

BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN

3.1 Metodologi

Pada bab ini akan dibahas metode dan langkah-langkah yang digunakan untuk melakukan deteksi pengenalan nominal uang yang menggunakan algoritma MKNN.

Penelitian dilakukan dengan langkah-langkah sebagai berikut:

1. Melakukan pengumpulan data dari uang kertas rupiah.
2. Melakukan studi literatur yang berkaitan dengan pengolahan citra digital, pengenalan pola, dan algoritma MKNN.
3. Merancang perangkat lunak untuk pengenalan nominal uang pada jenis uang kertas rupiah.
4. Implementasi perangkat lunak berdasarkan analisis dan perancangan yang dilakukan.
5. Melakukan pengujian terhadap perangkat lunak.
6. Melakukan evaluasi tingkat keberhasilan perangkat lunak dan analisis hasil pengujian.

3.2 Objek Penelitian

Penelitian ini dilakukan pada uang kertas rupiah yang masih berlaku sampai saat ini di Indonesia. Pada penelitian ini nantinya perangkat lunak yang dibuat dapat menganalisa atau mendiagnosa nominal yang tertera pada uang kertas rupiah.

3.3 Teknik Pengumpulan Data

Adapun teknik pengumpulan data yang dilakukan oleh penulis dalam penelitian, antara lain :

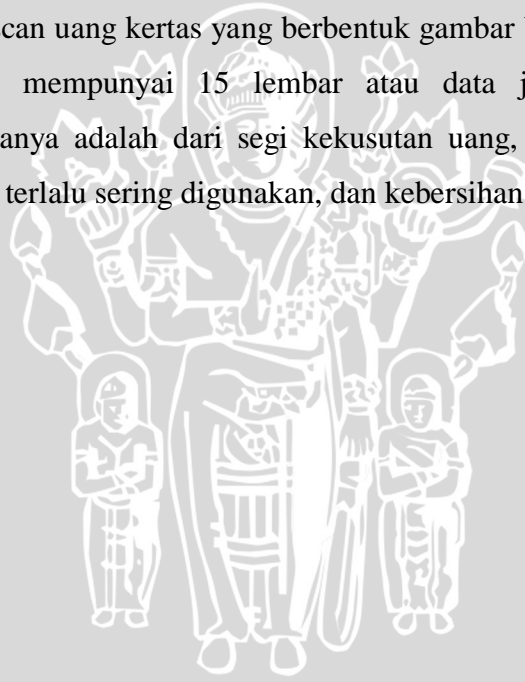
1. Studi Kepustakaan

Tinjauan literatur adalah bagian penting dari setiap proyek penelitian yang baik, dan jika tidak dilakukan dengan benar, proyek dapat berakhir sia-sia dengan mereplikasi penelitian sebelumnya. Metode ini dilakukan untuk mendapatkan

data literatur tambahan yang digunakan sebagai acuan pengenalan atau deteksi nominal uang kertas rupiah menggunakan metode *modified K-Nearest Neighbour*, sumber yang digunakan berupa buku, journal, karya ilmiah, dan situs-situs penunjang yang dapat membantu dalam penyelesaian laporan penelitian.

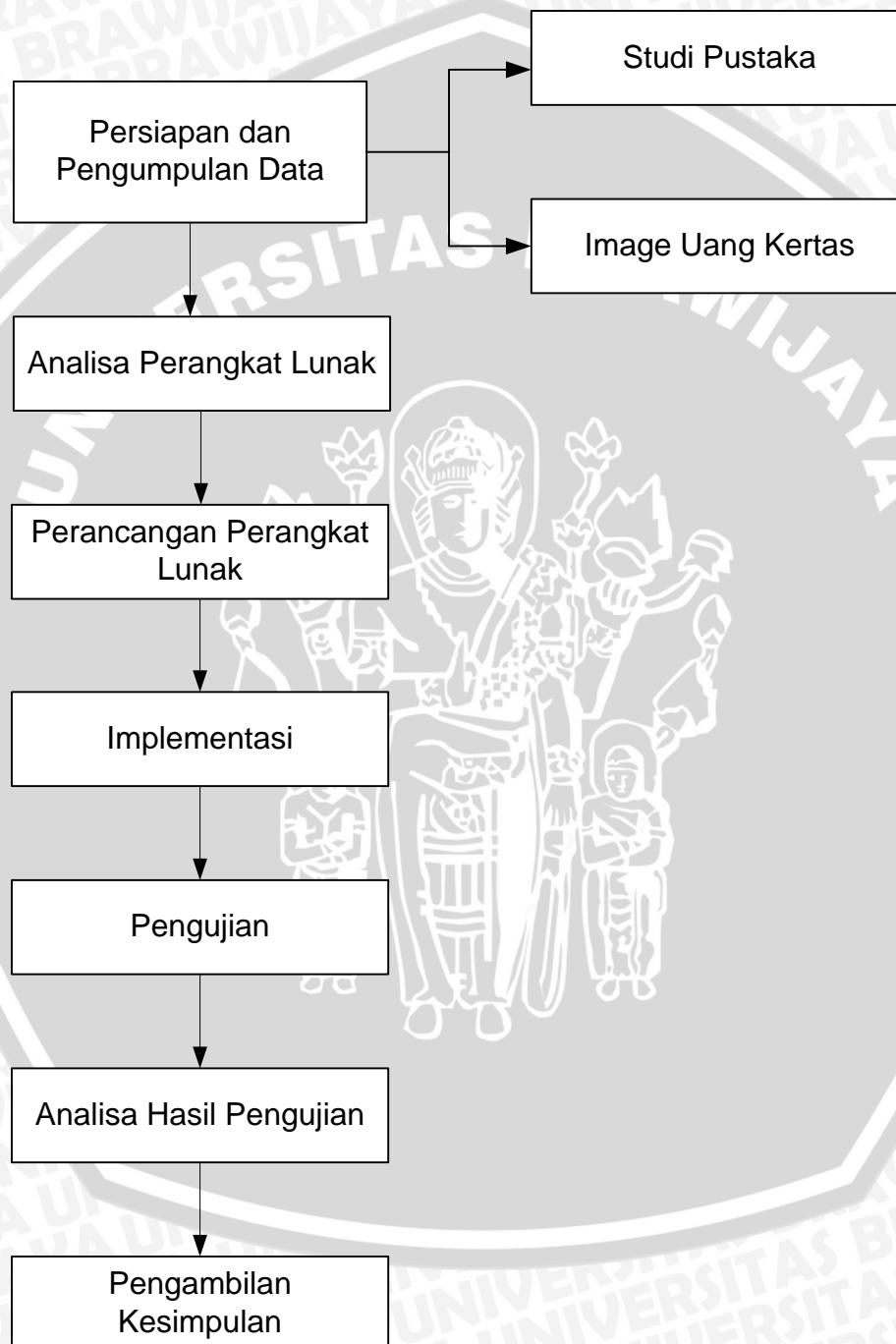
2. Data *Image* Uang Kertas

Data pada penelitian ini yang digunakan adalah berupa *image* uang kertas rupiah indonesia yang masih berlaku sampai saat ini. Data yang akan digunakan adalah data nilai RGB yang diambil dari proses pengolahan citra digital dari uang kertas rupiah. Jenis uang kertas rupiah yang digunakan adalah jenis pecahan 1000, 2000, 5000, 10000, 20000, 50000, dan 100000. Untuk mendapatkan nilai RGBnya, uang kertas kita scan lalu yang digunakan oleh perangkat lunak ini nantinya adalah hasil scan uang kertas yang berbentuk gambar berformat jpg/png. Setiap pecahan akan mempunyai 15 lembar atau data jenis uang. Yang mempengaruhi perbedaannya adalah dari segi kekusutan uang, warna uang yang sedikit berubah karena terlalu sering digunakan, dan kebersihan uang.



3.4 Tahapan Penelitian

Adapun tahapan penelitian dalam membangun perangkat lunak pengenalan nominal uang kertas adalah sebagai berikut :



Gambar 3. 1 Tahapan Penelitian

Penjelasan tahapan penelitian sebagai berikut :

1. Persiapan dan Pengumpulan Data

Penulis melakukan persiapan untuk sebelum melakukan pengumpulan data, kemudian penulis melakukan studi pustaka untuk mencari dan mempelajari tentang pengolahan citra *image*, dan mencari informasi tentang metode *modified K-nearest neighbour*, yang digunakan dalam perancangan perangkat lunak agar mempunyai tingkat kebenaran yang tinggi.

2. Analisa Perangkat Lunak

Pada tahap analisis perangkat lunak ini penulis melakukan analisa perangkat lunak sesuai data dan permasalahan yang telah dikumpulkan sebelumnya, guna sebagai acuan yang digunakan untuk merancang perangkat lunak sesuai dengan kebutuhan. Pada tahap analisa perangkat lunak ini pertama-tama akan dilakukan analisa apa saja yang dibutuhkan terlebih dahulu.

3. Perancangan

Merupakan tahap penulisan proses, data, aliran proses, perancangan antar muka, dan perhitungan manual. Rancangan system ini adalah berupa *prototype* yang nantinya akan digunakan untuk membuat aplikasi yang sesungguhnya. Perlunya membuat perancangan adalah untuk mempermudah dalam pembuatan sebuah aplikasi. Karena agar aplikasi tersebut dibuat sudah mempunyai acuan yaitu dari perancangan ini.

4. Implementasi

Pada tahap ini, penulis mengimplementasikan berdasarkan rancangan yang telah dibuat sebelumnya sesuai dengan kebutuhan dan metode yang digunakan. Dalam tahap ini semua rancangan akan kita pindahkan menjadi sebuah aplikasi pada komputer. Segala hal yang dikerjakan dalam implementasi ini mengacu para perancangan-perancangan sebelumnya.

5. Pengujian

Merupakan tahap pengujian terhadap perangkat lunak yang telah dibuat. Segala hal yang sudah dibuat yaitu aplikasi akan di uji pada tahap ini. Pengujian ini diperlukan untuk mengetahui tingkat akurasi dan performansi terhadap implementasi algoritma MKNN dalam menentukan nominal uang kertas rupiah.

6. Analisa Hasil Pengujian

Pada bagian ini akan dilakukan analisa hasil pengujian yaitu membuat analisa akurasi jika skenario pengujian kita ubah-ubah. Dalam bagian ini akan dilakukan beberapa skenario pengujian agar kita dapat mengetahui tingkat akurasi perangkat lunak yang sudah dibuat .

7. Pengambilan Kesimpulan

Penulis melakukan pengambilan kesimpulan terhadap aplikasi yang sudah dibuat yang mengacu berdasarkan metode yang digunakan. Pada tahap ini juga dijelaskan hasil utama akurasi perangkat lunak. Dan penulis juga memberikan saran terhadap penelitian yang telah dibuat berdasarkan metode yang dipakai.

3.5 Analisa Perangkat Lunak

3.5.1 Deskripsi Perangkat Lunak

Perangkat Lunak ini dibangun untuk mempermudah dalam identifikasi nominal uang kertas yang digunakan untuk suatu transaksi yang memanfaatkan teknologi supaya dapat meminimalisir suatu kesalahan manusia, serta menjadi alternatif bagi tunanetra untuk lebih mudah mengetahui nominal uang kertas. Hal ini akan menguntungkan masyarakat karena transaksi berjalan secara otomatis, akurat, cepat, dan validitas yang terjamin.

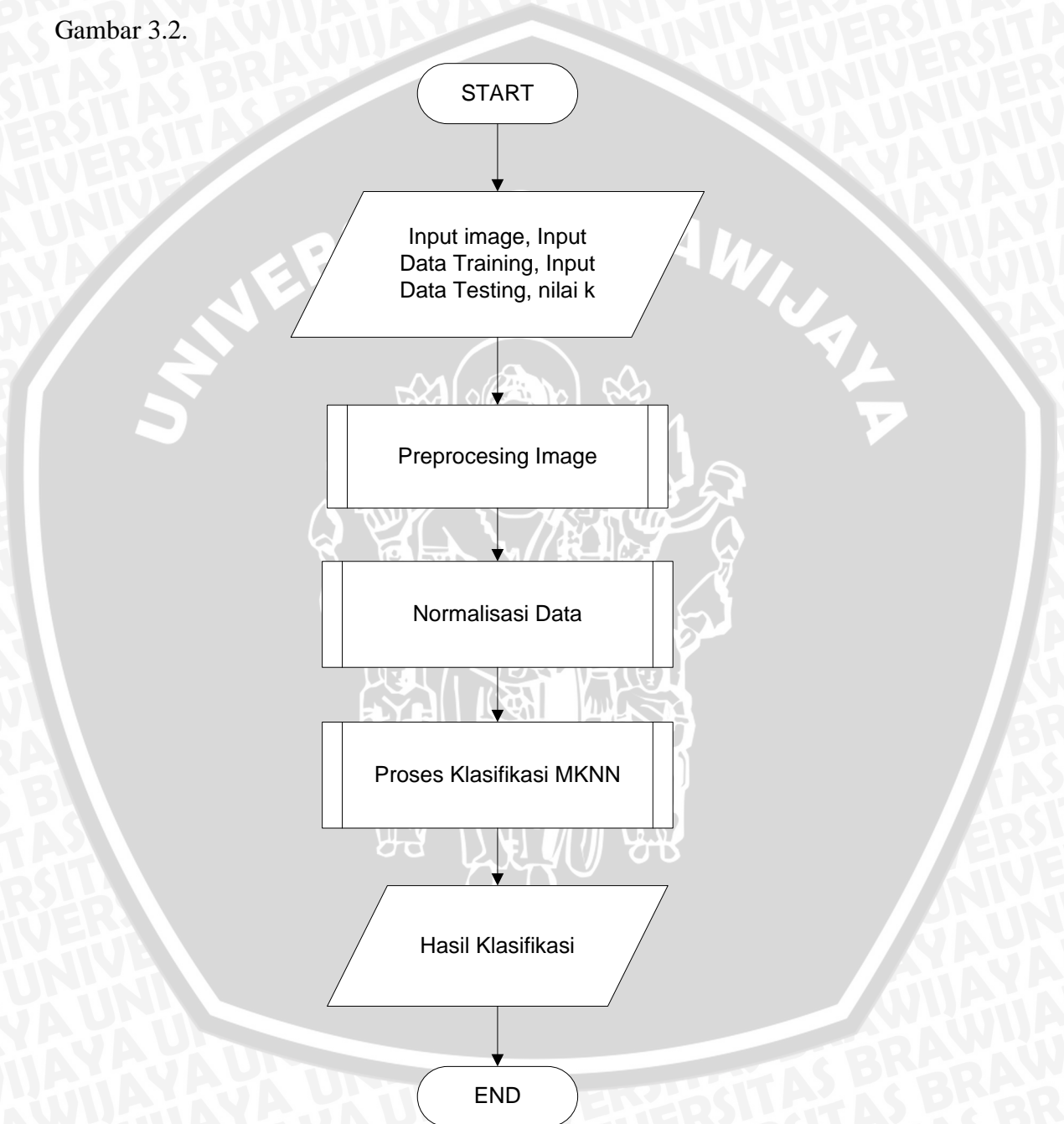
Perangkat lunak ini dibatasi hanya mengenali jenis uang kertas rupiah indonesia yang masih berlaku sekarang(2014). Data yang digunakan untuk mengenali nominal uang ini diambil dari nilai RGBnya.

3.5.2 Deskripsi Data

Pada penelitian ini, data yang akan digunakan adalah data nilai RGB yang diambil dari proses pengolahan citra digital dari uang kertas rupiah. Jenis uang kertas rupiah yang digunakan adalah jenis pecahan 1000, 2000, 5000, 10000, 20000, 50000, dan 100000. Untuk mendapatkan nilai RGBnya, uang kertas kita scan lalu yang digunakan oleh penelitian ini nantinya adalah hasil scan uang kertas yang berbentuk gambar berformat jpg/png. Setiap pecahan akan diambil kurang lebih 15 sampel. Yang mempengaruhi perbedaanya adalah dari segi kekusutan uang, warna uang yang sedikit berubah karena terlalu sering digunakan, dan kebersihan uang.

3.6 Perancangan Proses

Pada bagian ini akan dijelaskan tentang proses membangun aplikasi identifikasi nominal uang kertas rupiah. Metode yang digunakan adalah algoritma *Modified K Nearest Neighbour*. *Flowchart* deskripsi aplikasi digambarkan pada Gambar 3.2.



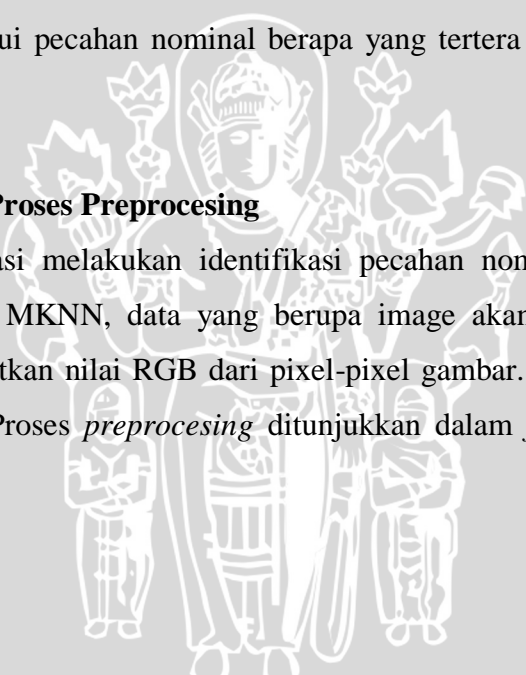
Gambar 3. 2 Flowchart Alir Proses Perangkat Lunak

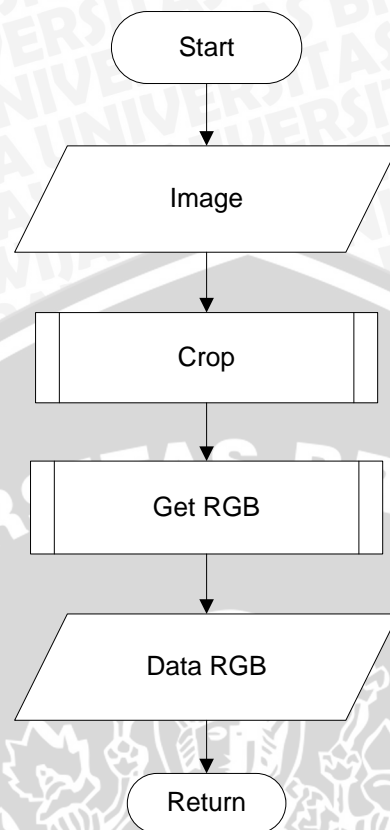
Untuk penjelasan flowchart di atas adalah sebagai berikut:

1. Input yang digunakan dalam aplikasi ini adalah berupa nilai k , image yaitu sebuah file gambar hasil scan uang kertas rupiah yang nantinya dapat digunakan sebagai data *training* dan data *testing*. Atau dapat juga input manual data *training* dan data *testing*.
2. Untuk data berupa image akan dilakukan *preprocessing* terlebih dahulu untuk mendapat hasil nilai RGB yang didapat dari pixel-pixel gambar.
3. Selanjutnya semua data *training* dan data *testing* dinormalisasikan menjadi range 0 sampai 1.
4. Metode yang digunakan adalah metode MKNN. Jadi data yang telah dinormalisasikan selanjutnya akan diproses menggunakan metode MKNN.
5. Output dari program ini adalah hasil identifikasi nominal pecahan uang kertas. Yaitu dapat diketahui pecahan nominal berapa yang tertera pada image yang diinputkan.

3.6.1 Perancangan Proses Preprocessing

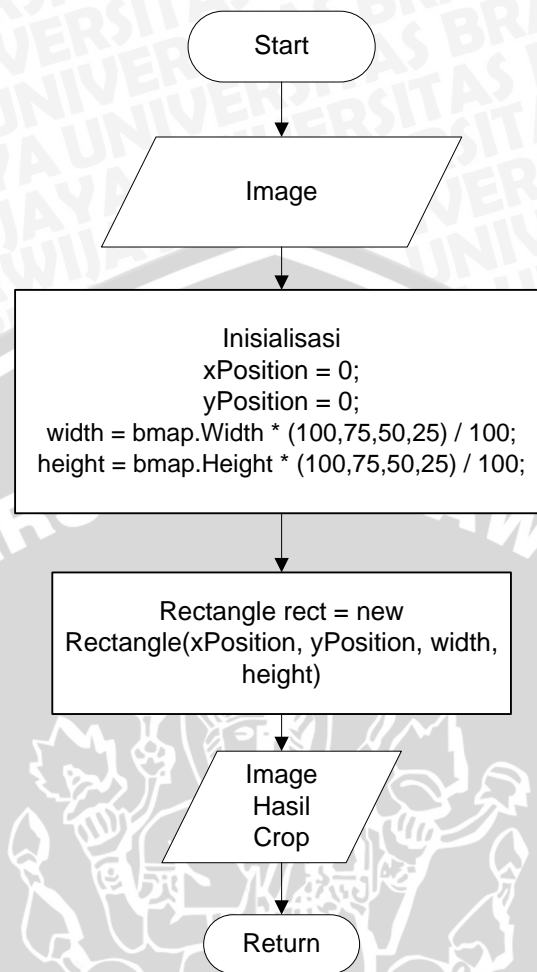
Sebelum aplikasi melakukan identifikasi pecahan nominal uang kertas menggunakan metode MKNN, data yang berupa image akan di olah terlebih dahulu untuk mendapatkan nilai RGB dari pixel-pixel gambar. Proses ini adalah proses *preprocessing*. Proses *preprocessing* ditunjukkan dalam *flowchart* sebagai berikut:





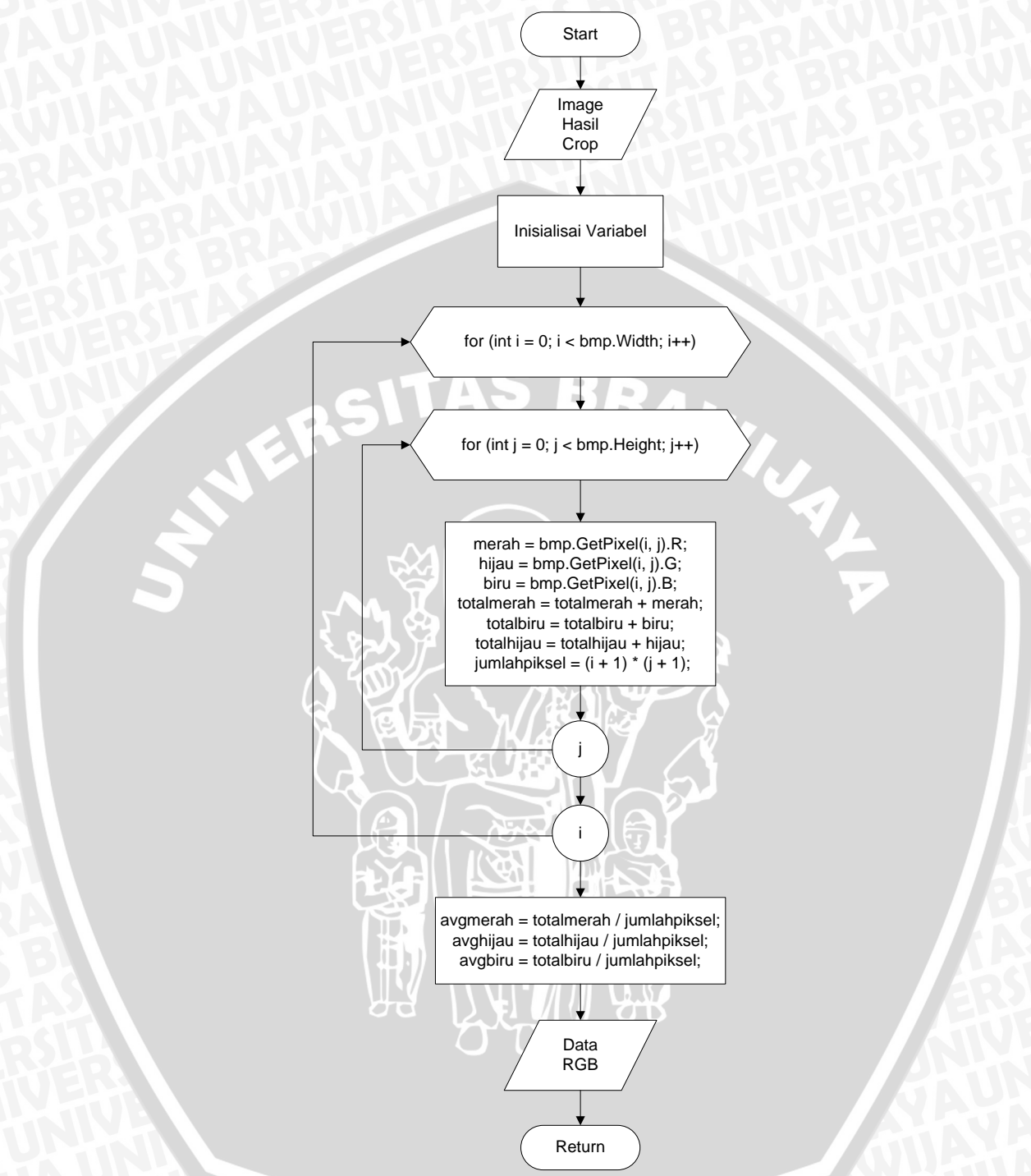
Gambar 3.3 Flowchart Preprocessing

1. Input berupa *image* hasil scan uang kertas.
2. Pada bagian ini gambar akan dilakukan proses pemotongan yaitu berupa tanpa potongan, potong 25%, potong 50%, dan potong 75%. Proses pemotongan dilakukan pengambilan dari kordinat (0,0) dari sudut atas kiri. Alasan dilakukan proses pemotongan ini karena penulis mempunyai hipotesa jika gambar diproses semua, maka akan didapatkan RGB yang kurang spesifik karena banyaknya warna yang didapat dari pixel gambar. Sedangkan jika hanya beberapa bagian gambar, warna yang didapat akan lebih spesifik karena pecahan yang satu dan yang lain pada bagian ini cukup berbeda, jadi dirasa akan lebih mendapatkan hasil yang optimal. Hal ini nantinya akan dibuktikan pada bagian pengujian. Agar penelitian ini menghasilkan gambar posisi pemotongan bagaimana yang menghasilkan hasil yang optimal. Proses pemotongan ditunjukkan dalam *flowchat* sebagai berikut:



Gambar 3. 4 Flowchart Crop pada Preprocessing

3. Get RGB adalah proses mendapatkan rata-rata nilai red, green, blue yang diperoleh dari pixel-pixel gambar. Jadi tiap titik pixel akan dihitung warnanya. Lalu akan warna yang diperoleh akan dibagi dengan jumlah total pixel. Maka akan diperoleh rata-rata RGB nya. Proses Get RGB ditunjukkan dalam *flowchart* sebagai berikut:

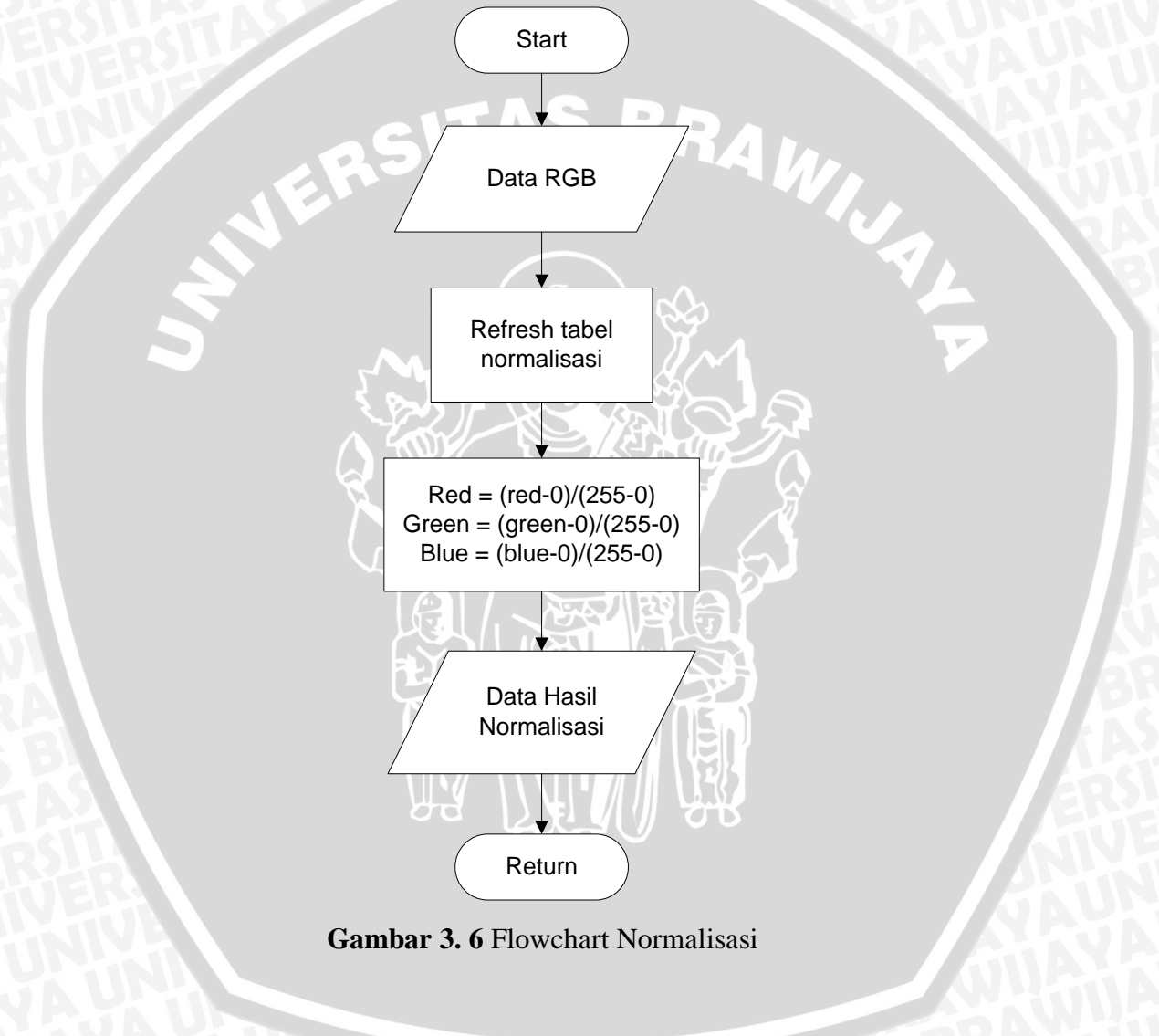


Gambar 3. 5 Flowchart GetRGB pada Preprocessing

4. Output berupa nilai rata-rata *red*, *green* dan *blue*. Nilai ini akan digunakan sebagai data identifikasi pecahan nominal uang dalam penelitian ini.

3.6.2 Perancangan Proses Normalisasi

Data yang digunakan (data input) yaitu data yang diperoleh dari *GetRGB* yaitu berupa nilai rata-rata *red*, *green*, dan *blue*. Data ini sebelum diproses harus di normalisai terlebih dahulu agar data berada pada *range* [0,1], sehingga sebaran datanya tidak terlalu jauh. Proses normalisai ditunjukkan dalam *flowchart* sebagai berikut:

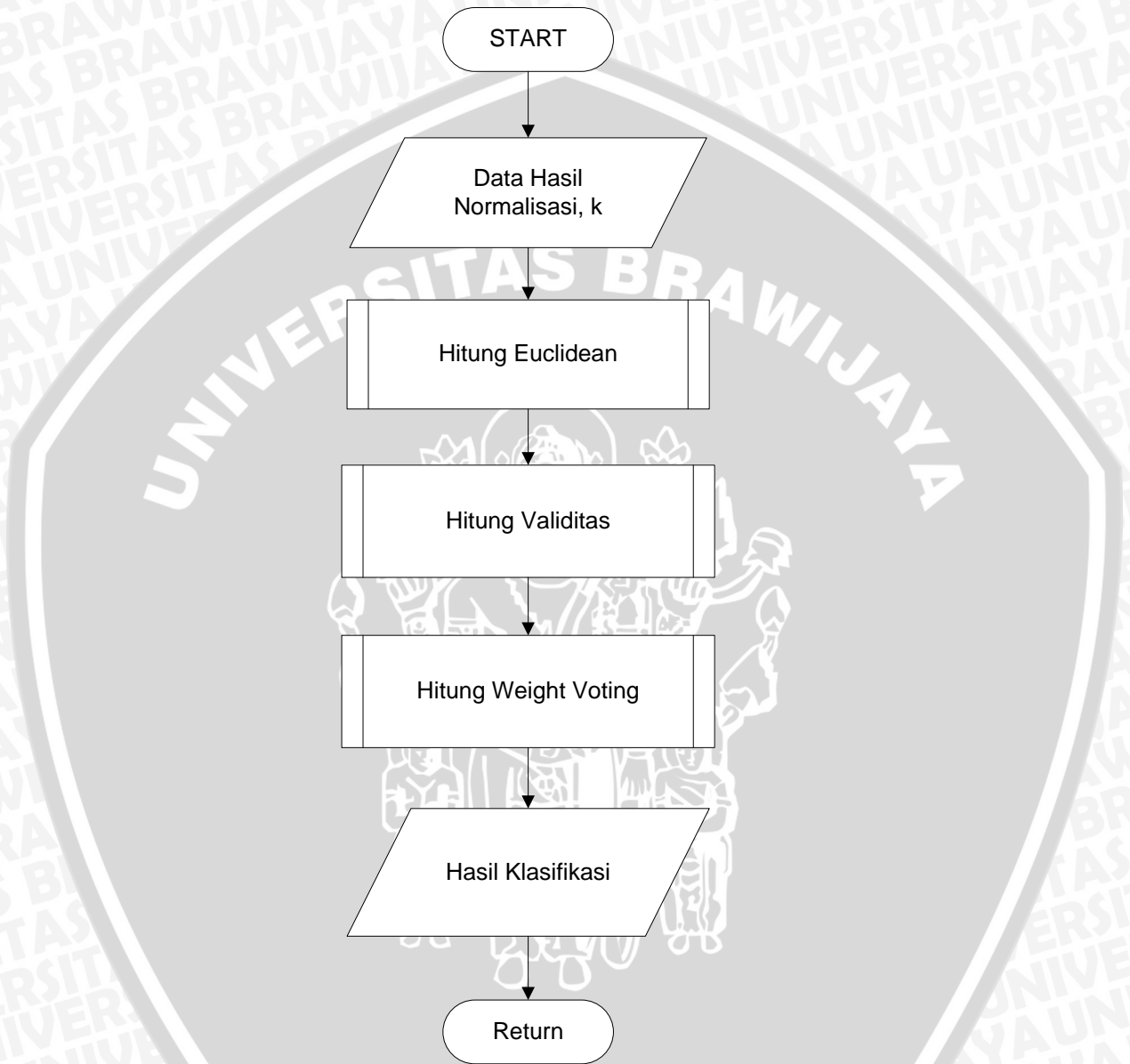


Gambar 3. 6 Flowchart Normalisasi

3.6.3 Perancangan Proses Metode Modified K-Nearest Neighbour

Aplikasi yang akan dibangun digunakan untuk mengidentifikasi pecahan nominal uang kertas. Metode *Modified K-Nearest Neighbour* digunakan untuk mengklasifikasikan objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tertentu.

Algoritma MKNN terdiri dari 3 proses utama, yaitu perhitungan jarak, perhitungan validitas data *training*, dan perhitungan *weight voting*. Proses algoritma MKNN ditunjukkan oleh Gambar 3.7 berikut.



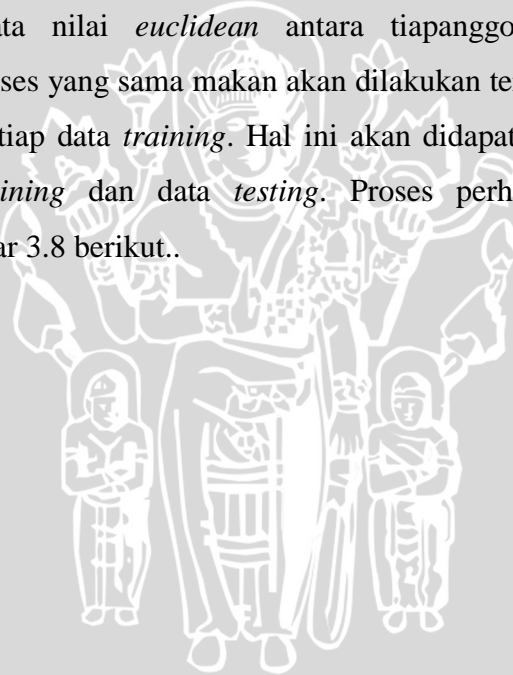
Gambar 3. 7 Flowchart proses metode MKNN

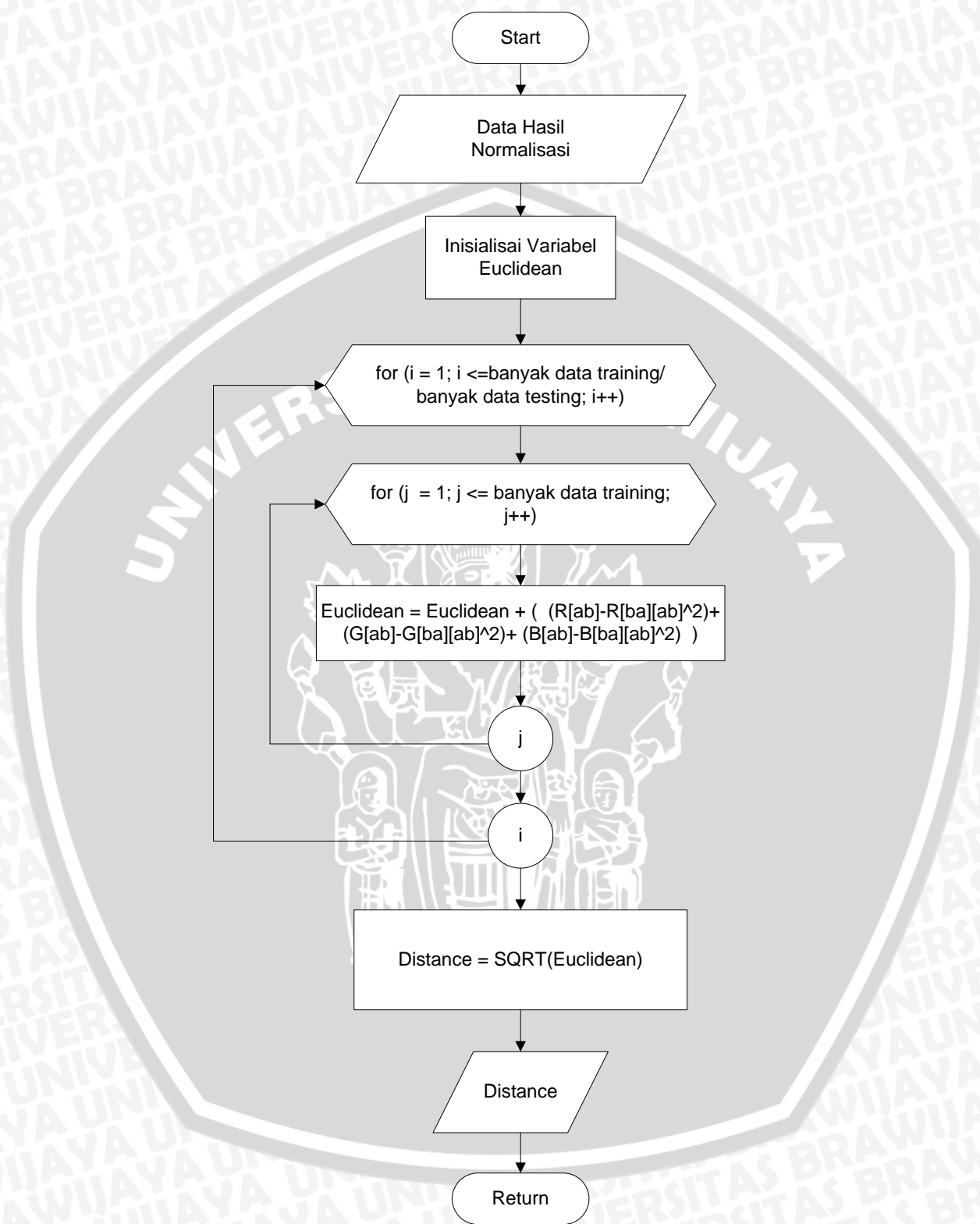
Pada MKNN terdapat 3 tahapan proses. Tahapan awal dimulai dengan memasukkan dataset RGB uang dan nilai k . Kemudian untuk dataset *training*, dan data *testing* dilakukan proses perhitungan jarak atau *euclidean*. Yaitu dihitung jarak antara tiap data *training* dan jarak antara data *testing* dengan tiap data *training*. Setelah dilakukan perhitungan *euclidean* lalu dilakukan perhitungan

validitas pada data *training*. Setelah itu dilakukan tahap perhitungan *weight voting* terhadap data uji. Tahapan perhitungan *weigh voting* ini adalah tahapan terakhir dari proses MKNN. Dari perhitungan *weigh voting* ini akan diambil hasil terbanyak sesuai dari nilai k yang sudah ditetapkan, maka akan dihasilkan output nominal pecahan uang kertas.

3.6.4 Perancangan Proses Hitung Jarak Euclidean

Perhitungan *euclidean* alur tahapannya terdiri dari beberapa tahapan antara lain dari dataset data *training* dan input data *testing*. Tahapan pertama adalah perhitungan terhadap dataset data *training* setelah input data akan dilakukan perhitungan *euclidean* sesuai dengan persamaan 2-1 sebanyak data maka akan didapatkan output data nilai *euclidean* antara tiap anggota data *training*. Selanjutnya dengan proses yang sama akan dilakukan terhadap data *testing* atau data uji terhadap tiap data *training*. Hal ini akan didapatkan output berupa jarak antara data *training* dan data *testing*. Proses perhitungan *euclidean* ditunjukkan oleh gambar 3.8 berikut..

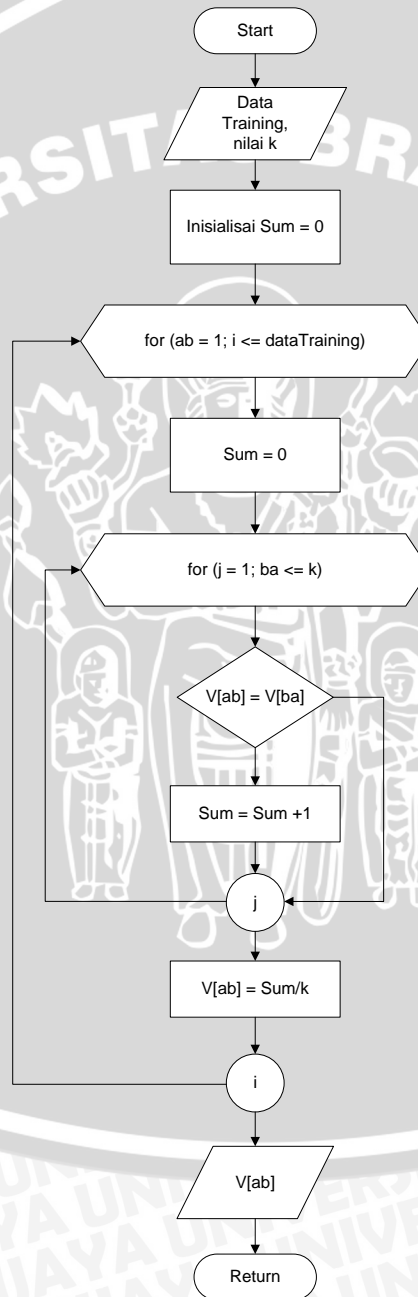




Gambar 3. 8 Flowchart proses Hitung Euclidean antara Tiap Anggota Data Training dan Data Testing dengan Tiap Anggota Data Training

3.6.5 Perancangan Proses Hitung Validitas

Langkah-langkah dalam proses perhitungan nilai validitas adalah memberikan input berupa dataset data *training* dan menentukan nilai k . Proses ini adalah proses untuk memperhitungkan nilai validitas dari data tetangga terdekatnya pada data *training*. Proses perhitungan validasi ditunjukkan oleh gambar berikut..

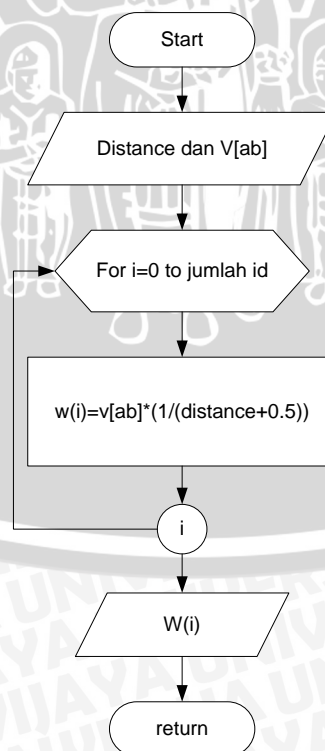


Gambar 3. 9 Flowchart proses Hitung Validasi

Pada alur flowchart perhitungan validitas dijelaskan alur tahapan yang terdiri dari beberapa tahapan yaitu menginputkan nilai k -nya yang dilakukan secara manual, lalu menghitung membandingkan kelas-kelas data *training* sesuai dengan ketepatan sebelumnya. Hal ini sesuai dengan persamaan validitas 2-2. Dengan ketentuan jika kelasnya sama, maka $V[x] = V[x]+1$ nilainya dan jika kelasnya berbeda maka $V[x] = V[x]$ nilainya 0 dilakukan perbandingan data sebanyak k yang sudah ditetapkan. Kemudian $V[x]$ akan dijumlah dan dibagi sebanyak k data yang telah ditetapkan. Maka akan didapatkan output data nilai validitas dari tiap data *training* pada proses identifikasi pecahan nominal uang kertas.

3.6.6 Perancangan Proses Hitung Weight Voting

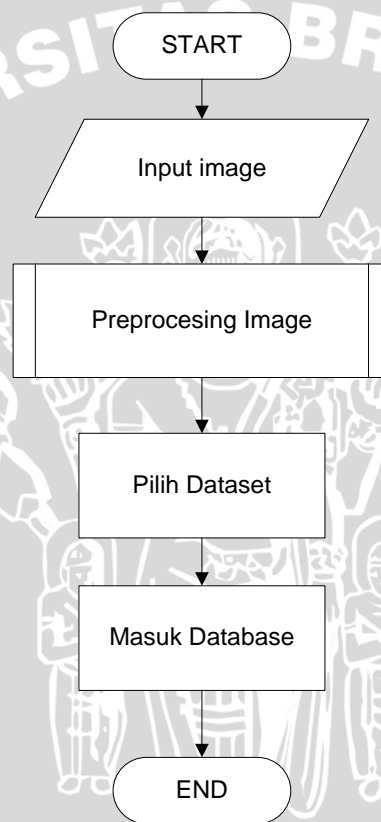
Langkah-langkah dalam proses perhitungan nilai *weight voting* adalah memberikan input berupa nilai jarak *euclidean* dan nilai validitas yang sudah didapat dari proses sebelumnya. Kemudian dihitung menggunakan persamaan (2-5). Proses ini adalah proses untuk memperhitungkan nilai pembobotan yang nantinya hanya akan diambil beberapa nilai teratas sesuai dengan k yang sudah ditetapkan. Proses perhitungan *euclidean* ditunjukkan oleh gambar berikut.



Gambar 3. 10 Flowchart proses hitung Weight Voting

3.7 Perancangan Proses Input Data Training

Langkah-langkah pada proses input data training ini adalah pertama memasukkan sebuah gambar uang lalu selanjutnya akan diproses dengan Preprocessing Image yaitu proses cropping dan mengkstrak warnanya untuk mendapatkan data sebuah RGB. Selanjutnya data RGB dimasukkan pada database data training dengan memilih database dataset. Proses perancangan ini nantinya akan berada pada form tersendiri yaitu form input data. Proses input data training ditunjukkan oleh gambar berikut.



Gambar 3. 11 Flowchart proses Input Data Training

3.8 Contoh Perhitungan Manual

Dari algoritma *Modified K-Nearest Neighbor* (MKNN) ini langkah – langkah dalam perhitungannya antara lain yaitu :

1. Menentukan nilai k atau tetangganya
2. Menghitung normalisasi data *training*
3. Menghitung jarak *euclidean* data *training*

4. Menghitung nilai validitas data *training*
5. Menghitung normalisasi data *testing*
6. Menghitung jarak *euclidean* data *testing*
7. Menghitung nilai pembobotan (*weight voting*)
8. Menentukan kelas dari data *testing* berdasarkan nilai k

Pada perhitungan manual ini, digunakan 14 dataset dimana dibagi menjadi 2 bagian yaitu 2 data *testing* dan 12 data *training*. Dataset yang digunakan dapat dilihat pada tabel 3.1 berikut.

Tabel 3. 1 Dataset

No	Jenis Pecahan	Red	Green	Blue
1	10000	165	92	143
2	20000	138	157	157
3	50000	116	124	172
4	100000	193	139	126
5	10000	155	100	151
6	20000	146	165	165
7	50000	124	132	180
8	100000	201	147	134
9	10000	182	109	160
10	20000	155	174	174
11	50000	133	140	189
12	100000	210	156	143
13	20000	140	169	166
14	100000	204	150	130

Tabel dataset diatas terdiri 4 pecahan saja. Nantinya pada implementasinya akan mempunyai 7 pecahan yaitu 1000, 2000, 5000, 10000, 20000, 50000, dan 100000. Pecahan diatas adalah sebagai klasifikasinya. Data yang digunakan mempunyai 3 variabel yaitu rata-rata warna merah sebagai red, rata-rata warna hijau sebagai green, dan rata-rata warna biru sebagai biru. Data red, green, dan blue ini didapat dari proses *preprocessing* yang sudah dijelaskan pada perancangan diatas.

Pada dataset untuk data nomor 1 hingga 12 digunakan sebagai data *training*. Sedangkan nomor 13 dan 14 digunakan sebagai data *testing* atau data

yang akan diuji. Pembagian data *training* dan data *testing* dapat dilihat pada tabel berikut:

Data Training :

Tabel 3. 2 Data Training

No	Jenis Pecahan	Red	Green	Blue
1	10000	165	92	143
2	20000	138	157	157
3	50000	116	124	172
4	100000	193	139	126
5	10000	155	100	151
6	20000	146	165	165
7	50000	124	132	180
8	100000	201	147	134
9	10000	182	109	160
10	20000	155	174	174
11	50000	133	140	189
12	100000	210	156	143

Data Testing :

Tabel 3. 3 Data Testing

No	Jenis Pecahan	Red	Green	Blue
13	20000	140	169	166
14	100000	204	150	130

3.8.1 Menentukan Nilai k

Proses pertama yang dilakukan adalah memasukkan nilai k yang nantinya digunakan sebagai pemilihan tetangga terdekat. Pada kasus ini penulis memberikan input nilai k=3.

3.8.2 Menghitung Normalisasi Data Training

Sebelum menghitung jarak *euclidean*, dataset dinormalisasikan terlebih dahulu. Proses normalisasi menggunakan persamaan 2-1 yaitu min-max normalisasi dengan rumus:

$$V' = \frac{V - \min A}{\max A - \min A} \times (new_{\max A} - new_{\min A}) + new_{\min A}$$

Perhitungan normalisasi atribut red no 1:

$$V' = \frac{V - \min A}{\max A - \min A} \times (\text{new}_{\max A} - \text{new}_{\min A}) + \text{new}_{\min A}$$

$$V' = \frac{165 - 0}{255 - 0} \times (1 - 0) + 0$$

$$V' = 0.64705882352941176470588235294118$$

Perhitungan normalisai atribut green no 1:

$$V' = \frac{V - \min A}{\max A - \min A} \times (\text{new}_{\max A} - \text{new}_{\min A}) + \text{new}_{\min A}$$

$$V' = \frac{92 - 0}{255 - 0} \times (1 - 0) + 0$$

$$V' = 0.36078431372549019607843137254902$$

Perhitungan normalisai atribut blue no 1:

$$V' = \frac{V - \min A}{\max A - \min A} \times (\text{new}_{\max A} - \text{new}_{\min A}) + \text{new}_{\min A}$$

$$V' = \frac{143 - 0}{255 - 0} \times (1 - 0) + 0$$

$$V' = 0.56078431372549019607843137254902$$

Jadi hasil perhitungan normalisasi adalah sebagai berikut:

Tabel 3. 4 Hasil Normalisai

No	Jenis Pecahan	Red	Green	Blue
1	10000	0.64706	0.36078	0.56078
2	20000	0.54118	0.61569	0.61569
3	50000	0.45490	0.48627	0.67451
4	100000	0.75686	0.54510	0.49412
5	10000	0.60784	0.39216	0.59216
6	20000	0.57255	0.64706	0.64706
7	50000	0.48627	0.51765	0.70588
8	100000	0.78824	0.57647	0.52549
9	10000	0.71373	0.42745	0.62745
10	20000	0.60784	0.68235	0.68235
11	50000	0.52157	0.54902	0.74118
12	100000	0.82353	0.61176	0.56078

3.8.3 Mengitung Jarak Euclidean Data Training

Proses pertama dari perhitungan jarak *euclidean* ini adalah menghitung jarak *euclidean* dari data *training* terlebih dahulu yaitu menghitung jarak

euclidean dari tiap data antara data *training*. Dan langkah berikutnya menghitung jarak *euclidean* data *testing* dengan tiap data *training*.

Untuk mendefinisikan jarak antara dua titik yaitu titik pada antara tiap data *training* (x) dan titik data *training* (x) dengan titik pada data *testing* (y) maka digunakan rumus *Euclidean*, seperti yang ditunjukkan pada persamaan 2-2 dengan rumus

$$d(i) = \sqrt{\sum_{i=1}^n (x_{2i} - x_{1i})^2}$$

Proses pertama menghitung jarak antara tiap data *training* yaitu sebagai berikut :

$$d(1,2) = \sqrt{\sum_{i=1}^n (x_{2i} - x_{1i})^2}$$

$$d(1,2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

$$d(1,2)$$

$$= \sqrt{(0.64706 - 0.54118)^2 + (0.36078 - 0.61569)^2 + (0.56078 - 0.61569)^2}$$

$$= 0.281425$$

$$d(1,3)$$

$$= \sqrt{(0.64706 - 0.45490)^2 + (0.36078 - 0.48627)^2 + (0.56078 - 0.67451)^2}$$

$$= 0.256136$$

Selanjutnya melakukan perhitungan yang sama untuk semua data *training*. Hasil perhitungan *euclidean* data *training* ini seperti ditunjukkan tabel 3.5 berikut.

Tabel 3. 5 Hasil Perhitungan Jarak Euclidean Data Training

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0.28 1425	0.25 6136	0.22 4662	0.05 9214	0.30 8136	0.26 7415	0.2601 86548	0.11 547	0.34 601	0.28 9347	0.30 6811
2		0	0.16 6286	0.25 7453	0.23 4443	0.05 4339	0.14 4088	0.2659 15898	0.25 5625	0.11 547	0.14 3446	0.28 7668
3			0	0.35 6626	0.19 7563	0.20 1112	0.05 4339	0.3761 02761	0.26 9563	0.24 8796	0.11 3251	0.40 5669
4				0	0.23 4967	0.26 0305	0.34 4697	0.0543 38849	0.18 2974	0.27 6547	0.34 1199	0.11 547
5					0	0.26 3125	0.20 8471	0.2663 7816	0.11 7057	0.30 389	0.23 2929	0.30 9407
6						0	0.16 6286	0.2574 53295	0.26 1807	0.06 1131	0.14 5151	0.26 7731
7							0	0.3566 25648	0.25 6945	0.20 606	0.05 8954	0.37 9015
8								0	0.19 5332	0.26 1454	0.34 4072	0.06 1131
9									0	0.28 1425	0.25 4237	0.22 4662
10										0	0.16 9355	0.25 7453
11											0	0.35 7293
12												0

Jadi hasil euclidean dapat dilihat pada tabel diatas yaitu dengan membacanya dilihat id dari kolom pertama paling kiri dihubungkan dengan id baris paling atas.

3.8.4 Menghitung Nilai Validitas Data Training

Nilai Validasi dihitung berdasarkan rata-rata dari nilai kedekatan data satu dengan tetangganya sebanyak k . Nilai kedekatan bernilai 0 dan 1. Nilai 0 diberikan jika data terdekat tidak dalam kategori yang sama, sedangkan sebaliknya nilai 1 diberikan jika data terdekat berada dalam satu kategori yang sama. Dilihat dari tabel jarak *euclidean* diatas, jarak terdekat data 1 adalah dengan data 5 yaitu 0.059214, karena data 1 dan data 2 berada dalam kategori yang sama yaitu 10.000, maka diberi nilai 1. Perhitungan ini ditentukan sampai batas k yang sudah ditentukan di awal. Pada kasus ini $k=3$. Jadi tiap data training ditentukan validasinya sampai 3 tetangga terdekat. Seperti pada persamaan 2-3, selanjutnya dihitung nilai validitasnya menggunakan rumus dibawah ini.

$$\begin{aligned}
 \text{Validitas}(x=1) &= \frac{1}{k} \sum_{i=1}^k S(\text{label}(x), (\text{label}(N_i(x)))) \\
 &= \frac{1}{3} \sum_{i=1}^3 S(\text{label}(x=3), (\text{label}(N_i(x=0)))) \\
 &= \frac{1}{3} \times (1+1+1) \\
 &= \frac{3}{3} \\
 &= 1
 \end{aligned}$$

Melakukan perhitungan yang sama untuk semua data training. Hasil lengkap ditunjukkan pada tabel 3.6 dibawah ini.

Tabel 3. 6 Validasi

data	k=1	k=2	k=3	sum(k)	validasi
1	1	1	0	2	0.666666667
2	1	1	0	2	0.666666667
3	1	1	0	2	0.666666667
4	1	1	0	2	0.666666667
5	1	1	0	2	0.666666667
6	1	1	0	2	0.666666667
7	1	1	0	2	0.666666667
8	1	1	0	2	0.666666667
9	1	1	0	2	0.666666667
10	1	1	0	2	0.666666667
11	1	1	0	2	0.666666667
12	1	1	0	2	0.666666667

3.8.5 Menghitung Normalisasi Data Testing

Data *testing* ini diambil dari dataset diatas. Peneliti mengambil 2 data untuk dilakukan *testing*. Data *testing* ditunjukkan dalam tabel 3.7 dibawah ini.

Tabel 3. 7 Data Testing

No	Jenis Pecahan	Red	Green	Blue
13	????	140	169	166
14	????	204	150	130

Langkah pertama adalah menormalisasikan data *testing* terlebih dahulu. Proses normalisasi data *testing* sama dengan proses normalisasi data training

diatas. Jadi hasil normalisasi data *testing* ditunjukkan dalam tabel 3.8 sebagai berikut.

Tabel 3. 8 Normalisai Data Testing

No	Jenis Pecahan	Red	Green	Blue
13	????	0.549019608	0.662745	0.65098
14	????	0.8	0.588235	0.509804

3.8.6 Menghitung Jarak Euclidean Data Testing

Jarak *euclidean* data *testing* didapat dari menghitung jarak data *testing* tersebut dengan semua data training. Proses perhitungan yang digunakan sama dengan proses menghitung jarak *euclidean* pada data training yaitu menggunakan persamaan 2-2. Hasil perhitungan untuk data *testing* pertama data(13) ditunjukkan tabel 3.9 berikut.

Tabel 3. 9 Hasil Perhitungan Jarak Euclidean Data Testing 13 dengan tiap Data Training

Data	Jarak
1	0.330041415
2	0.059344102
3	0.201379327
4	0.285736804
5	0.28308728
6	0.028549451
7	0.167345791
8	0.283575783
9	0.288175264
10	0.069490373
11	0.147723986
12	0.293410916
13	0

Untuk data *testing* kedua data(14) dilakukan perhitungan yang sama dengan proses perhitungan jarak *euclidean* seperti diatas. Hasil perhitungan data *testing* yang kedua ditunjukkan dalam tabel 3.10 berikut.

Tabel 3. 10 Hasil Perhitungan Jarak Euclidean Data Testing 14 dengan tiap Data Training

Data	Jarak
1	0.27879
2	0.280988
3	0.395748



4	0.06299
5	0.286623
6	0.27209
7	0.376634
8	0.022866
9	0.217108
10	0.274874
11	0.364136
12	0.060879
13	0

3.8.7 Menghitung Nilai Pembobotan (Weight Voting)

Nilai hasil *weight voting* digunakan untuk menentukan kelas dari data *testing*. Nilai *weight voting* dihitung dengan menggunakan persamaan 2-5. Sebagai contoh dilakukan *weight voting* untuk data *testing* pertama sebagai berikut.

$$\begin{aligned}
 W_{(1,1)} &= Validitas \times \frac{1}{d + 0.5} \\
 &= 1 \times \frac{1}{0.330041415 + 0,5} \\
 &= 1.204759
 \end{aligned}$$

Selanjutnya melakukan perhitungan *weight voting* yang sama untuk semua data *testing* terhadap tiap data training. Hasil perhitungan *weight voting* data *testing* pertama ditunjukkan pada tabel 3.11 sebagai berikut.

Tabel 3. 11 Hasil Perhitungan Weight Voting Data Testing 13

Data	Weight
1	0.803173
2	1.191872
3	0.950508
4	0.848461
5	0.851331
6	1.261314
7	0.998982
8	0.8508
9	0.845836
10	1.170637
11	1.029245
12	0.840254



Selanjutnya dengan proses yang sama dihitung weight voting pada data *testing* kedua. Dan hasil perhitungan weight voting data *testing* kedua ditunjukkan tabel 3.12 sebagai berikut.

Tabel 3. 12 Hasil Perhitungan Weight Voting Data Testing 14

Data	Weight
1	0.856029
2	0.85362
3	0.744257
4	1.184154
5	0.847504
6	0.863457
7	0.760485
8	1.275023
9	0.929661
10	0.860355
11	0.771483
12	1.18861

3.8.8 Menentukan Kelas Dari Data Testing Berdasarkan Nilai K

Setelah didapat nilai *weight voting*, kemudian nilai tersebut diurutkan dari yang terbesar samapi terkecil, kemudian diambil sebanyak k untuk menentukan kelasnya.

Hasil *weight voting* data testing pertama :

Tabel 3. 13 Weight Voting Data Testing 13

Data	Weight
1	0.803173
2	1.191872
3	0.950508
4	0.848461
5	0.851331
6	1.261314
7	0.998982
8	0.8508
9	0.845836
10	1.170637
11	1.029245
12	0.840254

Pada tabel weight voting diatas diambil 3 terbesar teratas sesuai dengan nilai k yang telah ditetapkan. Dan hasilnya ditunjukkan pada tabel 3.14 berikut.

Tabel 3. 14 Hasil Weight Voting 3 teratas Data Testing 13.

Data	Weight	Kelas
6	1.261314	20.000
2	1.191872	20.000
10	1.170637	20.000

Dari tabel 3.14, dapat diambil kesimpulan bahwa data *testing* pertama termasuk dalam kelas 20.000, karena jumlah dari 3 tetangga terdekat diatas menunjukkan semuanya menghasilkan kelas 20.000. Jadi hasil *testing* termasuk kelas 20.000

Perhitungan yang sama dilakukan untuk data *testing* kedua. Dari data tersebut diambil weigt voting 3 terbesar teratas sesuai dengan nilai k. Dan hasilnya ditunjukkan pada tabel 3.15 berikut.

Tabel 3. 15 Hasil Weight Voting 3 teratas Data Testing 13.

Data	Weight	Kelas
8	1.275023	100.000
12	1.18861	100.000
4	1.184154	100.000

Dari tabel 3.15, dapat diambil kesimpulan bahwa data *testing* kedua termasuk dalam kelas 100.000, karena jumlah dari 3 tetangga terdekat diatas menunjukan semuanya menghasilkan kelas 100.000. Jadi hasil *testing* termasuk kelas 100.000

3.9 Manajemen Data

Struktur basis data yang digunakan oleh Identifikasi Uang Kertas Rupiah Dengan Menggunakan Metode *Modified K-Nearest Neighbor* (MK-NN) adalah sebagai berikut:

3.9.1 Rancangan Tabel Dataset

Tabel 3. 16 Rancangan Tabel Dataset

Field	Tipe	Length	Keterangan
Id	Integer	3	Untuk menyimpan nomer urutan data.
red	Integer	5	Untuk menyimpan data rata-rata warna merah dari sebuah gambar.
green	Integer	5	Untuk menyimpan data rata-rata warna hijau dari sebuah gambar.
blue	Integer	5	Untuk menyimpan data rata-rata warna biru dari sebuah gambar.
jenis	Integer	7	Untuk menyimpan jenis kategori

nominal.

Tabel dataset digunakan untuk menyimpan data training yang akan digunakan sistem.

3.9.2 Rancangan Tabel Normalisasi

Tabel 3. 17 Rancangan Tabel Normalisasi

Field	Type	Length	Keterangan
id	Integer	3	Untuk menyimpan nomer urutan data.
red	Integer	5	Untuk menyimpan nilai hasil normalisasi dari data rata-rata red.
green	Integer	5	Untuk menyimpan nilai hasil normalisasi dari data rata-rata green.
blue	Integer	5	Untuk menyimpan nilai hasil normalisasi dari data rata-rata blue.
jenis	Integer	7	Untuk menyimpan jenis kategori nominal.

Tabel Normalisasi digunakan sebagai tempat menyimpan hasil perhitungan normalisasi dari data pada tabel dataset.

3.9.3 Rancangan Tabel Euclidean

Tabel 3. 18 Rancangan Tabel Euclidean

Field	Type	Length	Keterangan
idasal	Integer	4	Untuk menyimpan nomer data asal.
idtujuan	Integer	4	Untuk menyimpan nomer data tujuan.
jarak	Double	4	Untuk menyimpan hasil perhitungan jarak dari data asal ke data tujuan.

Tabel *Euclidean* digunakan sebagai tempat menyimpan hasil perhitungan *euclidean* data training.

3.9.4 Rancangan Tabel Validasi

Tabel 3. 19 Rancangan Tabel Validasi

Field	Type	Length	Keterangan
id	Integer	3	Untuk menyimpan nomer urutan data.
k1	Integer	3	Untuk menyimpan nilai dari hasil perhitungan tetangga terdekat 1.
k2	Integer	3	Untuk menyimpan nilai dari hasil perhitungan tetangga terdekat 2.
k3	Integer	3	Untuk menyimpan nilai dari

sum	Integer	3	hasil perhitungan tetangga terdekat 3.
valdiasi	Double		Untuk menyimpan nilai dari hasil penjumlahan k1, k2, k3 Untuk menyimpan nilai hasil perhitungan validasi.

Tabel validasi digunakan sebagai tempat menyimpan hasil perhitungan validasi data training.

3.9.5 Rancangan Tabel Testing

Tabel 3. 20 Rancangan Tabel Testing

Field	Type	Length	Keterangan
id	Integer	3	Untuk menyimpan nomer urutan data.
red	Integer	5	Untuk menyimpan data rata-rata warna merah dari sebuah gambar.
green	Integer	5	Untuk menyimpan data rata-rata warna hijau dari sebuah gambar.
blue	Integer	5	Untuk menyimpan data rata-rata warna biru dari sebuah gambar.
nred	Double		Untuk menyimpan nilai hasil normalisasi dari data rata-rata red.
ngreen	Double		Untuk menyimpan nilai hasil normalisasi dari data rata-rata green.
nblue	Double		Untuk menyimpan nilai hasil normalisasi dari data rata-rata blue.

Tabel *testing* digunakan untuk menyimpan data *testing* yang akan diuji.

3.9.6 Rancangan Tabel Testingeuclidean

Tabel 3. 21 Rancangan Tabel Testingeuclidean

Field	Type	Length	Keterangan
idasal	Integer	4	Untuk menyimpan nomer data asal.
idtujuan	Integer	4	Untuk menyimpan nomer data tujuan.
jarak	Double	4	Untuk menyimpan hasil perhitungan jarak dari data asal ke data tujuan.

Tabel Testing *euclidean* digunakan sebagai tempat menyimpan hasil perhitungan *euclidean* data *testing*.



3.9.7 Rancangan Tabel Testingweight

Tabel 3. 22 Rancangan Tabel Testingweight

Field	Type	Length	Keterangan
id	Integer	4	Untuk menyimpan nomer urutan data.
weight	double	3	Untuk menyimpan nilai dari hasil perhitungan pembobotan.
rank	Integer	4	Untuk menyimpan hasil urutan dari weight terbesar sampai terkecil.

Tabel *Testingweight* digunakan sebagai tempat menyimpan hasil perhitungan pembobotan data testing.

3.9.8 Rancangan Tabel Rank

Tabel 3. 23 Rancangan Tabel Rank

Field	Type	Length	Keterangan
Idasal	Integer	4	Untuk menyimpan nomer data asal.
Idtujuan	Integer	4	Untuk menyimpan nomer data tujuan.
Rank	Integer	4	Untuk menyimpan hasil urutan.

Tabel *Rank* digunakan sebagai tempat menyimpan hasil urutan dari urutan pertama sampai terakhir.

3.9.9 Rancangan Tabel K

Tabel 3. 24 Rancangan Tabel K

Field	Type	Length	Keterangan
Rank	Integer	4	Untuk menyimpan hasil urutan dari terkecil sampai terbesar berdasarkan input k.
Jenis	Integer	4	Untuk menyimpan nilai jenis.

K digunakan sebagai tempat menyimpan hasil perhitungan urutan sesuai dengan k yang diinputkan.

3.10 Perancangan Uji Coba

Proses pengujian ini dilakukan untuk mengetahui tingkat akurasi aplikasi. Proses pengujian ini dilakukan sebanyak n kali dengan menggunakan jumlah data tes yang telah ditetapkan. Pada tabel merupakan tabel yang akan digunakan untuk perhitungan akurasi untuk mencatat hasil dalam proses pengujian akurasi datanya. Hasil setiap kali uji coba akan dianalisis dan dihitung akurasinya. Dalam penelitian ini setiap pengujian akan dilakukan menggunakan nilai $k=1$ sampai $k=30$. Setiap pengujian juga akan dilakukan dengan menggunakan 3 skenario pada data *training* dengan data *testing* tetap yaitu 21 data. Jumlah data *training* yang digunakan untuk pengujian adalah 84, 63 dan 42. Dan juga akan dilakukan pengujian terhadap pengaruh data yang didapat dari *image*. Yaitu data yang diperoleh dari hasil *crop image* 75%, 50% dan 25%, dan data yang didapat tanpa *crop image*.

Tabel 3. 25 Rancangan Tabel Pengujian Akurasi

Data Image	Jumlah Data Training : Data Testing (%)	Nilai k	Nilai Akurasi (%)
Rata – Rata			

Keterangan :

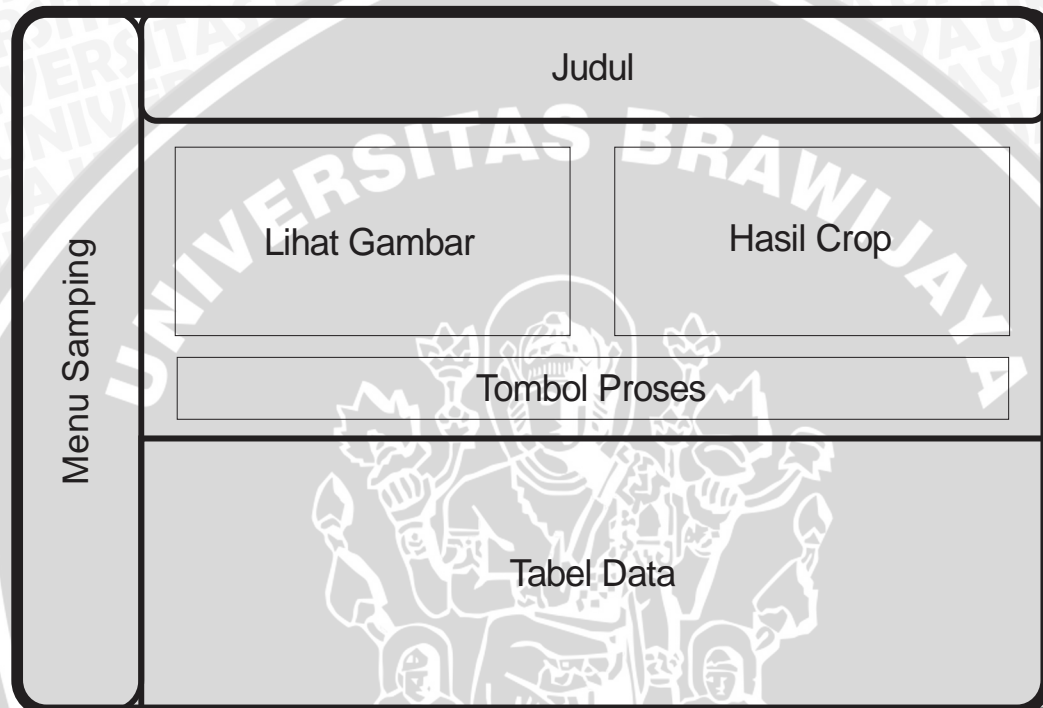
- Data Image : data kategori dengan crop dan tanpa crop
- Jumlah Data Trainig : jumlah data *training* yang digunakan adalah 84, 63, dan 41.
- Nilai Akurasi : jumlah hasil akurasi yang benar
- Rata – rata akurasi : jumlah akurasi data dibagi dengan jumlah data uji

3.11 Perancangan Antarmuka

Perancangan antarmuka pada penelitian identifikasi nominal pecahan uang ini adalah sebagai berikut :

1. Home

Rancangan antarmuka tampilan awal dapat dilihat pada gambar 3.11 berikut.



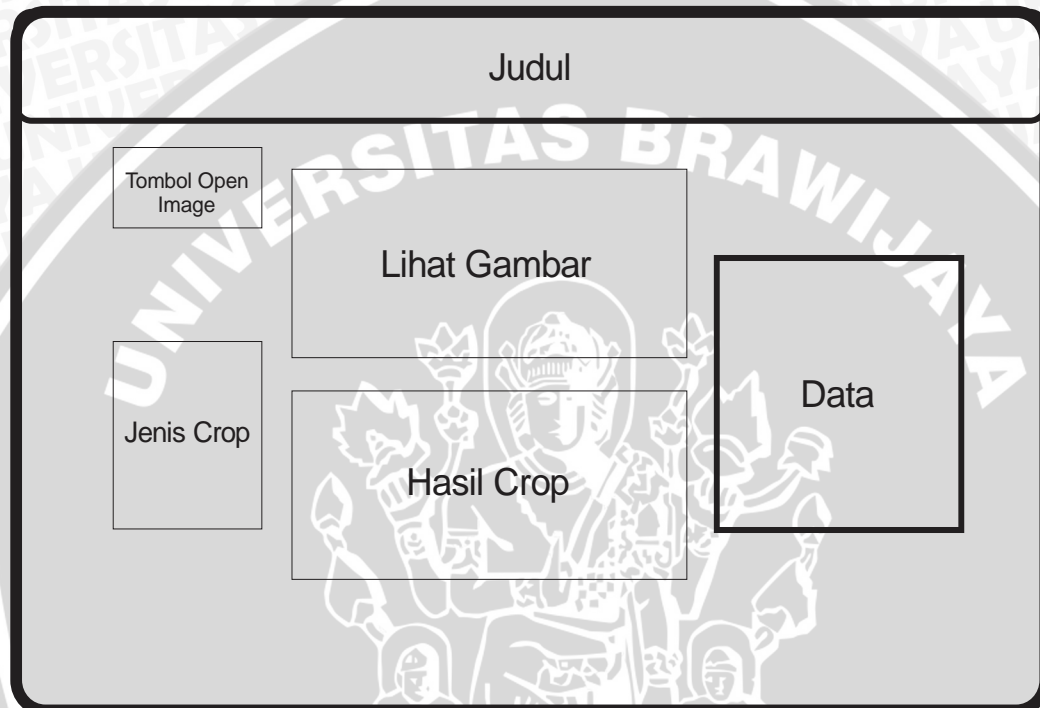
Gambar 3.12 Rancangan Antarmuka Home

Keterangan :

- Pada menu samping terdapat beberapa pilihan menu, yaitu:
 - Menu Input Data
 - Menu Satu Proses
 - Menu Pengujian
- Lihat Gambar adalah tempat tampilan saat membuka gambar uang yang akan di proses.
- Hasil Crop adalah tempat tampilan gambar uang yang kondisinya sudah di lakukan proses cropping.
- Tombol Proses adalah tempat tombol-tombol perintah yang digunakan dalam suatu proses.
- Tabel Data adalah tempat menampilkan data.

2. Menu Input Data

Menu Input Data ini adalah form yang digunakan untuk menginputkan nilai RGB yang didapat dari hasil ekstraksi RGB yang didapat dari gambar uang yang sudah di proses crop atau tanpa crop. Data RGB yang sudah didapatkan akan di masukkan pada tabel data *training* berdasarkan kategori jenis crop. Rancangan antarmuka Menu Input Data dapat dilihat pada gambar 3.12.



Gambar 3. 13 Rancangan Menu Input Data

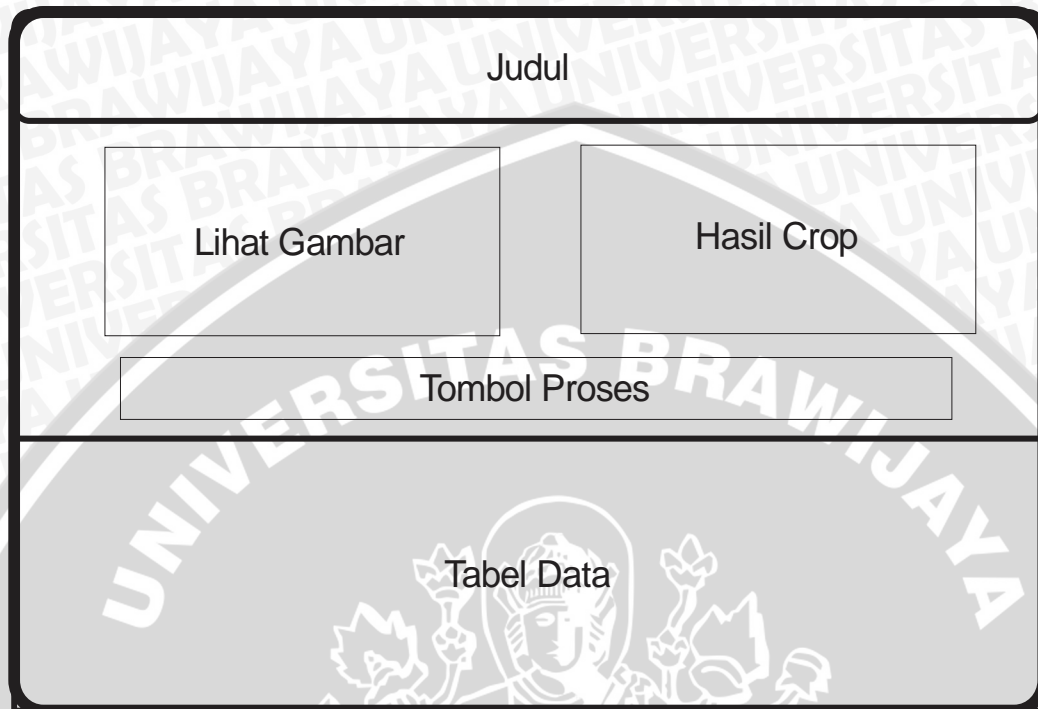
Keterangan :

- Tombol Open Image digunakan untuk membuka gambar yang akan dipilih. Selanjutnya akan langsung ditampilkan pada Lihat Gambar.
- Jenis Crop adalah pilihan untuk memilih crop yang mana yang akan digunakan pada gambar uang.
- Data adalah untuk menampilkan hasil rgb yang didapat dari hasil crop. Dan data ini yang akan di masukkan dalam tabel pada data *training*.

3. Menu Satu Proses

Menu Satu proses ini adalah form yang digunakan untuk mengidentifikasi uang kertas rupiah dengan hanya menggunakan satu tombol proses. Berbeda dengan form pada tampilan awal, jika pada tampilan awal setiap step proses

menggunakan satu tombol. Form ini hanya untuk mempercepat proses saja. Rancangan antarmuka menu satu proses dapat dilihat pada gambar 3.13 berikut.



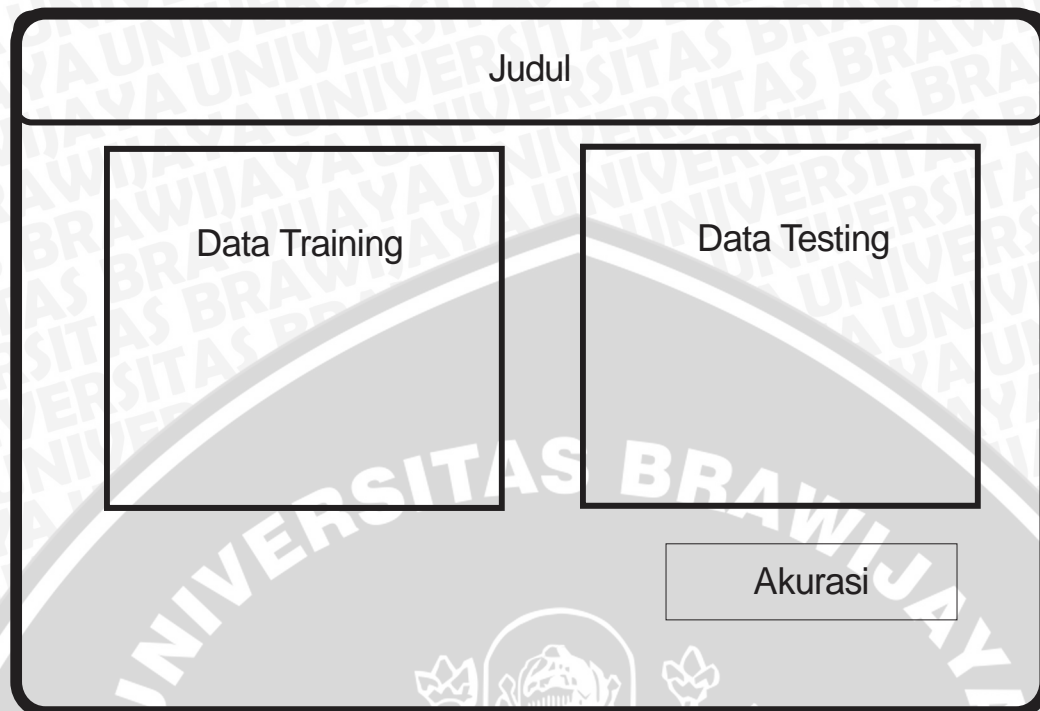
The image shows a wireframe of a home menu interface. It consists of a rounded rectangular container with a thick black border. At the top center is the label 'Judul'. Below this, there are two rectangular boxes side-by-side: 'Lihat Gambar' on the left and 'Hasil Crop' on the right. Underneath these two boxes is a single, wider rectangular box labeled 'Tombol Proses'. At the bottom of the container is a large, light gray area labeled 'Tabel Data'. The background of the wireframe features a faint watermark of the Universitas Brawijaya logo and name.

Gambar 3.14 Rancangan Antarmuka Home

Keterangan :

- a. Lihat Gambar adalah tempat tampilan saat membuka gambar uang yang akan di proses.
 - b. Hasil Crop adalah tempat tampilan gambar uang yang kondisinya sudah di lakukan proses cropping.
 - c. Tombol Proses adalah tempat tombol-tombol perintah yang digunakan dalam suatu proses.
 - d. Tabel Data adalah tempat menampilkan data.
4. Menu Pengujian

Menu pengujian ini adalah form yang digunakan untuk proses pengujian aplikasi. Pada form ini digunakan untuk mengetahui akurasi aplikasi yang berdasarkan variabel-variabel yang mempengaruhi. Rancangan antarmuka Menu Pengujian dapat dilihat pada gambar 3.14.



Gambar 3. 15 Rancangan Menu Pengujian

Keterangan :

- Data *Training* adalah tempat untuk menampilkan data *training*.
- Data *Testing* adalah tempat untuk menampilkan beberapa data *testing* yang akan ditesting.
- Akurasi adalah hasil akurasi dari pengujian pada data *training* yang hasil data *testing* dibandingkan dengan data asli.