

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

Menganalisis masalah merupakan langkah yang harus dilakukan dalam tahapan analisis sistem. Analisis sistem dapat didefinisikan sebagai penguraian dari sistem yang lengkap ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi permasalahan, hambatan, kesempatan, dan kebutuhan yang diharapkan dapat diusulkan perbaikannya.

Permasalahan yang ada dikarenakan belum adanya sistem yang dapat mengenali karakter number *Captcha*, pada aplikasi e-banking sebelumnya pengguna diharuskan untuk memasukkan karakter number *Captcha* dengan cara mengetik melalui keyboard, saat pengguna ingin memasukkan karakter number *Captcha* dengan mengetik melalui keyboard pengguna bisa melakukan kesalahan dalam memasukkan karakter number *Captcha* tersebut. Maka dibutuhkan sistem yang dapat membaca dan mengenali karakter number *Captcha*. Pada tahap pengumpulan data sebelumnya dilakukan proses pengunduhan citra *Captcha* dari halaman uji `ibank.bni.co.id` secara manual. Selanjutnya citra dicropping dan diresize perkarakturnya untuk mendapatkan citra dataset yang sudah berbentuk template karakter biner dari karakter 0-9 yang nantinya dijadikan dataset template acuan.

Dalam aplikasi ini, sistem akan dibagi dalam 3 tahapan utama, pertama adalah *pre-processing* data awal, pelabelan karakter dan yang kedua adalah pengenalan karakter dengan menggunakan algoritma *Template Matching* untuk mengenali pola citra karakter *Captcha*. Berdasarkan uraian maka penelitian ini ingin membuat suatu aplikasi yang berjudul ***“Pengolahan Citra digital Untuk Mendeteksi karakter Citra Captcha Pada Internet Banking Menggunakan Metode Template Matching”***.

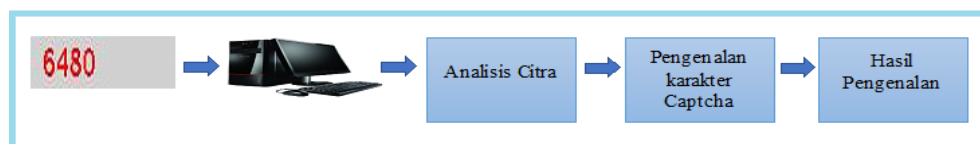
3.2. Hasil Analisis

Dari hasil analisis yang telah dilakukan diatas, maka dibutuhkan suatu sistem yang dapat mengenali karakter pola pada citra *Captcha*, serta mengetahui akurasi hasil pengenalan karakter. Dalam hal ini akan dibuat suatu sistem untuk mengenali citra karakter *Captcha* dengan menerapkan metode *Template Matching*. tahap berikutnya diperlukan data pembelajaran, data tersebut diperoleh dari hasil proses pengunduhan yang didapat dari halaman uji ibank.bni.co.id dengan resolusi citra asli 120 x 20 piksel, citra yang didapat kemudian dipotong, di resize menjadi 11 x 14 piksel dan dijadikan kebentuk citra biner untuk disimpan dalam database sebagai citra template acuan, sebelum nanti menuju tahapan pendeteksian karakter dengan algoritma *template matching*.

Pre-processing dilakukan sebagai persyaratan sebelum masuk pada proses pelabelan, tujuannya adalah untuk mendapatkan objek citra biner yang tanpa gangguan. Pada tahapan *pre-processing* terdapat beberapa proses diantaranya proses *grayscaleing*, *adjustmen*, *binary* dan *thersholding*. Setelah objek citra *biner* didapatkan, selanjutnya masuk pada proses pelabelan (*cropping*) per tiap karakter untuk memudahkan proses pencocokan citra karakter dengan dataset citra template yang sudah dibuat menggunakan algoritma *Template Matching*.

3.2.1. Deskripsi Sistem

Diskripsi sistem ini membahas tentang bagaimana proses dimulai hingga mampu menyelesaikan permasalahan yang dibuat. Berikut adalah gambaran dari perancangan sistem tersebut:



Gambar 3.1 Perancangan Umum Sistem


Dari gambar 3.1 diatas menunjukkan alur sistem yang akan dibuat menggunakan objek citra yang telah didapatkan dari proses pengunduhan dari halaman uji ibank.bni.co.id. Dari luas asli citra

Captcha 120 x 20 piksel akan dijadikan sebagai bahan untuk data sampel gambar (*image*) sehingga bisa dilakukan pemrosesan data menggunakan proses pengolahan citra (dalam hal ini memanfaatkan bahasa pemrograman **MATLAB R2013a 8.1.0.604** sebagai media pemrosesan data digital) dan juga menggunakan sistem operasi *Microsoft Windows 10 Premium 32-bit*. Kemudian dilakukan proses analisis citra untuk menghasilkan citra objek yang dapat mengenali citra karakter yang sesuai dengan syarat dan kondisi yang sudah ditentukan sebelumnya.

3.2.2 Metode Pengumpulan Data



Image yang digunakan dalam penelitian ini adalah data *image* standart *font* karakter berupa numeric *Captcha* yang telah diunduh secara manual dari halaman uji ibank.bni.co.id. kemudian untuk mendapatkan data set karakter, setiap citra dipotong berdasarkan wilayah objek perkarakturnya menggunakan *paint.net* dari angka 0 - 9 dan dikonversikan kebentuk citra biner untuk menghasilkan database *template* acuan dengan resolusi yang berbeda-beda. Seperti terlihat pada tabel 3.1 berikut :

Table 3.1 Template Dataset yang Digunakan

| Karakter | Citra Digital | Height (tinggi) | Widht (lebar) | Bwarea (luasan) | Perc bwarea / H x W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|--------------------|------------------|--------------------|------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----|---|--------|--------|
|  | <table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 12 | 9 | 54,000 | 0.5000 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | |
|-----------|---|---|---|---|---|---|---|---|----|----|---------|---------|--------|
| | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | |
| | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | |
| | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | |
| | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | | | | |
| | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | | | |
| | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | | | | |
| il | 0 | 0 | 1 | 1 | 0 | 0 | | | | 12 | 6 | 36,5000 | 0.5069 |
| | 1 | 1 | 1 | 1 | 0 | 0 | | | | | | | |
| | 1 | 1 | 1 | 1 | 0 | 0 | | | | | | | |
| | 0 | 0 | 1 | 1 | 0 | 0 | | | | | | | |
| | 0 | 0 | 1 | 1 | 0 | 0 | | | | | | | |
| | 0 | 0 | 1 | 1 | 0 | 0 | | | | | | | |
| | 0 | 0 | 1 | 1 | 0 | 0 | | | | | | | |
| | 0 | 0 | 1 | 1 | 0 | 0 | | | | | | | |
| | 0 | 0 | 1 | 1 | 0 | 0 | | | | | | | |
| | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | | |
| | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | | |
| 2 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 12 | 8 | 48,0000 | 0,5000 | |
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | | | | |
| | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | | | | | |
| | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | | | | | |
| | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | | | | | |
| | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | | |
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | | |

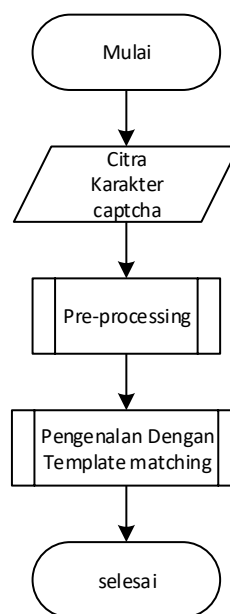
| | | | | | |
|-----------|-------------------|----|---|---------|--------|
| | 0 0 0 0 0 1 1 1 | | | | |
| | 0 0 0 0 0 0 1 1 | | | | |
| | 0 0 0 0 0 0 1 1 | | | | |
| | 1 0 0 0 0 1 1 1 | | | | |
| | 1 1 1 1 1 1 1 0 | | | | |
| | 0 1 1 1 1 1 0 0 | | | | |
| 07 | 0 0 0 1 1 1 1 0 0 | 12 | 9 | 64,000 | 0.5926 |
| | 0 1 1 1 1 1 1 1 0 | | | | |
| | 0 1 1 0 0 0 1 0 0 | | | | |
| | 1 1 1 0 0 0 0 0 0 | | | | |
| | 1 1 1 1 1 1 1 0 0 | | | | |
| | 1 1 1 1 1 1 1 1 0 | | | | |
| | 1 1 1 0 0 0 1 1 1 | | | | |
| | 1 1 0 0 0 0 0 1 1 | | | | |
| | 1 1 0 0 0 0 0 1 1 | | | | |
| | 0 1 1 0 0 0 1 1 1 | | | | |
| | 0 1 1 1 1 1 1 1 0 | | | | |
| | 0 0 0 1 1 1 1 0 0 | | | | |
| 7 | 1 1 1 1 1 1 1 1 1 | 12 | 8 | 37,3750 | 0.3893 |
| | 1 1 1 1 1 1 1 1 1 | | | | |
| | 0 0 0 0 0 0 1 1 1 | | | | |
| | 0 0 0 0 0 1 1 0 0 | | | | |
| | 0 0 0 0 1 1 0 0 0 | | | | |
| | 0 0 0 1 1 0 0 0 0 | | | | |
| | 0 0 0 1 1 0 0 0 0 | | | | |
| | 0 0 1 1 0 0 0 0 0 | | | | |
| | 0 0 1 1 0 0 0 0 0 | | | | |
| | 0 1 1 0 0 0 0 0 0 | | | | |
| | 0 1 1 0 0 0 0 0 0 | | | | |

| | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|---|----|---|---------|--------|
|  | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 12 | 9 | 67.1250 | 0.6215 |
| | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | | | |
| | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | | | | |
| | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | | | | |
| | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | | | | |
| | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | | | | |
| | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | | | | |
| | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | | | | |
| | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | |
| | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | | | | |
| | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | | | |
| | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | | | | |
|  | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 12 | 9 | 62,7500 | 0,5810 |
| | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | | | |
| | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | | | | |
| | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | |
| | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | |
| | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | | | | |
| | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | |
| | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | | | | |
| | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | | | |
| | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | | | | |

Yang digunakan dalam pencocokan nantinya adalah nilai *perc* yang didapat dengan menghitung nilai perbandingan antara jumlah piksel karakter dengan jumlah keseluruhan piksel.

3.2.3 Perancangan Sistem

Fungsi dari *flowchart* adalah memberikan gambaran tentang program yang akan dibuat pada penelitian ini, pada bagian ini akan dijelaskan bagaimana proses pengolahan data yang berupa citra dapat diolah menggunakan proses pengolahan citra hingga dapat menghasilkan kemampuan mengidentifikasi suatu objek. Untuk *flowchart* perancangan sistem dapat dilihat pada gambar 3.2.



Gambar 3.2 *flowchart* Perancangan Sistem

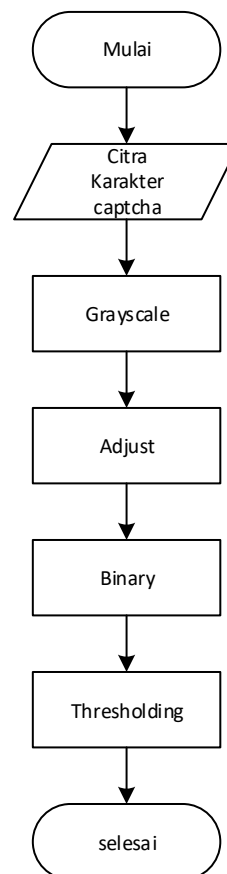
Pada perancangan sistem proses dimulai dari menginputkan citra karakter *Captcha* untuk masuk proses *pre-processing*, yang mana hasil *pre-processing* digunakan untuk proses pengenalan dengan algoritma *Template Matching*. Pada perancangan sistem data citra yang digunakan adalah citra yang diunduh secara manual dari halaman uji ibank.bni.co.id yang bisa dilihat pada gambar 3.3.



Gambar 3.3 Citra Data Uji

a. Pemrosesan Data Awal (*pre-processing*)

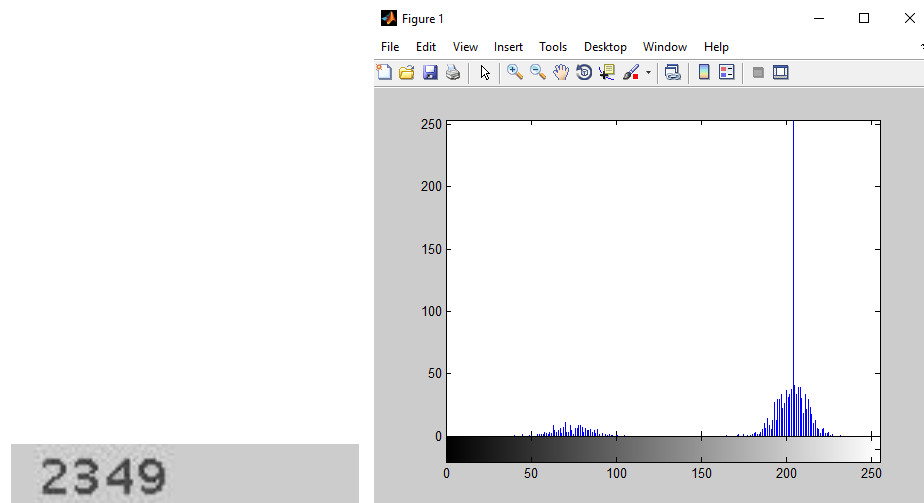
Tahapan *pre-processing* merupakan pra proses atau pengolahan awal sebelum nantinya akan dijalankan metode *Template Matching*, adapun alur *pre-processing* dapat dilihat pada gambar 3.4.



Gambar 3.4 Flowchart Pemrosesan data awal (*Pre-processing*)

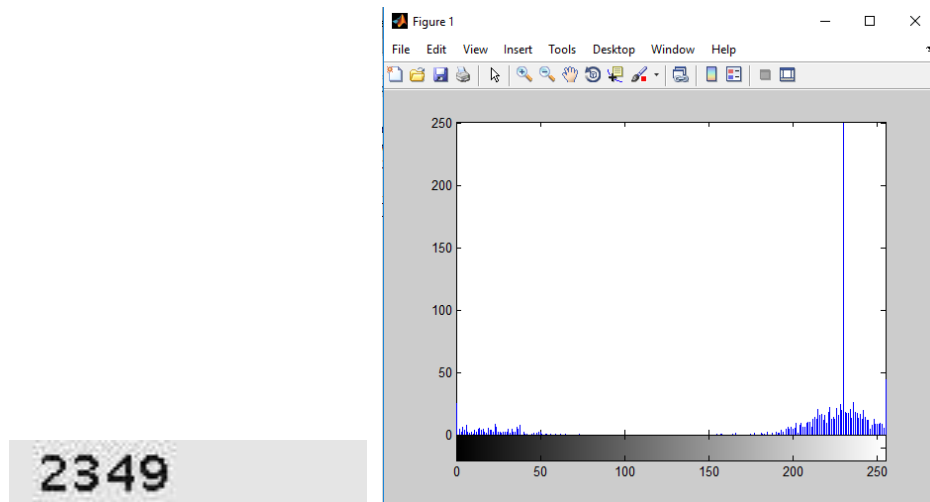
Seperti gambar 3.7 pengolahan data awal dimulai dengan Citra RGB, akan dikonversikan menjadi *grayscale* untuk mendapatkan citra *gray* (abu-abu). Dengan proses *grayscale* ini dapat mempermudah untuk memproses gambar lebih lanjut,

karena Citra *gray* hanya memiliki satu nilai kanal pada setiap pikselnya yang bernilai antara 0 – 255. Sedangkan citra RGB memiliki 3 kanal dalam setiap pikselnya yaitu R (Red) G (Green) B (Blue) sehingga didapatkan bit dalam satu kanalnya $((28)3) = 16.777.216$, dimana hal ini sangat mempersulit dan membuat proses semakin tidak optimal. Gambar citra uji *grayscale* dapat dilihat pada gambar 3.5.



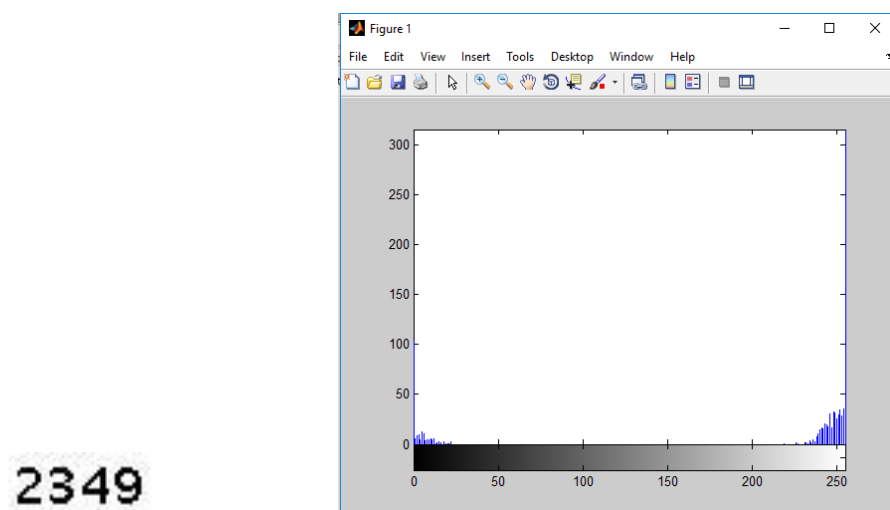
Gambar 3.5 Citra Uji *Grayscale* Beserta Frekuensi Nilai Intensitasnya

Proses selanjutnya adalah *adjust* atau *contrasting* bertujuan untuk menaikkan kontras citra sebelumnya, proses ini membantu untuk proses berikutnya agar kontras warna karakter lebih jelas. Gambar citra uji *adjust* beserta nilai intensitasnya dapat dilihat pada gambar 3.6.



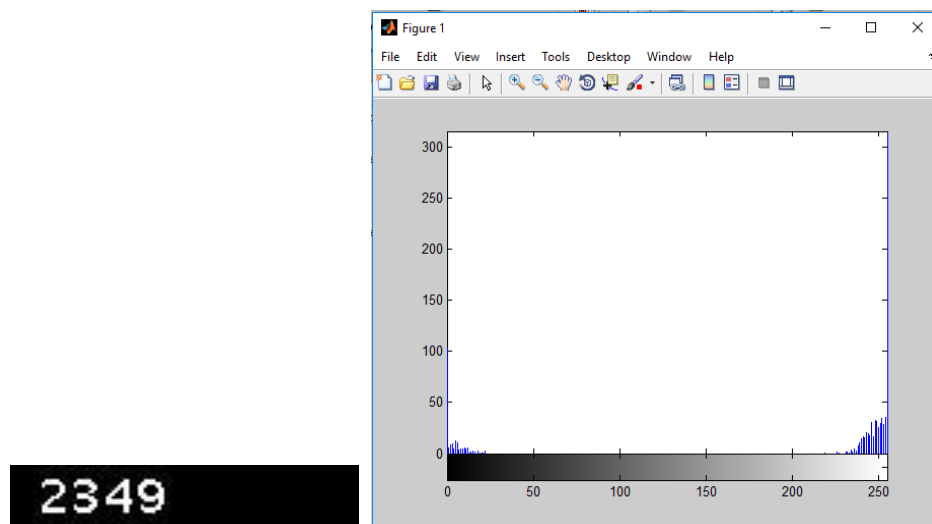
Gambar 3.6 Citra Uji *Adjust* Beserta Frekuensi Nilai Intensitasnya

Selanjutnya masuk pada proses *binary* adalah citra yang hanya mempunyai dua nilai derajat keabuan yaitu hitam dan putih. Citra biner banyak digunakan misalnya pada citra logo instansi (yang hanya memiliki warna hitam dan putih), citra kode batang (barcode), citra hasil pemindaian (scanner) dan lain-lain. Gambar citra uji *binary* beserta nilai intensitasnya dapat dilihat pada gambar 3.7.



Gambar 3.7 Citra Uji *binary* Beserta Frekuensi Nilai Intensitasnya

Kemudian masuk proses *thresholding* adalah dimana citra *grayscale* akan dibuat menjadi citra biner, yang mana intensitinya hanya berisi piksel 1 dan 0 hitam dan putih saja, proses ini bergantung pada nilai *threshold* dan kualitas citra, dimana dalam citra diharuskan mempunyai kontras warna yang berbeda antara karakter dan background. Gambar citra uji *threshold* dapat dilihat pada gambar 3.8.



Gambar 3.8 Citra Uji *Threshold* Beserta Frekuensi Nilai Intensitasnya

b. Proses Pengenalan Menggunakan Algoritma *Template Matching*

Pada tahap ini, umumnya citra hasil proses hanya bersisakan piksel karakter saja. Tahap ini merupakan tahap akhir dan utama dalam sistem, dan tahap sebelumnya bisa disebut dengan tahap pembantu atau tahapan awal saja. Tahap ini dimulai dengan proses labeling, pemotongan piksel tiap label, dan pencocokan pola piksel dengan dataset yang tersedia.

➤ Labeling dengan CCL (*Connected Component Labeling*)

Dari tahap *pre-processing* diatas seharusnya hasil yang didapat berupa citra piksel karakter biner. Selanjutnya

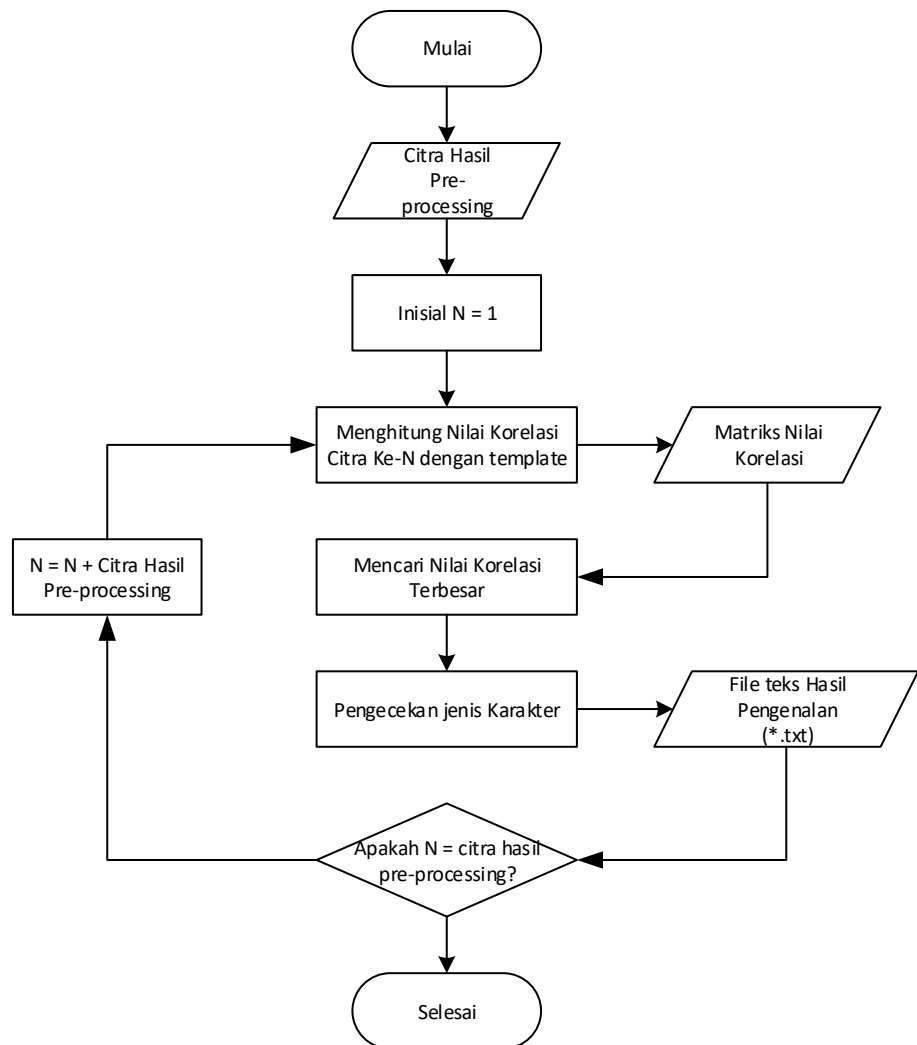
dilakukan labeling pemotongan piksel tiap label, dan pencocokan pola piksel dengan dataset yang tersedia, piksel karakter akan dipotong *rectangle* (persegi) sesuai batas tepi maksimal. Proses pengulangan akan berhenti berdasarkan jumlah indeks label, contohnya piksel karakter berjumlah 4 maka pengulangan akan terjadi sebanyak 4 kali dengan bentuk *rectangle*. Gambar citra label karakter dapat dilihat pada gambar 3.9.



Gambar 3.9 Gambar citra label Karakter

➤ Template Matching

Selanjutnya proses pencocokan dengan algoritma *template matching* dengan cara membandingkan citra yang sudah dipisah akan di uji dengan pola yang tersedia dalam dataset untuk mendapatkan informasi karakter. Proses berada dalam pengulangan sejumlah 10 indeks label karakter, bertujuan agar proses akan mengulangi sejumlah karakter yang ada. Gambar *flowchart* pengenalan karakter dapat dilihat pada gambar 3.10.



Gambar 3.10 Proses Pengenalan Karakter Dengan Metode *Template Matching*

Diberikan sampel citra karakter yang dilakukan proses *Template Matching*, untuk sampel yang bisa kita lihat berkarakter '2' yang merupakan indeks pertama dalam proses sebelumnya akan dicocokkan dengan template dataset. Setelahnya akan diambil intensitas piksel masing-masing untuk mencari jumlah yang paling mendekati atau bahkan sama dengan pola dasarnya. Gambar citra potongan karakter '2' dapat dilihat pada gambar 3.11 berikut.



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Gambar 3.11 Citra Matriks karakter '2'

Bwarea merupakan teknik pengolahan citra yang memperkirakan jumlah daerah dalam piksel yang terkandung pada sebuah gambar yang sudah dibinerisasi. Total = bwarea (BW) memperkirakan wilayah objek dalam citra biner BW. Total adalah skala yang nilainya sesuai dengan jumlah total pada piksel dalam gambar, tapi mungkin tidak persis sama karena pola yang berbeda dari bobot piksel.

Algoritma bwarea memperkirakan luas dari semua piksel di dalam gambar dengan menjumlahkan bidang setiap piksel dalam gambar. Bwarea sebagai alat ukur membantu visualisasi hasil dengan menghasilkan gambar dimana setiap bentuk diberi label dengan nilai wilayahnya. Bwarea tidak

hanya menghitung jumlah piksel diaktifkan namun sebaiknya, bobot bwarea poa piksel dihitung tidak merata. Bobot ini mengkompensasi distorsi yang melekat dalam merepresentasikan sebuah gambar terus-menerus dengan piksel diskrit, (Makalalag : 2012).

Yang akan dihitung adalah variabel *perc* yang berisi perbandingan antara jumlah piksel karakter dengan jumlah keseluruhan piksel. Jumlah piksel karakter didapatkan dengan bwarea pada variabel *n1* dan jumlah keseluruhan piksel diambil dari perkalian panjang dan lebar citra dalam variabel *n1*. Pada Program sebagai berikut.

$$\text{Perc} = \text{bwarea}(n1) / (\text{height} * \text{widht})$$

Diketahui hasil bwarea pada variabel *n1* yang berisi citra sampel adalah 48,0000 dengan panjang (*height*) 12 dan lebar (*widht*) 8. Sehingga sampel tadi

$$\text{Perc} = \text{bwarea}(n1) / (\text{height} * \text{width})$$

$$\text{Perc} = 48,0000 / (12 * 8)$$

$$\text{Perc} = 0,50000$$

Nilai *perc* ini akan dijadikan acuan, berikutnya variabel *LL*, awalnya berisi citra dataset template sesuai indeks pengulangan namun dalam setiap pengulangan variabel *LL* diubah menjadi citra gabungan antara *n1* dan template, namun sebelumnya *size* dari citra disamakan dengan dengan *size n1*.

Kemudian kembali dicari variabel *perc_v*, yang berisi perbandingan antara Jumlah piksel karakter dengan jumlah keseluruhan piksel pada variabel *LL*.

$$\text{Percv} = \text{bwarea(LL)} / (\text{height} * \text{width})$$

Nilai *percv* tidak akan mungkin melebihi nilai dari *perc*, dikarenakan *perc* adalah perbandingan jumlah piksel citra dasar sedangkan *percv* adalah perbandingan jumlah piksel citra proses. Sehingga yang dicari sebagai perbandingan citra yang tepat yakni citra proses yang memiliki nilai *percv* terbesar. Bisa dilihat dari table berikut, diketahui citra pola dasar memiliki *perc* = 0,50000. Untuk tabel proses template match dapat dilihat pada tabel 3.2 dibawah.

Tabel 3.2 Proses Template Match

| Indeks | LL | Size | Bwarea | Size | percv |
|----------|---------------|---------------|----------------|---------------|---------------|
| 0 | 12 x 9 | 12 x 8 | 25,0000 | 12 x 8 | 0,2604 |
| 1 | 12 x 6 | 12 x 8 | 27,0000 | 12 x 8 | 0,2852 |
| 2 | 12 x 8 | 12 x 8 | 48,0000 | 12 x 8 | 0,5000 |
| 3 | 12 x 8 | 12 x 8 | 35,1250 | 12 x 8 | 0,3659 |
| 4 | 12 x 9 | 12 x 8 | 18,8750 | 12 x 8 | 0,1966 |
| 5 | 12 x 8 | 12 x 8 | 29,8750 | 12 x 8 | 0,3112 |
| 6 | 12 x 9 | 12 x 8 | 24,8750 | 12 x 8 | 0,2591 |
| 7 | 12 x 8 | 12 x 8 | 25,2500 | 12 x 8 | 0,2630 |
| 8 | 12 x 9 | 12 x 8 | 29,1250 | 12 x 8 | 0,3034 |
| 9 | 12 x 9 | 12 x 8 | 33,8750 | 12 x 8 | 0,3529 |

Berdasarkan hasil yang tertera diatas bisa dilihat nilai variabel *percv* tertinggi dengan nilai 0,5000 berada pada indeks karakter “2”. Namun indeks bersifat numeric berdasarkan pada pengulangan variabel $t = 1$ hingga 10.

Berdasarkan indeks variabel t pada pengulangan karakter akan diambil pada array *letter*. Dibuat sebuah array *letters* sejumlah 36 baris yang berisi karakter alfabet dan numerik.

Sehingga, dengan indeks t yang didapat, akan disimpan pada variabel *bestmatch*. Nantinya karakter akan dipanggil pada array *letters* dengan indeks *bestmatch* untuk setiap pengujian. Sehingga didapatkan semua karakter.

3.3. Desain Antarmuka

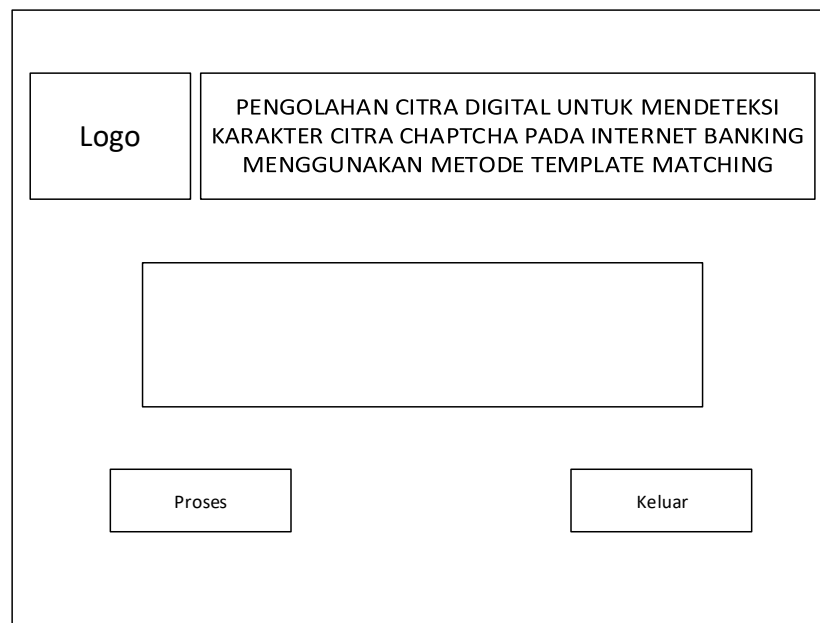
Perencanaan sistem merupakan desain antarmuka untuk menampilkan citra yang akan diproses dalam sistem yang akan dibuat. Desain antarmuka tersebut dapat dilihat dalam tahapan-tahapan sebagai berikut:

1. Menu Utama

Pada menu utama dalam tampilan ini terdapat beberapa tombol yang berfungsi untuk memproses objek secara jelas, dalam menu utama terdapat beberapa menu antara lain:

- a. Proses Pengujian
- b. Keluar

Untuk desain antar muka utama dapat dilihat pada gambar 3.12 sebagai berikut :



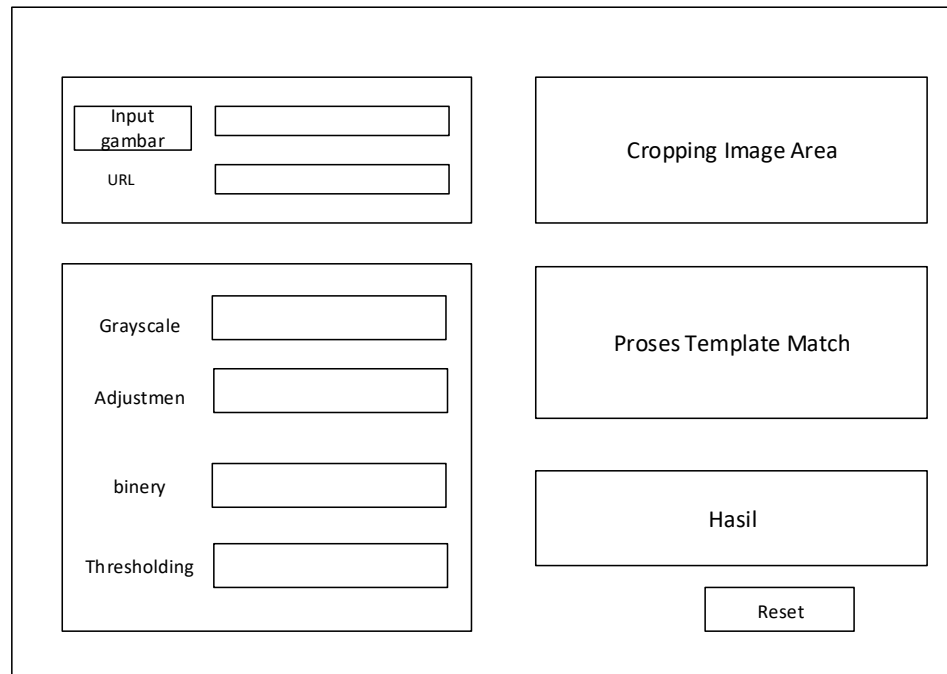
Gambar 3.12 Desain Antar Muka

a. Proses Pengujian

Menu Proses Pengujian : berfungsi untuk mengidentifikasi objek. Dalam proses ini terdapat dua button. Berikut adalah menu yang ada dalam proses pengujian :

1. Button Insert Gambar : Untuk memilih objek mana yang akan diidentifikasi. (Langkah pertama yaitu mencari lokasi data citra. Pada menu ini ada pilihan berupa Cari Data yang digunakan untuk mencari dimana lokasi citra disimpan. Klik Cari Data kemudian *browser* dan pilih data OK) dan secara otomatis akan memproses dengan sendirinya sampai memunculkan hasil pengenalan berupa texts.
2. Axes citra gambar *Grayscale* : merupakan tampilan untuk merubah citra *RGB* menjadi *Grayscale*.
3. Axes citra hasil proses *Adjuts* : merupakan tampilan citra hasil proses citra *Gray* menjadi *Adjust*.
4. Axes citra hasil proses *Binary* : merupakan tampilan citra hasil perubahan citra *Adjustmen* menjadi citra *Binary*.
5. Axes citra hasil proses *Thresholding* : merupakan tampilan citra hasil proses citra *Binary* menjadi *Thresholding*.
6. Axes citra hasil proses *Remove Border* : merupakan tampilan citra hasil penghilangan bagian gambar yang tidak perlu.
7. Axes pengenalan karakter dengan *template matching* : merupakan tampilan hasil pelabelan tiap karakter yang diulang 4 kali berdasarkan objek karakter yang dikenali, serta pencocokan karakter dengan algoritma *template matching*.
8. Menu hasil pengenalan : untuk menentukan hasil pengenalan yang sudah di proses menggunakan algoritma *template matching*. Dengan menghitung nilai error terkecil.
9. Menu Reset : Untuk mengembalikan aplikasi ke tampilan awal sistem. Desain Antarmuka Proses Pengujian.

Untuk desain antarmuka proses pengujian dapat dilihat pada gambar 3.13 sebagai berikut :



Gambar 3.13 Desain Antarmuka Proses Pengujian

3.4. Perancangan Pengujian Sistem

Pada penelitian penelitian ini scenario pengujian dilakukan dengan data sampel yang telah dikumpulkan berjumlah 150 citra karakter, yang diunduh dari halaman uji ibank.bni.co.id. Tahapan selanjutnya yakni melakukan pengujian terhadap 100 sampel uji, proses pengujian citra menggunakan *software* Matlab.

Adapun skenario pengujian pada penelitian ini sebagai berikut :

1. Masukkan satu data citra *Captcha* karakter
2. Lakukan Pre-processing yakni *Grayscaleing*, *Adjust*, *Binary*, *Thresholding* untuk mendapatkan citra biner yang tanpa gangguan.
3. Setelah citra biner sudah didapat selanjutnya dilakukan proses pemotongan atau penghilangan bagian citra biner yang tidak di inginkan.

