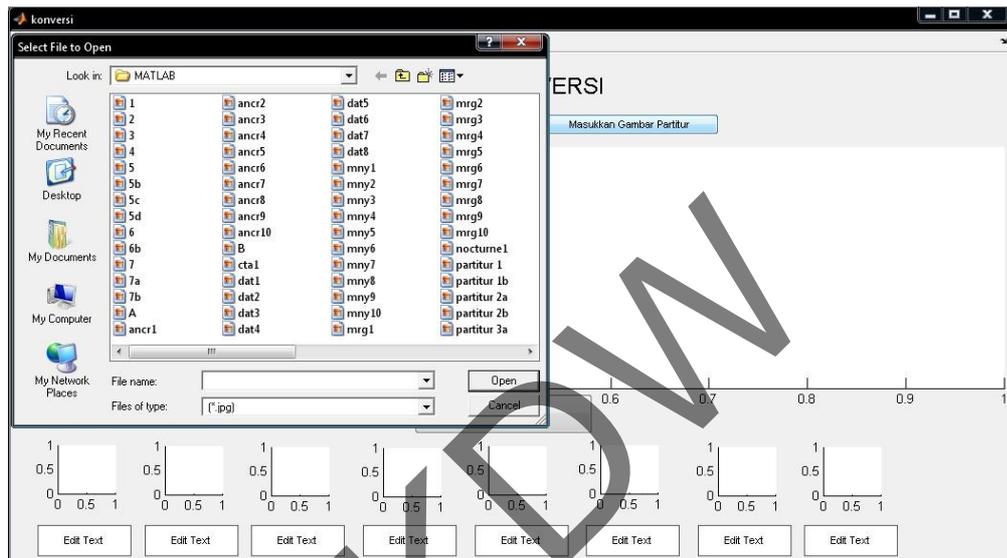
**Gambar 4.6** Tampilan Halaman Konversi

Untuk menginput gambar, pengguna dapat memilih tombol “Masukkan Gambar Partitur”, atau dapat juga dengan memilih menu *Edit* dan memilih submenu *Insert Partiture* seperti Gambar 4.7.



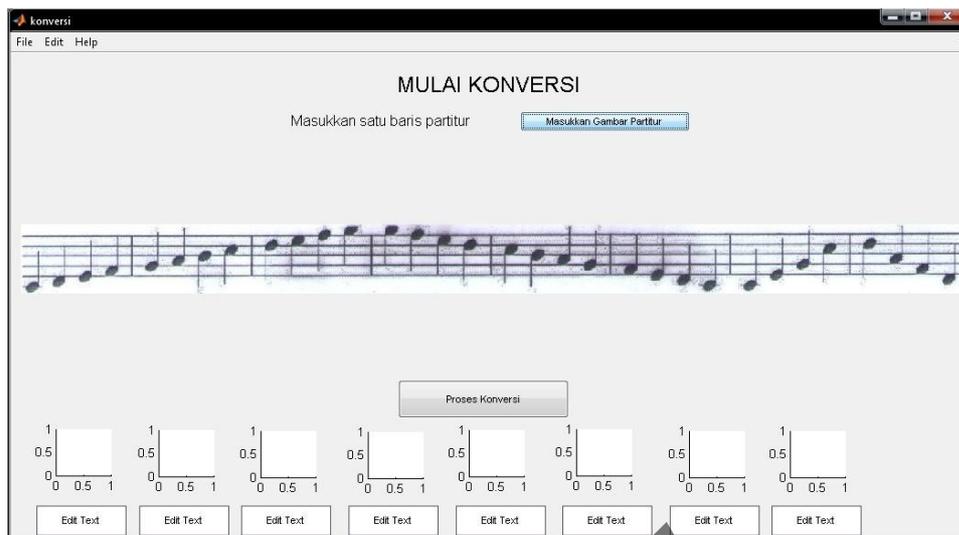
**Gambar 4.7** Tampilan Submenu Halaman Konversi

Setelah pengguna memilih metode penginputan gambar, akan muncul jendela untuk memilih gambar partitur yang hendak diinputkan. Pengguna dapat langsung memilih gambar yang dikehendaki kemudian memilih tombol *Open* seperti Gambar 4.8



**Gambar 4.8** Tampilan Penginputan Gambar

Kemudian gambar partitur tersebut akan ditampilkan pada halaman konversi untuk dilihat apakah sudah benar bahwa partitur tersebut yang hendak dikonversi oleh pengguna seperti pada Gambar 4.9



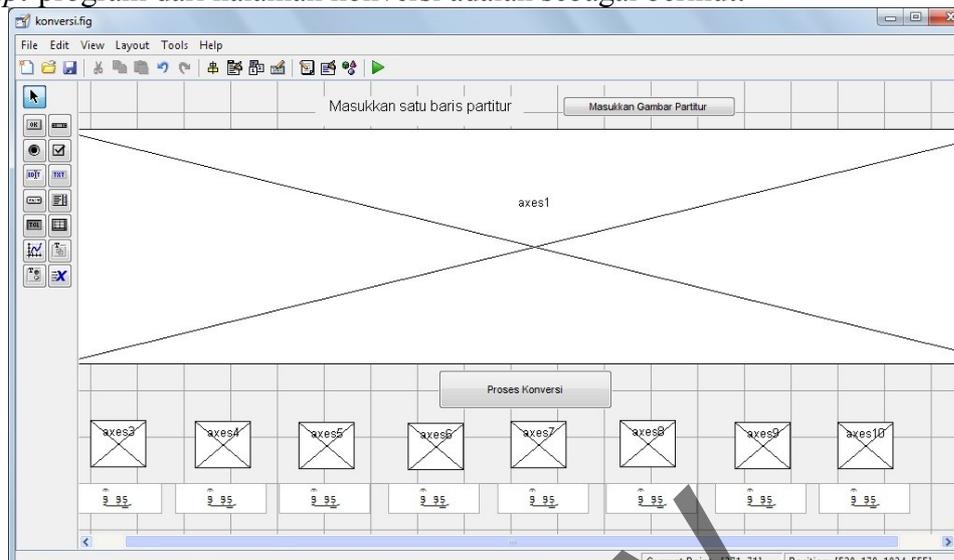
**Gambar 4.9** Tampilan Hasil Input

Jika gambar partitur sudah sesuai dengan keinginan pengguna, maka proses konversi dapat dilakukan dengan menekan tombol “Proses Konversi”, atau dengan menu *Edit* kemudian submenu *Conversion Processing* seperti pada Gambar 4.10.



**Gambar 4.10** Tampilan Proses Konversi

Script program dari halaman konversi adalah sebagai berikut:



**Gambar 4.11** Tampilan *Script* Halaman Konversi

Pada *button* “masukkan gambar partitur” disusun kode fungsi untuk menampilkan kotak direktori yang digunakan pengguna untuk memilih gambar \*.jpg dan akan ditampilkan pada kanvas atau bidang axes 1 sebagai tempat untuk menampilkan gambar. Uigetfile berfungsi untuk menampilkan kotak direktori yang hanya akan menampilkan file-file berformat jpg untuk dipilih pengguna. nmf akan menampung data berupa direktori gambar yang telah dipilih oleh pengguna dan ditampilkan di axes1 dengan fungsi handles.axes. kemudian I akan menampung file gambar yang telah terpilih dan ditampilkan pada axes.

```

% -----
function Untitled_5_Callback(hObject, eventdata, handles)
% hObject      handle to Untitled_5 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

global I
[n p]=uigetfile('*.jpg');
nmf=[p n];
axes(handles.axes1);
I=imread(nmf);
imshow(I);

```

**Gambar 4.12** Tampilan *Script* Tombol *Input* Gambar

Dilanjutkan dengan *button* “proses konversi” yang akan memproses keseluruhan konversi. Gambar yang telah ditampung dalam I tadi akan diubah menjadi citra keabuan (*gray*), kemudian baru menjadi black & white (BW) yang diolah oleh *rgb2gray* dan *im2bw*. Sehingga akan didapat *threshold* berupa nilai 1 untuk terang dan 0 untuk gelap. Proses dilanjutkan dengan mengganti ukuran dari citra dalam I tersebut menjadi berukuran 1600pixel x 130 pixel dengan *imresize*.

Terakhir, proses akan dilempar ke masing-masing proses konversi yang terbagi tiga, dan akan dibahas pada bagian proses konversi di bawah. Untuk menampilkan bagian segmentasi dari proses yang akan dilakukan pada bagian proses konversi tersebut, telah disediakan *axes3* hingga *axes10* yang berfungsi untuk menampilkan salah satu proses yang terjadi, yaitu segmentasi. Tampilan tersebut akan diproses oleh *handles.axes num2str(i+2)* karena dimulai dari *axes3*.

```
function Untitled_6_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%bagi region kolom menjadi 8 segmen
global I
I=rgb2gray(I);
I=im2bw(I);
I=imresize(I,[130,1600]);
% segmentasi 8
Is=delapan(I);
for i=1:size(Is,3)
    cmd=['axes(handles.axes' num2str(i+2) ')'];
    eval(cmd);
    cmd=['imshow(Is(:,:, num2str(i) ))'];
    eval(cmd);
end
```

**Gambar 4.13** Tampilan *Script* Tombol Konversi (1)

Proses selanjutnya adalah untuk menampilkan hasil konversi tersebut dalam bentuk *number* kedalam delapan buah *textbox* yang telah ditentukan. Maka perlu dibagi kedalam 4 buah bagian tersendiri yang didapat dari hasil proses konversi yang dilakukan terpisah pada bagian proses konversi. h1 sampai dengan h8 akan menentukan hasil yang akan ditampilkan pada delapan *textbox* tersebut.

```
tab=12:-1:1  
  
h1=[]  
for i=1:4  
    h1=[h1 num2str(tab(hasil(i)))];  
end  
h2=[]  
for i=5:8  
    h2=[h2 num2str(tab(hasil(i)))];  
end  
h3=[]  
for i=9:12  
    h3=[h3 num2str(tab(hasil(i)))];  
end  
h4=[]  
for i=13:16  
    h4=[h4 num2str(tab(hasil(i)))];  
end  
h5=[]  
for i=17:20  
    h5=[h5 num2str(tab(hasil(i)))];  
end  
h6=[]  
for i=21:24  
    h6=[h6 num2str(tab(hasil(i)))];  
end  
h7=[]  
for i=25:28  
    h7=[h7 num2str(tab(hasil(i)))];  
end  
h8=[]  
for i=29:32  
    h8=[h8 num2str(tab(hasil(i)))];  
end
```

**Gambar 4.14** Tampilan *Script* Tombol Konversi (2)

Terakhir adalah proses untuk menampilkan keseluruhan nilai yang ditampung pada h1 sampai h8 untuk ditampilkan pada delapan *textbox* yang tersedia. `Handles.edit num2str (i+11)` berfungsi untuk menampilkan nilai yang tertampung pada semua h tadi, dan dimulai dari i+11 karena *textbox* dimulai dari nama edit12 sampai dengan edit19.

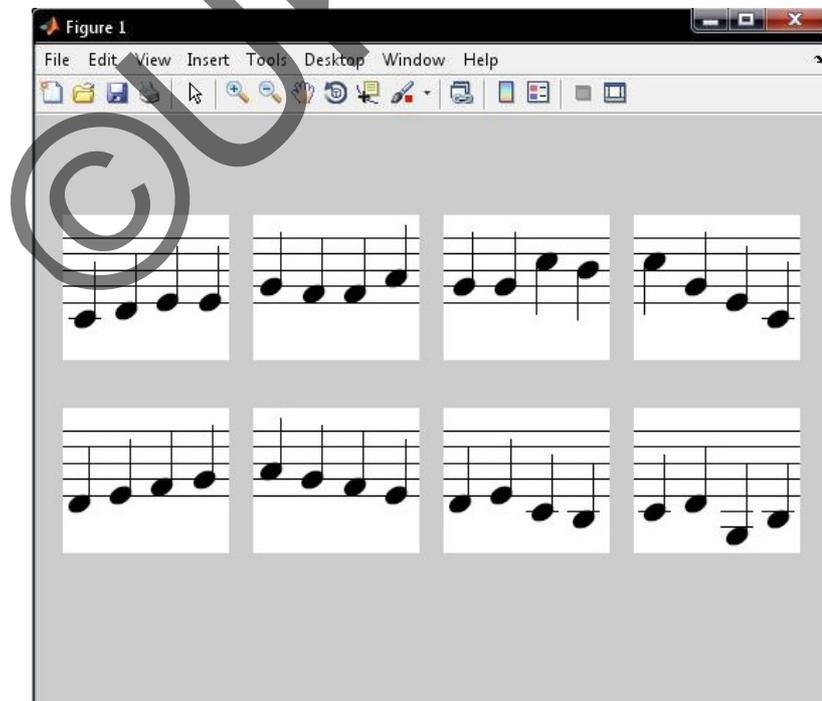
```
for i=1:8
    cmd=['set(handles.edit' num2str(i+11) ', ' char(39) 'string' char(39) ',h' num2str(i) ') ' '']
    eval(cmd);
end
```

**Gambar 4.15** Tampilan *Script Tombol Konversi* (3)

## 4.2 Proses Konversi

### 4.2.1 Proses Segmentasi (*Crop*) Kolom

Pada proses *crop* kolom, yang pertamakali dilakukan adalah membagi satu baris partitur menjadi delapan bagian birama yang masing-masing akan terdiri dari empat buah not seperti gambar 4.16.



**Gambar 4.16** Tampilan Segmentasi (*Crop*) Kolom (1)

Seperti dibahas sebelumnya, bahwa proses konversi akan dilakukan secara terpisah dan dipanggil dengan beberapa fungsi yang disiapkan, seperti `Is=delapan(I)`; akan memanggil proses segmentasi untuk melakukan *cropping* satu baris partitur lagu menjadi delapan bagian birama. Kemudian “Is” akan dibaca sebagai A untuk proses 1, dilanjutkan dengan memproses proses segmentasi kedua yang akan dibaca sebagai B. B= empat akan memanggil proses segmentasi untuk melakukan *cropping* satu birama menjadi 4 bagian agar menjadi masing-masing not tunggal. Terakhir adalah C=regio untuk memanggil proses region yang berfungsi untuk membaca posisi dari not yang didapat dalam garis paranada. Dari keseluruhan proses tersebut, akan diperoleh hasil yang akan ditampilkan kembali pada proses output pada *textbox* yang sudah dijelaskan pada bagian di atas.

```

% segmentasi 8
Is=delapan(I);
for i=1:size(Is,3)
    cmd=['axes(handles.axes' num2str(i+2) ' ');
    eval(cmd);
    cmd=['imshow(Is(:,:, ' num2str(i) ' ));
    eval(cmd);
end

A=(Is);

hasil=[];
for i=1:size(A,3)
    B=empat(A(:,:,i));
    for j=1:size(B,3)
        C=regio(B(:,:,j));
        hasil=[hasil C];
    end
end

hasilt=[''];

for i=1:length(hasil)
    hasilt=[hasilt num2str(hasil(i)) ' '];
end

```

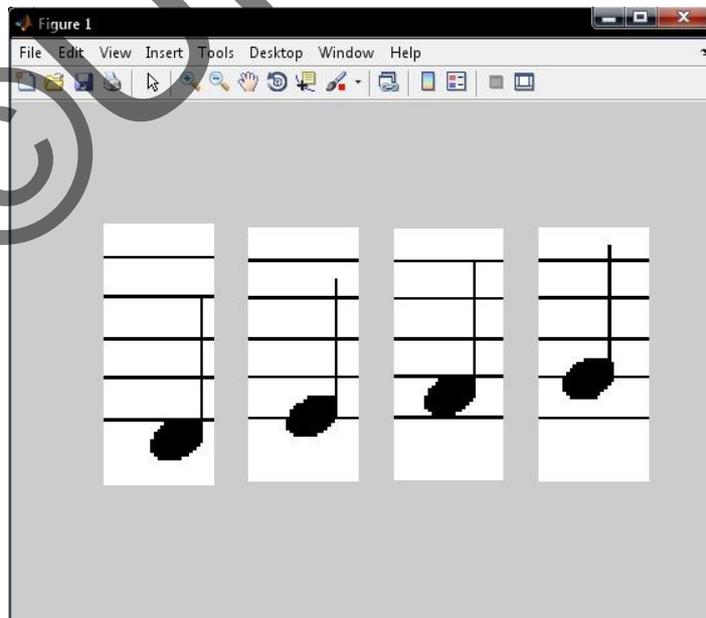
**Gambar 4.17** Tampilan *Script* Proses Konversi

Fungsi dari segmentasi “delapan” adalah membagi citra I yang sudah beresolusi 1600pixel x 130pixel menjadi delapan bagian potongan gambar tersendiri.

```
function A=delapan(I)
    A(:, :, 1)=I(:, 1:200);
    A(:, :, 2)=I(:, 201:400);
    A(:, :, 3)=I(:, 401:600);
    A(:, :, 4)=I(:, 601:800);
    A(:, :, 5)=I(:, 801:1000);
    A(:, :, 6)=I(:, 1001:1200);
    A(:, :, 7)=I(:, 1201:1400);
    A(:, :, 8)=I(:, 1401:1600);
```

**Gambar 4.18** Tampilan *Script* Segmentasi Kolom (1)

Selanjutnya setelah didapat masing-masing birama yang terdiri dari empat buah not, dilakukan proses *cropping* kolom kedua, yang membagi kembali empat buah not tersebut menjadi satu buah not untuk dibaca seperti gambar 4.19.



**Gambar 4.19** Tampilan Segmentasi (*Crop*) Kolom (2)

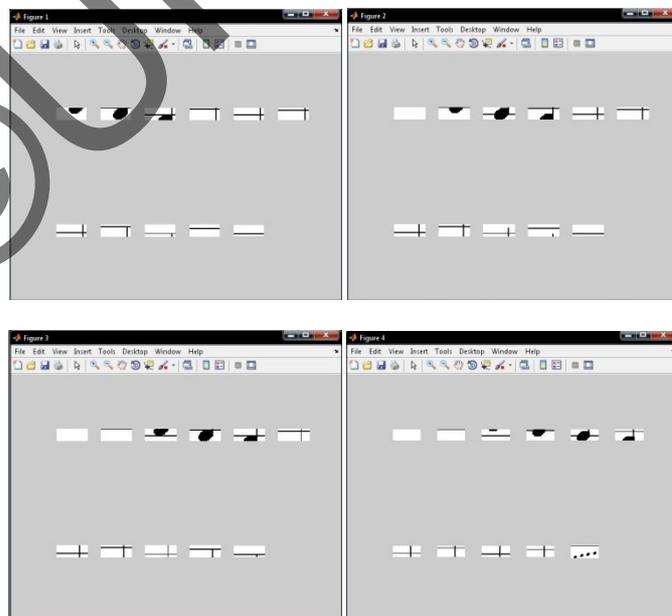
Fungsi dari segmentasi “empat” adalah membagi citra I yang sudah beresolusi 200pixel x 130pixel menjadi empat bagian potongan gambar tersendiri.

```
function B=empat(I);  
    I=imresize(I,[130,200]);  
    B(:,:,1)=I(:,1:50);  
    B(:,:,2)=I(:,51:100);  
    B(:,:,3)=I(:,101:150);  
    B(:,:,4)=I(:,151:200);
```

**Gambar 4.20** Tampilan *Script* Segmentasi Kolom (2)

#### 4.2.2 Proses *Region*

Setelah mendapatkan masing-masing not tunggal, kemudian dilanjutkan dengan proses *Region*, yaitu membagi baris dari not tersebut menjadi 11 *Regions* (Gambar 4.21) untuk mendapatkan posisi dari not-not tersebut sehingga dapat diterjemahkan kedalam not angka.



**Gambar 4.21** Pembagian *Regions*

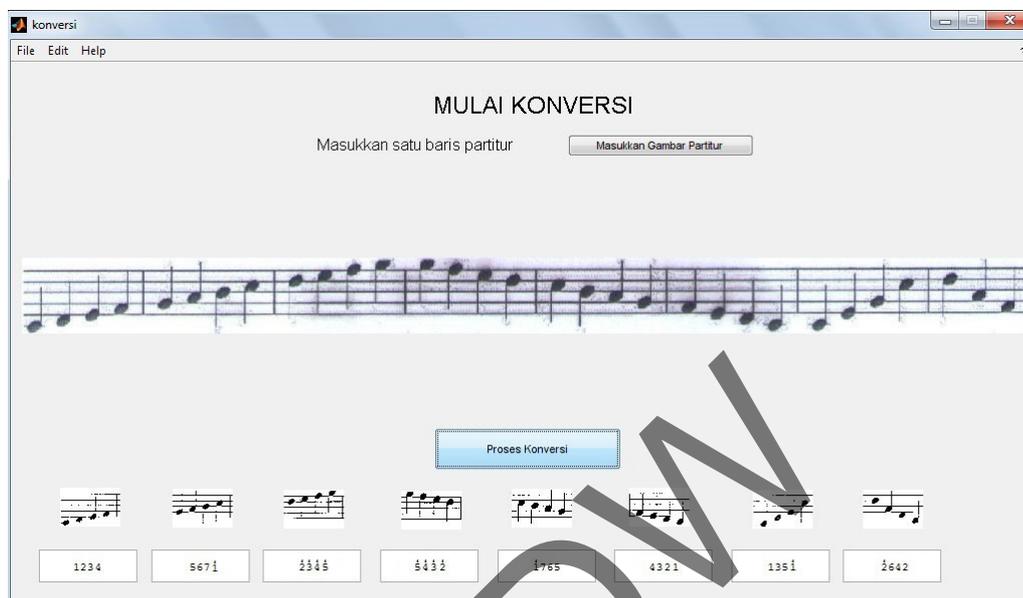
Fungsi dari region “regio” adalah membagi citra I yang sudah beresolusi 50pixel x 130pixel menjadi dua belas bagian region untuk mendapatkan posisi not dalam garis paranada pada citra.

```
function b=regio(I)
    I=imresize(I,[130,50]);
    for i=1:130
        if sum(sum(I(i,:)))==0
            I(i,:)=ones(1,50);
        end
    end
    C1=I( 1:20,:);
    C2=I(10:30,:);
    C3=I(20:40,:);
    C4=I(30:50,:);
    C5=I(40:60,:);
    C6=I(50:70,:);
    C7=I(60:80,:);
    C8=I(70:90,:);
    C9=I(80:100,:);
    C10=I(90:110,:);
    C11=I(100:120,:);
    C12=I(110:130,:);
    area=[];
    for i=1:12
        cmd=['area(' num2str(i) ') = sum(sum(C' num2str(i) '));'];
        eval(cmd);
    end
    [a b]=min(area);
```

**Gambar 4.22** Tampilan *Script* Pembagian *Regions*

### 4.2.3 Hasil Konversi

Hasil akhir berupa not angka dalam satu birama hasil konversi.



Gambar 4.23 Hasil Konversi

### 4.3 Hasil Analisis Sistem

Uji coba pembacaan citra pola not balok dilakukan dengan pengujian 30 citra partitur yang berbeda beserta tambahan efek distorsi dari masing-masing citra. Dari hasil uji coba dengan sistem ini, didapati proses pembacaan not balok dalam kondisi citra bersih (tanpa ada efek distorsi) dengan kondisi not *single* (dicoba untuk masing-masing not tersendiri, dalam *range* satu oktaf tangga nada C ditambah 4 nada tinggi re, mi, fa, sol) didapatkan hasil rata-rata keakuratan sebesar 83,34% ( $\frac{\text{Total Not Terbaca}}{\text{Total Not Yang Ada}} \times 100\%$ , terbaca 10 not dari total 12 not). Sedangkan dalam kondisi citra yang terdistorsi dengan kondisi not *single* bisa dilihat dari tabel perbandingan 4.1 berikut ini:

**Tabel 4.1**

Perbandingan Hasil Distorsi Masing-Masing Not (per satu not)

No.	Nama Distorsi	Hasil Deteksi (dalam %)
1	Blur	66,67
2	Noise	75
3	Film Grain	58,33
4	Terkena cairan (minyak)	41,66
5	Fragment	16,67
6	Mosaic	8,33
7	Terkena cairan (teh)	33,33
8	Terlipat / tertekuk / lusuh	58,33
9	Kemiringan	16,66
10	Wind	33,33
11	Dark Strokes	33,33
12	Smudge Stick	41,67
13	Tersobek dan ditambal dengan selotip	25
14	Photocopy	75

Sedangkan pembacaan not balok dalam citra bersih dengan kondisi campuran (dalam masing-masing baris partitur lagu), didapatkan hasil rata-rata keakuratan sebesar 75% ( $\frac{\text{total not terbaca dalam 30 citra}}{\text{total seluruh not dalam 30 citra}} \times 100\%$ , terbaca 720 not dari total 960 not). Dan hasil pembacaan not balok dalam citra terdistorsi dalam kondisi campuran dapat dilihat pada tabel 4.2 berikut:

**Tabel 4.2**

Perbandingan Hasil Distorsi Not dalam Masing-Masing Baris Partitur

No.	Nama Distorsi	Hasil Deteksi (dalam %)
1	Blur	53,65
2	Noise	63,75
3	Film Grain	50,94
4	Terkena cairan (minyak)	10,83
5	Fragment	10,21
6	Mosaic	7,81
7	Tekena cairan (teh)	10,21
8	Terlipat / tertekuk / lusuh	45,63,
9	Kemiringan	4,79
10	Wind	30,63
11	Dark Strokes	33,13
12	Smudge Stick	39,80
13	Tersobek dan ditambal dengan selotip	7,81
14	Photocopy	71,46

Dapat diambil kesimpulan dari tabel di atas bahwa selama garis paranada dan bulatan not dalam citra masih nampak cukup jelas oleh sistem, maka pembacaan pola not balok masih dapat dilakukan untuk kemudian diproses menjadi not angka.

#### **4.4 Kelebihan dan Kekurangan Sistem**

##### **4.4.1 Kelebihan Sistem**

- a) Sistem ini dapat menerima inputan data dalam bentuk foto, *scan*, atau proses citra lainnya selama masih dalam format \*.jpg untuk pengenalan partitur not balok.
- b) Sistem dapat masih dapat membaca posisi not balok walau citra telah mengalami distorsi dalam tingkat tertentu.
- c) Sistem memberikan kebebasan bagi pengguna untuk memilih proses penginputan dan pemrosesan konversi.
- d) Sistem dapat langsung menampilkan hasil konversi dalam not angka yang sudah dibagi berdasarkan masing-masing birama.

##### **4.4.2 Kekurangan Sistem**

- a) Dalam proses segmentasi dan *region* masih ditemukan kendala beberapa not yang belum akurat yang disebabkan sistem masih bersifat statis (belum dapat secara dinamis dalam pengolahan citra) karena jumlah birama dalam 1 baris sudah ditetapkan dan pembagian *region* yang terkadang masih melenceng (tergeser).
- b) Sistem tidak dapat membaca apabila gambar partitur yang diinputkan tidak dalam posisi yang lurus.

## LAMPIRAN

### I. Lampiran Coding

#### a) Menu Utama

##### Menampilkan Logo UKDW Pada Menu Utama

```
% --- Executes just before menuutama is made visible.
function menuutama_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% varargin   command line arguments to menuutama (see
VARARGIN)

% Choose default command line output for menuutama
handles.output = hObject;
axes(handles.axes1);
imshow(imread('ukdw.jpg'));
grid off;
axis off;
```

##### Untuk Masuk Halaman Konversi Pada Tombol “Mulai”

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
figure(konversi)
```

##### Untuk Masuk Halaman Konversi Pada Menu Bar “Mulai”

```
% -----
function Untitled_2_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
figure(konversi)
```

## b) Halaman Konversi

### Menginputkan dan Menampilkan Citra yang Dipilih Pengguna

Menginputkan melalui tombol “Masukkan Gambar Partitur”

Menampilkan Citra yang dipilih pada kanvas axes1

```
% --- Executes on button press in pushbutton1.  
function pushbutton1_Callback(hObject, eventdata, handles)  
% hObject handle to pushbutton1 (see GCBO)  
% eventdata reserved - to be defined in a future version of  
MATLAB  
% handles structure with handles and user data (see  
GUIDATA)  
global I  
[n p]=uigetfile('*.jpg');  
nmf=[p n];  
axes(handles.axes1);  
I=imread(nmf);  
imshow(I);
```

### Melakukan Proses Konversi

Proses konversi melalui tombol “Proses Konversi”

Menampilkan hasil segmentasi pada axes3 sampai axes 10

Menampilkan hasil konversi pada edit text (edit12 sampai edit19)

```
% --- Executes on button press in pushbutton2.  
function pushbutton2_Callback(hObject, eventdata, handles)  
% hObject handle to pushbutton2 (see GCBO)  
% eventdata reserved - to be defined in a future version of  
MATLAB  
% handles structure with handles and user data (see  
GUIDATA)  
  
%bagi region kolom menjadi 8 segmen  
global I  
I=rgb2gray(I);  
I=im2bw(I);  
I=imresize(I, [130,1600]);  
% segmentasi 8  
Is=delapan(I);  
for i=1:size(Is,3)  
cmd=['axes(handles.axes' num2str(i+2) ')'];  
eval(cmd);  
cmd=['imshow(Is(:, :, ' num2str(i) '))'];  
eval(cmd);
```

```

end
A=(Is);

    hasil=[];
    for i=1:size(A,3)
        B=empat(A(:,:,i));
        for j=1:size(B,3)
            C=regio(B(:,:,j));
            hasil=[hasil C];
        end
    end

hasil_t=[''];

for i=1:length(hasil)
    hasil_t=[hasil_t num2str(hasil(i)) ' '];
end

tab=12:-1:1

h1=[]
for i=1:4
    h1=[h1 num2str(tab(hasil(i)))];
end
h2=[]
for i=5:8
    h2=[h2 num2str(tab(hasil(i)))];
end
h3=[]
for i=9:12
    h3=[h3 num2str(tab(hasil(i)))];
end
h4=[]
for i=13:16
    h4=[h4 num2str(tab(hasil(i)))];
end
h5=[]
for i=17:20
    h5=[h5 num2str(tab(hasil(i)))];
end
h6=[]
for i=21:24
    h6=[h6 num2str(tab(hasil(i)))];
end
h7=[]
for i=25:28
    h7=[h7 num2str(tab(hasil(i)))];
end
h8=[]
for i=29:32
    h8=[h8 num2str(tab(hasil(i)))];
end

for i=1:8

```

```

    cmd=['set(handles.edit' num2str(i+11) ',' char(39)
'string' char(39) ',h' num2str(i) ') ' ' ' ]
    eval(cmd);
end

```

### Pemanggilan Fungsi Segmentasi “delapan”

Segmentasi membagi satu baris partitur menjadi 8 birama (A=delapan)

```

function A=delapan(I)
    A(:,:,1)=I(:,1:200);
    A(:,:,2)=I(:,201:400);
    A(:,:,3)=I(:,401:600);
    A(:,:,4)=I(:,601:800);
    A(:,:,5)=I(:,801:1000);
    A(:,:,6)=I(:,1001:1200);
    A(:,:,7)=I(:,1201:1400);
    A(:,:,8)=I(:,1401:1600);

```

### Pemanggilan Fungsi Segmentasi “empat”

Segmentasi membagi satu baris partitur menjadi 4 not (B=empat)

```

function B=empat(I);
    I=imresize(I,[130,200]);
    B(:,:,1)=I(:,1:50);
    B(:,:,2)=I(:,51:100);
    B(:,:,3)=I(:,101:150);
    B(:,:,4)=I(:,151:200);

```

### Pemanggilan Fungsi region “regio”

Region membagi satu not menjadi 12 region (C=regio)

```

function b=regio(I)
    I=imresize(I,[130,50]);
    for i=1:130
        if sum(sum(I(i,:)))==0
            I(i,:)=ones(1,50);
        end
    end
    C1=I(1:20,:);
    C2=I(10:30,:);
    C3=I(20:40,:);
    C4=I(30:50,:);
    C5=I(40:60,:);

```

```

C6=I(50:70,:);
C7=I(60:80,:);
C8=I(70:90,:);
C9=I(80:100,:);
C10=I(90:110,:);
C11=I(100:120,:);
C12=I(110:130,:);
area=[];
for i=1:12
    cmd=['area(' num2str(i) ')=sum(sum(C' num2str(i)
    '));'];
    eval(cmd);
end
[a b]=min(area);

```

## II. Lampiran Hasil Penelitian

### Citra dalam kondisi bersih

Terbaca: 720 not





**Citra dalam kondisi distorsi**

**a. Blur**

Terbaca: 686 not

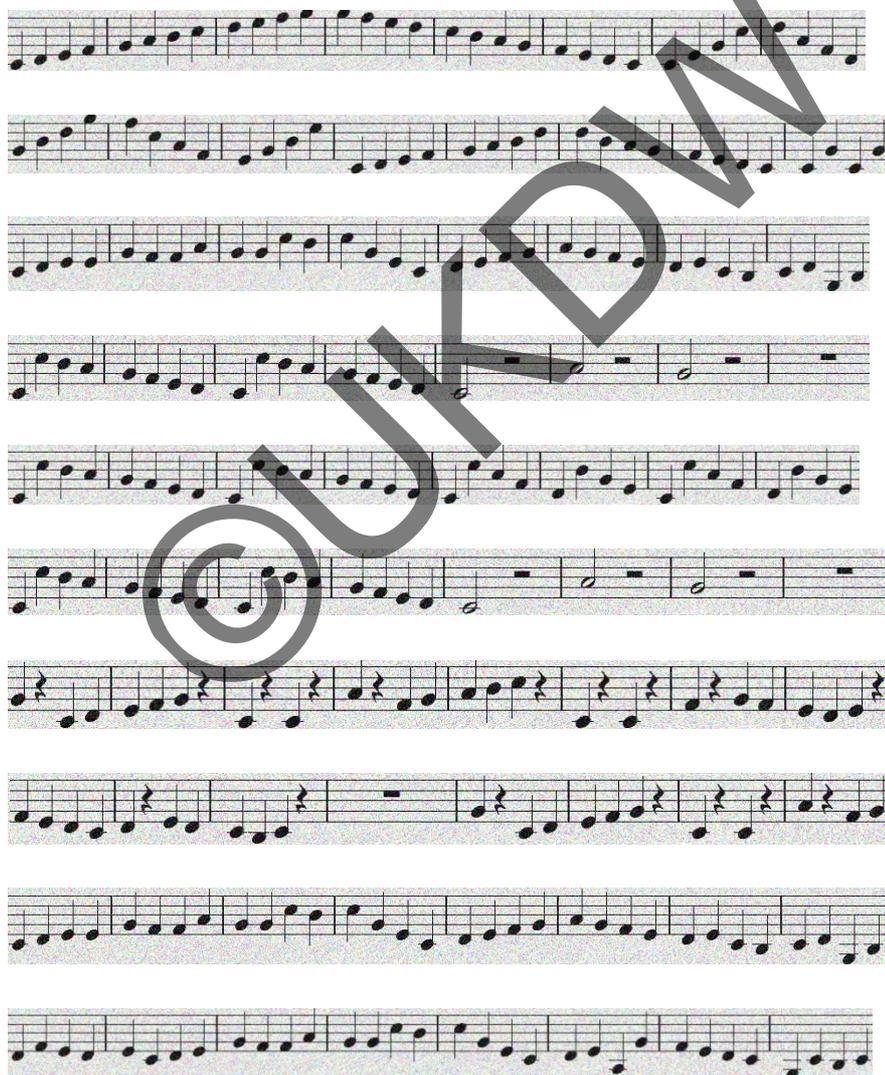


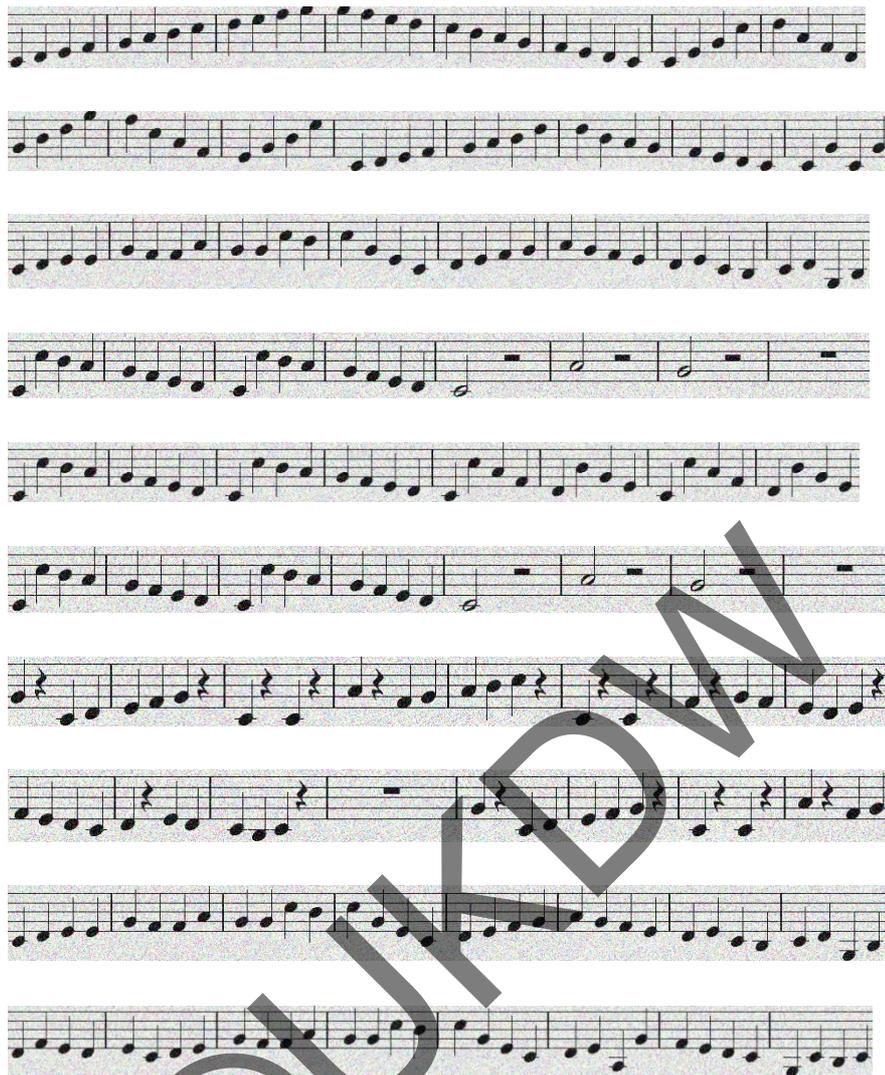
A musical score consisting of 15 staves of music. The notation includes various note values, rests, and bar lines. A large, semi-transparent watermark reading '© UKDM' is overlaid diagonally across the center of the page, partially obscuring the musical notation.



**b. Noise**

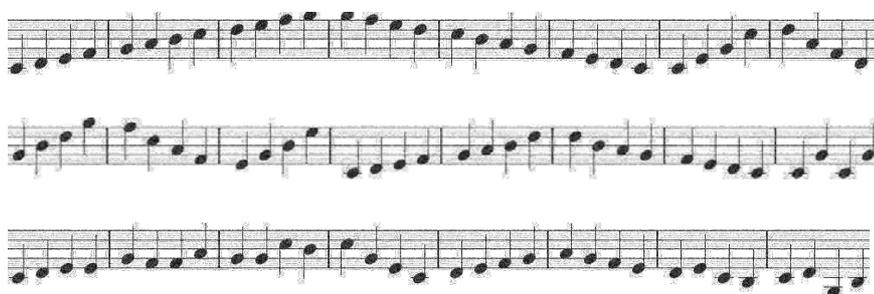
Terbaca : 612 not



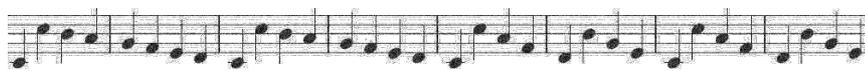
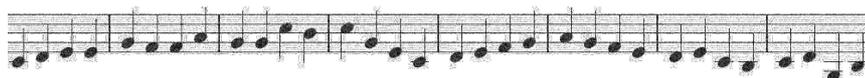


**c. Film Grain**

Terbaca: 489 not

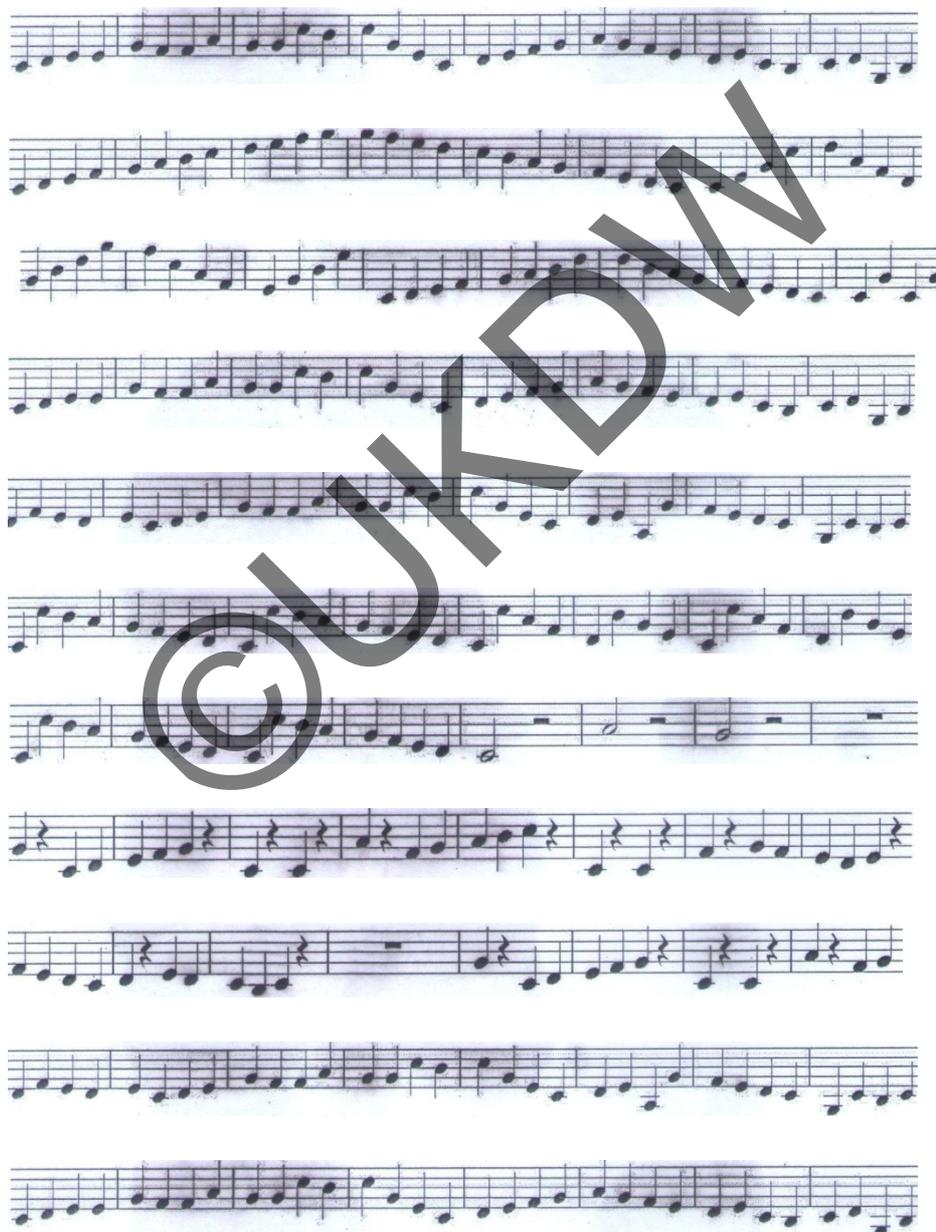


This image displays a musical score for guitar, consisting of 15 horizontal staves. The notation includes various rhythmic values such as eighth and sixteenth notes, as well as rests. A large, semi-transparent watermark with the text '© UNKOWN' is overlaid diagonally across the center of the page, partially obscuring the musical notation.



**d. Terkena cairan (minyak)**

Terbaca: 104 not





**e. Fragment**

Terbaca: 98 not



A musical score consisting of 15 staves of music. The notation includes various rhythmic values, stems, and beams. A large, semi-transparent watermark with the text '© UKDM' is overlaid diagonally across the center of the page, partially obscuring the musical notation.



**f. Mosaic**

Terbaca: 82 not





**g. Terkena cairan (teh)**

Terbaca: 98 not







**h. Terlipat / tertekuk / lusuh**  
Terbaca: 438 Not



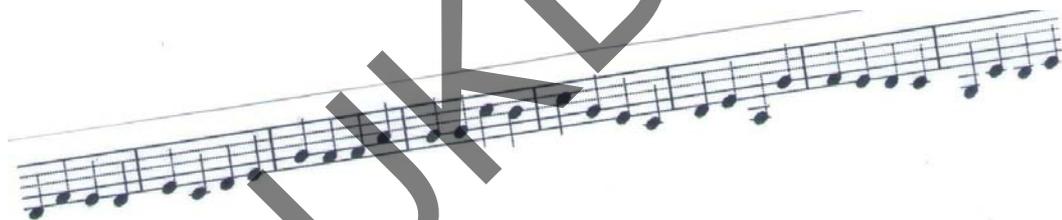
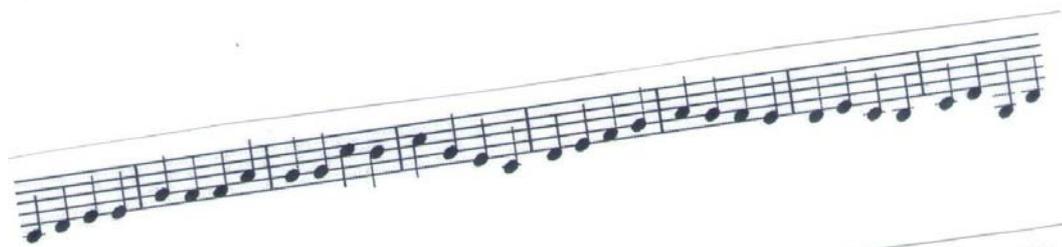


i. **Kemiringan**

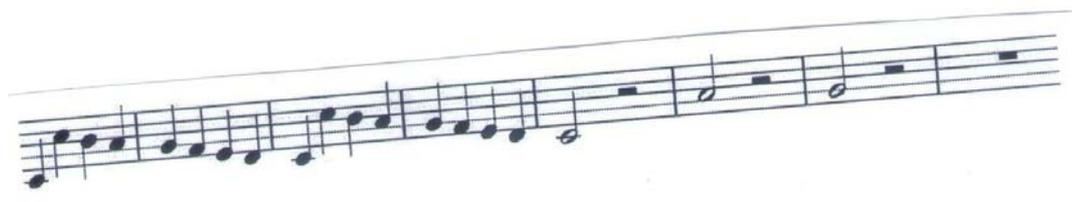
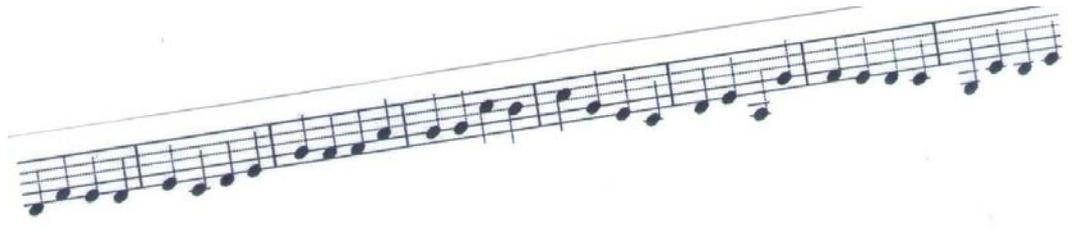
Terbaca: 46 Not







© UKDM



© UKDM

j. **Wind**

Terbaca: 294 Not

The image displays a musical score for a piece titled "Wind". The score is written on 14 staves, organized into seven pairs. Each pair consists of a treble clef staff and a bass clef staff. The music is written in a single system, with a large, semi-transparent watermark "© UKDM" overlaid diagonally across the center of the page. The notation includes various note values, rests, and bar lines, typical of a musical score. The piece is identified as having 294 notes.



k. **Dark Strokes**

Terbaca: 318 Not



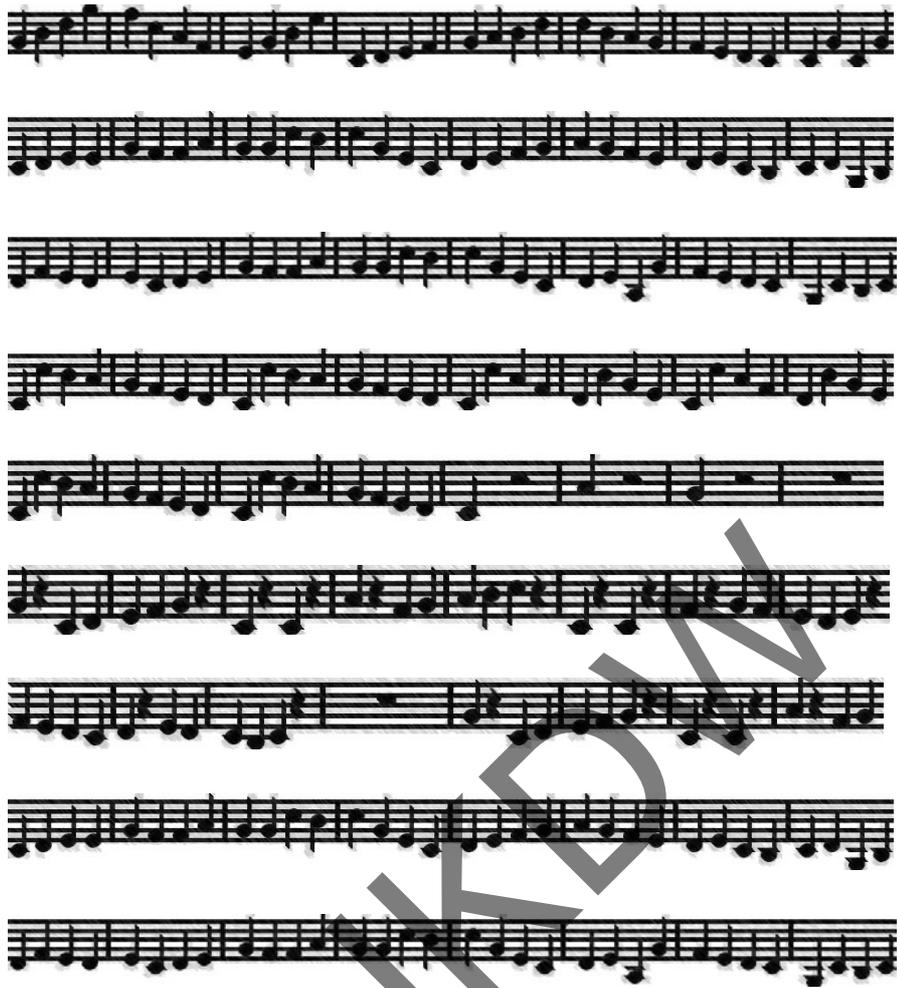
This image displays a page of musical notation, likely a score for a piece of music. The notation is arranged in 15 horizontal staves. Each staff contains a sequence of notes, rests, and bar lines, representing a melodic or harmonic line. The notes are primarily eighth and sixteenth notes, with some quarter notes and rests. The watermark 'SUKUN' is prominently displayed in the center, oriented diagonally from the bottom-left towards the top-right.



I. Smudge Stick

Terbaca 382 Not





© UKRIN

m. Tersobek dan ditambal dengan selotip

Terbaca: 75







n. Photocopy

Terbaca: 686 not



The image displays ten staves of musical notation, likely for a guitar or piano. The notation includes various rhythmic values such as eighth and sixteenth notes, as well as rests. A large, semi-transparent watermark with the text '© SUKRON' is positioned diagonally across the middle of the page, partially obscuring the musical staves.