BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Metode pengenalan karakter pada citra natural salah satunya adalah *template matching correlation.* Metode ini digunakan untuk mengenali tulisan dalam citra yang diperoleh dari kamera perangkat *mobile.* Dalam penelitian (Saha Satyajit S., dkk, 2013) menggunakan *grayscale, thresholding, blurring* untuk dalam *preprocessing*. Tahap selanjutnya yaitu *character recognition* yang terdiri dari *thinning*, segmentasi, *scaling* dan kemudian *template matching*. Sebelum setiap karakternya dikenali, terlebih dahulu citranya disegmen agar karakter - karakter penting dapat terpisah dari *background*. Setelah itu barulah karakter - karakter yang sudah disegmen diubah menjadi ukuran yang standar.

Penerapan metode ini diakui dapat mengurangi waktu dan tenaga, sekaligus didapati pengolahan data yang terbukti cepat. Hal ini selaras dengan penelitian yang membandingkan penerapan *template matching correlation* dengan *phase angle method* untuk identifikasi obyek dalam suatu citra natural (Ahuja & Tuli, 2013). membutuhkan waktu yang lebih singkat untuk mengenali obyek yang berbeda ukuran seperti yang akan diterapkan dalam sistem ini.

Pada dasarnya, tahap-tahap pengenalan teks pada citra meliputi *preprocessing,* ekstraksi, dan pengenalan karakter. Pada pengenalan lembar jawab (Risanto & Bahri, 2013), tahap *preprocessing* meliputi *grayscale*, binerisasi dan *cropping*. Proses pengenalan karakter tersebut dilanjutkan dengan tahap ekstraksi dan pengenalan menggunakan *template* yang disimpan melalui tahap pelatihan.

Proses yang serupa juga dilakukan untuk mengenali karakter dalam dokumen cetak dengan variasi bentuk dan ukuran (Hartanto, Sugiharto, & Endah, 2011). Dalam jurnal ini dijelaskan lebih detail dalam proses ekstraksi

menggunakan segmentasi dan normalisasi. Rata-rata tingkat keberhasilan pengenalan dari sistem ini dinyatakan sebesar 92,90%.

Namun pada pengenalan dokumen cetak kamus bilingual (Ferdinandus, Hermawan, Tedjokusumo, & Zaman, 2011), terdapat perbedaan dengan jurnal-jurnal sebelumnya dalam penerapan tahap-tahap pengenalan karakternya. Setelah melewati tahap pengenalan pola menggunakan *template matching,* penulis mendapati banyak karakter yang gagal dikenali. Hal ini dikarenakan karakter-karakter tersebut gagal dipisahkan, diproses kembali dengan histogram projection dan dilakukan template matching kembali menggunakan CCOEFF. Karakter yang masih gagal dikenali karena perbedaan format *bold* dan *italic* dilakukan *postprocessing* untuk memperbaiki format karakter kemudian dilakukan *template matching* kembali.

Penelitian yang saya lakukan adalah pada *preprocessing* menggunakan *grayscale,* binerisasi, hsv, binerisasi dari hsv, komplemen, menggabungkan hasil binerisasi dari *grayscale* dengan hasil komplemen dari hsv dengan operasi and. Pada pencocokan menggunakan *template matching* yang sebelumnya dibuat daerah-daerah potong 70, 20, dan 12. Daerah-daerah tersebut yang nantinya digunakan untuk mendapatkan nilai korelasi. Penggunaan daerah potong tersebut bertujuan untuk mengurangi penggunaan memori dan mengetahui kecocokan citra uji dari ciri khas karakter yang dicocokan dengan *template.*

2.2 Landasan Teori

Penelitian ini didasari oleh penelitian-penelitian *Template Matching Correlation* namun dengan *template* yang berbeda yaitu huruf dan angka yang terdapat dalam tampilan skor mesin PIU seri Prime. Perbedaan dengan penelitian tersebut adalah proses ekstraksi dalam penelitian ini dilakukan terhadap citra warna

2.2.1 Preprocessing

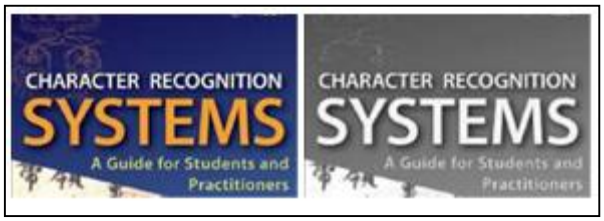
Preprocessing merupakan tahapan awal dalam mengolah data input sebelum memasuki proses tahapan utama yaitu *citra processing*, dan *text recognition*. Penggunaaan operasi *preprocessing* agar citra siap untuk diekstraksi. Dalam project ini langkah-langkah preprocessing yang dilakukan adalah *grayscaling*, binerisasi citra, dan *noise removal*, (Cheriet dkk., 2007).

2.2.1.1 Grayscaling

Grayscale adalah suatu citra yang nilai dari setiap pikselnya merupakan *sample* tunggal. Citra yang dihasilkan dari citra *grayscale* ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat seperti pada gambar 2.1. Citra *grayscale* berbeda dengan citra ”hitam-putih”, citra hitam putih hanya terdiri atas 2 warna saja yaitu ”hitam” dan ”putih” saja.

Pada citra *grayscale* warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak. Citra *grayscale* seringkali merupakan perhitungan dari intensitas cahaya pada setiap piksel pada spektrum elektromagnetik *single band* seperti pada persamaan 2.1. Citra *grayscale* disimpan dalam format 8 bit untuk setiap *sample pixel*, yang memungkinkan sebanyak 256 intensitas (Purnomo dan Muntasa, 2010).

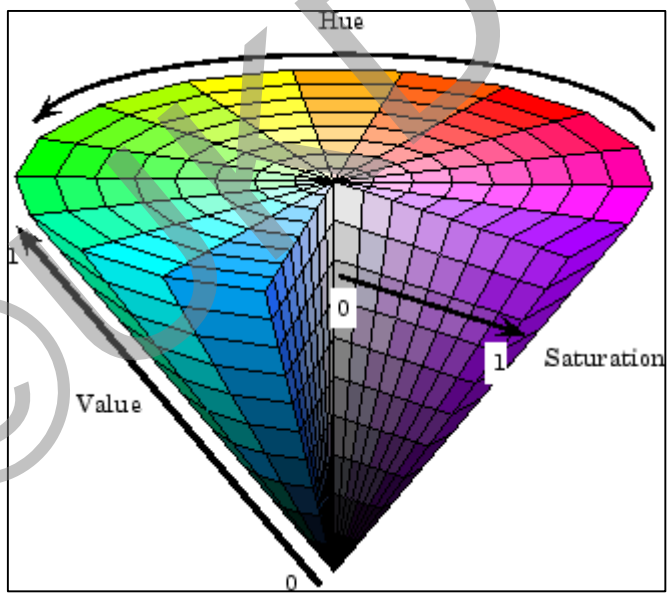
$$\text{CitraGray} = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad \text{..... [2.1]}$$



Gambar 2.1. Hasil konversi citra berwarna menjadi citra grayscale
Dikutip dari: Cheriet, M; Nawwaf, K; Liu, C; Suen, C, Y. (2007). *Character Recognition Systems: A Guide for Students and Practitioners.*

2.2.1.3 HSV

Ruang Warna HSV menunjukkan ruang warna dalam tiga komponen utama yaitu *Hue, Saturation* dan *Value*. Hue menunjukkan jenis warna seperti merah, biru atau kuning. Hue bernilai mulai dari 0 sampai 360 derajat bisa dilihat seperti pada gambar 2.2. *Saturation* adalah ukuran seberapa besar kemurnian dari warna tersebut. *Saturation* bernilai 0 sampai dengan 1. *Saturation* ini menunjukkan nilai abu dari warna dimana 0 menunjukkan abu-abu dan 1 menunjukkan warna murni. *Value* merupakan ukuran untuk menunjukan besar kecerahan dari suatu warna atau seberapa besar cahaya yang berasal dari suatu warna (Putra, 2010). Perhitungan HSV sendiri terpisah yaitu untuk *Hue* perhitungannya adalah pada persamaan 2.2, perhitungan Saturasinya yaitu pada persamaan 2.3 dan untuk *Value* menggunakan persamaan 2.4.



Gambar 2.2 Gambar ruang warna HSV

Dikutip dari: Mathworks.com (2017).*Convert from HSV to RGB Color Space* (https://www.mathworks.com/help/images/convert-from-hsv-to-rgb-color-space.html)

$$H = tan\left[\frac{3(G-B)}{(R-G)+(R-B)}\right] \quad \text{..... [2.2]}$$

$$S = 1 - \frac{\min(R,G,B)}{V} \quad \text{..... [2.3]}$$

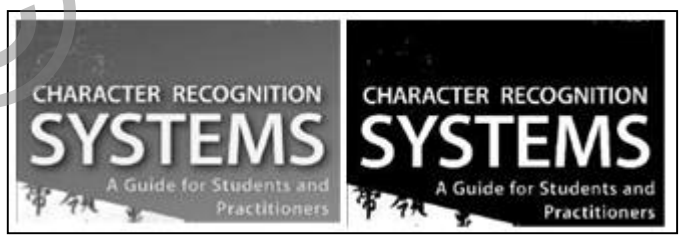
$$V = \frac{B}{R+G+B} \quad \text{..... [2.4]}$$

2.2.1.4 Binerisasi Citra

Citra biner adalah citra yang setiap pikselnya hanya memiliki 2 kemungkinan derajat keabuan yakni 0 dan 1. Dengan merubah ke bentuk biner, citra hanya akan mempunyai 2 warna yakni hitam dan putih seperti pada gambar 2.3. Dengan proses ini, citra RGB juga akan menjadi 1 matriks penyusun saja.

Proses pembineran dilakukan dg membulatkan keatas atau kebawah untuk setiap nilai keabuan dari piksel yg berada diatas atau bawah harga ambang. Metode untuk menentukan besarnya harga ambang disebut *thresholding.*

Thresholding digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Dengan menggunakan *thresholding* maka derajat keabuan bisa diubah sesuai keinginan, misalkan diinginkan menggunakan derajat keabuan 16, maka tinggal membagi nilai derajat keabuan dengan 16. (Purnomo dan Muntasa, 2010). Dalam kasus ini *threshold* yang digunakan untuk skor sebesar 180 dan untuk judul sebesar 120.



Gambar 2.3 Hasil konversi citra *grayscale* menjadi citra biner

Dikutip dari: Cheriet, M; Nawwaf, K; Liu, C; Suen, C, Y. (2007). *Character Recognition Systems: A Guide for Students and Practitioners.*

2.2.1.5 Threshold HSV

Proses Threshold dari HSV bertujuan untuk mengubah citra HSV menjadi biner. Proses Threshold HSV ini dapat diimplementasikan dengan cara menghubungkan warna HSV yang sudah dilakukan memiliki *threshold*.

2.2.1.6 *Image Complement*

Dalam komplemen dari citra biner, nol menjadi satu dan yang menjadi nol, hitam dan putih terbalik. Dalam komplemen dari intensitas atau gambar RGB, setiap nilai piksel dikurangi dari nilai piksel maksimum yang didukung oleh kelas (atau 1, 0 untuk gambar presisi ganda) dan perbedaan digunakan sebagai nilai piksel pada gambar *output*. Pada gambar *output*, daerah gelap menjadi lebih ringan dan area terang menjadi gelap (The MathWorks, Inc, 2016).

2.2.1.7 *Noise Reduction*

Menghilangkan kotoran pada citra merupakan salah satu cara untuk membantu dalam memperbagus citra agar mudah diproses ke tahap selanjutnya. Selain memperbagus juga dapat membantu dalam meningkatkan detail dari suatu citra.

2.2.1.8 Pemotongan dengan histogram

Pemotongan digunakan untuk mempermudah pengidetifikasian citra pada bagian tertentu. Bagian - bagian citra sendiri memiliki informasi yang berbeda - beda sehingga pemotongan sangat berpengaruh dalam mengelompokkan citra.

2.2.1.9 *Resize*

Operasi *resize* adalah operasi pengurangan atau pertambahan ukuran suatu citra. Operasi *resize* yang digunakan dalam kasus ini adalah operasi yang mengubah ukuran citra baru yang memiliki nilai tinggi dan lebar, perubahan ukuran tidak akan mengubah bentuk dari citra awal namun hanya ukurannya saja yaitu dengan mempertahankan perbandingan skalanya (The MathWorks, Inc., 2016).

2.2.2 Ekstraksi

2.2.2.1 Region of Interest

Region of Interest (ROI) merupakan pengkodean pada area tertentu dari citra digital, sehingga mempunyai kualitas yang lebih baik dari area sekitarnya. Area yang dipilih diidentifikasi untuk tujuan tertentu sehingga fitur ini menjadi sangat penting. Bagian tertentu dari citra digital yang dirasakan lebih penting dari bagian yang lainnya tersebutlah yang nantinya akan menjadi fokus untuk diproses (Brinkmann, 1999). Dalam kasus ini ROI yang dipakai adalah *regionprops* dengan *bonding box.*

2.2.3 Klasifikasi

2.2.3.1 *Template Matching Correlation*

Template matching correlation adalah suatu metode klasifikasi citra untuk mengenali teks/karakter yang memiliki tingkat keberhasilan yang tinggi. Metode ini membutuhkan *memory* dan *storage* yang cukup besar untuk melakukan pemrosesan. Dalam metode ini, keseluruhan citra akan diproses untuk dicari objek yang memiliki kemiripan objek dengan *lexicon* sampel.

Metode *Template Matcing Correlation* dilakukan dengan cara mencocokkan matriks - matriks citra dengan teknik statistik yang bersifat kuantitatif yaitu dengan persamaan 2.5. Proses ini memerlukan kekuatan komputasi yang cukup tinggi untuk mencapai hasil optimum, namun hal ini membuat tingkat akurasi menjadi lebih baik daripada metode - metode lain karena algoritma ini dilakukan dengan cara mencocokkan setiap piksel pada suatu matriks citra digital dengan citra yang menjadi *template*. Dalam library Matlab dapat digunakan dengan:

```
Corr2(Data 1, Data 2);
```

Algoritma dari Corr2 sendiri dirumuskan sebagai berikut:

$$r = \frac{\sum_{k=1}^{n}(x_{ik}-\bar{x}_i).(x_{jk}-\bar{x}_j)}{\sqrt{\left[\sum_{k=1}^{n}(x_{ik}-\bar{x}_i)^2.\sum_{k=1}^{n}(x_{jk}-\bar{x}_j)^2\right]}} \quad \text{..... [2.5]}$$

Dimana $\bar{x}_i$ dirumuskan dengan persamaan 2.6

dan $\bar{x}_j$ dirumuskan dengan persamaan 2.7

$$\bar{x}i = \frac{1}{n}\sum_{k=1}^{n} xik \quad \text{..... [2.6]}$$

$$\bar{x}j = \frac{1}{n}\sum_{k=1}^{n} xjk \quad \text{..... [2.7]}$$

Keterangan :

r adalah nilai korelasi antara dia buah matriks (nilainya antara -1 atau +1).

x_{ik} adalah nilai piksel ke-*k* dalam matriks i.

x_{jk} adalah nilai piksel ke-*k* dalam matriks j.

$\bar{x}_i$ adalah rata-rata nilai piksel matriks i.

$\bar{x}_j$ adalah rata-rata nilai piksel matriks j.

n menyatakan jumlah piksel dalam satuan matriks.

BAB 3

RENCANA RISET DAN PERANCANGAN SISTEM

3.1. Spesifikasi Pembuatan Sistem

Spesifikasi pembuatan sistem merupakan penjelasan secara detail mengenai perangakat yang digunakan dalam pembuatan sistem yang terdiri dari spesifikasi perangkat keras dan spesifikasi perangkat lunak. Spesifikasi perangkat keras dan spesifikasi perangkat lunak yang digunakan dalam pembuatan sistem ini adalah adalah sebagai berikut:

a) Perangkat Keras

Perangkat keras yang digunakan dalam pembuatan sistem Pencatatan Skor ini yaitu dengan menggunakan *notebook* dengan detail spesifikasi *device* sebagai berikut:

i. CPU : Intel Core i5 (2.5GHz)
ii. RAM : 6 GB
iii. Hard Disk :500 GB

b) Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan sistem Pencatatan Skor ini yaitu sebagai berikut:

i. Sistem Operasi Windows 8.1 (64-bit)
ii. Matlab R2016a

c) Fungsi - fungsi Matlab

Perangkat lunak yang dibangun menggunakan fungsi - fungsi matlab antara lain:

i. im2gray(Citra RGB)
ii. rgb2hsv(Citra RGB)
iii. imcomplement(Citra Biner)
iv. bwareaopen(Citra Biner)
v. imresize(Citra)

3.2 Rencana Riset

Riset yang akan dilakukan adalah mendapatkan *threshold* binerisasi yang terbaik agar setiap karakter dapat terpisah dan menjadi biner dengan hasil yang baik. Sehingga hasil binerisasi tersebut dapat meminimalkan foreground tetapi tanpa mengurangi informasi penting yang ada di dalamnya.

Riset selanjutnya adalah dengan melihat apakah *noise* yang berupa cahaya bisa di pisahkan yaitu menggunakan warna HSV yang pada penerapannya menggunakan fungsi dari matlab. Fungsi dari riset ini adalah untuk mengurangi dampak dari *noise* cahaya yang mengganggu informasi yang ada dalam citra sehingga sulit untuk dikenali.

Riset yang ketiga adalah untuk mengetahui apakah citra sudah cukup bersih dari *noise* yang memiliki piksel kecil maupun besar yaitu dengan membuat batas maksimal dan minimal untuk luas piksel informasi *foreground*. Sehingga apabila suatu *foreground* memiliki luas kurang dari atau lebih dari batas luas yang ditentukan maka piksel *foreground* tersebut akan hilang

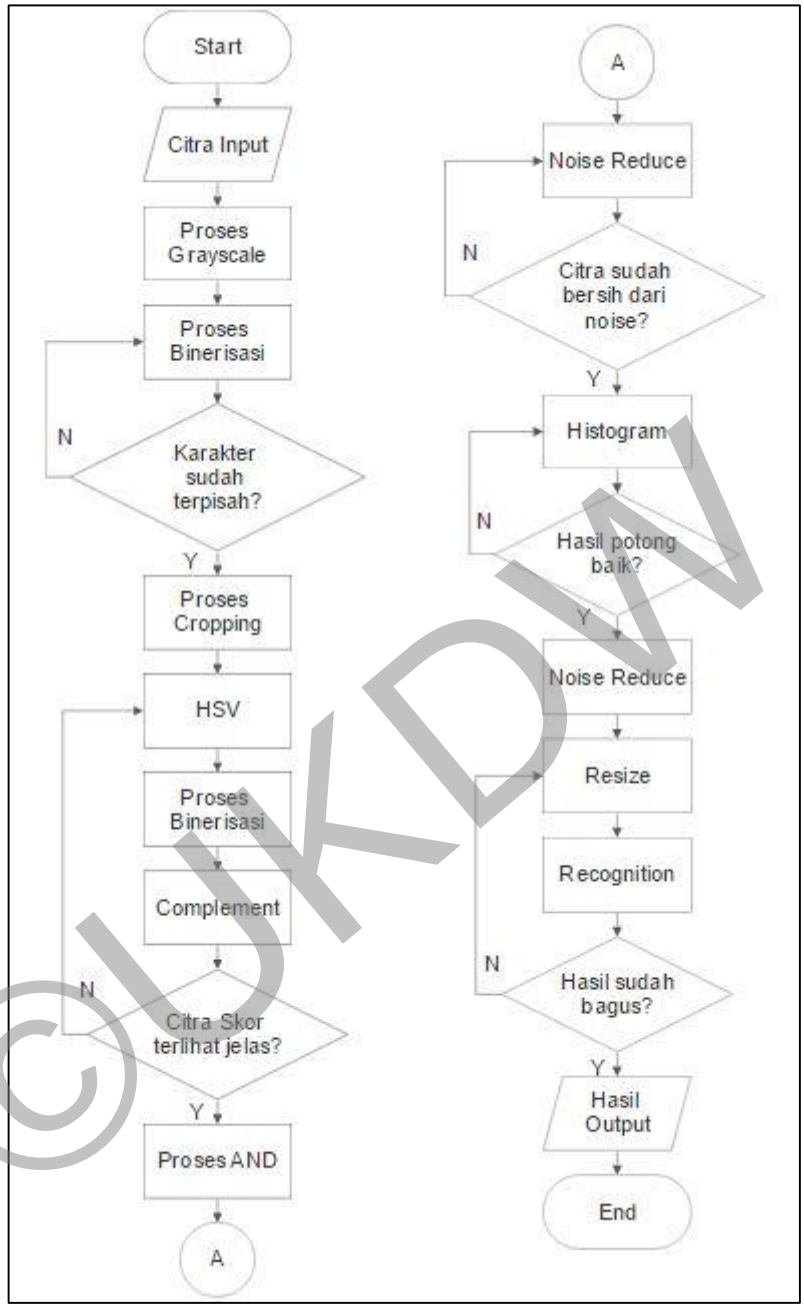
Riset berikutnya adalah untuk menentukan tinggi potong citra perbaris. Riset ini dilakukan untuk mempermudah informasi yang ada pada citra dalam pengelompokan perbaris. Apabila tingginya tidak pas maka perpotongannya tidak bagus yang menyebabkan hasil potong yang kurang maksimal dan memperburuk hasil sebelum di ekstraksi. Sehingga tinggi potong ini harus pas agar dapat menghasilkan hasil potong yang baik, karena proses ini sangat menentukan berhasil tidaknya citra dalam pengelompokan datanya.

Riset terakhir yang akan saya lakukan adalah dengan proses *template matching* dalam ukuran *template* yang berbeda yang akan dibagi menjadi beberapa daerah yang nantinya akan dicocokan dengan daerah citra hasil ekstraksi. Riset ini bertujuan untuk mengetahui akankah sistem menghasilkan hasil output yang lebih baik atau semakin buruk.

Riset tersebut dilakukan dengan 2 perbedaan ukuran yaitu 20x20 px dan 60x60. Inilah yang nantinya menjadi dasar apakah ukuran mempengaruhi hasil dari penggunaan *template matching.* Setelah ukurannya ditentukan kemudian yang akan digunakan untuk mengetahui hasil dari template mathcing sendiri adalah menggunakan 3 macam daerah potongan. Masing - masing ukuran akan dicoba untuk mengetahui hasilnya dengan 70 potongan area, 20 potongan area, dan 12 potongan area. Perpotongan area ini bertujuan untuk mengurangi jumlah memori yang digunakan untuk mengidentifikasi citra. Oleh karena *resize* yang digunakan adalah 13px dan untuk dicocokan dengan *template*, citra di pindahkan ke ukuran yang sama dan dimulai dari kordinat (2,2) sehingga untuk tinggi *foreground* hanya 14px oleh karena itu yang digunakan adalah 70, 20, dan 12 bukan 100 (per 2 piksel), 25 (per 4 piksel) dan, 16 (per 5 piksel) area.

Dari semua riset seperti pada gambar 3.1 yang dilakukan maka akan diketahui berapakah *threshold,* batas luas hingga batas tinggi yang menjadi acuan nilai yang akan digunakan. Hasil korelasi dari ketiga potongan area dengan *template* dari masing - masing ukuran juga akan diketahui manakah hasil yang baik untuk digunakan untuk mengenali karakter dengan hasil yang terbaik.

Setelah hasil pencocokan selesai maka akan dilakukan pengujian yaitu dengan melihat hasil per karakter dan hasil per tempat. Sehingga akan diketahui hasil akurasi ketepatan dari sistem.



Gambar 3.1 Pohon riset

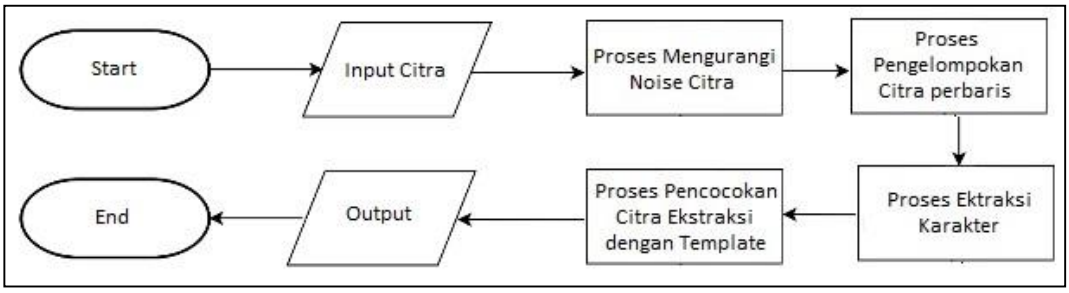
3.3 Rancangan Sistem

3.3.1 Deskripsi Sistem

Sistem yang akan dibuat adalah sistem pengenalan karakter untuk mencatat judul dan angka skor yang terdapat dalam suatu citra. Citra input sistem ini merupakan citra yang berasal dari kamera perangkat mobile yang mengandung layar skor *Pump It Up Prime*. Sistem akan mengolah citra awal tersebut menjadi citra yang dapat digunakan untuk proses pengenalan karakter menggunakan metode template matching correlation. Hasil dari proses pengenalan karakter adalah judul, jumlah *perfect, great, good, bad, miss, combo,* dan skor. Output dari proses pengenalan tersebut akan dimasukkan kedalam array dan disimpan kedalam *database*. Sistem yang akan dibuat ini dapat digunakan tidak hanya untuk versi prime saja tetapi juga dapat digunakan untuk versi yang terbaru.

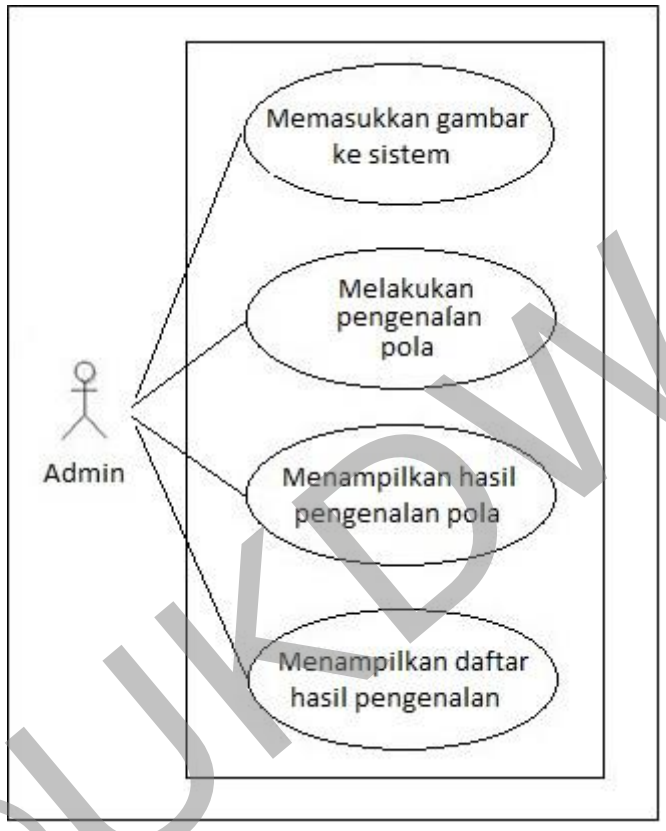
3.3.1.1 Alur Sistem Secara Umum

Awalnya citra dimasukkan kedalam sistem yang kemudian akan diproses untuk mengurangi *noise.* Setelah *noise* diproses, tahap selanjutnya adalah pengelompokan. Pengelompokkan digunakan untuk memudahkan dalam mengidentifikasi informasi yang ada pada citra yang memiliki informasi yang berbeda setiap garisnya. Tujuan dari pemotongan pergaris adalah untuk mempermudah pengelompokan skor antara untuk *perfect, great, good, bad, miss, max combo.* Setelah itu lakukan ektraksi untuk mendapatkan karakter lebih akurat dan dilanjutkan untuk menormalisasi hasil ekstraksi agar semua menjadi ukuran yang sama. Setelah itu barulah dilakukan proses pengenalan dengan template.



Gambar 3.2 Flowchart Sistem

3.3.2 *Use Case*



Gambar 3.3 Use Case

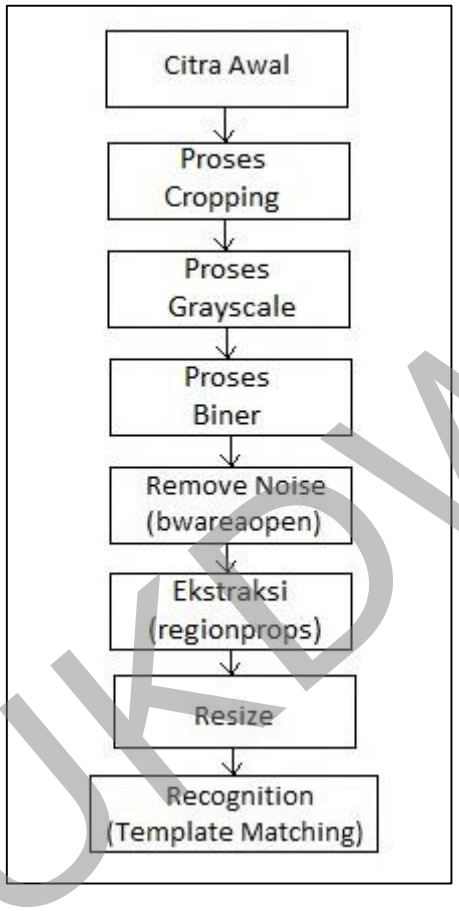
3.3.3 Blok Diagram

Blok diagram dari sistem yang akan dibuat ini ada 2 yaitu untuk proses pengenalan judul maupun proses pengenalan skor yang dimulai dari pengambilan data berupa foto hasil permainan. Setelah data tersebut berhasil diambil, proses selanjutnya adalah *grayscale.*

3.3.2.1 Pengenalan Judul

Sistem untuk bagian judul ini terdiri *croppping*, *grayscale*, binerisasi, *remove noise*, *regionprops*, resize, dan kemudian character recognition. Terdapat

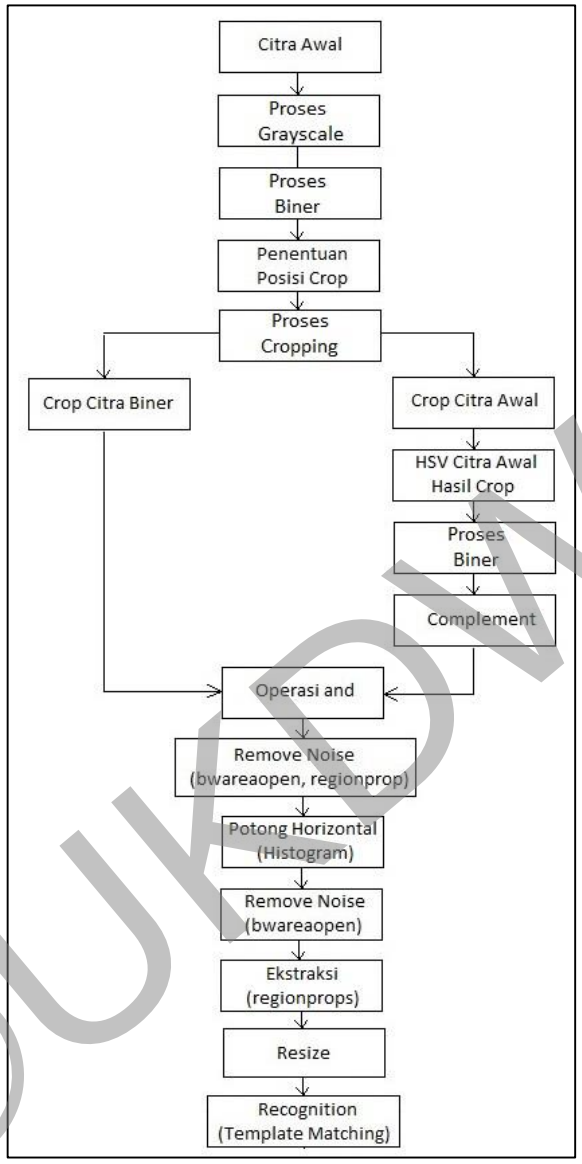
beberapa perbedaan antara proses untuk bagian skor karena untuk bagian judul karakter *foreground* berwarna dasar biru.



Gambar 3.4 Blok Diagram Proses bagian Judul

3.3.2.2 Pengenalan Skor

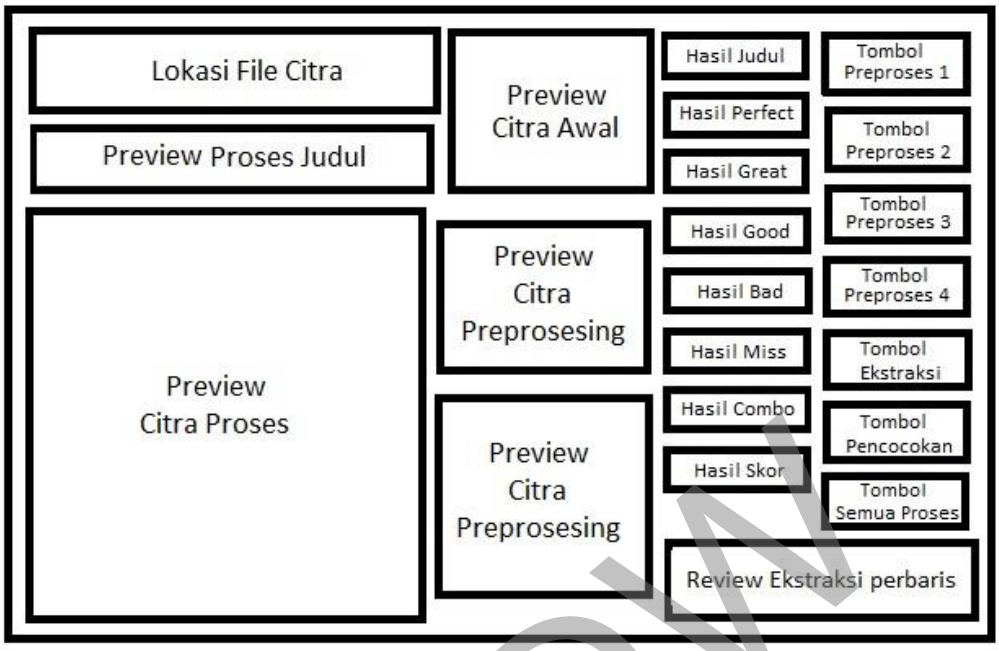
Sistem untuk bagian skor terdiri *grayscale*, binerisasi, penentuan posisi sebelum *cropping*, HSV, operasi and, *remove noise*, pemotongan dengan histogram, *regionprops*, resize, dan kemudian character recognition. Terdapat beberapa perbedaan antara proses untuk bagian judul karena untuk bagian skor karakter *foreground* berwarna dasar putih.



Gambar 3.5 Blok Diagram Proses bagian Skor

3.3.4 Rancangan Antarmuka

Rancangan antarmuka berfungsi untuk menampilkan proses deteksi skor PIU pada sistem. Seperti Gambar 3.6 *mock up* dari rancangan tampilan antarmuka sistem pencatatan skor PIU.



Gambar 3.6 Rencana antarmuka sistem

BAB 4

IMPLEMENTASI, VALIDASI DAN ANALISIS SISTEM

4.1 Implementasi dan Validasi Sistem

4.1.1 Implementasi sistem

4.1.1.1 Binerisasi

Sebelum masuk kepada proses binerisasi citra diubah dahulu menjadi *grayscale* baru kemudian dapat diproses ke binerisasi. Proses binerisasi citra dibagi menjadi 2, yaitu bagian judul dan bagian skor. Perbedaan ini dilihat dari warna dasar kedua karakter judul maupun skor yang memiliki warna yang berbeda. Perbedaan warna dasar keduanya mengakibatkan munculnya kebutuhan proses binerisasi terbagi menjadi dua bagian, yaitu bagian judul dan skor.

4.1.1.1.1 Judul

Pada bagian judul yang dilakukan terlebih dahulu adalah dengan memotong bagian atas citra. Proses tersebut dilakukan untuk mempermudah dalam pemprosesan selanjutnya. Sehingga untuk pemprosesan judul sendiri tidak ada hubungannya dengan bagian skor. Pengujian pada tabel 4.1 merupakan pengujian dengan gambar yang sama. Binerisasi pada bagian judul ini memiliki warna terang yang sedikit kebiruan. Hal ini mengakibatkan perlu adanya *threshold* yang pas agar karakter dapat diproses lebih lanjut seperti pada tabel 4.1.

Tabel 4.1
Tabel *threshold* binerisasi judul

Percobaan	***Threshold***	**Hasil**
1	100	Karakter yang terbaca adalah 'Heel and Toe' tetapi huruf 'e' masih tersambung.
2	110	Karakter yang terbaca adalah 'Heel and Toe' tetapi huruf 'e' masih tersambung.

Tabel 4.1
Tabel *threshold* binerisasi judul (sambungan)

Percobaan	*Threshold*	Hasil
3	120	Karakter yang terbaca 'Heel and Toe' tanpa ada yang tersambung antar katakter.
4	130	Karakter yang terbaca 'Heel and Toe' tanpa ada yang tersambung antar katakter.

Gambar 4.1 Percobaan judul *threshold 100*

Gambar 4.2 Percobaan judul *threshold 110*

Gambar 4.3 Percobaan judul *threshold 120*

Gambar 4.4 Percobaan judul *threshold 130*

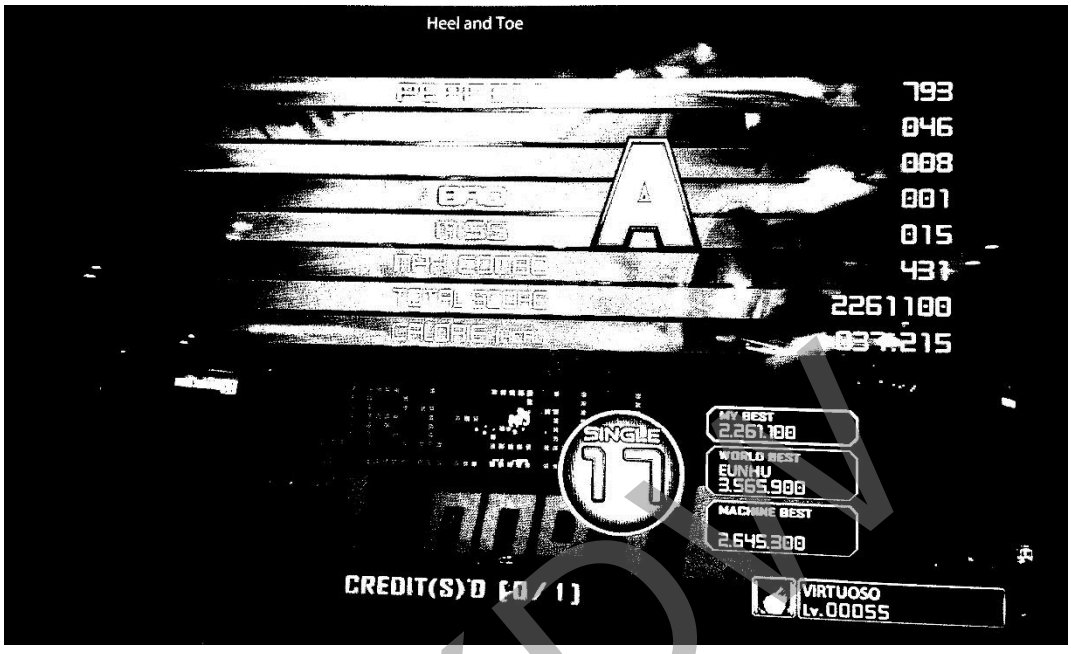
Dari beberapa percobaan diatas, pada gambar 4.1 masih ada karakter yang tidak terpisah, pada gambar 4.2 juga masih ada yang tidak terpisah, gambar 4.3 semua karakter sudah terpisah dan gambar 4.4 juga sudah terpisah semua. Karena pada percobaan ke-3 dan ke-4 memiliki hasil yang sama bagusnya sehingga *threshold digunakan* adalah 120.

4.1.1.1.2 Skor

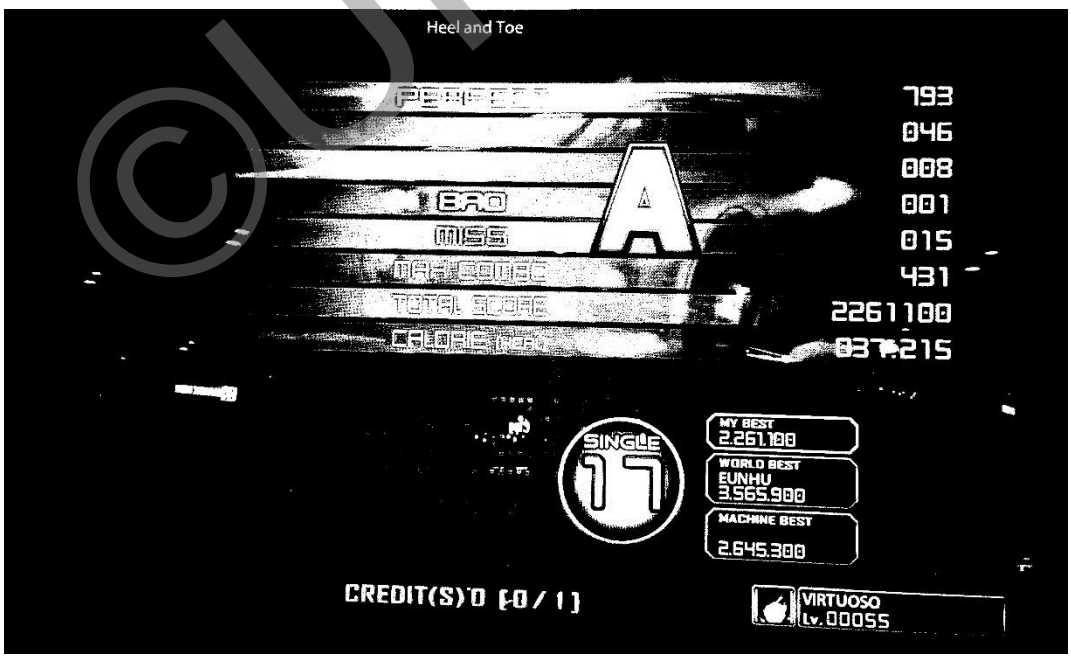
Berbeda dari bagian judul, untuk pemprosesan bagian judul menggunakan citra ukuran awal yang nantinya akan dipotong. Pengujian pada tabel 4.2 menggunakan gambar yang sama. Binerisasi pada bagian skor ini memiliki warna terang ke arah putih. Sehingga perlu adanya threshold yang pas juga agar karakter dapat diproses lebih lanjut.

Tabel 4.2
Tabel *threshold* binerisasi skor

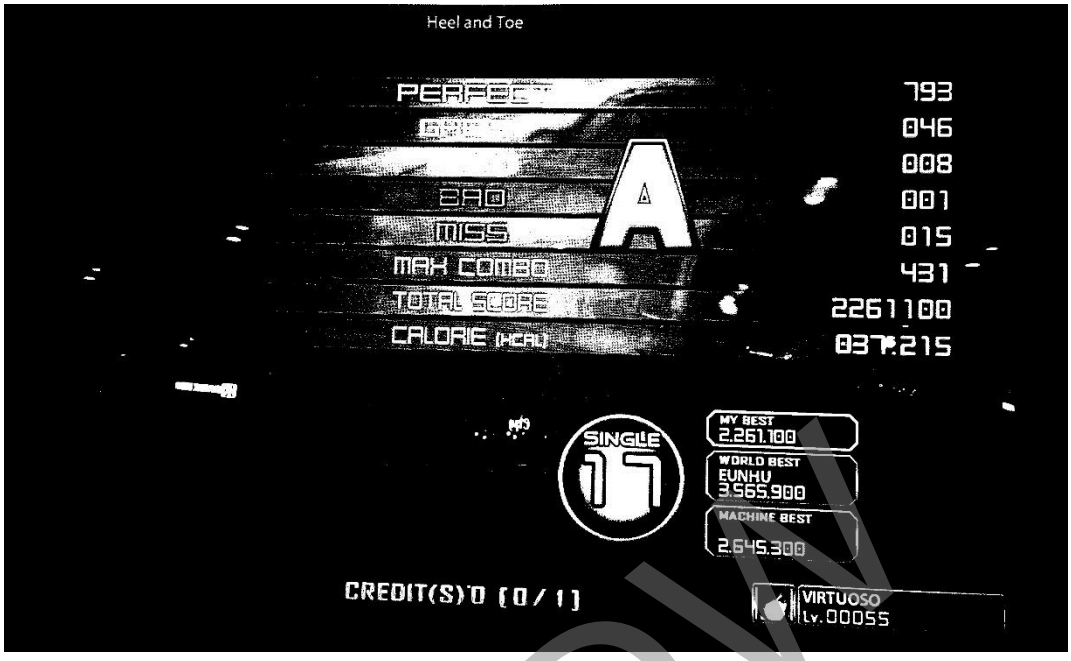
Percobaan	***Threshold***	**Hasil**
1	100	*Foreground* menjadi lebih banyak sehingga mengganggu karakter
2	120	*Foreground* masih terlihat banyak dan mengganggu karakter.
3	150	*Foreground* sudah berkurang tetapi masih mengganggu baris karakter.
4	170	*Foreground* sudah berkurang dan sudah tidak mengganggu baris karakter.
5	180	*Foreground* sudah berkurang dan sudah tidak mengganggu baris karakter.
6	190	*Foreground* sudah berkurang dan sudah tidak mengganggu baris karakter tetapi sudah agak tipis.



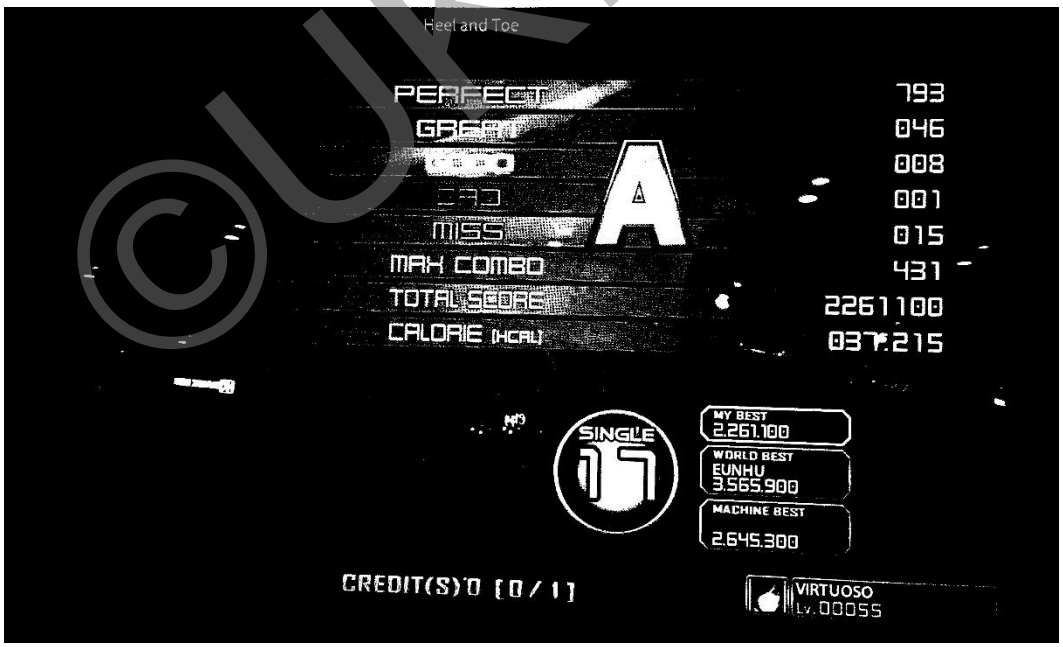
Gambar 4.5 Percobaan skor *threshold* 100



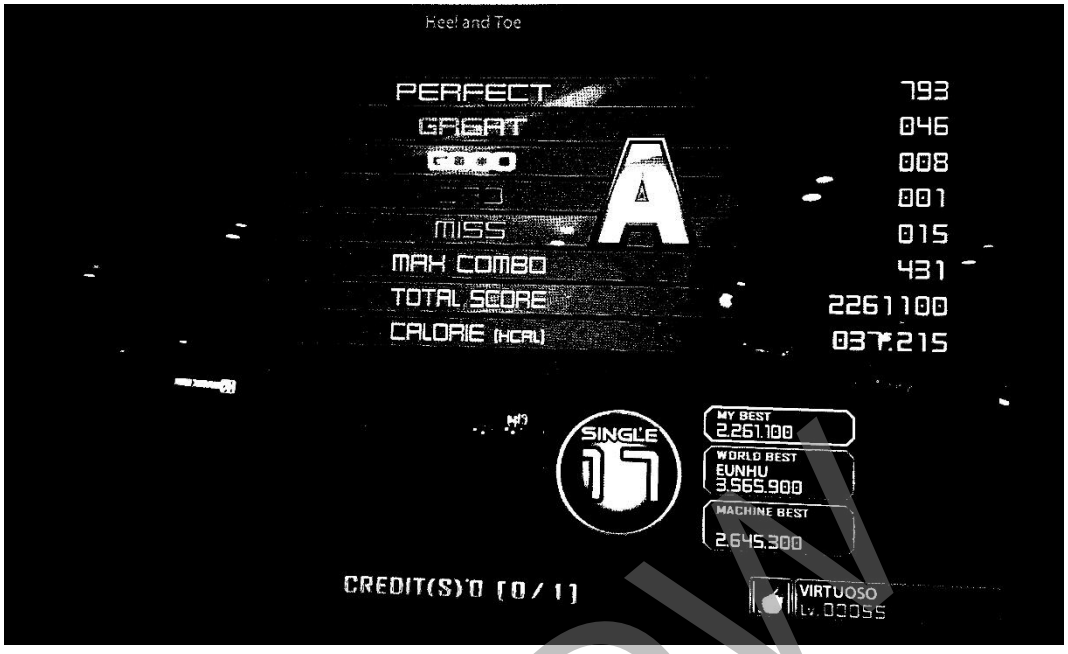
Gambar 4.6 Percobaan skor *threshold* 120



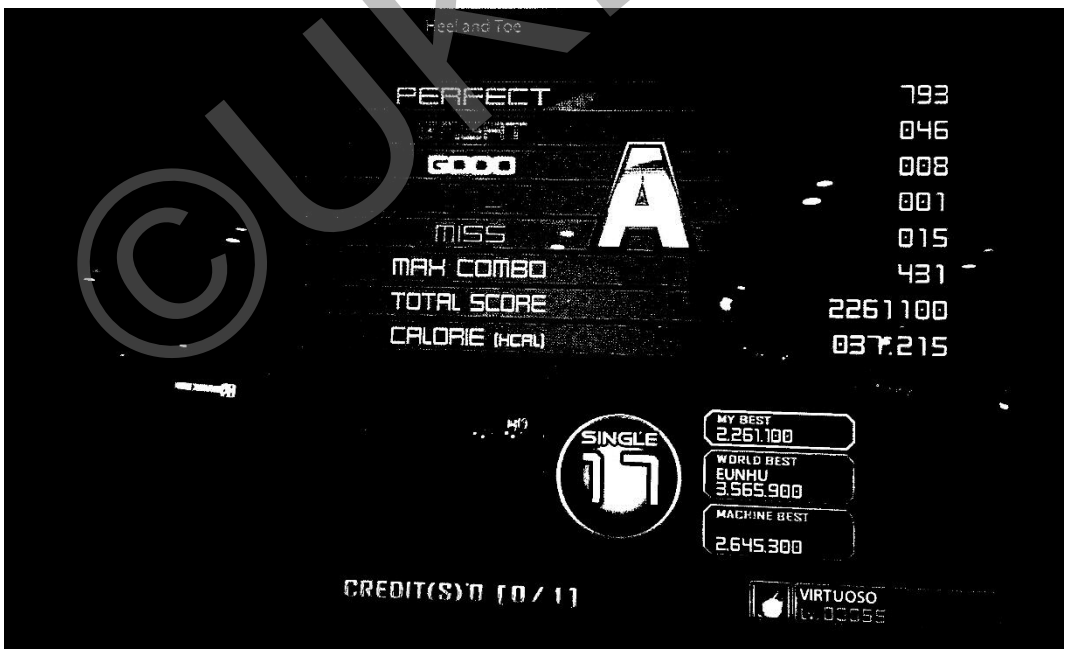
Gambar 4.7 Percobaan skor *threshold* 150



Gambar 4.8 Percobaan skor *threshold* 170



Gambar 4.9 Percobaan skor *threshold* 180



Gambar 4.10 Percobaan skor *threshold* 190

Dari beberapa percobaan yang telah dilakukan, seperti pada gambar 4.5 jumlah *foreground* masih terlalu banyak, gambar 4.6 jumlah *foreground masih agak banyak* , gambar 4.7 jumlah *foreground sudah berkurang tetapi masih menganggu*, gambar 4.8 jumlah *foreground sudah bagus*, gambar 4.9 jumlah *foreground* sama dengan gambar 4.8 dan gambar 4.10 jumlah *foreground sudah mulai berkurang. Sehingga threshold* binerisasi pada citra bagian skor menggunakan *threshold* sebesar 180. Karena pada *threshold* 180 bagian skor sudah lebih bersih sehingga menjadi lebih akurat dan hasilnya akan menjadi optimal.

4.1.1.2 HSV threshold

Citra yang diproses tidak lepas dari adanya *noise* cahaya. Apabila hanya menggunakan proses binerisasi saja maka hasilnya tidak akan optimal karena *threshold grayscale* ke biner cukup sederhana. Oleh sebab digunakanlah proses HSV sehingga *noise* yang berupa pantulan cahaya lampu tersebut dapat diidentifikasi dan diproses agar tidak mempengaruhi hasil dari sistem ini. Untuk mengetahui *threshold* HSV yang paling efektif dalam sistem ini, maka dilakukanlah percobaan-percobaan pada nilai *threshold* seperti pada tabel 4.3.

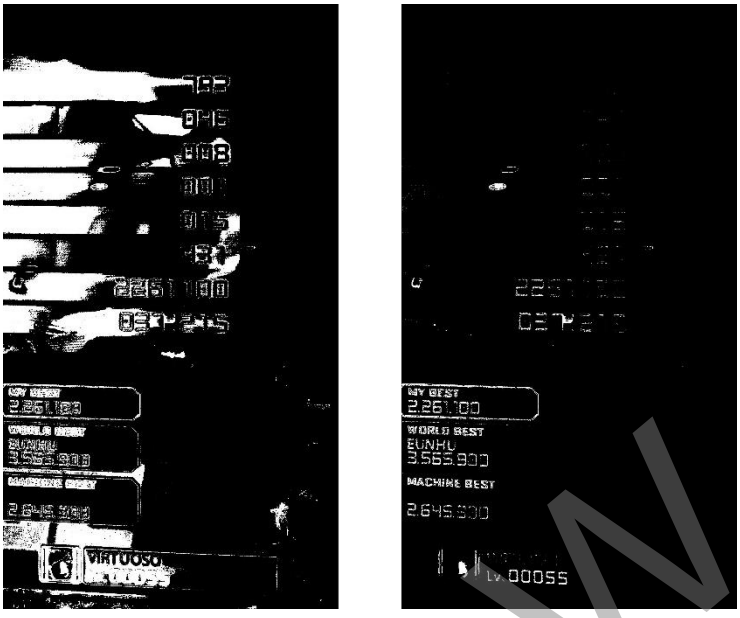
Tabel 4.3
Tabel *threshold* HSV

Percobaan	***Threshold***						**Hasil**
	H1	**H2**	**S1**	**S2**	**V1**	**V2**	
1	0	0,1	0	0,1	0	0,1	Dari 25 karakter terdapat 22 karakter yang dikenali.
2	0,1	0,5	0,1	0,5	0,1	0,5	Dari 25 karakter tidak ada karakter yang dikenali.
3	0,5	0,9	0,5	0,9	0,5	0,9	Dari 25 karakter tidak ada yang dikenali.

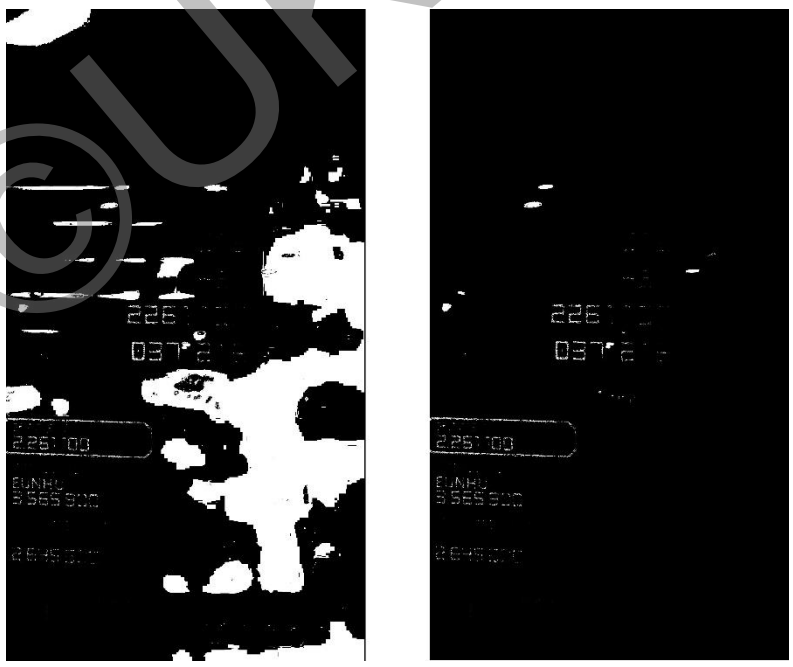
Tabel 4.3
Tabel *threshold* HSV (sambungan)

Percobaan	***Threshold***						**Hasil**
	H1	**H2**	**S1**	**S2**	**V1**	**V2**	
4	0	0,1	0,5	0,9	0,5	0,9	Dari 25 karakter terdapat 25 karakter yang dikenali.
5	0,5	0,9	0	0,1	0,5	0,9	Dari 25 karakter tidak ada karakter yang dikenali.
6	0,5	0,9	0,5	0,9	0	0,1	Dari 25 karakter tidak ada karakter yang dikenali.

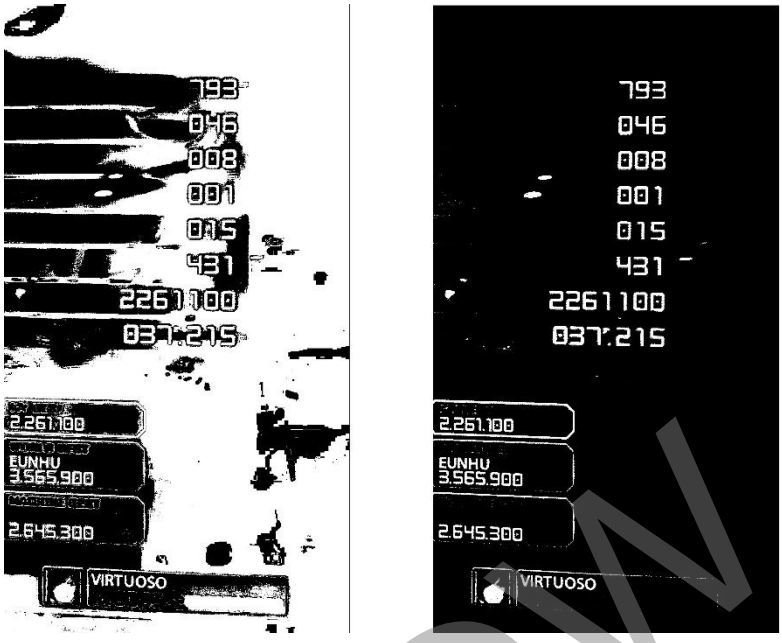
Gambar 4.11 Percobaan 1 sebelum dan setelah operasi "and"



Gambar 4.12 Percobaan 2 sebelum dan setelah operasi “and”

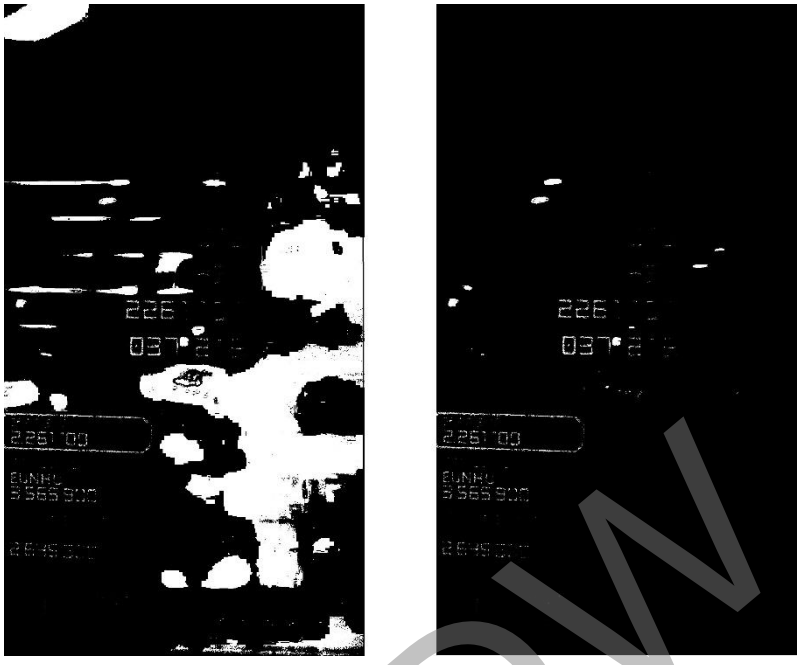


Gambar 4.13 Percobaan 3 sebelum dan setelah operasi “and”



Gambar 4.14 Percobaan 4 sebelum dan setelah operasi “and”

Gambar 4.15 Percobaan 5 sebelum dan setelah operasi “and”



Gambar 4.16 Percobaan 6 sebelum dan setelah operasi “and”

Dari percobaan yang telah dilakukan seperti pada gambar 4.11 ada *foreground* penting yang hilang, gambar 4.12 ada banyak *foreground* yang hilang, gambar 4.13 ada banyak *foreground* yang hilang, gambar 4.14 semua *foreground* terlihat, gambar 4.15 ada banyak *foreground* yang hilang dan gambar 4,16 ada banyak *foreground* yang hilang. Maka *threshold* HSV citra menggunakan theshold seperti pada percobaan ke-4. Hal ini berdasarkan hasil perbandingan dengan percobaan lain hasilnya tampak tidak dapat diproses lebih lanjut.

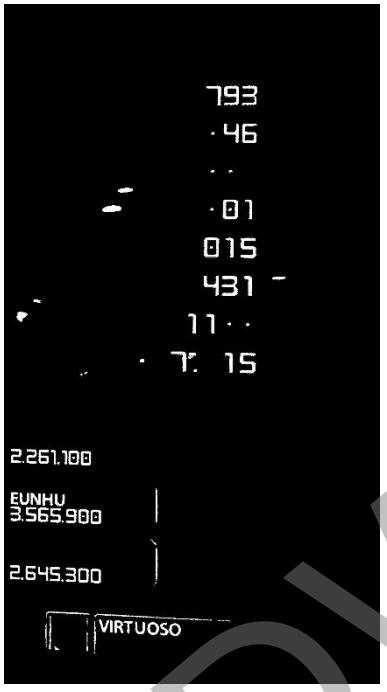
4.1.1.3 Regionprops

Regionprops merupakan *syntax* dalam matlab untuk mengekstraksi area. Proses regionprops pada tahap ini dimaksudkan untuk menghilangkan area yang tidak mengandung informasi skor. Tabel 4.4 merupakan daftar percobaan untuk mengetahui nilai minimal dan maksimal yang optimal dalam sistem ini. Selain untuk menghilangkan area proses ini juga dapat digunakan untuk mengekstraksi informasi sebelum masuk kedalam tahap *resize.*

Tabel 4.4
Tabel area *regionprops* untuk menghilangkan *noise*

Percobaan	Area minimal	Area maksimal	Hasil
1	50	1000	Dari 25 karakter hanya dikenali 4 karakter saja.
2	50	2000	Dari 25 karakter hanya dikenali 15 karakter saja.
3	100	2000	Dari 25 karakter hanya dikenali 15 karakter saja.
4	200	3000	Dari 25 karakter dikenali 25 karakter.

Gambar 4.17 Hasil *regionprops* percobaan 1



Gambar 4.18 Hasil *regionprops* percobaan 2

Gambar 4.19 Hasil *regionprops* percobaan 3

Gambar 4.20 Hasil *regionprops* percobaan 4

Dari beberapa percobaan yang telah dilakukan, pada gambar 4.17 banyak karakter yang hilang, pada gambar 4.18 masih ada beberapa karakter yang hilang, pada gambar 4.19 hasilnya sama seperti gambar 4.18, dan gambar 4.20 hasilnya semua karakter tidak ada yang hilang. Sehingga area *regionprops* citra menggunakan batas - batas area seperti pada percobaan ke-3. Hal ini berdasarkan hasil perbandingan dengan percobaan lain yang secara kasat mata citra hasil mengandung informasi skor.

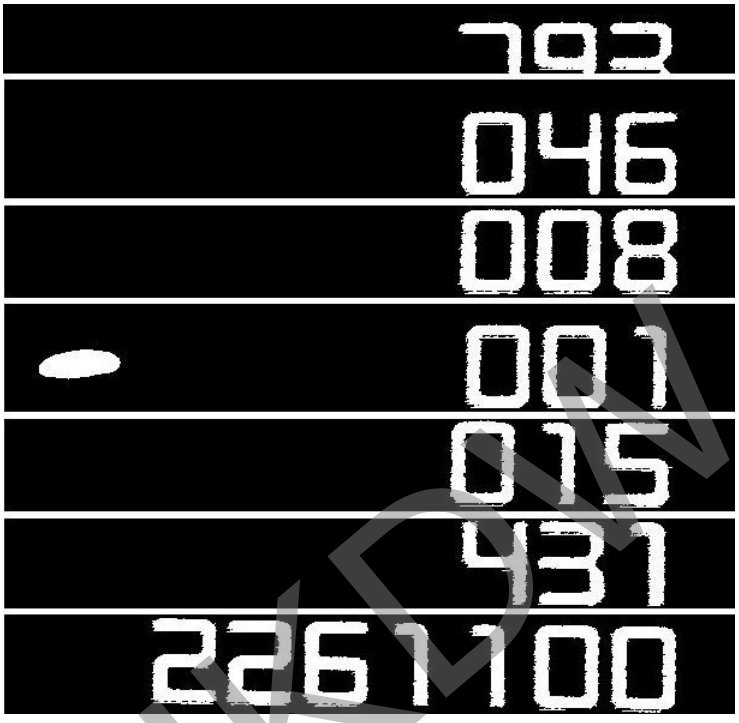
4.1.1.4 Histogram

Setelah citra hanya mengandung informasi skor, maka tahap selanjutnya adalah mengekstraksi baris citra yang dianggap mengandung informasi skor. Proses analisa histogram dalam sistem ini dilakukan untuk ekstraksi area baris yang dianggap mengandung karakter. Tabel 4.5 berisi daftar percobaan jumlah piksel yang dianggap sebagai karakter. Pengecekan dilakukan terhadap jumlah piksel secara horizontal dan vertikal.

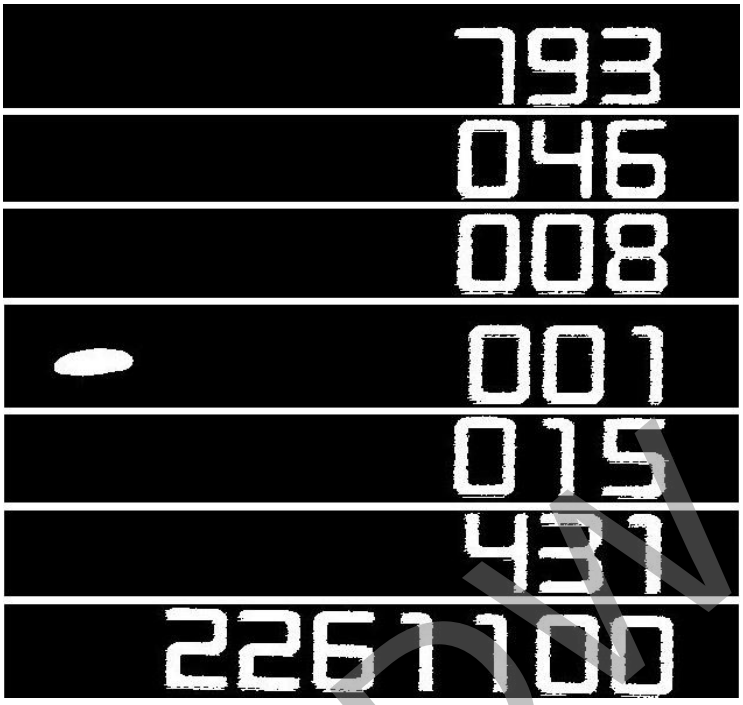
Tabel 4.5
Tabel ukuran histogram pemotongan perbaris

Percobaan	Jumlah Piksel Horizontal		Hasil
	Batas mulai	Batas akhir	
1	> 35	< 36	Dari 25 karakter hanya 22 karakter yang dapat terpotong dengan baik.
2	> 25	< 26	Dari 25 karakter hanya 22 karakter yang dapat terpotong dengan baik.
3	> 15	< 16	Semua karakter dapat terpotong dengan baik.
4	> 10	< 11	Semua karakter dapat terpotong dengan baik.

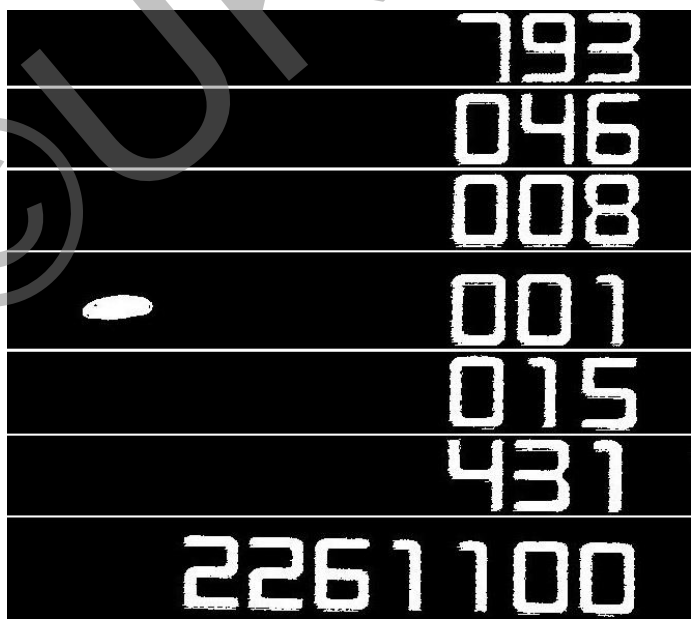
Gambar 4.21 Hasil pemotongan histogram percobaan 1



Gambar 4.22 Hasil pemotongan histogram percobaan 2



Gambar 4.23 Hasil pemotongan histogram percobaan 3



Gambar 4.24 Hasil permotongan histogram percobaan 4

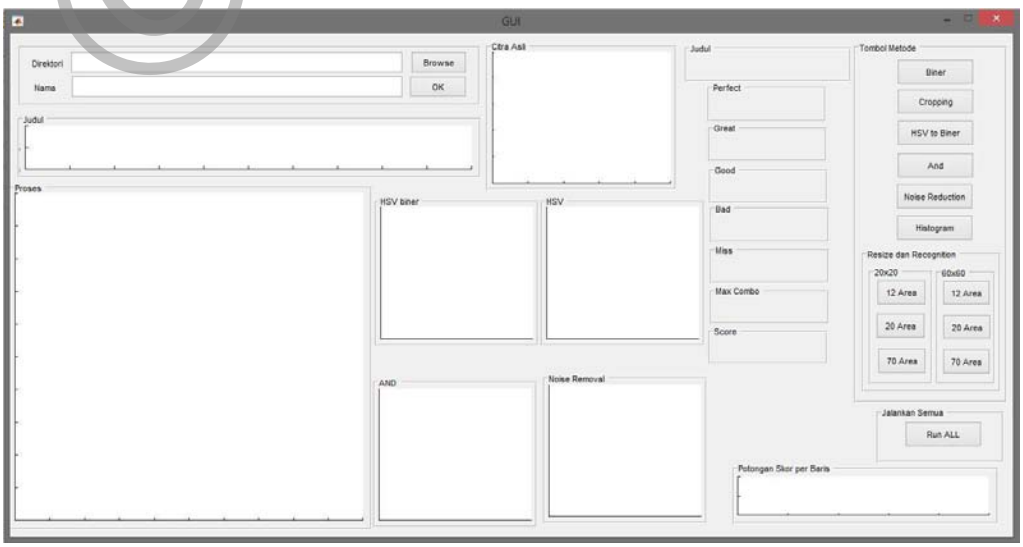
Pada beberapa percobaan yang dilakukan, seperti pada gambar 4.21 ada dua baris yang tidak jelas, pada gambar 4.22 ada satu baris yang tidak jelas, gambar 4.23 semua perpotongannya bagus, gambar 4.24 semua perpotongannya bagus. Sehingga untuk pemotongan menggunakan histogram menggunakan seperti pada percobaan 3.

4.1.1.5 Resize

Tahap terakhir sebelum pengenalan karakter adalah merubah ukuran citra yang bertujuan untuk normalisasi. Ukuran citra yang sudah terpotong diseragamkan agar proses perbandingan dengan template ditahap berikutnya berjalan dengan baik. berdasarkan ukuran template yang sudah dibuat, maka ukuran optimal adalah 13 untuk ukuran template 20 dan 39 untuk ukuran template 60.

4.1.1.6 Antarmuka

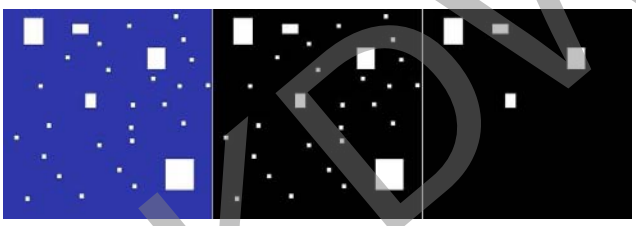
Sistem yang dijalankan menggunakan antarmuka seperti gambar 4.24. Sistem akan dapat mengambil gambar melalui tombol *browse* dan dapat langsung diproses hanya dengan menekan tombol tersedia. Apabila citra sudah selesai diproses maka hasilnya dari pencocokkan tiap karakternya akan langsung di munculkan.



Gambar 4.25 Hasil antarmuka sistem

4.1.2 Validasi Sistem

Sistem dibuat untuk mencatat hasil skor maupun judul permainan PIU sehingga perlu untuk memproses sebuah citra permainan agar dapat memperoleh hasil yang benar. Yang pertama dilakukan adalah memasukkan citra untuk diproses kemudian hilangkan *noise* yang ada agar hasil dari pemrosesannya menjadi lebih baik. Setelah itu citra bisa langsung diekstraksi apabila informasi yang diperlukan tidak ada pengelompokan. Seperti gambar 4.25 adalah contoh penghilangan noise.



Gambar 4.26 Proses validasi menghilangkan noise

Setelah itu yang dilakukan adalah mengekstraksi gambar yang telah dibersihkan dari *noise*. Proses ekstraksi dilakukan untuk mendapatkan daerah atau karakter tertentu. Setiap karakter yang telah diambil kemudian diresize menjadi ukuran yang diharapkan. Tahap terakhir adalah di *recognize* yaitu agar karakter yang sudah di proses dapat dikenali olek komputer seperti pada gambar 4.26.

Gambar 4.27 Hasil validasi pencocokan

4.2 Analisis Pengujian Sistem

Bagian ini memuat daftar hasil pengujian sistem. Sistem ini diuji dengan menggunakan 30 foto skor yang berbeda. Seluruh foto tersebut satu persatu diunggah ke sistem untuk menguji ukuran template dan jumlah potongan optimum yang diperlukan untuk proses pengenalan karakter menggunakan metode *classification-based template matching.* Pengujian ini dibagi menjadi 2 bagian yaitu pengujian ukuran *template* 20x20*px* dan 60x60px. Sedangkan jumlah potongan bagian masing-masing template diuji dengan jumlah potongan 70, 20, dan 12 bagian.

4.2.1 Pengujian Template 20x20

Pada pengujian ini, data uji dilakukan pengenalan karakter dengan tiga kelompok daerah potong masing-masing dicocokkan terhadap *template* berukuran 20x20*px*. Dalam tabel 4.6 disertakan total karakter tiap gambar uji beserta jumlah hasil pengenalan karakter yang benar untuk setiap pengelompokan daerah potong. Pada akhir tabel juga disertakan nilai rata-rata persentase ketepatan pengenalan karakter.

Tabel 4.6
Tabel pengujian citra menggunakan ukuran *template* 20x20

No.	Nama	Daerah Potong

	Gambar	70			20			12		
		Jml karakter	Jml dikenali	%	Jml karakter	Jml dikenali	%	Jml karakter	Jml dikenali	%
1.	1.jpg	35	34	97,14	36	32	88,89	36	34	97.14
2.	2.jpg	32	31	96,88	32	30	93,75	32	28	87.5
3.	3.jpg	32	32	100	33	32	96,97	33	28	87.5
4.	4.jpg	31	31	100	33	31	93,94	33	27	87.1
5.	5.jpg	35	33	94,29	36	32	88,89	36	28	80
6.	6.jpg	28	28	100	28	27	96,43	28	27	96.43
7.	7.jpg	32	32	100	33	31	93,94	33	20	62.5
8.	8.jpg	33	32	96,97	34	31	91,18	34	25	78.13
9.	9.jpg	31	29	93,55	31	28	90,32	31	28	90.32
10.	10.jpg	33	32	96,97	34	33	97,06	34	31	93.94
11.	11.jpg	30	30	100	31	29	93,55	31	25	83.33
12.	12.jpg	32	32	100	33	31	93,94	33	18	56.25
13.	13.jpg	33	33	100	34	31	91,18	34	28	84.85
14.	14.jpg	30	29	96,67	31	30	96,77	31	23	76.67
15.	15.jpg	32	29	90,63	33	28	84,85	33	16	50
16.	16.jpg	29	29	100	29	28	96,55	29	21	72.41
17.	17.jpg	33	31	93,94	34	31	91,18	34	24	72.73
18.	18.jpg	32	31	96,88	34	31	91,18	34	21	67.74
19	19.jpg	37	30	81,08	38	29	76,32	38	23	63.89
20	20.jpg	29	29	100	30	25	83,33	29	25	86.21
21	21.jpg	33	33	100	35	32	91,43	35	27	81.82
22	22.jpg	34	33	97,06	35	33	94,29	35	25	73.53
23	23.jpg	41	40	97,56	41	39	95,12	41	38	92.68
24	24.jpg	32	29	90,63	34	27	79,41	34	25	78.13
25	25.jpg	27	27	100	28	25	89,28	28	24	88.89
26.	26.jpg	31	27	87,1	32	26	81,25	32	25	78,12
27.	27.jpg	34	27	79,41	35	27	77,14	35	18	51,42
28.	28.jpg	35	32	91,42	36	30	83,33	36	29	80,55
29.	29.jpg	31	30	96,77	32	30	93,75	32	19	59,37
30.	30.jpg	33	31	93,93	35	30	85,71	35	29	82,85
Rata-rata persentase ketepatan		95.63			90.03			76,1		

4.2.2 Pengujian Template 60x60

Pada pengujian ini, data uji dilakukan pengenalan karakter dengan tiga kelompok daerah potong masing-masing dicocokkan terhadap *template* berukuran 60x60px. Dalam tabel 4.7 disertakan total karakter tiap gambar uji beserta jumlah hasil pengenalan karakter yang benar untuk setiap pengelompokan daerah potong. Pada akhir tabel juga disertakan nilai rata-rata persentase ketepatan pengenalan karakter.

Tabel 4.7
Tabel pengujian citra menggunakan ukuran *template* 60x60

No.	Nama Gambar	Daerah Potong								
		70			20			12		
		Jml karakter	Jml dikenali	%	Jml karakter	Jml dikenali	%	Jml karakter	Jml dikenali	%
1.	1.jpg	35	33	94,29	36	34	94,44	36	33	91,67
2.	2.jpg	32	32	100	32	31	96,88	32	32	100
3.	3.jpg	32	32	100	33	32	96,97	33	32	96,97
4.	4.jpg	32	31	96,88	34	30	88,235	34	28	82,35
5.	5.jpg	35	34	97,14	36	32	88,889	36	35	97,22
6.	6.jpg	28	28	100	28	28	100	28	28	100
7.	7.jpg	32	32	100	33	31	93,939	33	31	93,94
8.	8.jpg	32	31	96,88	33	31	93,939	33	30	90,91
9.	9.jpg	31	31	100	31	29	93,548	31	29	93,55
10.	10.jpg	33	32	96,97	34	33	97,059	34	31	91,18
11.	11.jpg	30	30	100	31	27	87,097	31	28	90,32
12.	12.jpg	32	32	100	33	30	90,909	33	30	90,91
13.	13.jpg	33	33	100	34	32	94,118	34	30	88,24
14.	14.jpg	30	30	100	31	28	90,323	31	27	87,1
15.	15.jpg	32	29	90,63	33	27	81,818	33	28	84,85
16.	16.jpg	29	28	96,55	29	29	100	29	28	96,55
17.	17.jpg	33	31	93,94	34	31	91,18	34	30	88,24
18.	18.jpg	32	31	96,88	34	28	82,35	34	30	88,24
19	19.jpg	37	30	81,08	38	29	76,32	38	27	71,05
20	20.jpg	30	28	93,33	31	20	64,52	30	27	90
21	21.jpg	33	33	100	35	30	85,71	34	28	82,35
22	22.jpg	34	33	97,06	35	31	88,57	35	30	85,71
23	23.jpg	41	40	97,56	41	39	95,12	41	38	92,68
24	24.jpg	32	29	90,63	34	29	85,29	32	27	84,38
25	25.jpg	27	27	100	28	25	89,29	28	24	85,71
26.	26.jpg	31	28	90,32	32	27	84,38	32	27	84,38
27.	27.jpg	34	26	76,47	35	24	68,57	35	26	74,29
28.	28.jpg	35	32	91,43	36	29	80,56	35	30	85,71
29.	29.jpg	31	31	100	32	30	93,75	32	28	87,5
30.	30.jpg	33	32	96,97	35	32	91,43	35	31	88,57
Rata-rata persentase ketepatan		95,83			88,84			88,82		

Berdasarkan hasil pengujian dari tabel 4.6 dan tabel 4.7 maka template yang memiliki ketepatan tertinggi adalah dengan menggunakan ukuran 60 dengan potongan area sebanyak 70. Namun, dilihat dari selisih dari penggunaan ukuran 20 dan 60 dengan menggunakan 70 potongan piksel sangat kecil yaitu 0,2.

Sehingga penggunaan template ukuran 20 juga tidak kalah bagus dengan ukuran 60.

4.2.3 Pengujian per tempat Template 20x20

Pada pengujian ini, data uji dilakukan pengenalan karakter dengan tiga kelompok daerah potong masing-masing dicocokkan terhadap *template* berukuran 20x20*px*. Dalam tabel 4.8 dilakukan pengenalan karakter setiap bagian, apabila benar maka bernilai 1 dan jika salah bernilai 0 yang kemudian dirata-ratakan untuk melihat keberhasilan secara keseluruhan.

Tabel 4.8
Hasil perbaris ukuran *template* 20x20

Citra	**Ukuran**	**Judul**	***Perfect***	***Great***	***Good***	***Bad***	***Miss***	***Combo***	**Skor**	**%**
1	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	0	1	1	1	0	62,5
	12	1	1	1	0	1	1	1	0	75
2	70	0	1	1	1	1	1	1	1	87,5
	20	0	0	1	1	1	1	1	1	75
	12	0	1	1	1	0	0	1	1	62,5
3	70	1	1	1	1	1	1	1	1	100
	20	1	1	1	0	1	1	1	1	87,5
	12	0	1	1	0	1	1	0	0	50
4	70	1	1	1	1	1	1	1	1	100
	20	1	1	1	0	1	1	1	1	87,5
	12	0	0	1	0	1	1	0	1	50
5	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	0	1	1	1	1	75
	12	0	0	0	0	1	1	1	0	37,5
6	70	1	1	1	1	1	1	1	1	100
	20	1	1	1	1	1	1	1	1	100
	12	1	1	1	1	1	1	1	1	100
7	70	1	1	1	1	1	1	1	1	100
	20	1	0	1	0	1	1	1	1	75
	12	1	0	1	0	1	0	0	0	37,5
8	70	1	1	1	1	0	1	1	1	87,5
	20	0	1	1	1	0	1	1	1	75
	12	0	1	1	1	0	1	1	1	75
9	70	0	1	1	1	1	1	1	1	87,5
	20	0	0	1	1	1	1	0	1	62,5
	12	0	1	1	1	1	1	1	1	87,5

Tabel 4.8
Hasil perbaris ukuran *template* 20x20 (sambungan 1)

Citra	**Ukuran**	**Judul**	***Perfect***	***Great***	***Good***	***Bad***	***Miss***	***Combo***	**Skor**	**%**

10	70	1	1	1	1	1	1	1	1	100
	20	1	1	1	0	1	1	1	1	87,5
	12	0	1	1	0	1	1	1	1	75
11	70	1	1	1	1	1	1	1	1	100
	20	0	1	1	0	1	1	1	1	75
	12	0	1	1	0	0	1	1	1	62,5
12	70	1	1	1	1	1	1	1	1	100
	20	1	1	1	0	1	1	1	1	87,5
	12	0	1	0	0	0	1	0	0	25
13	70	1	1	1	1	1	1	1	1	100
	20	0	0	1	0	1	1	1	1	62,5
	12	0	1	0	0	1	0	1	0	37,5
14	70	0	1	1	1	1	1	1	1	87,5
	20	1	1	1	1	1	1	1	1	100
	12	0	1	0	1	1	1	1	1	75
15	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	0	1	1	1	1	75
	12	0	1	1	0	1	0	0	0	37,5
16	70	1	1	1	1	1	1	1	1	100
	20	1	1	1	1	1	1	1	1	100
	12	1	0	1	1	1	0	0	1	62,5
17	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	1	1	1	1	1	87,5
	12	0	1	1	0	0	1	0	0	37,5
18	70	1	1	1	0	1	1	1	1	87,5
	20	1	1	1	0	1	1	1	1	87,5
	12	1	1	1	0	1	0	1	0	62,5
19	70	0	1	1	0	1	1	1	1	75
	20	0	1	1	0	1	1	1	1	75
	12	0	1	1	0	0	1	1	0	50
20	70	1	1	1	1	1	1	1	1	100
	20	1	1	1	0	1	1	0	0	62,5
	12	1	1	0	0	1	0	1	1	62,5
21	70	1	1	1	1	1	1	1	1	100
	20	0	1	1	1	1	1	1	1	87,5
	12	0	0	1	0	1	1	0	1	50
22	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	0	1	1	1	1	75
	12	0	1	1	0	1	1	0	0	50
23	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	1	1	1	1	0	75
	12	0	1	1	1	1	1	1	1	87,5
24	70	0	0	1	1	1	1	1	1	75
	20	0	0	1	0	1	1	1	1	62,5
	12	0	0	0	0	1	0	1	1	37,5
25	70	1	1	1	1	1	1	1	1	100
	20	0	1	1	0	1	1	1	1	75
	12	1	1	1	0	1	1	1	0	75

Tabel 4.8
Hasil perbaris ukuran *template* 20x20 (sambungan 2)

Citra	Ukuran	Judul	*Perfect*	*Great*	*Good*	*Bad*	*Miss*	*Combo*	Skor	%
26	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	1	0	1	1	1	75
	12	0	1	1	1	0	1	1	1	75
27	70	0	0	1	1	1	1	1	1	75
	20	0	0	1	0	1	1	1	1	62,5
	12	0	1	1	0	0	0	1	0	37,5
28	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	1	0	1	1	0	62,5
	12	0	1	0	1	0	1	1	1	62,5
29	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	0	0	1	1	1	62,5
	12	0	1	1	0	0	1	0	1	50
30	70	1	1	1	0	1	1	1	1	87,5
	20	1	1	1	0	1	1	1	1	87,5
	12	1	1	1	0	1	1	1	1	87,5
		Rata-rata 70		**92,38**	**Rata-rata 20**		**78,53**	**Rata-rata 12**		**60,3**

4.2.4 Pengujian per tempat Template 60x60

Pada pengujian ini, data uji dilakukan pengenalan karakter dengan tiga kelompok daerah potong masing-masing dicocokkan terhadap *template* berukuran 60x60*px*. Dalam tabel 4.9 dilakukan pengenalan karakter setiap bagian, apabila benar maka bernilai 1 dan jika salah bernilai 0 yang kemudian dirata-ratakan untuk melihat keberhasilan secara keseluruhan.

Tabel 4.9
Hasil perbaris ukuran *template* 60x60

Citra	Ukuran	Judul	*Perfect*	*Great*	*Good*	*Bad*	*Miss*	*Combo*	Skor	%
1	70	0	1	1	1	1	1	1	1	87,5
	20	1	1	1	1	1	1	1	0	87,5
	12	1	1	1	1	1	0	1	0	75
2	70	1	1	1	1	1	1	1	1	100
	20	0	1	1	1	1	1	1	1	87,5
	12	1	1	1	1	1	1	1	1	100
3	70	1	1	1	1	1	1	1	1	100
	20	1	1	1	0	1	1	1	1	87,5
	12	1	1	1	0	1	1	1	1	87,5
4	70	1	1	1	0	1	1	1	1	87,5
	20	1	1	1	0	1	1	1	1	87,5
	12	1	1	1	0	1	1	1	1	87,5

Tabel 4.9
Hasil perbaris ukuran *template* 60x60 (sambungan 1)

Citra	Ukuran	Judul	*Perfect*	*Great*	*Good*	*Bad*	*Miss*	*Combo*	Skor	%

5	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	0	1	1	1	1	75
	12	1	1	1	0	1	1	1	1	87,5
6	70	1	1	1	1	1	1	1	1	100
	20	1	1	1	1	1	1	1	1	100
	12	1	1	1	1	1	1	1	1	100
7	70	1	1	1	1	1	1	1	1	100
	20	1	1	1	0	1	1	0	1	75
	12	0	1	1	0	1	1	1	1	75
8	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	1	0	1	1	1	75
	12	0	0	1	1	0	1	1	1	62,5
9	70	1	1	1	1	1	1	1	1	100
	20	0	0	1	1	1	1	1	1	75
	12	0	0	1	1	1	1	1	1	75
10	70	0	1	1	1	1	1	1	1	87,5
	20	1	1	1	0	0	1	1	1	75
	12	0	1	1	0	0	1	1	1	62,5
11	70	1	1	1	1	1	1	1	1	100
	20	0	0	1	0	1	1	1	1	62,5
	12	0	1	1	0	1	1	1	1	75
12	70	1	1	1	1	1	1	1	1	100
	20	1	1	1	0	1	1	1	1	87,5
	12	1	1	1	0	1	1	1	0	75
13	70	1	1	1	1	1	1	1	1	100
	20	1	1	1	0	1	1	1	1	87,5
	12	0	1	1	0	1	1	1	0	62,5
14	70	1	1	1	1	1	1	1	1	100
	20	0	1	1	1	1	1	1	1	87,5
	12	0	1	1	1	0	0	1	1	62,5
15	70	0	1	1	1	1	1	1	1	87,5
	20	0	0	0	1	1	1	1	1	62,5
	12	0	1	0	1	1	1	1	1	75
16	70	0	1	1	1	1	1	1	1	87,5
	20	1	1	1	1	1	1	1	1	100
	12	0	1	1	1	1	1	1	1	87,5
17	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	0	1	1	1	1	75
	12	0	1	1	0	1	1	1	1	75
18	70	1	1	1	0	1	1	1	1	87,5
	20	0	1	1	0	1	1	1	1	75
	12	0	1	1	0	1	1	1	1	75
19	70	0	1	1	0	1	1	1	1	75
	20	0	1	1	0	1	1	1	1	75
	12	0	1	1	0	1	1	1	1	75
20	70	0	1	1	0	1	1	1	1	75
	20	0	1	1	0	1	1	1	1	75
	12	0	1	1	0	1	1	1	1	75

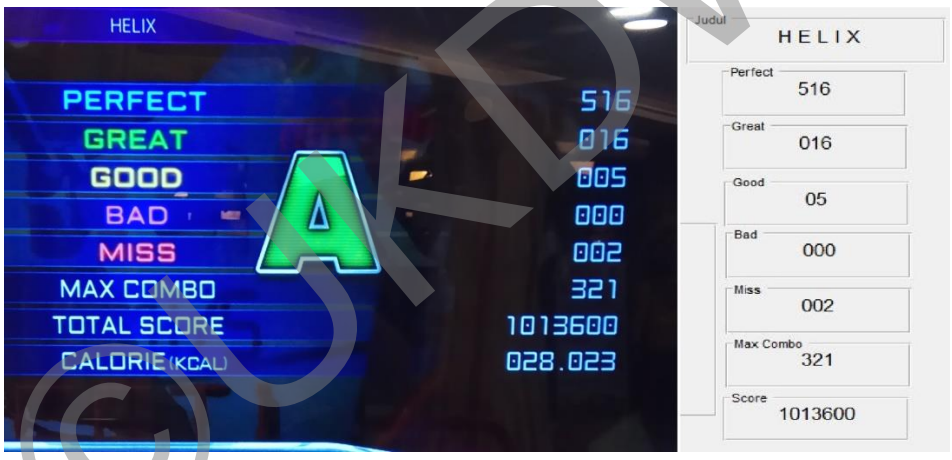
Tabel 4.9
Hasil perbaris ukuran *template* 60x60 (sambungan 2)

Citra	Ukuran	Judul	*Perfect*	*Great*	*Good*	*Bad*	*Miss*	*Combo*	Skor	%
21	70	1	1	1	1	1	1	1	1	100
	20	0	1	1	0	1	1	1	1	75
	12	0	0	1	0	1	1	1	1	62,5
22	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	0	1	1	1	1	75
	12	0	1	1	0	1	1	1	1	75
23	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	1	1	1	1	1	87,5
	12	0	1	1	1	1	1	1	1	87,5
24	70	0	0	1	1	1	1	1	1	75
	20	1	0	0	1	1	1	1	1	75
	12	0	0	0	1	1	1	1	1	62,5
25	70	1	1	1	1	1	1	1	1	100
	20	1	0	0	0	1	1	1	1	62,5
	12	1	0	0	0	1	1	1	0	50
26	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	1	0	1	1	1	75
	12	0	1	1	1	0	1	1	1	75
27	70	0	0	1	1	1	1	1	1	75
	20	0	1	1	0	1	1	1	1	75
	12	0	0	1	0	1	1	1	1	62,5
28	70	0	1	1	1	1	1	1	1	87,5
	20	0	1	1	1	0	1	1	1	75
	12	0	1	1	1	0	1	1	1	75
29	70	1	1	1	1	1	1	1	1	100
	20	0	1	1	0	1	1	1	1	75
	12	1	1	1	0	1	1	1	1	87,5
30	70	1	1	1	0	1	1	1	1	87,5
	20	1	1	1	0	1	1	1	1	87,5
	12	0	1	1	0	1	1	1	1	75
		Rata-rata 70		**90,83**	Rata-rata 20		**79,17**	Rata-rata 12		**75,42**

Berdasarkan hasil pengujian dari tabel 4.8 dan tabel 4.9 maka template yang memiliki ketepatan tertinggi per tempatnya adalah dengan menggunakan ukuran 20 dengan potongan area sebanyak 70, yaitu dengan rata-rata sebesar 92,38. Namun, dilihat dari selisih dari penggunaan ukuran 20 dan 60 dengan menggunakan 70 potongan piksel yaitu 2,55. Sehingga penggunaan template ukuran 60 juga tidak kalah bagus dengan ukuran 60.

4.3 Evaluasi Sistem

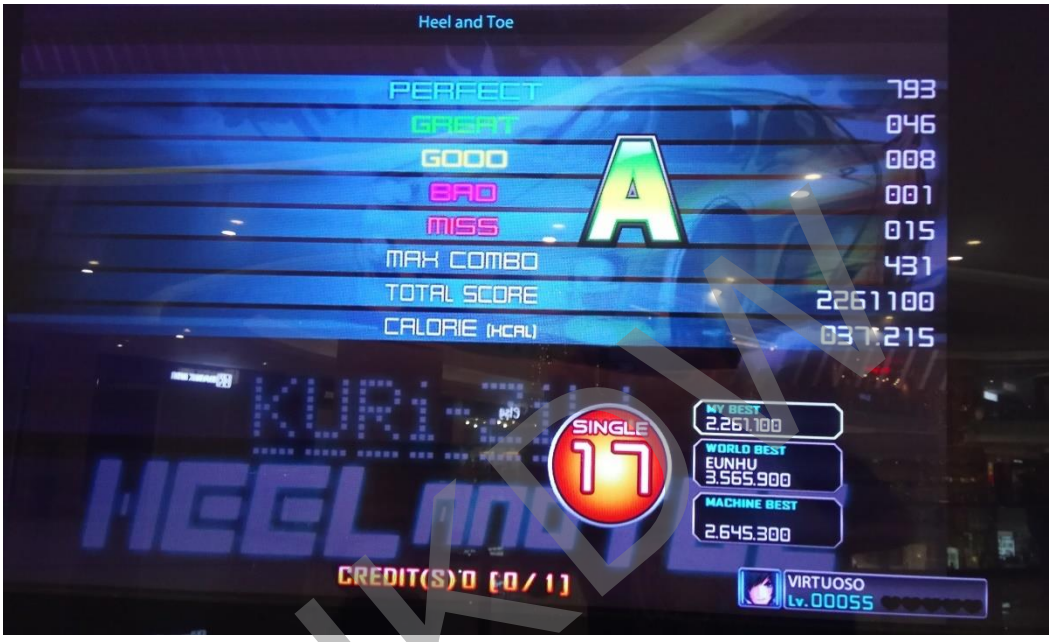
Sistem yang telah dibuat sudah berjalan dengan baik. Sistem dapat mengenali karakter melalui *Template Matching* berbasis Korelasi. Ada beberapa faktor eksternal yang menyebabkan karakter citra menjadi tidak bisa maksimal seperti cahaya yang terlalu terang maupun kurang terang yang menyebabkan warna hasil foto kamera tidak konsisten karena pengaruh *auto focus.* Selain faktor eksternal juga terdapat faktor lain seperti adanya noise yang masih terlewati yang dikenali menjadi sebuah karakter. Hampir semua karakter dapat dikenali dengan benar. Sistem yang dibuat dapat digunakan untuk versi *Pump It Up* yang terbaru, karena memiliki karakter huruf dan angka yang memiliki bentuk yang sama seperti pada gambar 4.24.



Gambar 4.28 Hasil penerapan sistem pada versi lain

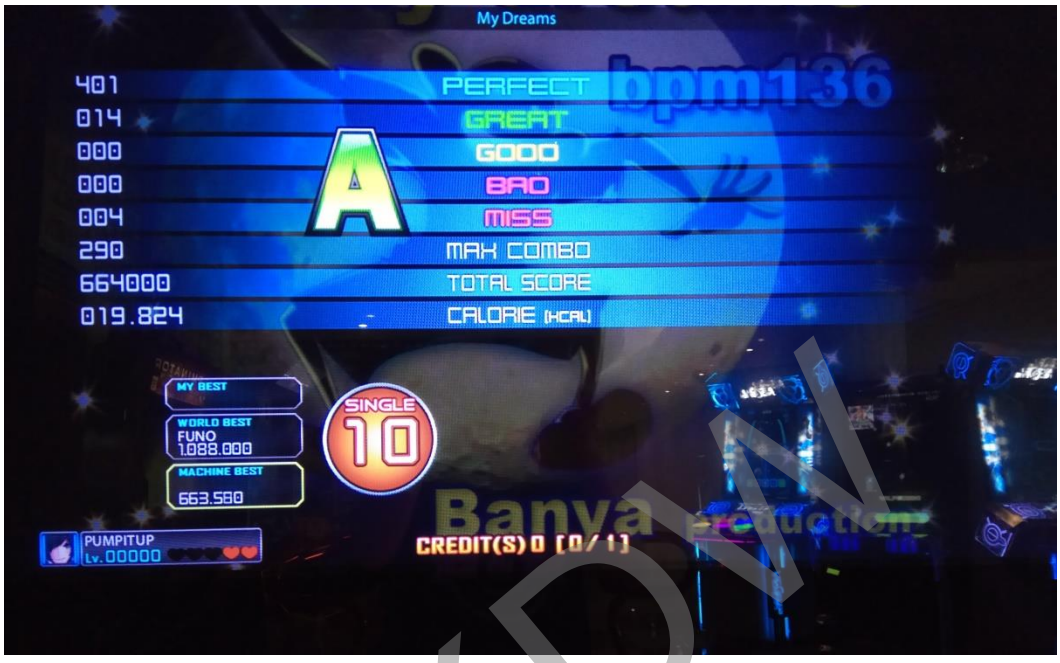
Lampiran A: Gambar Uji

Gambar 1

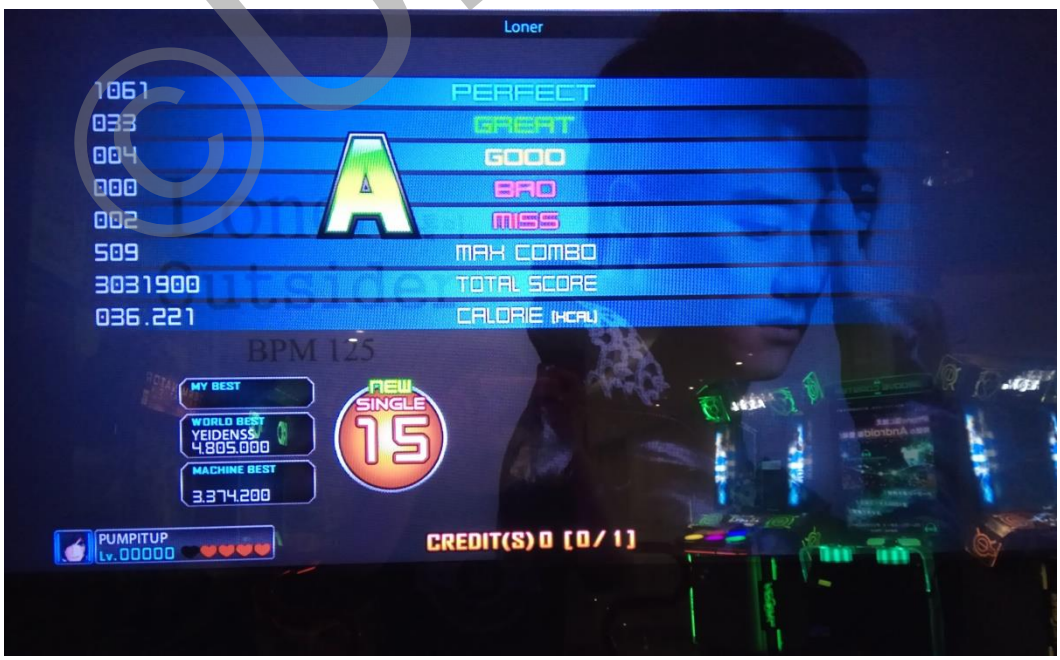


Gambar 2

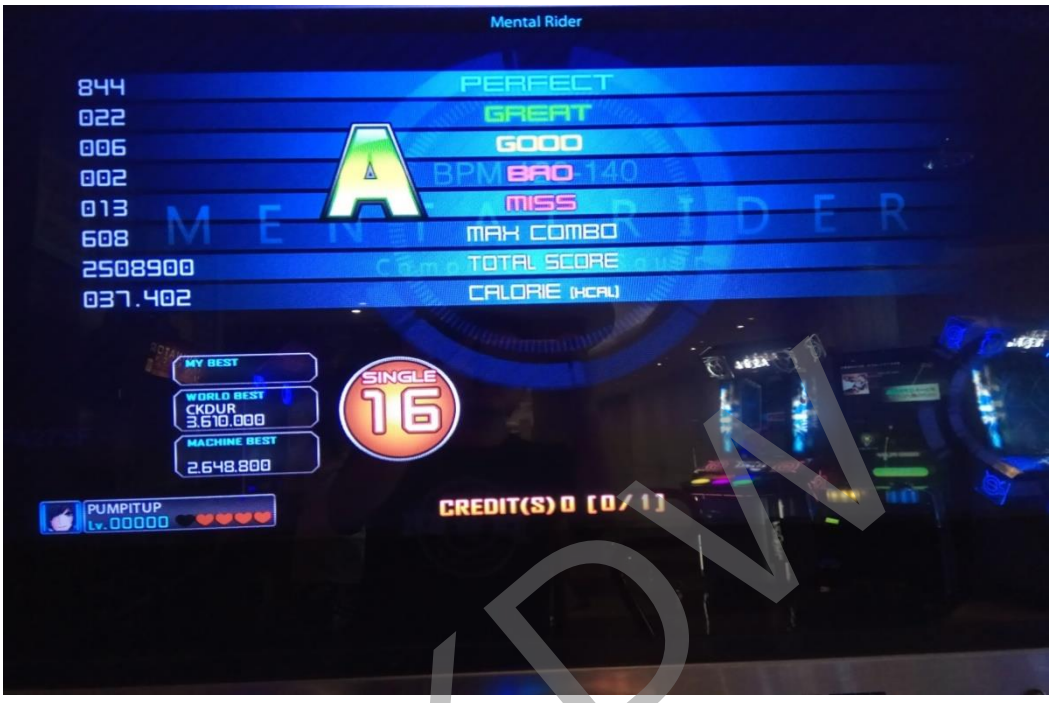
Gambar 3



Gambar 4

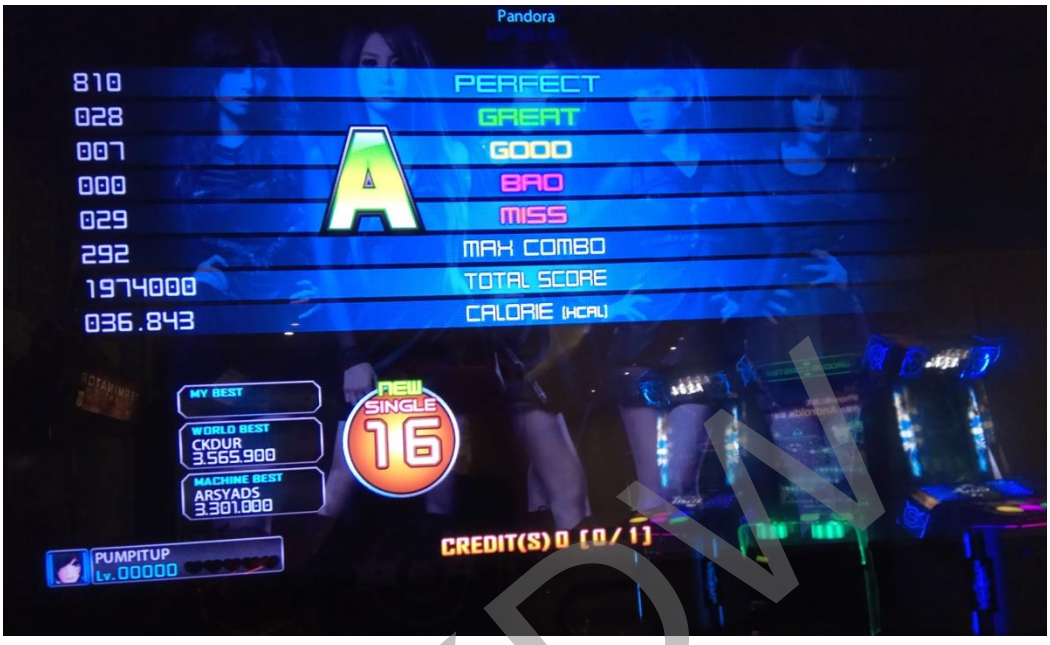


Gambar 5



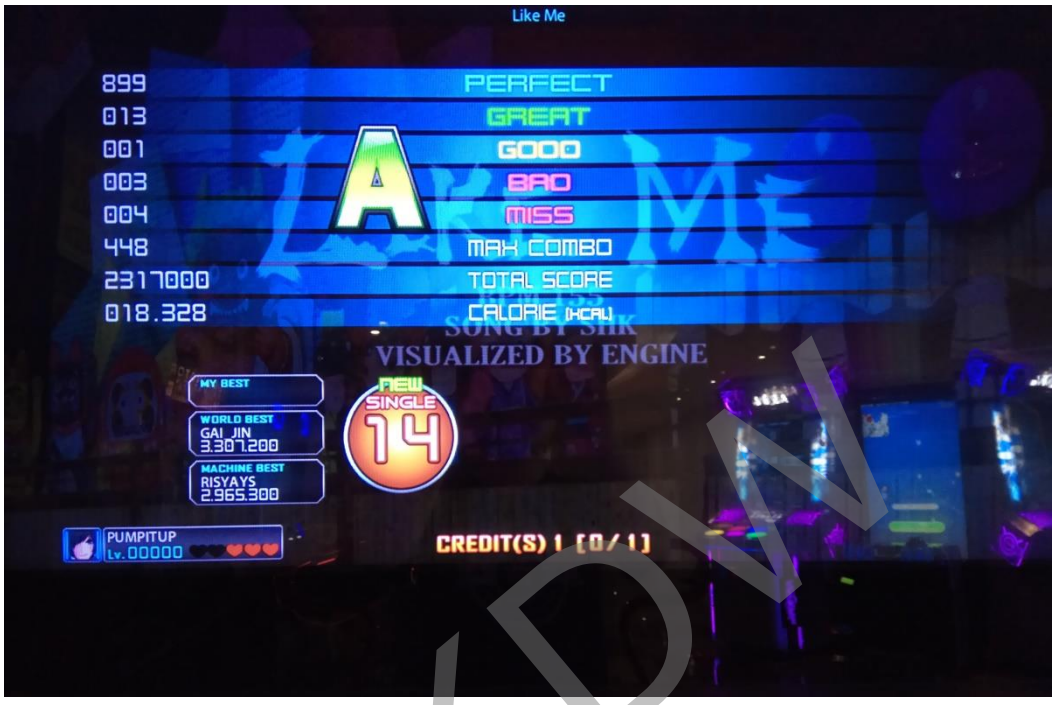
Gambar 6

Gambar 7

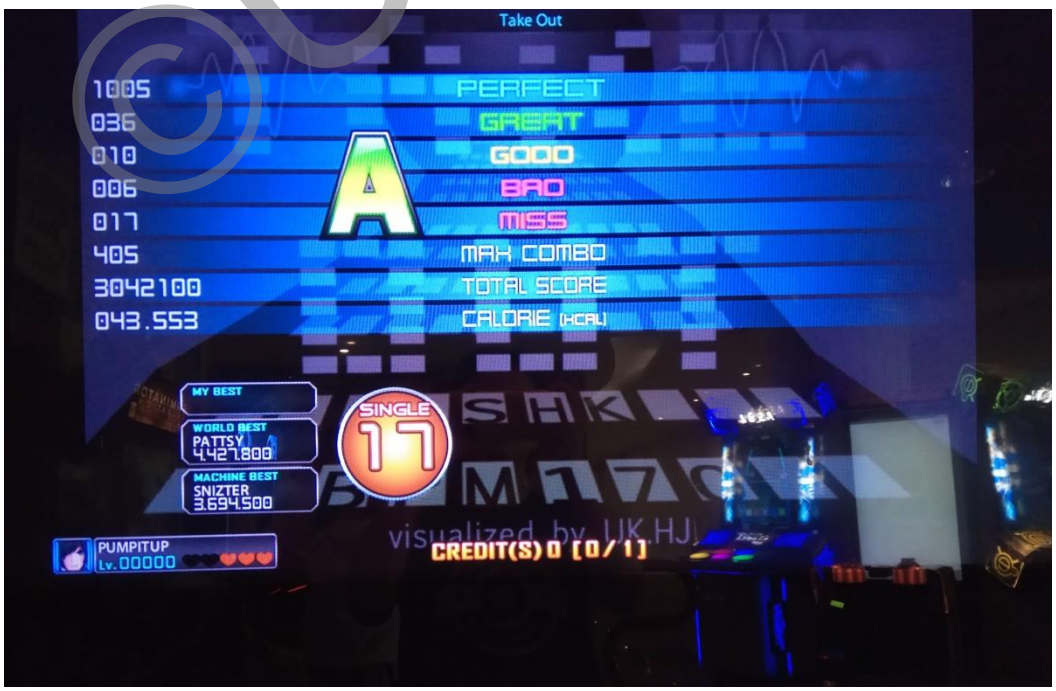


Gambar 8

Gambar 9

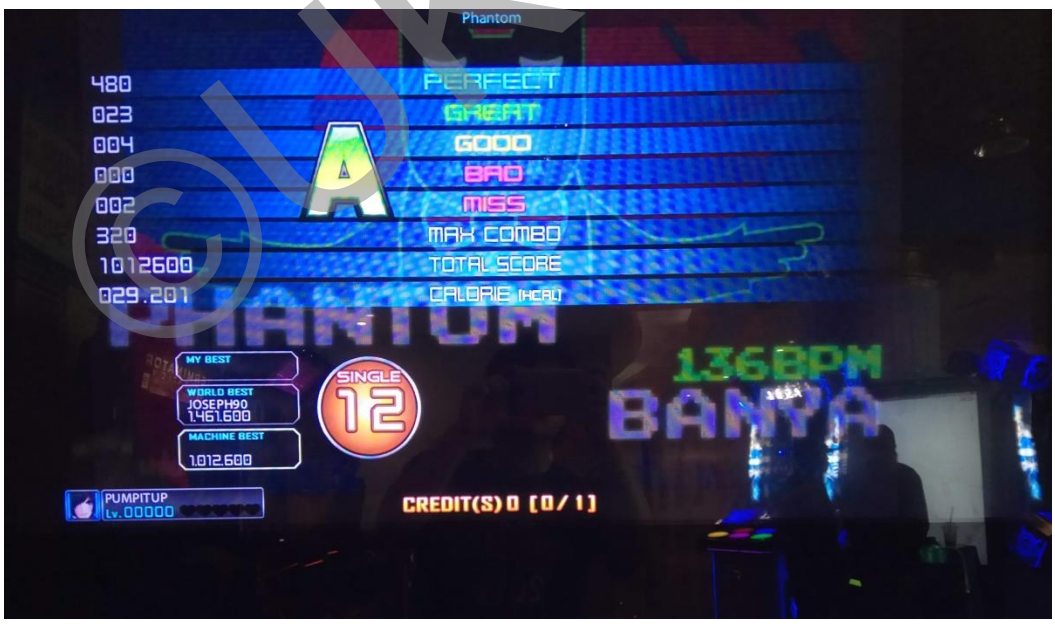


Gambar 10

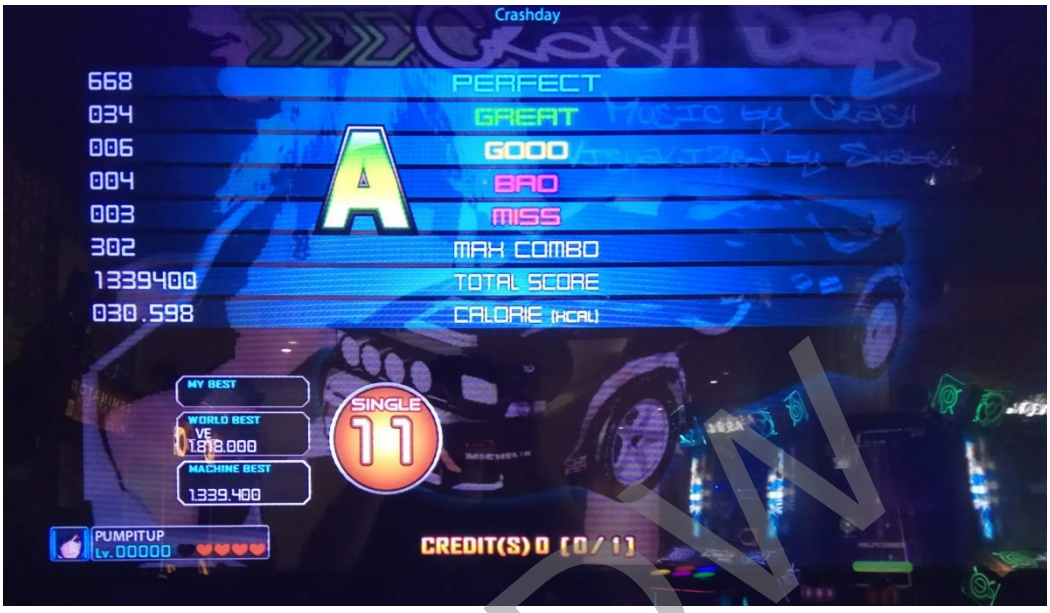


Gambar 11

Gambar 12



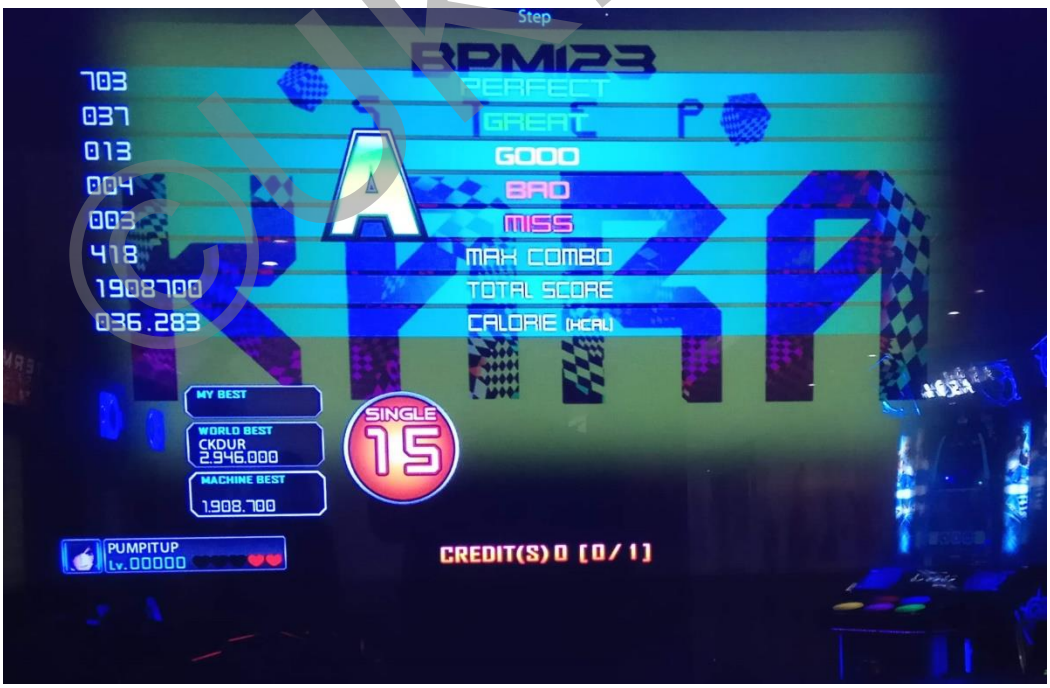
Gambar 13



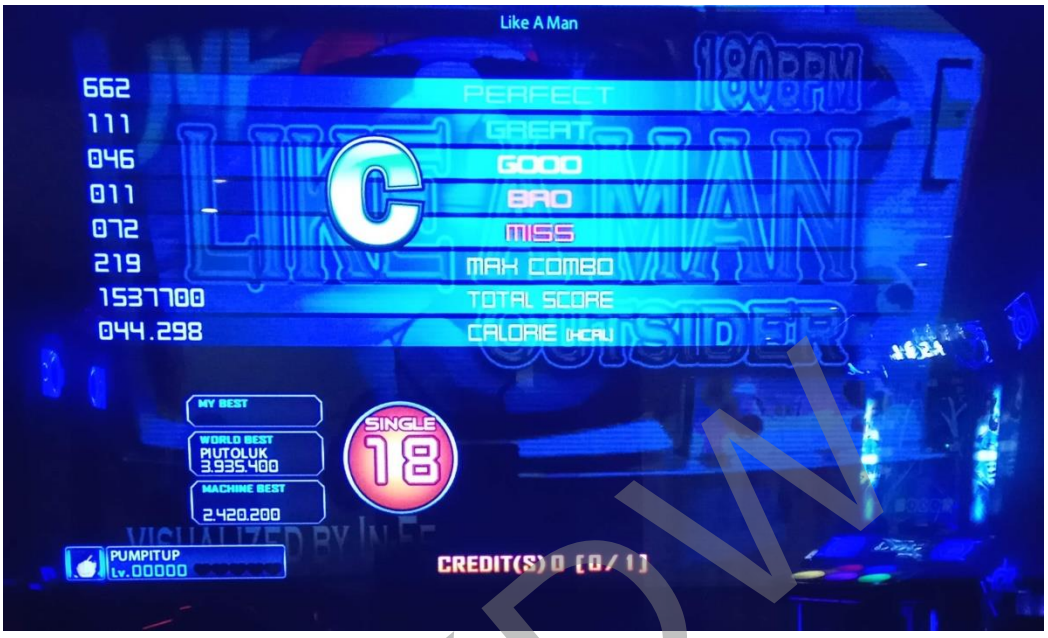
Gambar 14

Gambar 15

Gambar 16



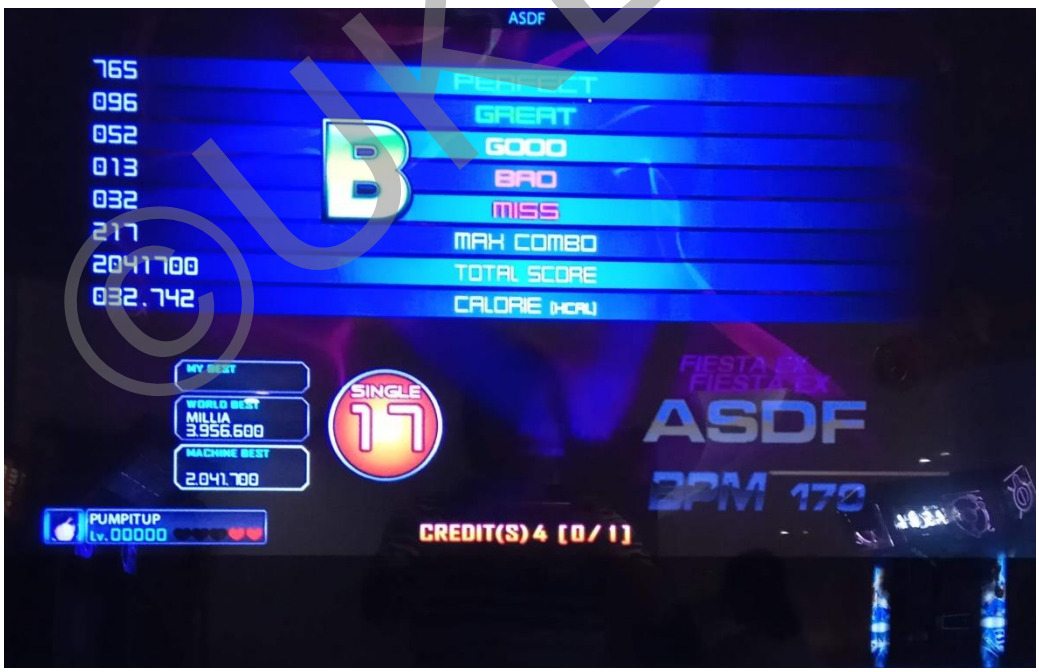
Gambar 17



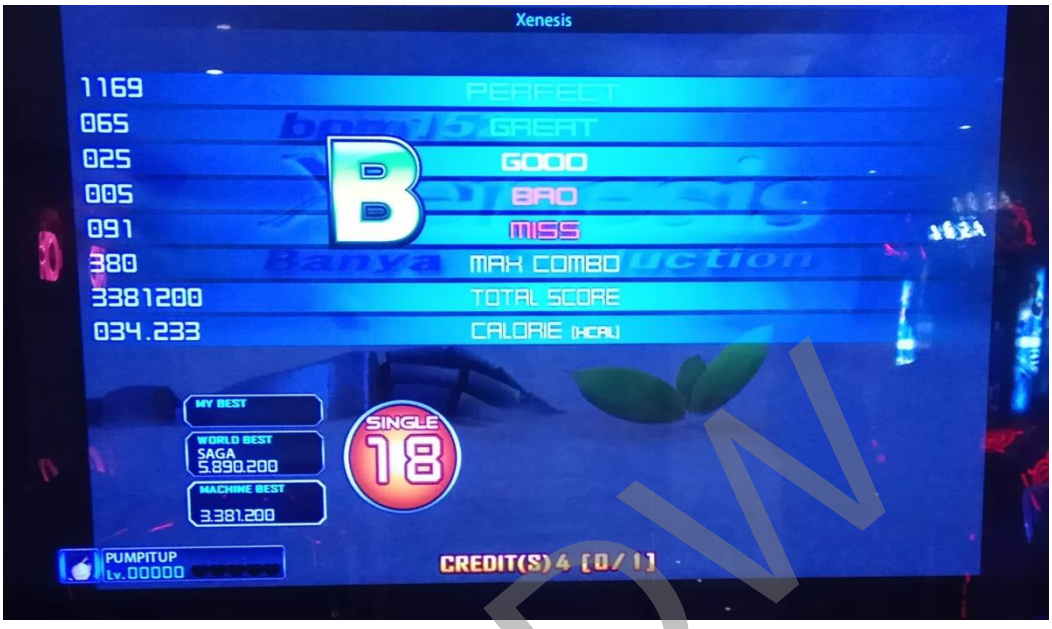
Gambar 18

Gambar 19

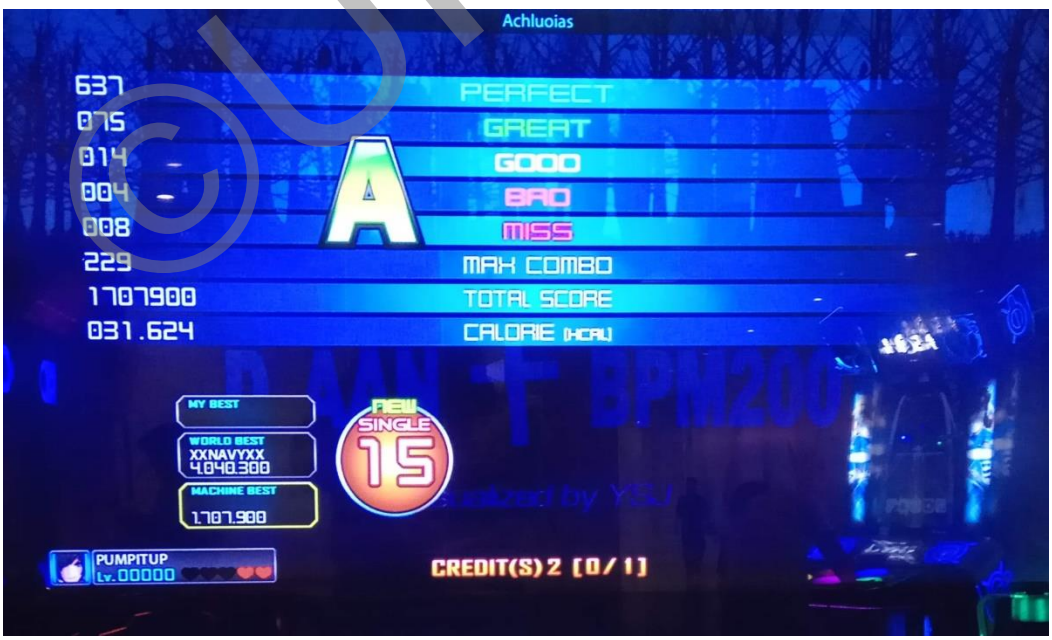
Gambar 20



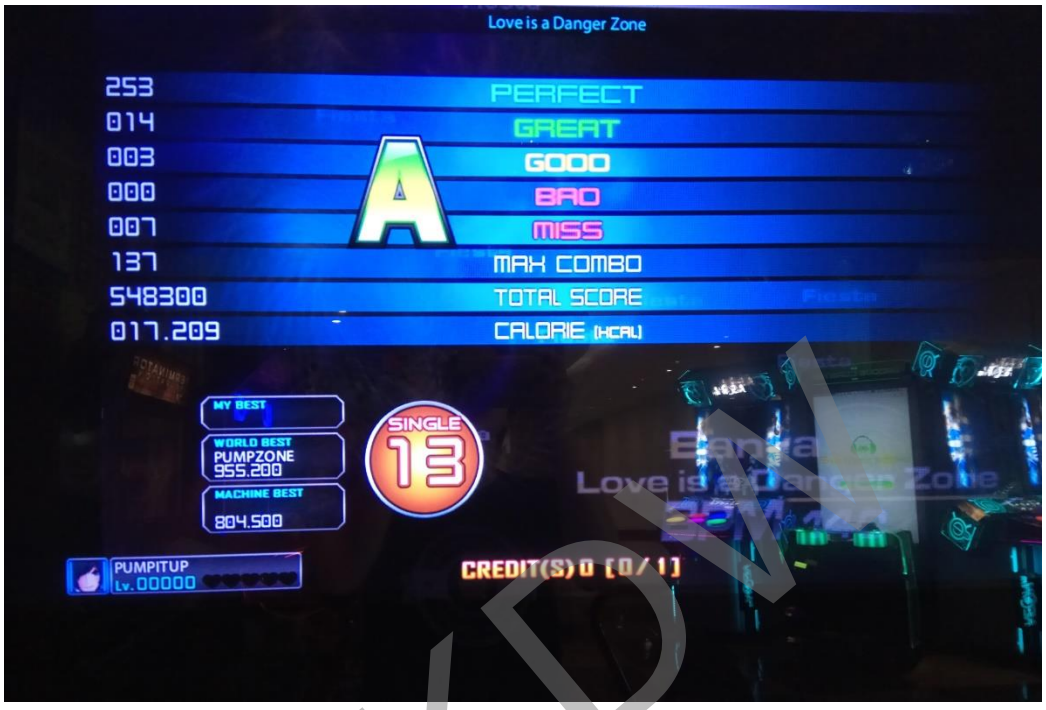
Gambar 21



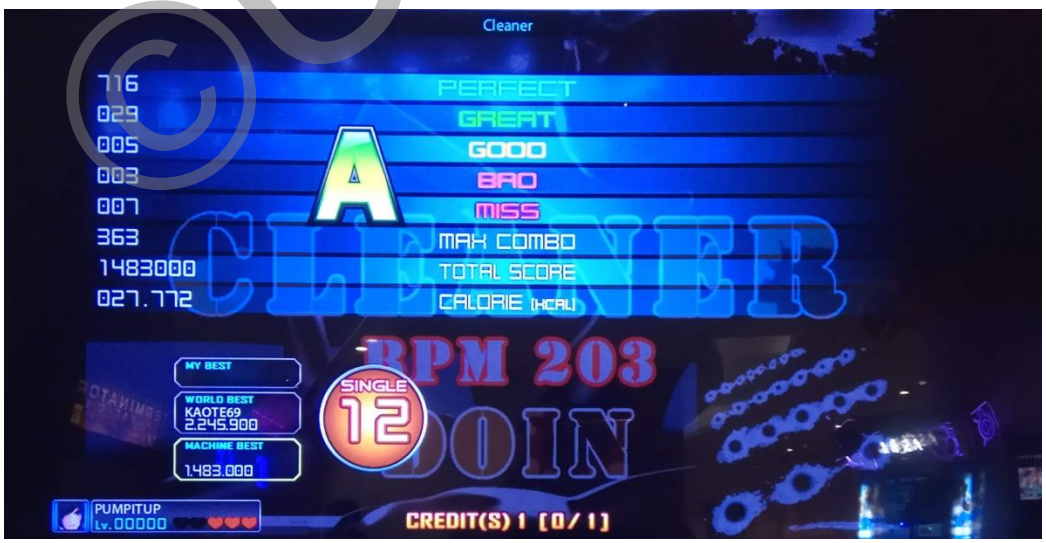
Gambar 22



Gambar 23

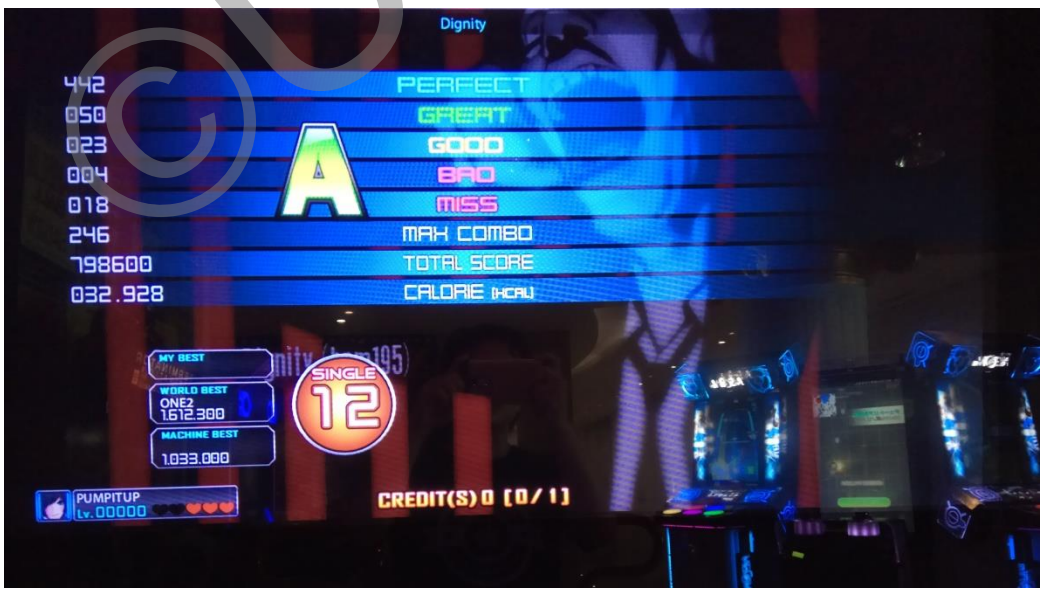


Gambar 24

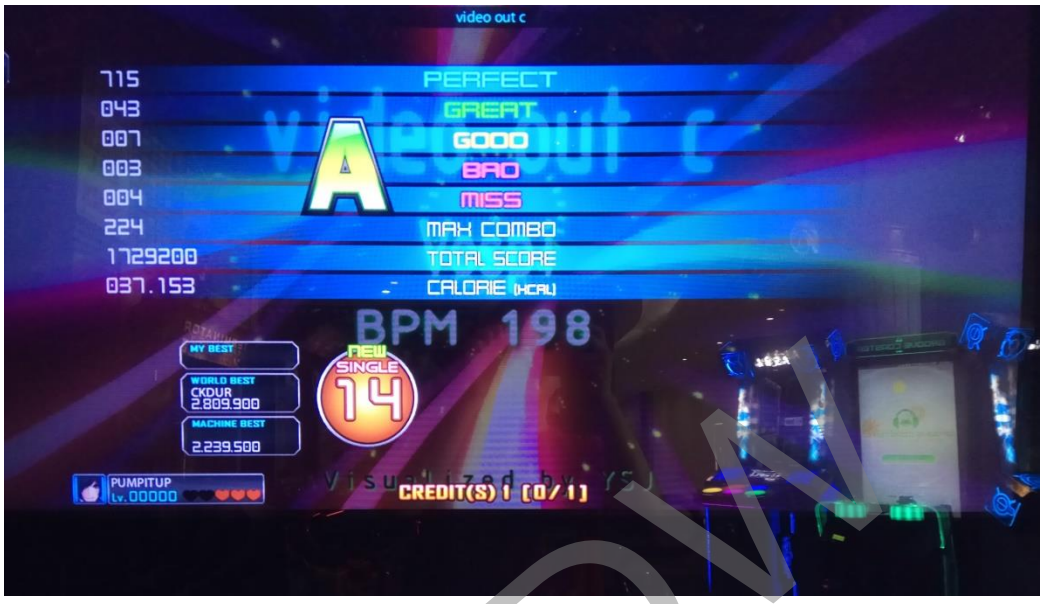


Gambar 25

Gambar 26

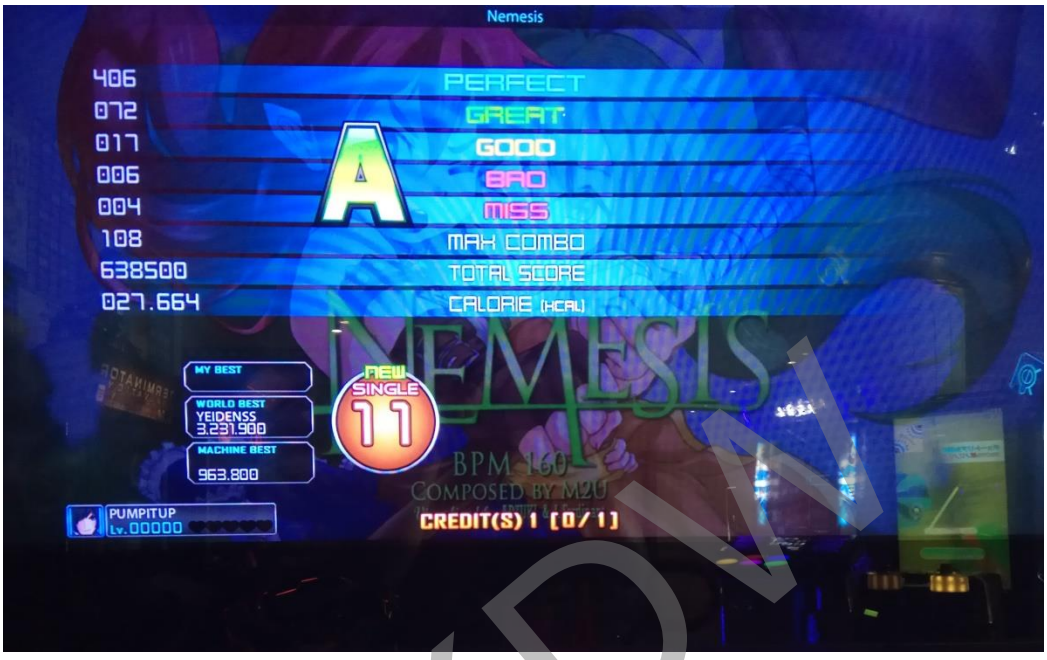


Gambar 27



Gambar 28

Gambar 29



Gambar 30

Lampiran B: Listing Program

biner.m

```
global namaFile;
global gambar;
global gambarSkorBW;
global gambarJudulBW;

%baca gambar
gambar = imread(namaFile);

imshow(gambar,'Parent',handles.axesAsli);

%menentukan w dan h
lebarGambar = size(gambar,2);
tinggiGambar = size(gambar,1);

%potong judul
judul=gambar(1:120,1:end,:);

lebarJudul = size(judul,2);
tinggiJudul = size(judul,1);

%menentukan threshold gray to biner
gambarSkorGray = rgb2gray(gambar);

gambarSkorBW = gambarSkorGray;

for forTinggiSkorBW=1:tinggiGambar;
    for forLebarSkorBW=1:lebarGambar;
            if gambarSkorBW(forTinggiSkorBW,forLebarSkorBW) < 170
                gambarSkorBW(forTinggiSkorBW,forLebarSkorBW) = 0;

            else
                gambarSkorBW(forTinggiSkorBW,forLebarSkorBW) = 255;
            end
    end
end

imshow(gambarSkorBW,'Parent',handles.axesBiner);
imshow(gambarSkorBW,'Parent',handles.axesProses);

gambarJudulBW = rgb2gray(judul);
for forTinggiJudulBW=1:tinggiJudul;
    for forLebarJudulBW=1:lebarJudul;
            if gambarJudulBW(forTinggiJudulBW,forLebarJudulBW) < 120
               gambarJudulBW(forTinggiJudulBW,forLebarJudulBW) = 0;
            else
                gambarJudulBW(forTinggiJudulBW,forLebarJudulBW) = 255;
            end


    end
end
imshow(gambarJudulBW,'Parent',handles.axesJudul);
```

pixelcount.m

```
global gambar;
global gambarSkorBW;
global hasilPixelCount;
global gambarSkorHasilCrop;

pemotong=fix(size(gambar,2)/3);

kiri=gambarSkorBW(:,1:pemotong,:); %BW
kiriAsli=gambar(:,1:pemotong,:); %RGB

kanan=gambarSkorBW(:,(2*pemotong)+1:end,:); %BW
kananAsli=gambar(:,(2*pemotong)+1:end,:); %RGB

%cek pixel
lebarGambarKiri=fix(size(kiri,2));
tinggiGambarKiri=fix(size(kiri,1));

lebarGambarKanan=fix(size(kanan,2));
tinggiGambarKanan=fix(size(kanan,1));

kiri=double(kiri); % perlu digunakan variabel lain?
pixKiri=0;

kanan=double(kanan); % perlu digunakan variabel lain?
pixKanan=0;

for forLebarKiri=1:lebarGambarKiri
    for forTinggiKiri=1:tinggiGambarKiri
        if kiri(forTinggiKiri,forLebarKiri)==0

        else %hitung foreground (255)
            pixKiri=pixKiri+1;
        end
    end
end

for forLebarKanan=1:lebarGambarKanan
    for forTinggiKanan=2:tinggiGambarKanan
        if kanan(forTinggiKanan,forLebarKanan)==0

        else %hitung foreground (255)
            pixKanan=pixKanan+1;
        end
    end
end

if(pixKiri > pixKanan)
    gambarSkorHasilCrop = kiriAsli; %RGB
    hasilPixelCount = kiri; %Biner
    imshow(hasilPixelCount,'Parent',handles.axesProses);
else
    gambarSkorHasilCrop = kananAsli; %RGB
    hasilPixelCount = kanan; %Biner
    imshow(hasilPixelCount,'Parent',handles.axesProses);
end
```

hsvbiner.m

```
global gambarSkorHasilCrop;
global hasilSkorHsv;

gambarSkorHsv = rgb2hsv(im2double(gambarSkorHasilCrop));

hThresh = gambarSkorHsv(:,:,1) > 0 & gambarSkorHsv(:,:,1) < 0.1;
sThresh = gambarSkorHsv(:,:,2) > 0.5 & gambarSkorHsv(:,:,2) < 0.9;
vThresh = gambarSkorHsv(:,:,3) > 0.5 & gambarSkorHsv(:,:,3) < 0.9;

hasilSkorHsv = hThresh | sThresh | vThresh;

%imcomplement
hasilSkorHsv = ~(hasilSkorHsv);

imshow(gambarSkorHsv,'Parent',handles.axesHSV);
imshow(hasilSkorHsv,'Parent',handles.axesProses);
```

and.m

```
global hasilAnd;
global hasilSkorHsv;
global hasilPixelCount;

%Proses AND
hasilAnd = and(hasilSkorHsv,hasilPixelCount);

imshow(hasilAnd,'Parent',handles.axesAND);
imshow(hasilAnd,'Parent',handles.axesProses);
```

reducenoise.m

```
global hasilAnd;
global gambarSkorBW1;
global gambarJudulBW;
global gambarJudulLabel;

%% Judul
gambarJudulLabel=gambarJudulBW;
gambarJudulLabel = bwareaopen(gambarJudulLabel,50);

imshow(gambarJudulLabel,'Parent',handles.axesJudul);

%% Skor
gambarSkorBW=hasilAnd;

gambarSkorBW = bwareaopen(gambarSkorBW,200);

[G1, bykLabel1]=bwlabel(gambarSkorBW);
```

```
propied=regionprops(G1,'Area');
area_values = [propied.Area];
idx = find((200 <= area_values) & (area_values <= 3000));

gambarSkorBW1 = ismember(G1, idx);

imshow(gambarSkorBW1,'Parent',handles.axesNoiseRemove);
imshow(gambarSkorBW1,'Parent',handles.axesProses);
```

hist.m

```
global gambarSkorBW1;
global gambarSkorSel;

potongBarisHistogram =sum(gambarSkorBW1,2);

countHistogram=0;
counterPixelHistogram=0;

    for forBarisHistogram=1:size(potongBarisHistogram,1)
        if potongBarisHistogram(forBarisHistogram) > 15
          %terdapat tulisan (byk pixel foreground)
            countHistogram=countHistogram+1; %menambahkan tinggi
            if potongBarisHistogram(forBarisHistogram+1) < 16
               %pixel selanjutnya < 16 artinya character selesai
                if countHistogram > 30 %(setelah tinggi > 31)
                    if countHistogram < 100 %tingi tertentu(31 - 109)
                        counterPixelHistogram=counterPixelHistogram+1;

                            koordinat(counterPixelHistogram,:)={forBarisHistogr
                            am-(countHistogram+3), countHistogram+6};
                        %kordinat awal, panjang
                        countHistogram=0;
                    end

                end
            end
        end
    end

    barisGambarSkor= 7; %untuk menentukan banyak baris
    gambarSkorSel = cell(barisGambarSkor,2);

    %memindahkan citra baris dari koordinat yang sudah didapatkan
   for forBarisGambar=1:barisGambarSkor;
        koordinatLebarGambar=cell2mat(koordinat(forBarisGambar,2));
        koordinatTinggiGambar=cell2mat(koordinat(forBarisGambar,1));

       gambarHasilHistogram=logical((gambarSkorBW1(koordinatTinggiGambar:koordi
       natLebarGambar+koordinatTinggiGambar,1:end)));

        gambarSkorSel{forBarisGambar,1} = gambarHasilHistogram;
        gambarSkorSel{forBarisGambar,2} = forBarisGambar;
```

```
        pause(0.5);
        imshow(gambarHasilHistogram,'Parent',handles.axesSkorperBaris);
    end
```

resize.m

```
global gambarJudulLabel;
global gambarSkorSel;
global namaPlayer;
global data;

tampungan=cell(0,1);
tampunganJ=cell(0,1);
HasilCorelasi=cell(0,3);

load('template6070.mat');
load('database.mat');


    [J1, bykLabelJ1]=bwlabel(gambarJudulLabel);
    propiedJ=regionprops(J1,'Area');
        area_valuesJ = [propiedJ.Area];
        idxJ = find((50 <= area_valuesJ) & (area_valuesJ <= 3000));
    hasilLabelJ = (ismember(J1, idxJ));


    %% RESIZE

    for forResizeJ=1:bykLabelJ1
      [J2,bykLabelJ2] = find(J1==forResizeJ);

gambarCharacterJudul=(hasilLabelJ(min(J2):max(J2),min(bykLabelJ2):max(bykLabelJ2
)));
      %gambar yang diambil

      ukuranTempJT=60;
      ukuranTempJL=60;
      daerahTempJ=6; %ukuran untuk tiap daerah

      region=ukuranTempJT/daerahTempJ;

      tinggiJudulLama = fix(size(gambarCharacterJudul,1));
      tinggiMaksimalJ = 39;
      ratioJ = tinggiMaksimalJ/tinggiJudulLama;
      ubahUkuranJ = imresize(gambarCharacterJudul,ratioJ);
      ubahUkuranJ = im2double(ubahUkuranJ);

       %% TEMPLATE

      tempJ = (zeros(ukuranTempJT,ukuranTempJL));
      tinggiMaksJudul=tinggiMaksimalJ; %tinggi
      lebarMaksJudul=size(ubahUkuranJ,2); %lebar

      pixJ1=0;
      pixJ2=0;
      pixJ3=0;
      pixJ4=0;
      pixJ5=0;
      pixJ6=0;
```

```
pixJ7=0;
pixJ8=0;
pixJ9=0;
pixJ10=0;
pixJ11=0;
pixJ12=0;
pixJ13=0;
pixJ14=0;
pixJ15=0;
pixJ16=0;
pixJ17=0;
pixJ18=0;
pixJ19=0;
pixJ20=0;
pixJ21=0;
pixJ22=0;
pixJ23=0;
pixJ24=0;
pixJ25=0;
pixJ26=0;
pixJ27=0;
pixJ28=0;
pixJ29=0;
pixJ30=0;
pixJ31=0;
pixJ32=0;
pixJ33=0;
pixJ34=0;
pixJ35=0;
pixJ36=0;
pixJ37=0;
pixJ38=0;
pixJ39=0;
pixJ40=0;
pixJ41=0;
pixJ42=0;
pixJ43=0;
pixJ44=0;
pixJ45=0;
pixJ46=0;
pixJ47=0;
pixJ48=0;
pixJ49=0;
pixJ50=0;
pixJ51=0;
pixJ52=0;
pixJ53=0;
pixJ54=0;
pixJ55=0;
pixJ56=0;
pixJ57=0;
pixJ58=0;
pixJ59=0;
pixJ60=0;
pixJ61=0;
pixJ62=0;
pixJ63=0;
pixJ64=0;
pixJ65=0;
pixJ66=0;
pixJ67=0;
```

```
pixJ68=0;
pixJ69=0;
pixJ70=0;


if(lebarMaksJudul> ukuranTempJL)
      %apabila image > dari wadah maka dianggap noise
else
      tempJ(2:tinggiMaksJudul+1,2:lebarMaksJudul+1)                    =
[ubahUkuranJ(1:end,1:end)];

      for forTempJ1=1:ukuranTempJT %baris
          for forTempJ2=1:ukuranTempJL %kolom
              if tempJ(forTempJ1,forTempJ2)<1 %apabila background
                  %nothing
              else
                  if forTempJ1 < (daerahTempJ+1)
                      if(forTempJ2 < (daerahTempJ+1))
                          pixJ1=pixJ1+1;
                      elseif(forTempJ2 < ((2*daerahTempJ)+1))
                          pixJ2=pixJ2+1;
                      elseif(forTempJ2 < ((3*daerahTempJ)+1))
                          pixJ3=pixJ3+1;
                      elseif(forTempJ2 < ((4*daerahTempJ)+1))
                          pixJ4=pixJ4+1;
                      elseif(forTempJ2 < ((5*daerahTempJ)+1))
                          pixJ5=pixJ5+1;
                      elseif(forTempJ2 < ((6*daerahTempJ)+1))
                          pixJ6=pixJ6+1;
                      elseif(forTempJ2 < ((7*daerahTempJ)+1))
                          pixJ7=pixJ7+1;
                      elseif(forTempJ2 < ((8*daerahTempJ)+1))
                          pixJ8=pixJ8+1;
                      elseif(forTempJ2 < ((9*daerahTempJ)+1))
                          pixJ9=pixJ9+1;
                      elseif(forTempJ2 > (9*daerahTempJ))
                          pixJ10=pixJ10+1;
                      end


                  elseif(forTempJ1 < ((2*daerahTempJ)+1))
                      if(forTempJ2 < (daerahTempJ+1))
                          pixJ11=pixJ11+1;
                      elseif(forTempJ2 < ((2*daerahTempJ)+1))
                          pixJ12=pixJ12+1;
                      elseif(forTempJ2 < ((3*daerahTempJ)+1))
                          pixJ13=pixJ13+1;
                      elseif(forTempJ2 < ((4*daerahTempJ)+1))
                          pixJ14=pixJ14+1;
                      elseif(forTempJ2 < ((5*daerahTempJ)+1))
                          pixJ15=pixJ15+1;
                      elseif(forTempJ2 < ((6*daerahTempJ)+1))
                          pixJ16=pixJ16+1;
                      elseif(forTempJ2 < ((7*daerahTempJ)+1))
                          pixJ17=pixJ17+1;
                      elseif(forTempJ2 < ((8*daerahTempJ)+1))
                          pixJ18=pixJ18+1;
                      elseif(forTempJ2 < ((9*daerahTempJ)+1))
                          pixJ19=pixJ19+1;
                      elseif(forTempJ2 > (9*daerahTempJ))
                          pixJ20=pixJ20+1;
```

```
    end

elseif(forTempJ1 < ((3*daerahTempJ)+1))
    if(forTempJ2 < (daerahTempJ+1))
        pixJ21=pixJ21+1;
    elseif(forTempJ2 < ((2*daerahTempJ)+1))
        pixJ22=pixJ22+1;
    elseif(forTempJ2 < ((3*daerahTempJ)+1))
        pixJ23=pixJ23+1;
    elseif(forTempJ2 < ((4*daerahTempJ)+1))
        pixJ24=pixJ24+1;
    elseif(forTempJ2 < ((5*daerahTempJ)+1))
        pixJ25=pixJ25+1;
    elseif(forTempJ2 < ((6*daerahTempJ)+1))
        pixJ26=pixJ26+1;
    elseif(forTempJ2 < ((7*daerahTempJ)+1))
        pixJ27=pixJ27+1;
    elseif(forTempJ2 < ((8*daerahTempJ)+1))
        pixJ28=pixJ28+1;
    elseif(forTempJ2 < ((9*daerahTempJ)+1))
        pixJ29=pixJ29+1;
    elseif(forTempJ2 > (9*daerahTempJ))
        pixJ30=pixJ30+1;
    end

elseif(forTempJ1 < ((4*daerahTempJ))+1)
    if(forTempJ2 < (daerahTempJ+1))
        pixJ31=pixJ31+1;
    elseif(forTempJ2 < ((2*daerahTempJ)+1))
        pixJ32=pixJ32+1;
    elseif(forTempJ2 < ((3*daerahTempJ)+1))
        pixJ33=pixJ33+1;
    elseif(forTempJ2 < ((4*daerahTempJ)+1))
        pixJ34=pixJ34+1;
    elseif(forTempJ2 < ((5*daerahTempJ)+1))
        pixJ35=pixJ35+1;
    elseif(forTempJ2 < ((6*daerahTempJ)+1))
        pixJ36=pixJ36+1;
    elseif(forTempJ2 < ((7*daerahTempJ)+1))
        pixJ37=pixJ37+1;
    elseif(forTempJ2 < ((8*daerahTempJ)+1))
        pixJ38=pixJ38+1;
    elseif(forTempJ2 < ((9*daerahTempJ)+1))
        pixJ39=pixJ39+1;
    elseif(forTempJ2 > (9*daerahTempJ))
        pixJ40=pixJ40+1;
    end

elseif(forTempJ1 < ((5*daerahTempJ))+1)
    if(forTempJ2 < (daerahTempJ+1))
        pixJ41=pixJ41+1;
    elseif(forTempJ2 < ((2*daerahTempJ)+1))
        pixJ42=pixJ42+1;
    elseif(forTempJ2 < ((3*daerahTempJ)+1))
        pixJ43=pixJ43+1;
    elseif(forTempJ2 < ((4*daerahTempJ)+1))
        pixJ44=pixJ44+1;
    elseif(forTempJ2 < ((5*daerahTempJ)+1))
        pixJ45=pixJ45+1;
    elseif(forTempJ2 < ((6*daerahTempJ)+1))
        pixJ46=pixJ46+1;
```

```
                elseif(forTempJ2 < ((7*daerahTempJ)+1))
                    pixJ47=pixJ47+1;
                elseif(forTempJ2 < ((8*daerahTempJ)+1))
                    pixJ48=pixJ48+1;
                elseif(forTempJ2 < ((9*daerahTempJ)+1))
                    pixJ49=pixJ49+1;
                elseif(forTempJ2 > (9*daerahTempJ))
                    pixJ50=pixJ50+1;
                end

            elseif(forTempJ1 < ((6*daerahTempJ))+1)
                if(forTempJ2 < (daerahTempJ+1))
                    pixJ51=pixJ51+1;
                elseif(forTempJ2 < ((2*daerahTempJ)+1))
                    pixJ52=pixJ52+1;
                elseif(forTempJ2 < ((3*daerahTempJ)+1))
                    pixJ53=pixJ53+1;
                elseif(forTempJ2 < ((4*daerahTempJ)+1))
                    pixJ54=pixJ54+1;
                elseif(forTempJ2 < ((5*daerahTempJ)+1))
                    pixJ55=pixJ55+1;
                elseif(forTempJ2 < ((6*daerahTempJ)+1))
                    pixJ56=pixJ56+1;
                elseif(forTempJ2 < ((7*daerahTempJ)+1))
                    pixJ57=pixJ57+1;
                elseif(forTempJ2 < ((8*daerahTempJ)+1))
                    pixJ58=pixJ58+1;
                elseif(forTempJ2 < ((9*daerahTempJ)+1))
                    pixJ59=pixJ59+1;
                elseif(forTempJ2 > (9*daerahTempJ))
                    pixJ60=pixJ60+1;
                end

            elseif(forTempJ1 > (6*daerahTempJ))
                if(forTempJ2 < (daerahTempJ+1))
                    pixJ61=pixJ61+1;
                elseif(forTempJ2 < ((2*daerahTempJ)+1))
                    pixJ62=pixJ62+1;
                elseif(forTempJ2 < ((3*daerahTempJ)+1))
                    pixJ63=pixJ63+1;
                elseif(forTempJ2 < ((4*daerahTempJ)+1))
                    pixJ64=pixJ64+1;
                elseif(forTempJ2 < ((5*daerahTempJ)+1))
                    pixJ65=pixJ65+1;
                elseif(forTempJ2 < ((6*daerahTempJ)+1))
                    pixJ66=pixJ66+1;
                elseif(forTempJ2 < ((7*daerahTempJ)+1))
                    pixJ67=pixJ67+1;
                elseif(forTempJ2 < ((8*daerahTempJ)+1))
                    pixJ68=pixJ68+1;
                elseif(forTempJ2 < ((9*daerahTempJ)+1))
                    pixJ69=pixJ69+1;
                elseif(forTempJ2 > (9*daerahTempJ))
                    pixJ70=pixJ70+1;
                end

            end
        end
    end
end
```

```
            idxJudul=size(tampunganJ,1)+1;


             if (region==10)
               tampunganJ{idxJudul,1} = [pixJ1 pixJ2 pixJ3 pixJ4 pixJ5 pixJ6
pixJ7
                                         pixJ8 pixJ9 pixJ10...
                           pixJ11 pixJ12 pixJ13 pixJ14 pixJ15 pixJ16 pixJ17
pixJ18
                             pixJ19 pixJ20...
                           pixJ21 pixJ22 pixJ23 pixJ24 pixJ25 pixJ26 pixJ27
pixJ28
                             pixJ29 pixJ30...
                           pixJ31 pixJ32 pixJ33 pixJ34 pixJ35 pixJ36 pixJ37
pixJ38
                             pixJ39 pixJ40...
                           pixJ41 pixJ42 pixJ43 pixJ44 pixJ45 pixJ46 pixJ47
pixJ48
                             pixJ49 pixJ50...
                           pixJ51 pixJ52 pixJ53 pixJ54 pixJ55 pixJ56 pixJ57
pixJ58                       pixJ59 pixJ60...
                           pixJ61 pixJ62 pixJ63 pixJ64 pixJ65 pixJ66 pixJ67
pixJ68
                             pixJ69 pixJ70...
                           ];
             elseif (region==5)
                 tampunganJ{idxJudul,1} = [pixJ1 pixJ2 pixJ3 pixJ4 pixJ5...
                           pixJ11 pixJ12 pixJ13 pixJ14 pixJ15...
                           pixJ21 pixJ22 pixJ23 pixJ24 pixJ25...
                           pixJ31 pixJ32 pixJ33 pixJ34 pixJ35...
                           ];

             elseif (region==4)
                 tampunganJ{idxJudul,1} = [pixJ1 pixJ2 pixJ3 pixJ4...
                           pixJ11 pixJ12 pixJ13 pixJ14...
                           pixJ21 pixJ22 pixJ23 pixJ24...
                           ];
             end

             simpanKorelasiJ=0;

             for forPengecekanTempJ=1:size(template,1)

          KorelasiJ=corr2(tampunganJ{idxJudul,1},template{forPengecekanTempJ,1});
                    tampunganKorelasiJ=KorelasiJ;

                    if(tampunganKorelasiJ > simpanKorelasiJ)
                        simpanKorelasiJ = tampunganKorelasiJ;
                        nilaiTemplateJ = template{forPengecekanTempJ,2};
                    end
             end

             if(simpanKorelasiJ > 0.75)
                 hasilJudul=size(HasilCorelasi,1)+1;
                        %untuk ganti apabila idx sudah terisi
                 HasilCorelasi{hasilJudul,1} = 1;
                        %identifikasi ke judul maka baris ke 1
                 HasilCorelasi{hasilJudul,2} = nilaiTemplateJ;
                 HasilCorelasi{hasilJudul,3} = simpanKorelasiJ;
             end
         end
```

```
   end

%SCORE
for forLabel2=1:7 %for sebanyak 7 baris dari yang sudah dipotong
    gambarSkorperBaris = gambarSkorSel{forLabel2,1};

    %untuk menghilangkan hasil potongan horizontal yang bisa menjadi noise.
    gambarSkorperBaris = bwareaopen(gambarSkorperBaris,200);

    [G3, bykLabel3]=bwlabel(gambarSkorperBaris);
    propied=regionprops(G3,'Area');
        area_values = [propied.Area];
        idx = find((200 <= area_values) & (area_values <= 3000));
    hasilLabel2 = (ismember(G3, idx));

    %% RESIZE

     for forResize=1:bykLabel3 %diambil tiap character tiap baris

      [G4,bykLabel4] = find(G3==forResize);

gambarCharacter=(hasilLabel2(min(G4):max(G4),min(bykLabel4):max(bykLabel4)));

      ukuranTempT=60;
      ukuranTempL=60;
      daerahTemp=6; %ukuran untuk tiap daerah

      tinggiGambarLama = fix(size(gambarCharacter,1));
      tinggiMaksimal = 39;
      ratio = tinggiMaksimal/tinggiGambarLama;
      ubahUkuran = imresize(gambarCharacter,ratio);
      ubahUkuran = im2double(ubahUkuran);

      tempS = (zeros(ukuranTempT,ukuranTempL));
      tinggiMaksimalGambar=tinggiMaksimal; %tinggi
      lebarMaksimalGambar=size(ubahUkuran,2); %lebar

      pix1=0;
      pix2=0;
      pix3=0;
      pix4=0;
      pix5=0;
      pix6=0;
      pix7=0;
      pix8=0;
      pix9=0;
      pix10=0;
      pix11=0;
      pix12=0;
      pix13=0;
      pix14=0;
      pix15=0;
      pix16=0;
      pix17=0;
      pix18=0;
      pix19=0;
      pix20=0;
      pix21=0;
      pix22=0;
      pix23=0;
      pix24=0;
```

```
pix25=0;
pix26=0;
pix27=0;
pix28=0;
pix29=0;
pix30=0;
pix31=0;
pix32=0;
pix33=0;
pix34=0;
pix35=0;
pix36=0;
pix37=0;
pix38=0;
pix39=0;
pix40=0;
pix41=0;
pix42=0;
pix43=0;
pix44=0;
pix45=0;
pix46=0;
pix47=0;
pix48=0;
pix49=0;
pix50=0;
pix51=0;
pix52=0;
pix53=0;
pix54=0;
pix55=0;
pix56=0;
pix57=0;
pix58=0;
pix59=0;
pix60=0;
pix61=0;
pix62=0;
pix63=0;
pix64=0;
pix65=0;
pix66=0;
pix67=0;
pix68=0;
pix69=0;
pix70=0;

if(lebarMaksimalGambar> ukuranTempL)
      %apabila image > dari wadah maka dianggap noise
else
      tempS(2:tinggiMaksimalGambar+1,2:lebarMaksimalGambar+1)             =
[ubahUkuran(1:end,1:end)];

      for forTemp1=1:ukuranTempT %baris
          for forTemp2=1:ukuranTempT %kolom
              if tempS(forTemp1,forTemp2)<1
                  %skip
              else
                  if(forTemp1 < (daerahTemp+1))
                      if(forTemp2 < (daerahTemp+1))
                          pix1=pix1+1;
```

```
    elseif(forTemp2 < ((2*daerahTemp)+1))
        pix2=pix2+1;
    elseif(forTemp2 < ((3*daerahTemp)+1))
        pix3=pix3+1;
    elseif(forTemp2 < ((4*daerahTemp)+1))
        pix4=pix4+1;
    elseif(forTemp2 < ((5*daerahTemp)+1))
        pix5=pix5+1;
    elseif(forTemp2 < ((6*daerahTemp)+1))
        pix6=pix6+1;
    elseif(forTemp2 < ((7*daerahTemp)+1))
        pix7=pix7+1;
    elseif(forTemp2 < ((8*daerahTemp)+1))
        pix8=pix8+1;
    elseif(forTemp2 < ((9*daerahTemp)+1))
        pix9=pix9+1;
    elseif(forTemp2 > (9*daerahTemp))
        pix10=pix10+1;
    end

elseif(forTemp1 < ((2*daerahTemp)+1))
    if(forTemp2 < (daerahTemp+1))
        pix11=pix11+1;
    elseif(forTemp2 < ((2*daerahTemp)+1))
        pix12=pix12+1;
    elseif(forTemp2 < ((3*daerahTemp)+1))
        pix13=pix13+1;
    elseif(forTemp2 < ((4*daerahTemp)+1))
        pix14=pix14+1;
    elseif(forTemp2 < ((5*daerahTemp)+1))
        pix15=pix15+1;
    elseif(forTemp2 < ((6*daerahTemp)+1))
        pix16=pix16+1;
    elseif(forTemp2 < ((7*daerahTemp)+1))
        pix17=pix17+1;
    elseif(forTemp2 < ((8*daerahTemp)+1))
        pix18=pix18+1;
    elseif(forTemp2 < ((9*daerahTemp)+1))
        pix19=pix19+1;
    elseif(forTemp2 > (9*daerahTemp))
        pix20=pix20+1;
    end

elseif(forTemp1 < ((3*daerahTemp)+1))
    if(forTemp2 < (daerahTemp+1))
        pix21=pix21+1;
    elseif(forTemp2 < ((2*daerahTemp)+1))
        pix22=pix22+1;
    elseif(forTemp2 < ((3*daerahTemp)+1))
        pix23=pix23+1;
    elseif(forTemp2 < ((4*daerahTemp)+1))
        pix24=pix24+1;
    elseif(forTemp2 < ((5*daerahTemp)+1))
        pix25=pix25+1;
    elseif(forTemp2 < ((6*daerahTemp)+1))
        pix26=pix26+1;
    elseif(forTemp2 < ((7*daerahTemp)+1))
        pix27=pix27+1;
    elseif(forTemp2 < ((8*daerahTemp)+1))
        pix28=pix28+1;
    elseif(forTemp2 < ((9*daerahTemp)+1))
```

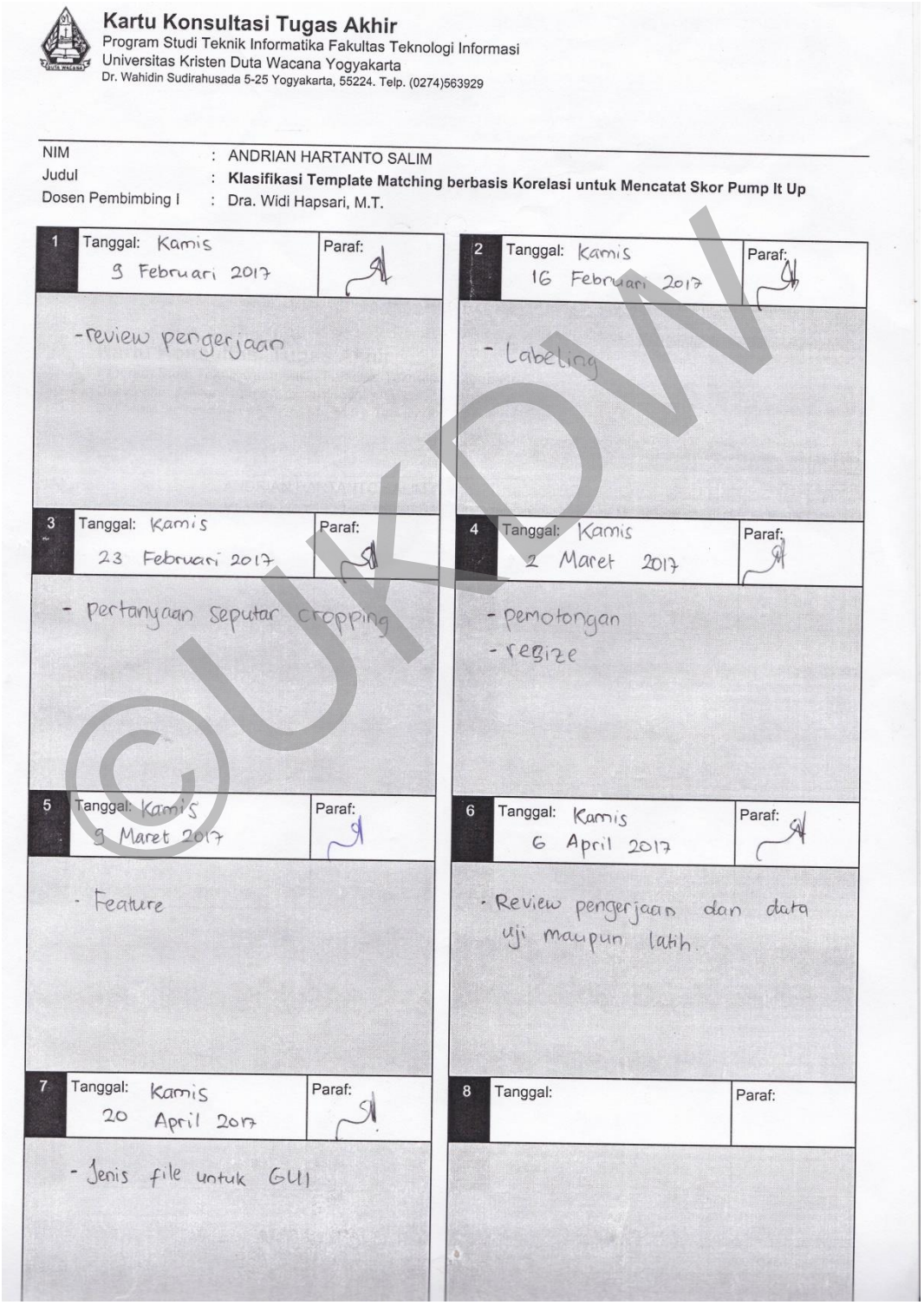
```
            pix29=pix29+1;
        elseif(forTemp2 > (9*daerahTemp))
            pix30=pix30+1;
        end

    elseif(forTemp1 < ((4*daerahTemp))+1)
        if(forTemp2 < (daerahTemp+1))
            pix31=pix31+1;
        elseif(forTemp2 < ((2*daerahTemp)+1))
            pix32=pix32+1;
        elseif(forTemp2 < ((3*daerahTemp)+1))
            pix33=pix33+1;
        elseif(forTemp2 < ((4*daerahTemp)+1))
            pix34=pix34+1;
        elseif(forTemp2 < ((5*daerahTemp)+1))
            pix35=pix35+1;
        elseif(forTemp2 < ((6*daerahTemp)+1))
            pix36=pix36+1;
        elseif(forTemp2 < ((7*daerahTemp)+1))
            pix37=pix37+1;
        elseif(forTemp2 < ((8*daerahTemp)+1))
            pix38=pix38+1;
        elseif(forTemp2 < ((9*daerahTemp)+1))
            pix39=pix39+1;
        elseif(forTemp2 > (9*daerahTemp))
            pix40=pix40+1;
        end

    elseif(forTemp1 < ((5*daerahTemp))+1)
        if(forTemp2 < (daerahTemp+1))
            pix41=pix41+1;
        elseif(forTemp2 < ((2*daerahTemp)+1))
            pix42=pix42+1;
        elseif(forTemp2 < ((3*daerahTemp)+1))
            pix43=pix43+1;
        elseif(forTemp2 < ((4*daerahTemp)+1))
            pix44=pix44+1;
        elseif(forTemp2 < ((5*daerahTemp)+1))
            pix45=pix45+1;
        elseif(forTemp2 < ((6*daerahTemp)+1))
            pix46=pix46+1;
        elseif(forTemp2 < ((7*daerahTemp)+1))
            pix47=pix47+1;
        elseif(forTemp2 < ((8*daerahTemp)+1))
            pix48=pix48+1;
        elseif(forTemp2 < ((9*daerahTemp)+1))
            pix49=pix49+1;
        elseif(forTemp2 > (9*daerahTemp))
            pix50=pix50+1;
        end

    elseif(forTemp1 < ((6*daerahTemp))+1)
        if(forTemp2 < (daerahTemp+1))
            pix51=pix51+1;
        elseif(forTemp2 < ((2*daerahTemp)+1))
            pix52=pix52+1;
        elseif(forTemp2 < ((3*daerahTemp)+1))
            pix53=pix53+1;
        elseif(forTemp2 < ((4*daerahTemp)+1))
            pix54=pix54+1;
        elseif(forTemp2 < ((5*daerahTemp)+1))
```

```
                                pix55=pix55+1;
                            elseif(forTemp2 < ((6*daerahTemp)+1))
                                pix56=pix56+1;
                            elseif(forTemp2 < ((7*daerahTemp)+1))
                                pix57=pix57+1;
                            elseif(forTemp2 < ((8*daerahTemp)+1))
                                pix58=pix58+1;
                            elseif(forTemp2 < ((9*daerahTemp)+1))
                                pix59=pix59+1;
                            elseif(forTemp2 > (9*daerahTemp))
                                pix60=pix60+1;
                            end

                        elseif(forTemp1 < ((7*daerahTemp))+1)
                            if(forTemp2 < (daerahTemp+1))
                                pix61=pix61+1;
                            elseif(forTemp2 < ((2*daerahTemp)+1))
                                pix62=pix62+1;
                            elseif(forTemp2 < ((3*daerahTemp)+1))
                                pix63=pix63+1;
                            elseif(forTemp2 < ((4*daerahTemp)+1))
                                pix64=pix64+1;
                            elseif(forTemp2 < ((5*daerahTemp)+1))
                                pix65=pix65+1;
                            elseif(forTemp2 < ((6*daerahTemp)+1))
                                pix66=pix66+1;
                            elseif(forTemp2 < ((7*daerahTemp)+1))
                                pix67=pix67+1;
                            elseif(forTemp2 < ((8*daerahTemp)+1))
                                pix68=pix68+1;
                            elseif(forTemp2 < ((9*daerahTemp)+1))
                                pix69=pix69+1;
                            elseif(forTemp2 > (9*daerahTemp))
                                pix70=pix70+1;
                            end
                        end
                    end
                end
            end

            idxIsi=size(tampungan,1)+1;

            if (region==10)
                tampungan{idxIsi,1} = [pix1 pix2 pix3 pix4 pix5 pix6 pix7 pix8
pix9
                                       pix10...
                                    pix11 pix12 pix13 pix14 pix15 pix16 pix17 pix18
                                       pix19 pix20 ...
                                    pix21 pix22 pix23 pix24 pix25 pix26 pix27 pix28
                                       pix29 pix30 ...
                                    pix31 pix32 pix33 pix34 pix35 pix36 pix37 pix38
                                       pix39 pix40 ...
                                    pix41 pix42 pix43 pix44 pix45 pix46 pix47 pix48
                                       pix49 pix50 ...
                                    pix51 pix52 pix53 pix54 pix55 pix56 pix57 pix58
                                       pix59 pix60 ...
                                    pix61 pix62 pix63 pix64 pix65 pix66 pix67 pix68
                                       pix69 pix70 ...
                                    ];
             elseif (region==5)
                tampungan{idxIsi,1} = [pix1 pix2 pix3 pix4 pix5...
```

```
                                pix11 pix12 pix13 pix14 pix15...
                                pix21 pix22 pix23 pix24 pix25...
                                pix31 pix32 pix33 pix34 pix35...
                                ];

            elseif (region==4)
                 tampungan{idxIsi,1} = [pix1 pix2 pix3 pix4...
                                pix11 pix12 pix13 pix14...
                                pix21 pix22 pix23 pix24 ...
                                ];
            end


            simpanKorelasi=0;

            for forPengecekanTemplate=1:size(template,1)

Korelasi=corr2(tampungan{idxIsi,1},template{forPengecekanTemplate,1});
                  tampunganKorelasi=Korelasi;

                  if(tampunganKorelasi > simpanKorelasi)
                      simpanKorelasi = tampunganKorelasi;
                      nilaiTemplate = template{forPengecekanTemplate,2};
                  end
            end

             if(simpanKorelasi>0.75)
                hasilisi=size(HasilCorelasi,1)+1; %utk idx array
                HasilCorelasi{hasilisi,1} = forLabel2+1; %menunjuk isi
                HasilCorelasi{hasilisi,2} = nilaiTemplate;
                HasilCorelasi{hasilisi,3} = simpanKorelasi;
             end

     %masukin hasil ke dalam array utk database
     %di for kemudian str cat untuk digabungkan berdasarkan garis dan banyak
     %huruf

      end

    end
end
    namaJudul='';
    perfect='';
    great='';
    good='';
    bad='';
    miss='';
    maxCombo='';
    nilaiScore='';


    %simpan ke database
    for bykdata=1:size(HasilCorelasi,1)
        if cell2mat(HasilCorelasi(bykdata,1))==1

            namaJudul=strcat(namaJudul,HasilCorelasi(bykdata,2));

        elseif cell2mat(HasilCorelasi(bykdata,1))==2

            perfect=strcat(perfect,HasilCorelasi(bykdata,2));
```

```
    elseif cell2mat(HasilCorelasi(bykdata,1))==3
        great=strcat(great,HasilCorelasi(bykdata,2));
    elseif cell2mat(HasilCorelasi(bykdata,1))==4
        good=strcat(good,HasilCorelasi(bykdata,2));
    elseif cell2mat(HasilCorelasi(bykdata,1))==5
        bad=strcat(bad,HasilCorelasi(bykdata,2));
    elseif cell2mat(HasilCorelasi(bykdata,1))==6
        miss=strcat(miss,HasilCorelasi(bykdata,2));
    elseif cell2mat(HasilCorelasi(bykdata,1))==7
        maxCombo=strcat(maxCombo,HasilCorelasi(bykdata,2));
    elseif cell2mat(HasilCorelasi(bykdata,1))==8
        nilaiScore=strcat(nilaiScore,HasilCorelasi(bykdata,2));
    end
end

%cell ke cell
bykdatabase=(size(data,1))+1;
%9 = nama | judul lagu | score | dkk
data(bykdatabase,1)={namaPlayer};
data(bykdatabase,2)={namaJudul};
data(bykdatabase,3)={nilaiScore};
data(bykdatabase,4)={perfect};
data(bykdatabase,5)={great};
data(bykdatabase,6)={good};
data(bykdatabase,7)={bad};
data(bykdatabase,8)={miss};
data(bykdatabase,9)={maxCombo};

save('database.mat','data');

%cell ke matrik sehingga langsung terambil isi dari cell nya dan bisa
   ditampilkan
namaJudul=cell2mat(namaJudul);
perfect=cell2mat(perfect);
great=cell2mat(great);
good=cell2mat(good);
bad=cell2mat(bad);
miss=cell2mat(miss);
maxCombo=cell2mat(maxCombo);
nilaiScore=cell2mat(nilaiScore);

set(handles.tabJudul,'string',namaJudul);
set(handles.tabPerfect,'string',perfect);
set(handles.tabGreat,'string',great);
set(handles.tabGood,'string',good);
set(handles.tabBad,'string',bad);
set(handles.tabMiss,'string',miss);
set(handles.tabCombo,'string',maxCombo);
set(handles.tabScore,'string',nilaiScor
```

Lampiran C: Kartu Konsultasi dan Formulir Revisi



Kartu Konsultasi Tugas Akhir
Program Studi Teknik Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana Yogyakarta
Dr. Wahidin Sudirahusada 5-25 Yogyakarta, 55224. Telp. (0274)563929

NIM : ANDRIAN HARTANTO SALIM
Judul : **Klasifikasi Template Matching berbasis Korelasi untuk Mencatat Skor Pump It Up**
Dosen Pembimbing I : Dra. Widi Hapsari, M.T.

1	Tanggal: Kamis 9 Februari 2017	Paraf:	- review pengerjaan
2	Tanggal: Kamis 16 Februari 2017	Paraf:	- Labeling
3	Tanggal: Kamis 23 Februari 2017	Paraf:	- pertanyaan seputar cropping
4	Tanggal: Kamis 2 Maret 2017	Paraf:	- pemotongan - resize
5	Tanggal: Kamis 9 Maret 2017	Paraf:	- Feature
6	Tanggal: Kamis 6 April 2017	Paraf:	- Review pengerjaan dan data uji maupun latih.
7	Tanggal: Kamis 20 April 2017	Paraf:	- Jenis file untuk GUI
8	Tanggal:	Paraf:	

Kartu Konsultasi Tugas Akhir

Program Studi Teknik Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana Yogyakarta
Dr. Wahidin Sudirahusada 5-25 Yogyakarta, 55224. Telp. (0274)563929

NIM : ANDRIAN HARTANTO SALIM
Judul : **Klasifikasi Template Matching berbasis Korelasi untuk Mencatat Skor Pump It Up**
Dosen Pembimbing II : Nugroho Agus Haryono, M.Si

1	Tanggal: Kamis 9 Februari 2017	Paraf:
	- review pengerjaan	
2	Tanggal: Kamis 16 Februari 2017	Paraf:
	- Labeling	
3	Tanggal: Kamis 2 Maret 2017	Paraf:
	- pemotongan - resize	
4	Tanggal: Kamis 9 Maret 2017	Paraf:
	- Feature	
5	Tanggal: Kamis 30 Maret 2017	Paraf:
	- Pengumpulan data template	
6	Tanggal: Kamis 6 April 2017	Paraf:
	- Review pengerjaan dan data uji maupun latih.	
7	Tanggal: Kamis 20 April 2017	Paraf:
	- Jenis file untuk GUI	
8	Tanggal:	Paraf:

Program Studi Teknik Informatika
Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana Yogyakarta
Dr. Wahidin Sudirahusada 5-25 Yogyakarta, 55224. Telp. (0274)563929

FORMULIR PERBAIKAN (REVISI) SKRIPSI

Strata-1 Program Studi Teknik Informatika

Yang bertanda tangan di bawah ini:

Nama	:	ANDRIAN HARTANTO SALIM
N I M	:	71130037
Judul Skripsi	:	KLASIFIKASI TEMPLATE MATCHING BERBASIS KORELASI UNTUK MENCATAT SKOR PUMP IT UP
Tanggal Pendadaran	:	31 Mei 2017 08:00 WIB

Telah melakukan perbaikan tugas akhir dengan lengkap.

Demikian pernyataan kami agar dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 6 Juni 2017

Dosen Pembimbing I	Dosen Pembimbing II
Widi Hapsari, Dra. M.T.	Nugroho Agus Haryono, M.Si

Dicetak tanggal: *6 Juni 2017 14:28 WIB*