Bab 2
TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Pada bagian tinjauan pustaka ini, penulis akan merangkum beberapa jurnal yang berkaitan dengan tugas akhir yang dibuat oleh penulis. Sub bab ini berisi tentang penelitian yang telah dilakukan peneliti lain, yaitu tentang segmentasi dan algoritma *Connected Component Labeling.*

Segmentasi karakter pernah digunakan dalam beberapa penelitian, seperti yang dilakukan oleh Putra dan Prapitasari (2011), pada penelitiannya tentang segmentasi karakter pada skrip bahasa Bali menggunakan metode *Canny Edge Detection.* Algoritma ini digunakan dalam penelitian ini karena mampu mendeteksi tepi dengan hasil tepian yang tampak lebih jelas dan *background* citra terlihat lebih nyata daripada algoritma yang lainnya. Hal ini didukung dengan hasil penelitian yang telah dilakukan pada *input* yang baik dan berdasarkan uji coba pada sistem tersebut menunjukkan bahwa sistem mampu memotong tulisan aksara Bali menjadi beberapa karakter secara sempurna.

Ardiansyah (2011) melakukan penelitian tentang lokalisasi citra plat nomor kendaraan dan segmentasi karakter menggunakan algoritma *Mean Shift*. Hal penting dalam proses pengenalan plat nomor kendaraan adalah proses pencarian lokasi plat nomor dan segmentasi karakter pada plat nomor. Proses ini dilakukan karena terdapat bermacam-macam kandidat huruf atau angka yang akan dicari oleh sistem. Dengan dilakukan proses tersebut sistem tidak perlu menganalisa semua objek tersebut untuk dikenali sebagai bagian dari karakter citra plat nomor. Berdasarkan penelitian yang telah dilakukan menunjukkan bahwa sistem dapat mendeteksi lokasi plat nomor dengan akurasi 78% dan menunjukkan adanya pengaruh segmentasi dengan metode *Mean Shift Algorithm* yang cukup signifikan dalam proses penentuan lokasi plat nomor.

Penelitian lain tentang segmentasi plat nomor kendaraan pernah dilakukan oleh Liliana, Budhi, dan Hendra (2010) yang menggunakan *Run-Length Smearing Algorithm* untuk mencari lokasi plat nomor dan segmentasi karakter. *Smearing* merupakan suatu metode yang bertujuan untuk mengekstraksi lokasi teks pada suatu gambar, proses pencarian dilakukan dengan melakukan *scan line* baik secara vertikal maupun horisontal yang dilakukan secara bergantian. Algoritma ini hanya dapat mendeteksi bidang dengan berbentuk kotak dengan *range* tertentu. Selain itu dalam penelitian ini ditemukan adanya kesulitan yaitu warna mobil yang ada di Indonesia cenderung gelap sehingga menyulitkan dalam proses segmentasi karena latar belakang plat nomor juga berwarna gelap. Untuk mengatasi hal tersebut peneliti menggunakan warna biru sebagai ganti warna *grayscale* dan hasil akhirnya pun lebih baik. Kelemahan lain pada penelitian ini yaitu proses pengambilan citra mobil harus memiliki pencahayaan yang baik, citra mobil harus diambil pada jarak kurang lebih dua meter, dan posisi plat nomor juga harus tegak lurus. Ketiga hal ini harus diperhatikan karena akan mempengaruhi hasil akhir proses segmentasi plat nomor.

Selain untuk segmentasi karakter, segmentasi juga dapat dilakukan untuk segmentasi warna. Penelitian ini pernah dilakukan oleh Aryuanto, Limpraptono, dan Yamada (2010) dalam penelitiannya tentang segmentasi warna untuk mengekstraksi simbol dan karakter pada citra rambu lalu lintas. Pada jurnal ini membahas metode segmentasi warna RGB berbasis *RGB Chromaticity Diagram* ternormalisasi. Metode ini mengusulkan pemisahan warna biru pada latar belakang rambu lalu lintas dan memanfaatkan histogram yang dikembangkan pada diagram kromatis untuk penentuan nilai ambang segmentasi secara otomatis. Selain itu, teknik morfologi citra dan proyeksi histogram digunakan untuk ekstraksi simbol dan karakter. Dari hasil uji coba diperoleh bahwa metode ini dapat mengekstrak simbol dan karakter dengan rata-rata ekstraksi 97,3 %. Kesalahan ekstraksi karakter terjadi disebabkan karena susunan karakter yang saling tumpang tindih dan kesalahan ekstraksi simbol yang menyebabkan kesalahan dalam analisis proyeksi histogram.

Karthikeyan, Vijayalakshmi, dan Jeyakumar (2013) menggunakan segmentasi karakter dengan *Connected Component Analysis* sebelum dilakukan pengenalan plat nomor kendaraan bermotor. *Connected Component Analysis* merupakan metode pelabelan piksel-piksel dari citra biner menjadi komponen-komponen berdasarkan piksel yang saling terhubung. Sebelum dilakukan segmentasi dan pengenalan plat nomor dilakukan *thresholding biner* yang bertujuan untuk memisahkan objek dengan *background*. Komponen yang saling terhubung menganalisis dan menyaring komponen panjang dan lebar serta hanya menyisakan komponen yang telah ditentukan sebelumnya. Dari penelitian tersebut dapat disimpulkan bahwa metode ini dapat mendeteksi secara efisien, akurat dan cepat.

Rizki, Nugroho, Jamal, Handoko, Gunawan, Witjaksono, et al. (2010) mengusulkan suatu metode baru pencarian karakter plat nomor menggunakan *Connected Component Analysis*. Karakter plat nomor pada citra mobil disegmentasi tanpa harus mencari letak posisi plat nomornya. Metode ini akan melakukan proses *filtering* karakter yang akan menentukan objek terlabel mana yang merupakan karakter yang dicari. Berdasarkan hasil uji coba diperoleh kesimpulan tingkat keberhasilan mencapai 85% dari 20 citra yang diujikan. Kegagalan yang terjadi pada proses segmentasi disebabkan oleh karakter yang rusak dan saling terhubung.

Ada beberapa metode lain yang mempunyai fungsi yang sama seperti algoritma *Connected Component Labeling* yang telah ada sebelumnya. Salah satunya diusulkan oleh Rakhmadi, Othman, dan Bade (2011) yang menjelaskan suatu pendekatan komputasional yang disebut algoritma *neighbors-scan labeling* untuk *connected component labeling.* Algoritma ini menggunakan *forward scanning* dalam proses pelabelan dan menggunakan konsep 8-ketetanggaan. Berdasarkan hasil penelitian yang dilakukan pada citra biner diperoleh kesimpulan bahwa algoritma ini mampu meningkatkan kecepatan hingga 67,4% dari dua kali *scanning.* Selain itu algoritma ini juga mempunyai kelebihan yaitu proses pelabelan lebih akurat, waktu yang digunakan untuk proses pelabelan lebih

cepat, dan metodenya lebih sederhana dibandingkan dengan metode-metode yang telah ada sebelumnya.

2.2 Landasan Teori

Pengolahan citra digital *(digital image processing)* merupakan pemrosesan citra dengan menggunakan komputer yang bertujuan untuk menjadikan kualitas citra lebih baik. Selain bertujuan untuk memperbaiki kualitas citra, tujuan dari pengolahan citra digital adalah memungkinkan manusia mengambil informasi yang terdapat pada citra, proses pengambilan informasi dari citra tersebut dapat dilakukan dengan cara segmentasi citra (Basuki, Palandi & Fatchurrochman, 2005).

Sebelum dilakukan proses segmentasi citra, maka citra tersebut akan dilakukan *image preprocessing* yang bertujuan untuk mempermudah di dalam melakukan proses selanjutnya. Pada bagian ini akan dijelaskan *image preprocessing* yang akan digunakan sebelum dilakukan proses pelabelan piksel dan segmentasi Nomor Induk Mahasiswa (NIM) menggunakan algoritma *Connected Component Labeling*.

2.2.1 Citra

Citra merupakan salah satu komponen multimedia yang memegang peran penting sebagai informasi visual. Kelebihan citra dibanding data teks adalah citra dapat memberikan informasi lebih banyak daripada informasi yang disajikan dalam bentuk teks (Munir, 2004). Menurut Gonzales dan Woods (2002), citra merupakan fungsi intensitas cahaya dua dimensi, yang ditunjukkan dengan *f(x,y)* di mana nilai atau amplitudo *f* memberikan intensitas kecerahan citra pada koordinat *(x,y)*.

Selain citra merupakan fungsi intensitas cahaya dua dimensi, Basuki et al. (2005) mengatakan bahwa citra merupakan suatu fungsi *f(x,y)* di mana *x* menyatakan nomor baris dan y menyatakan nomor kolom, dan f menyatakan nilai

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

Gambar 2.1 Contoh Matriks Citra

Dikutip dari: Munir, R. (2004). *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika.

derajat keabuan dari citra. Dengan demikian, *(x,y)* adalah posisi dari piksel dan *f* adalah nilai derajat keabuan pada titik *(x,y)*. Citra digital yang berukuran *N x M* biasanya dinyatakan sebagai matriks yang berukuran *N* baris dan *M* kolom seperti yang ditunjukkan pada Gambar 2.1.

2.2.2 Citra *Bitmap* (BMP)

Menurut Munir (2004), format citra yang baku digunakan pada sistem operasi Microsoft Windows dan IBM OS/2 adalah berkas *bitmap* (BMP). Format BMP memiliki ukuran berkas relatif besar dibanding format citra yang lain karena berkas BMP tidak dikompresi. Pada penelitian ini, citra *input* yang digunakan adalah dalam format *bitmap* (BMP) dengan kedalaman warna 24 bit. Citra-24 bit setiap piksel memiliki panjang 24 bit, karena setiap piksel langsung menyatakan komponen warna merah, hijau, dan biru. Citra 24 bit disebut juga citra 16 juta warna, karena citra ini mampu menghasilkan 2^{24} = 16.777.216 kombinasi warna.

2.2.3 Model Pewarnaan *Grayscale*

Menurut Basuki et al. (2005) proses awal yang sering dilakukan di dalam *image processing* adalah mengubah citra RGB menjadi citra *grayscale* yang bertujuan untuk menyederhanakan model citra. Citra berwarna mempunyai 3 layer matrik, yaitu R-*layer*, G-*layer*, dan B-*layer*. Sigit et al. (2005) menjelaskan bahwa di dalam citra *grayscale* tidak ada warna, melainkan derajat keabuan. Untuk mengubah citra berwarna menjadi citra *grayscale* adalah dengan mengambil rata-rata dari nilai R, G, dan B dari citra yang dapat dirumuskan sebagai berikut:

$$s = \frac{R + G + B}{3} \qquad [2.1]$$

2.2.4 Citra Biner

Hasil dari konversi citra RGB menjadi citra *grayscale* digunakan untuk proses selanjutnya yaitu untuk mengubah citra *grayscale* menjadi citra biner (hitam-putih). Menurut Basuki et al. (2005), citra biner merupakan citra yang banyak dimanfaatkan untuk keperluan *pattern recognition* yang sederhana seperti pengenalan angka atau huruf. Untuk mengubah citra *grayscale* ke citra biner, proses yang dilakukan sama dengan *threshold* yaitu mengubah kuantisasi citra. Pada operasi pengambangan *(threshold),* nilai piksel dua dipetakan ke salah satu dari dua nilai, a_1 atau a_2, berdasarkan nilai ambang *(threshold)* T yang dapat dituliskan sebagai berikut:

$$f(x,y)' = \begin{cases} a_1, f(x,y) < T \\ a_2, f(x,y) \geq T \end{cases} \qquad [2.2]$$

Jika $a_1 = 0$ dan $a_2 = 1$, maka operasi pengambangan mentransformasikan citra hitam-putih ke citra biner. Dengan demikian, nilai intensitas piksel semula dipetakan ke dua nilai saja yaitu hitam dan putih. Operasi pengambangan citra biner dengan fungsi transformasi adalah sebagai berikut:

$$f(x,y)' = \begin{cases} 0, f(x,y) < T \\ 1, f(x,y) \geq T \end{cases} \qquad [2.3]$$

2.2.5 *Median Filter*

Ada berbagai macam teknik yang digunakan untuk mengurangi (reduksi) *noise,* salah satunya dengan menggunakan *median filter*. *Noise* merupakan suatu nilai piksel yang berbeda dengan semua tetangga di sekitar piksel tersebut, maka dapat dikatakan bahwa *noise* merupakan nilai-nilai yang berada pada frekuensi tinggi (Basuki et al., 2005). Pada *median filter,* suatu "jendela" *(window)* memuat sejumlah piksel yang berjumlah ganjil. *Filtering* dilakukan dengan menggeser

jendela tersebut piksel demi piksel pada seluruh daerah citra. Setiap pergeseran dibuat jendela baru. Nilai-nilai pada piksel tersebut diurutkan dari yang paling kecil ke yang paling besar, setelah itu diambil nilai tengahnya. Titik pusat dari "jendela" ini diubah dengan nilai tengah *(median)* dari "jendela" tersebut (Munir, 2004).

2.2.6 Hubungan Antar Piksel

Hubungan antar piksel merupakan hal yang penting di dalam citra digital. Seperti yang sudah dijelaskan sebelumnya, sebuah citra digital didefinisikan sebagai fungsi *f(x,y)*. Hal ini menunjuk pada sebuah piksel tertentu yang diberi notasi *p* atau *q* (Gonzales & Woods, 2002).

Menurut Gonzales dan Woods (2002), suatu piksel *p* pada koordinat *(x,y)* mempunyai empat tetangga, baik pada arah horisontal maupun vertikal, di mana koordinatnya dapat dinyatakan sebagai berikut:

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)$$

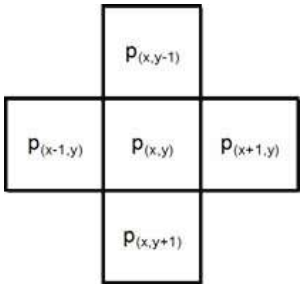
Kumpulan piksel tersebut disebut 4-tetangga dari *p*, yang dinotasikan $N_4(p)$. Selain itu, piksel *p* mempunyai empat tetangga diagonal yang disebut diagonal tetangga, koordinatnya adalah sebagai berikut:

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)$$

Kumpulan piksel dari diagonal tetangga dinotasikan $N_D(p)$. Jika piksel 4-tetangga dan diagonal tetangga digabungkan, maka akan menjadi piksel 8-tetangga dari *p* yang dinotasikan $N_8(p)$.

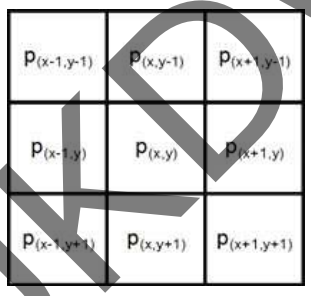
Konektivitas piksel merupakan metode yang biasanya digunakan dalam pengolahan citra untuk menganalisa suatu piksel yang saling terhubung dengan piksel yang lainnya dan berada di sekitar piksel tersebut. Dua piksel dianggap saling terhubung satu sama lain jika mereka berdekatan satu sama lain dan memiliki sekumpulan nilai yang sama. Sebuah nilai piksel pada citra biner adalah elemen dari himpunan {0, 1}, di mana 0 adalah nilai piksel *background* dan 1 adalah nilai piksel *foreground* (Sutheebanjard & Premchaiswadi, 2011).

Ada dua jenis konektivitas piksel, yaitu 4-konektivitas (N_4) yang ditunjukkan pada Gambar 2.2 dan 8-konektivitas (N_8) yang ditunjukkan pada Gambar 2.3.



Gambar 2.2 4-konektivitas (N_4)

Dikutip dari : Sutheebanjard, P. & Premchaiswadi, W. (2011). Efficient scan mask techniques for connected component labeling algorithm. *EURASIP Journal on Image and Video Processing*. Diakses pada tanggal 30 Januari 2014 dari http://jivp.eurasipjournals.com/content/2011/1/14



Gambar 2.3 8-konektivitas (N_8)

Dikutip dari : Sutheebanjard, P. & Premchaiswadi, W. (2011). Efficient scan mask techniques for connected component labeling algorithm. *EURASIP Journal on Image and Video Processing*. Diakses pada tanggal 30 Januari 2014 dari http://jivp.eurasipjournals.com/content/2011/1/14

2.2.7 Segmentasi

Segmentasi merupakan langkah awal yang biasanya digunakan sebelum proses analisis terhadap suatu citra. Menurut Basuki et al. (2005), tujuan dari segmentasi citra adalah untuk memisahkan objek-objek yang terdapat dalam suatu citra, misalkan memisahkan objek dengan latar belakangnya. Segmentasi terdiri dari beberapa jenis, salah satunya adalah segmentasi karakter.

Segmentasi karakter adalah proses awal yang harus dilakukan sebelum dilakukan proses pengenalan karakter dalam suatu citra. Di dalam segmentasi karakter terdapat banyak metode yang dapat dilakukan, salah satu metode tersebut

adalah dengan cara pelabelan piksel menggunakan *Connected Component Labeling.*

2.2.8 *Connected Component Labeling*

Connected Component Labeling merupakan salah satu metode segmentasi citra. Ada empat tahap di dalam algoritma tersebut, pertama adalah memasukkan citra (RGB atau *grayscale*) yang kemudian diproses *filtering* untuk memisahkan objek dari *background*. Biasanya citra yang digunakan adalah citra biner. Kedua, proses pelabelan komponen terhubung yang digunakan untuk menetapkan label masing-masing daerah yang unik. Ketiga, masing-masing daerah tersebut diproses berdasarkan labelnya untuk mengekstrak sejumlah fitur objek yang direpresentasikan oleh daerah *(region)*. Dan pada tahap yang terakhir, fitur tersebut digunakan untuk mengklasifikasikan setiap daerah menjadi satu atau lebih kelas (Bailey & Johnston, 2007).

Menurut Sasirekha dan Chandra (n.d.), *connected component* adalah suatu algoritma dari aplikasi teori graf, yang berisi kumpulan komponen terhubung yang telah dilabeli berdasarkan heuristik. Hasil dari pelabelan tersebut dapat digunakan misalnya untuk pengenalan pola. Di dalam *computer vision, connected component labeling* digunakan untuk mendeteksi daerah yang saling terhubung pada citra biner.

Sebuah komponen terhubung adalah sekumpulan piksel di mana semua piksel yang terhubung satu sama lain dengan piksel di sekitarnya. Pelabelan komponen terhubung adalah suatu metode untuk pemberian label pada semua piksel yang saling terhubung menjadi komponen-komponen yang sama berdasarkan konektivitas piksel dan menandai setiap komponen dengan label yang berbeda (Sutheebanjard & Premchaiswadi, 2011).

Implementasi pelabelan komponen terhubung pada citra biner merupakan hal penting yang digunakan dalam pengolahan citra, pengenalan pola, dan *computer vision*. Secara umum, algoritma ini dikategorikan menjadi empat kelas: algoritma satu-*scan*, algoritma dua-*scan*, algoritma multi-*scan*, dan

algoritma *tracing contour.* Namun yang akan dibahas dan dipakai pada penelitian ini adalah algoritma dua-*scan.* Algoritma dua-*scan* merupakan algoritma yang sederhana dalam implementasinya dan lebih efisien dalam perhitungan waktu prosesnya (Sutheebanjard & Premchaiswadi, 2011).

2.3 Implementasi Sistem *Connected Component Labeling*

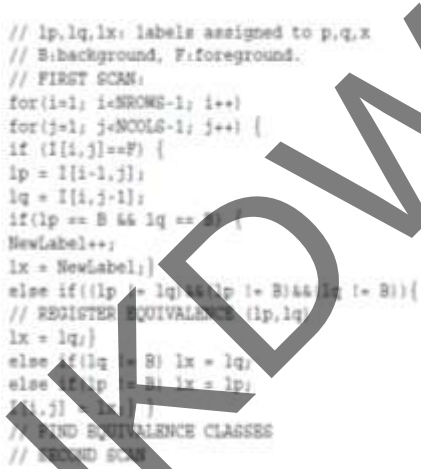
Bagian ini menjelaskan implementasi algoritma *Connected Component Labeling* (CCL) yang telah dimodifikasi oleh Stefano dan Bulgarelli (1999) dalam melakukan segmentasi Nomor Induk Mahasiswa (NIM) dari citra Kartu Tanda Mahasiswa (KTM) Universitas Kristen Duta Wacana. Pada penelitian ini, proses pelabelan piksel dilakukan pada citra hasil *cropping* yaitu berupa citra NIM.

Pada proses ini, setiap objek dengan nilai 1 (dalam penelitian ini warna hitam dianggap sebagai objek) yang saling terhubung dianggap objek dan akan diberikan label yang sama, sedangkan piksel yang bernilai 0 (warna putih) dianggap sebagai *background* dan tidak diberikan label. Pelabelan dilakukan dengan cara memindai *(scan)* piksel dari atas ke bawah, mulai dari kiri ke kanan (dari y ke x). Menurut Bailey dan Johnston (2007), proses pelabelan piksel menggunakan *Connected Component Labeling* klasik terdiri dari dua tahap, yaitu:

1. Pelabelan tahap 1, yaitu proses pemindaian piksel dan pengecekkan piksel pada citra biner dengan menggunakan konsep 4-konektivitas. Jika pada proses pemindaian piksel ketemu piksel *foreground*, maka cek nilai tetangga atas dan kirinya. Jika ketemu piksel *background,* maka piksel tersebut diberi label 0. Terdapat tiga kondisi pada proses pelabelan tahap 1, yaitu sebagai berikut:
    a. Jika tetangga atas dan kiri adalah *background*, maka *increment* nilai label.
    b. Jika bertemu dengan tetangganya hanya atas atau kiri, maka beri dengan nilai tetangganya.
    c. Jika bertemu dengan tetangga atas dan kiri, maka beri nilai tetangganya yang lebih kecil.

2. Pelabelan tahap 2, yaitu penggantian label dengan label yang baru sesuai dengan data konversi labelnya.

Metode *Connected Component Labeling* yang dimodifikasi oleh Stefano dan Bulgarelli (1999) merupakan metode pelabelan dua-*scan* yang dikembangkan dari metode pelabelan dua-*scan* klasik. Gambar 2.4 merupakan metode CCL dua-*scan* klasik dan Gambar 2.5 adalah metode CCL dua-*scan* yang dimodifikasi oleh Stefano dan Bulgarelli (1999).



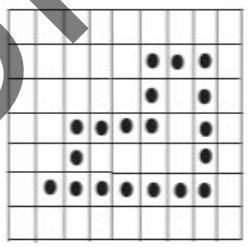
Gambar 2.4 Algoritma CCL Dua-*Scan* Klasik
Dikutip dari : Stefano, L. D. & Bulgarelli, A. (1999). A Simple and Efficient Connected Components Labeling Algorithm. *Proceeding of the International Conference on Image Analysis and Processing, Sept. 27-29, IEEE Computer Society, Washington DC., USA.,* 322-327


```
// C has been initialised ( C[i]=i );
// FIRST SCAN
for(i=1; i<NROWS-1; i++)
for(j=1; j<NCOLS-1; j++){
if (I[i,j] == F){
lp = I[i-1,j];
lq = I[i,j-1];
if(lp == B && lq == B){
NewLabel++;
lx = NewLabel;}
else if((C[lp]!=C[lq])&&(lp != B)&&(lq != B)){
for(k=0; k<=Newlabel; k++)
if (C[k] == C[lp]) C[k]=C[lq];
lx = lq;}
else if(lq != B) lx = lq;
else if(lp != B) lx = lp;
I[i,j] = lx;}}
// SECOND SCAN
for(i=1; i<NROWS-1; i++)
for(j=1; j<NCOLS-1; j++)
if (I[i,j] != B) I[i,j]=C[I[i,j]];
```

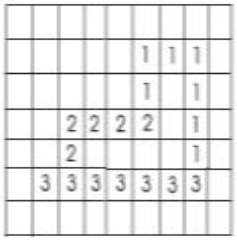
Gambar 2.5 Algoritma CCL Dua-*Scan* yang Dimodifikasi
Dikutip dari : Stefano, L. D. & Bulgarelli, A. (1999). A Simple and Efficient Connected Components Labeling Algorithm. *Proceeding of the International Conference on Image Analysis and Processing, Sept. 27-29, IEEE Computer Society, Washington DC., USA.,* 322-327.

Berdasarkan Gambar 2.4 dan 2.5 terdapat perbedaan antara metode CCL klasik dengan metode CCL yang sudah dimodifikasi ini. Perbedaannya hanya terdapat pada proses pelabelan tahap pertama. Jika pada metode yang sudah dimodifikasi oleh Stefano dan Bulgarelli (1999), selama proses pelabelan piksel tahap pertama dilakukan proses ekuivalensi dengan penggabungan kelas-kelas ekuivalen dengan segera ketika ekuivalen yang baru ditemukan. Proses tersebut menunjukkan adanya peningkatan secara signifikan pada efisiensi proses pelabelan yang tidak terjadi pada metode CCL klasik. Struktur data yang digunakan untuk melakukan proses ekuivalensi tersebut adalah berupa *array* satu dimensi (Stefano & Bulgarelli, 1999).

Berikut ini merupakan contoh efisiensi dari penerapan pelabelan piksel dengan menggunakan metode CCL yang telah dimodifikasi Stefano dan Bulgarelli (1999). Gambar 2.6 menunjukkan citra biner yang akan dijadikan contoh untuk penerapan pelabelan piksel pada tahap pertama. Setelah dilakukan proses pelabelan piksel tahap pertama, hasil pelabelannya dapat dilihat pada Gambar 2.7.



Gambar 2.6 Citra Biner
Dikutip dari : Stefano, L. D. & Bulgarelli, A. (1999). A Simple and Efficient Connected Components Labeling Algorithm. *Proceeding of the International Conference on Image Analysis and Processing, Sept. 27-29, IEEE Computer Society, Washington DC., USA.,* 322-327.



Gambar 2.7 Hasil Akhir Proses Pelabelan Piksel Tahap 1
Dikutip dari : Stefano, L. D. & Bulgarelli, A. (1999). A Simple and Efficient Connected Components Labeling Algorithm. *Proceeding of the International Conference on Image Analysis and Processing, Sept. 27-29, IEEE Computer Society, Washington DC., USA.,* 322-327.

Misalkan Gambar 2.6 memiliki *Class Array* (C) [1 2 3]; setelah ditemukan label ekuivalen (2, 1), C melakukan pembaruan menjadi [2 2 3]; kemudian setelah ditemukan label ekuivalen kembali (3, 2) maka C menjadi [3 3 3]. Jadi, ketika ditemukan ekuivalen (3, 1) pada baris terakhir, hal ini tidak ditangani oleh metode CCL modifikasi sejak C[3] = C[1]. Berdasarkan Gambar 2.7 tersebut menunjukkan bahwa metode CCL yang telah dimodifikasi tidak melakukan penanganan ekuivalensi yang tidak berguna seperti yang dilakukan oleh metode CCL klasik yang diproses pada pelabelan tahap pertama ini seperti yang telah dijelaskan sebelumnya. Selama proses pelabelan tahap pertama terdapat dua label menyebabkan *multiple conflict* (suatu keadaan di mana suatu piksel *foreground* bertemu dengan dua tetangga, yaitu tetangga kiri dan atas yang mempunyai label berbeda). Dengan metode CCL modifikasi, *Class Array* (C) akan selalu diperbarui sesuai dengan ekuivalen yang diberikan. Jadi, ada beberapa kejadian baru pada ekuivalen ini yang diabaikan oleh metode CCL modifikasi.

Bab 3
ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini penulis akan menjelaskan analisis dan perancangan sistem pada tugas akhir ini. Bagian pertama akan dijelaskan spesifikasi sistem dalam tugas akhir ini, baik spesifikasi perangkat keras maupun perangkat lunak yang akan digunakan untuk mengerjakan tugas akhir ini. Selain itu, akan dijelaskan pula diagram alir utama sistem, diagram alir *preprocessing* citra, diagram alir segmentasi Nomor Induk Mahasiswa (NIM), dan diagram alir algoritma *Connected Component Labeling* yang digunakan untuk proses pelabelan dan segmentasi Nomor Induk Mahasiswa (NIM). Bagian akhir dari bab ini akan dijelaskan perancangan antarmuka sistem yang akan dibuat.

3.1 Alat Penelitian

Alat penelitian menjelaskan kebutuhan yang digunakan di dalam pengerjaan tugas akhir ini. Spesifikasi sistem dibedakan menjadi dua bagian yaitu spesifikasi perangkat keras dan spesifikasi perangkat lunak. Spesifikasi perangkat keras akan menjelaskan kebutuhan perangkat keras komputer yang digunakan untuk mengerjakan tugas akhir ini, sedangkan spesifikasi perangkat lunak akan menjelaskan kebutuhan perangkat lunak berupa *software* atau bahasa pemrograman yang akan digunakan untuk menyelesaikan pembuatan sistem dalam tugas akhir ini.

3.1.1 Perangkat Keras

Perangkat keras yang digunakan dalam mengerjakan tugas akhir ini adalah sebagai berikut:

1. Satu unit *notebook* Compaq Presario CQ41 dengan *processor* Intel(R) Core(TM) i3 M350 @ 2.27 GHz.

2. RAM 2 GB
3. Harddisk
4. Scanner

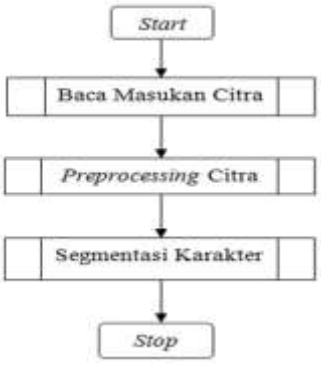
3.1.2 Perangkat Lunak

Perangkat lunak yang digunakan dalam mengerjakan tugas akhir ini adalah sebagai berikut:

1. Sistem Operasi Windows 7 Home Premium 32-bit
2. Microsoft Visual Studio 2008

3.2 Rancangan Sistem

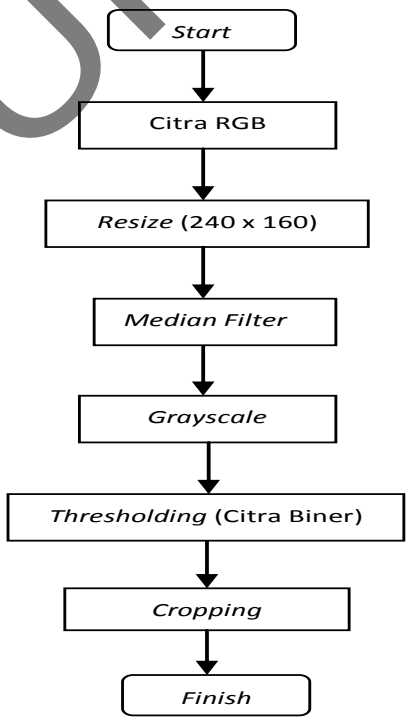
Rancangan sistem menjelaskan diagram alir utama sistem secara keseluruhan dari tugas akhir ini. Cara kerja sistem pada penelitian ini dibagi menjadi dua bagian, yaitu *preprocessing* citra dan segmentasi Nomor Induk Mahasiswa (NIM) menggunakan *Connected Component Labeling* yang dikembangkan oleh Stefano dan Bulgarelli (1999). *Preprocessing* citra sebagai proses awal yang harus dilakukan sebelum citra tersebut akan disegmentasi. Setelah itu citra hasil dari *preprocessing* akan dilakukan segmentasi karakter. Gambar 3.1 merupakan diagram alir utama sistem yang akan digunakan pada pembuatan sistem ini.



Gambar 3.1 Diagram Alir Utama Sistem

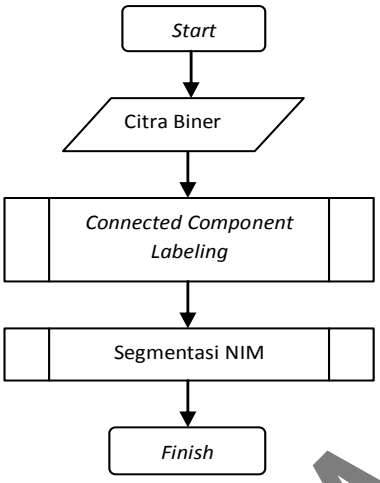
3.2.1 *Preprocessing* Citra

Pada bagian *preprocessing* citra terdiri dari beberapa proses yang harus dilakukan sebelum dilakukan proses segmentasi. Proses tersebut adalah sistem membaca citra yang dimasukkan sebagai *input* awal yaitu berupa citra RGB Kartu Tanda Mahasiswa (KTM) yang berformat *bitmap* (BMP) 24 bit. Selanjutnya, citra RGB dilakukan proses *resize* ukuran citra menjadi 240 x 160 piksel. Citra hasil proses *resize* dilakukan proses *filtering* menggunakan metode *median filter,* kemudian diubah menjadi citra *grayscale* dengan menggunakan persamaan [2.1]. Hasil dari konversi citra ke *grayscale* dilakukan *thresholding* biner untuk mengubah citra *grayscale* menjadi citra biner. Pengguna dapat mengetikkan nilai *threshold* yang diinginkan dan setelah itu dilakukan *thresholding* biner sesuai dengan ketentuan pada persamaan [2.3]. Kemudian dilakukan proses *cropping* untuk memotong bagian citra KTM yang tidak akan disegmentasi dan untuk mendapatkan citra NIM yang akan disegmentasi. Diagram alir untuk *preprocessing* citra dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Alir *Preprocessing* Citra

3.2.2 Segmentasi Nomor Induk Mahasiswa



Gambar 3.3 Diagram Alir Segmentasi NIM

Diagram alir segmentasi karakter dapat dilihat pada Gambar 3.3. Proses terakhir setelah *image preprocessing* adalah segmentasi NIM. Pada proses segmentasi ini terdapat dua proses penting yang harus dilakukan, yaitu proses pelabelan dengan *Connected Component Labeling* dan ekstrak karakter hasil proses pelabelan piksel untuk melakukan segmentasi NIM. Pada proses pelabelan piksel menggunakan konsep 4-konektivitas.

3.2.3 *Connected Component Labeling*

Seperti yang telah dijelaskan pada Bab 2 bagian implementasi sistem, menurut Bailey dan Johnston (2007) proses pelabelan piksel menggunakan *Connected Component Labeling* klasik ada dua tahap, yaitu sebagai berikut:

1. Pelabelan tahap 1, yaitu pemindaian dan pemberian label sementara pada citra biner. Jika pada proses pemindaian piksel ketemu nilai 1 *(foreground)*, maka cek nilai tetangga atas dan tetangga kirinya. Terdapat tiga kondisi pada proses pelabelan piksel tahap pertama, yaitu sebagai berikut:
    a. Jika tetangga kiri dan tetangga atas adalah *background*, maka *increment* nomor label dan berilah nomor label yang baru.

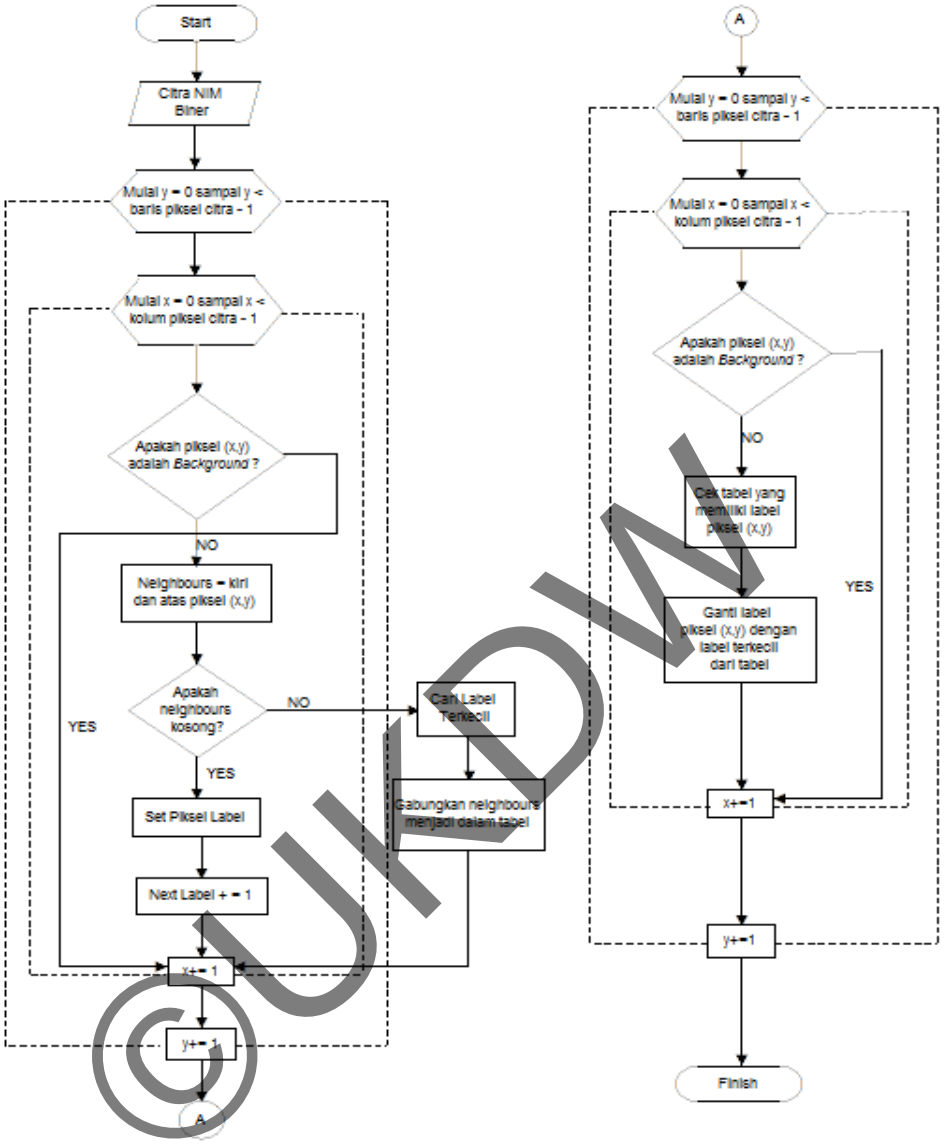
b. Jika hanya bertemu dengan tetangga kiri atau tetangga atas dan hanya salah satu saja yang memiliki nomor label, maka beri nomor label sesuai dengan nilai tetangga yang memiliki nomor label.
c. Jika bertemu dengan tetangga kiri dan tetangga atas, maka beri nomor label dengan nomor label tetangga yang lebih kecil.

2. Pelabelan tahap kedua, yaitu penggantian label dengan label yang sesuai dengan nomor label yang terdapat pada tabel ekuivalen.

Algoritma pelabelan *Connected Component Labeling* yang dimodifikasi oleh Stefano dan Bulgarelli (1999) dengan menggunakan konsep 4-konektivitas, adalah sebagai berikut:

1. *Scan* dan cek semua piksel dari citra biner, dari atas ke bawah dan mulai dari kiri ke kanan (mulai dari y ke x).
2. Jika ketemu piksel bernilai 1 *(foreground),* cek tetangga kiri dan tetangga atas.
3. Jika piksel tetangga kiri dan tetangga atas bernilai 0 *(background),* maka *increment* nomor label dan beri nomor label yang baru.
4. Jika bertemu hanya dengan tetangga kiri atau tetangga atas di mana hanya salah satu tetangga yang memiliki nomor label, maka beri nomor label sesuai dengan tetangga yang memiliki nomor label.
5. Jika bertemu dengan tetangga kiri dan tetangga atas, maka beri nomor label dengan nomor label tetangga yang lebih kecil.
6. Selama proses pelabelan tahap pertama dilakukan proses ekuivalensi dengan penggabungan kelas-kelas ekuivalen dengan segera ketika ekuivalen yang baru ditemukan.
7. Penggantian nomor label dengan nomor label yang ekuivalen sesuai dengan tabel ekuivalen.
8. Selesai.

Untuk memudahkan memahami algoritma *Connected Component Labeling* di atas, maka akan disajikan diagram alir *(flowchart) Connected Component Labeling* yang dapat dilihat pada Gambar 3.4.



Gambar 3.4 Diagram Alir *Connected Component Labeling*
Dikutip dari: Hartono (2013). Konversi Citra Papan Penunjuk Jalan menjadi Karakter ASCII dengan Optical Character Recognition. (Skripsi S1, Universitas Kristen Duta Wacana, 2013). Diakses pada tanggal 28 Februari 2014 dari http://sinta.ukdw.ac.id

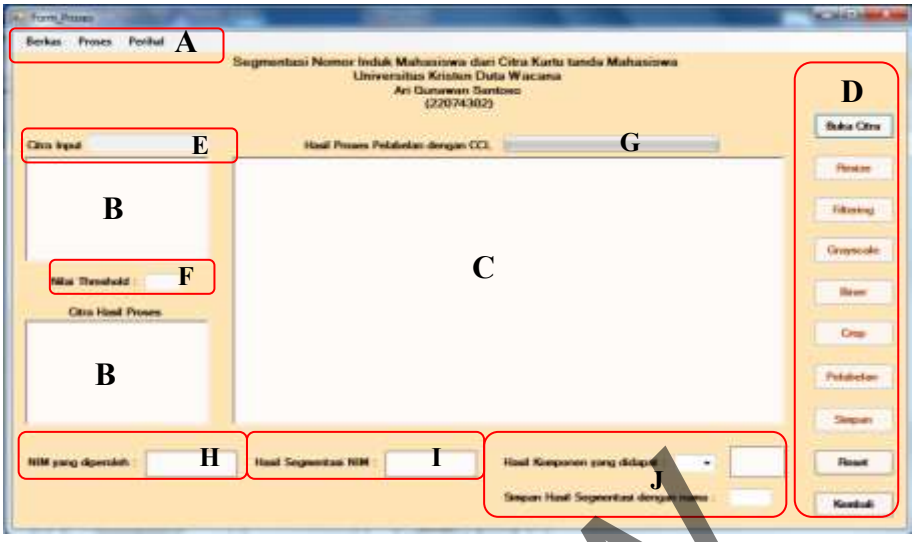
3.3 Perancangan Antarmuka Sistem

Antarmuka sistem ini terdiri dari empat *form,* yaitu Form_Utama, Form_ Proses, Form_TentangPembuat, dan Form_TentangProgram. Form_Utama merupakan antarmuka sistem yang akan tampil pertama kali ketika aplikasi ini dijalankan. Perancangan antarmuka untuk Form_Utama dapat dilihat pada Gambar 3.5.

Gambar 3.5 Rancangan Antarmuka Form_Utama

Gambar 3.5 merupakan rancangan tampilan *form* yang akan muncul ketika sistem dijalankan pertama kali. Di dalam rancangan antarmuka Form_Utama di atas hanya berisi tentang informasi nama aplikasi dan juga pembuat serta terdapat dua tombol yaitu tombol Keluar dan tombol Buka. Tombol Keluar digunakan apabila pengguna ingin keluar dari aplikasi, sedangkan tombol Buka untuk membuka Form_Proses. Selain kedua tombol tersebut, *form* ini juga dilengkapi dengan menu untuk membuka Form_TentangProgram dan Form_TentangPembuat.

Rancangan antarmuka *form* yang kedua adalah Form_Proses yang terdiri dari empat bagian utama, yaitu *menu strip,* area citra, tombol proses, dan area untuk menampilkan hasil proses pelabelan yaitu berupa *rich text box*. *Menu strip* memuat perintah-perintah proses untuk melakukan suatu fungsi program dan pengguna dapat mengakses dengan melakukan klik pada perintah proses yang dikehendaki. Tombol proses mempermudah pengguna di dalam memilih suatu proses yang dikehendaki dan pengguna cukup melakukan klik pada tombol proses. Area citra merupakan bagian yang digunakan untuk menampilkan citra setelah pengguna memilih suatu proses yang dikehendaki. *Rich text box* digunakan untuk menampilkan proses pelabelan piksel pada citra biner menggunakan *connected component labeling.* Rancangan antarmuka Form_Proses yang akan dibuat dapat dilihat pada Gambar 3.6.



Gambar 3.6 Rancangan Antarmuka Form_Proses

Selain empat bagian di atas, rancangan antarmuka Form_Utama juga dilengkapi dengan sebuah *textbox* yang akan berisi *file name* dari citra yang dipilih oleh pengguna dan letak *textbox* ditunjukkan pada huruf E. Sebuah *editbox* yang digunakan untuk memasukkan nilai *threshold* yang dapat diketikkan pengguna sebelum melakukan konversi ke citra biner dan letak *editbox* ditunjukkan pada huruf F. Huruf G adalah *progress bar* untuk menunjukkan bahwa proses pelabelan dan segmentasi sedang melakukan proses. Huruf H adalah *picture box* yang digunakan untuk menampilkan citra hasil *cropping*. Huruf I menunjukkan *picture box* yang digunakan untuk menampilkan citra hasil pelabelan piksel dan segmentasi. Huruf J menunjukkan *combo box* yang berisi nomor label yang didapat dari proses pelabelan piksel, *picture* box yang digunakan untuk menampilkan citra hasil dari segmentasi NIM, dan *text box* yang digunakan untuk mengetikkan *file name* pada waktu menyimpan citra hasil segmentasi.

Berikut ini akan dijelaskan untuk masing-masing bagian pada tampilan antarmuka *Form* Proses yang akan dibuat:

1. **Menu (A),** terdiri dari:
    a. Berkas, dengan submenu:
        - Buka Citra : digunakan untuk membuka citra Kartu Tanda Mahasiswa (KTM) UKDW.

- Kembali : digunakan untuk menutup Form_Proses dan kembali ke Form_Utama.

b. Proses, dengan submenu:
   - Resize : digunakan untuk mengatur ukuran citra *input* sesuai dengan ukuran yang telah ditentukan yaitu 240 x 160 piksel.
   - Filtering : digunakan untuk mengurangi *noise* pada citra hasil *resize* menggunakan metode *median filter.*
   - Grayscale : digunakan untuk mengkonversi citra RGB menjadi citra *grayscale.*
   - Biner : digunakan untuk mengubah citra *grayscale* menjadi citra biner.
   - Crop : digunakan untuk melakukan proses crop dan menentukan lokasi NIM pada citra KTM UKDW.
   - Pelabelan dengan CCL : digunakan untuk melakukan proses pelabelan piksel dan segmentasi NIM menggunakan *Connected Component Labeling*.
   - Simpan Hasil Segmentasi : digunakan untuk menyimpan hasil segmentasi NIM.
   - Reset : digunakan untuk mengembalikan sistem dalam keadaan *default.*

c. Perihal, dengan submenu:
   - Pembuat : digunakan untuk menampilkan Form_TentangPembuat yang berisi tentang informasi pembuat sistem.
   - Program : digunakan untuk menampilkan Form_Tentang Program yang berisi penjelasan singkat langkah-langkah proses dari sistem yang dibuat.

2. **Area Citra (B)**, terdiri dari:
    a. Citra Input : untuk menampilkan citra RGB KTM.
    b. Citra Hasil Proses : untuk menampilkan citra hasil proses konversi ke citra *grayscale,* proses *median filter,* konversi ke citra biner, proses pelabelan dan segmentasi NIM.

3. **Area Hasil Proses Pelabelan (C)**, yaitu area yang digunakan untuk menampilkan hasil dari proses pelabelan menggunakan *Connected Component Labeling,* baik proses pelabelan tahap pertama maupun tahap kedua.

4. **Tombol Proses (D)**, terdiri dari:
    a. Buka Citra : digunakan untuk membuka citra KTM UKDW.
    b. Resize : digunakan untuk mengatur ukuran citra *input* sesuai dengan ukuran yang telah ditentukan yaitu 240 x 160 piksel.
    c. Filtering : digunakan untuk mengurangi *noise* pada citra hasil *resize* dengan menggunakan metode *median filter.*
    d. Grayscale : digunakan untuk mengkonversi citra RGB menjadi citra *grayscale.*
    e. Biner : digunakan untuk mengubah citra *grayscale* menjadi citra biner.
    f. Crop : digunakan untuk melakukan proses crop dan menentukan lokasi NIM pada citra KTM UKDW.
    g. Pelabelan : digunakan untuk melakukan proses pelabelan piksel segmentasi NIM dengan *Connected Component Labeling*.
    h. Simpan : digunakan untuk menyimpan hasil segmentasi NIM.
    i. Reset : digunakan untuk mengembalikan sistem dalam keadaan *default.*
    j. Kembali : digunakan untuk menutup Form_Proses dan kembali ke Form_Utama.

Gambar 3.7 Rancangan Antarmuka Form_TentangProgram

Rancangan antarmuka *form* yang ketiga di dalam aplikasi ini adalah Form_TentangProgram yang ditunjukkan pada Gambar 3.7. Rancangan antarmuka *form* ini memberikan informasi langkah-langkah yang dilakukan dalam proses segmentasi NIM dari citra KTM UKDW. Rancangan antarmuka *form* tersebut juga terdapat dua tombol, yaitu tombol Tutup dan Keluar. Tombol Tutup digunakan untuk menutup *form* ini dan tombol Keluar digunakan untuk menutup *form* ini dan keluar dari aplikasi.

Rancangan antarmuka *form* yang keempat adalah Form_TentangPembuat dapat dilihat pada Gambar 3.8. Rancangan antarmuka *form* ini berisi tentang identitas pembuat dari aplikasi ini. Selain terdapat informasi identitas pembuat, *form* Pembuat terdapat dua tombol yaitu tombol Tutup dan Keluar. Tombol Tutup digunakan untuk menutup Fom_TentangPembuat dan tombol Keluar digunakan untuk menutup *form* ini dan keluar dari aplikasi.

Gambar 3.8 Rancangan Antarmuka Form_TentangPembuat

Bab 4
IMPLEMENTASI DAN ANALISIS SISTEM

Bab ini terdiri dari implementasi sistem, hasil uji coba sistem, dan analisis sistem. Implementasi sistem berisi tentang hasil implementasi sistem berdasarkan perancangan sistem pada Bab 3 sebelumnya. Hasil uji coba sistem akan menjelaskan hasil uji coba terhadap sistem yang telah dibangun. Uji coba dilakukan pada data yang telah dikumpulkan yaitu berupa hasil pemindaian Kartu Tanda Mahasiswa (KTM) UKDW. Bagian terakhir dari bab ini adalah analisis sistem yang berisi tentang analisis berdasarkan hasil penelitian yang telah dilakukan terhadap sistem yang telah dibangun.

4.1 Implementasi Sistem

Subbab ini akan menjelaskan implementasi sistem yang terdiri dari implementasi antarmuka sistem, implementasi proses dari *input* sampai *output,* dan implementasi algoritma yang digunakan di dalam membangun sistem ini. Berikut ini akan dijelaskan masing-masing implementasi dari sistem yang telah dibangun berdasarkan analisis dan perancangan antarmuka sistem yang telah dijelaskan pada Bab 3 sebelumnya.

4.1.1 Antarmuka Sistem

Implementasi antarmuka sistem menjelaskan hasil implementasi sistem berdasarkan perancangan antarmuka sistem yang telah dijelaskan pada Bab 3 sebelumnya. Antarmuka pada sistem ini terdiri dari 4 *form,* yaitu Form_Utama, Form_Proses, Form_TentangPembuat, dan Form_TentangProgram. *Form* yang pertama adalah Form_Utama. Form_Utama adalah *form* yang akan ditampilkan pada saat pertama aplikasi ini dijalankan. Tampilan *form* pada saat aplikasi ini dijalankan oleh pengguna dapat dilihat pada Gambar 4.1.

Gambar 4.1 Antarmuka Form_Utama

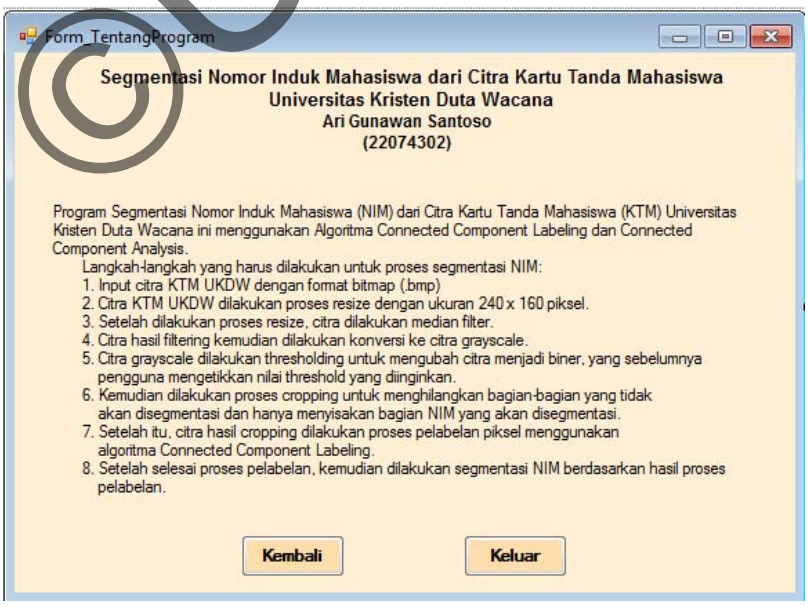
Form_Utama berisi judul tugas akhir dan nama pembuat sistem. *Form* ini mempunyai dua menu yang bertujuan untuk membuka Form_TentangProgram dan Form_TentangPembuat. Selain itu, *form* ini juga mempunyai dua tombol, yaitu tombol Keluar dan tombol Buka. Tombol Keluar digunakan apabila pengguna ingin keluar dari aplikasi, sedangkan tombol Buka digunakan apabila pengguna ingin membuka Form_Proses.

Form kedua adalah Form_Proses. Form_Proses merupakan *form* yang berisi semua proses untuk melakukan segmentasi NIM dari citra KTM UKDW, mulai dari *input* sampai *output*. Seluruh proses tersebut hasilnya akan ditampilkan pada *form* ini. Implementasi dari *form* ini dapat dilihat pada Gambar 4.2.

Gambar 4.2 Antarmuka Form_Proses

Bagian menu yang terdapat pada Gambar 4.2 terdiri dari tiga bagian, yaitu menu Berkas, Proses, dan Perihal. Menu Berkas memiliki dua sub menu yaitu Buka Citra dan Kembali. Menu Proses terdiri dari proses-proses utama dalam sistem yang meliputi sub menu proses Resize, Filtering, Grayscale, Biner, Crop, Pelabelan dengan CCL, Simpan Segmentasi NIM, dan Reset. Menu Perihal terdiri dari dua bagian yaitu Program dan Pembuat. Bagian area citra digunakan untuk menampilkan citra *input,* citra hasil proses dari proses *resize,* citra hasil *grayscale,* dan citra hasil biner. Citra hasil *cropping* NIM dan citra hasil segmentasi NIM ditampilkan pada *picture box* yang berbeda. Tombol proses terdiri dari tombol proses Buka Citra, Resize, Filtering, Grayscale, Biner, Crop, Pelabelan, Simpan, Reset, dan Keluar. Penjelasan dan fungsi masing-masing submenu dan tombol tersebut dapat dilihat pada Subbab 3.3.

Form yang ketiga pada sistem ini adalah Form_TentangProgram. Form_TentangProgram berisi penjelasan singkat langkah-langkah yang dilakukan dalam proses segmentasi NIM dari citra KTM UKDW mulai dari pengguna memasukkan citra KTM UKDW sampai didapatkan hasil segmentasi NIM. Gambar 4.3 menunjukkan tampilan implementasi dari Form_TentangProgram.



Gambar 4.3 Antarmuka Form_Tentang Program

Form ini akan ditampilkan jika pengguna menekan menu Tentang Program pada Form_Utama atau menekan menu Perihal dan selanjutnya memilih sub menu Program pada Form_Proses. Di dalam Form_TentangProgram juga terdapat dua tombol, yaitu tombol Kembali dan tombol Keluar. Tombol Kembali digunakan apabila pengguna ingin menutup *form* ini dan kembali ke Form_Utama. Tombol Keluar digunakan apabila pengguna ingin menutup semua *form* yang dibuka dan keluar dari aplikasi ini.

Form terakhir yang terdapat pada sistem ini adalah Form_TentangPembuat. Form_TentangPembuat adalah *form* yang memberi informasi tentang pembuat aplikasi Segmentasi Nomor Induk Mahasiswa dari Citra Kartu Tanda Mahasiswa Universitas Kristen Duta Wacana. Informasi tersebut berisi nama, NIM, *e-mail*. Form_TentangPembuat akan ditampilkan jika pengguna menekan menu Tentang Pembuat pada Form_Utama atau menekan menu Perihal dan selanjutnya memilih sub menu Pembuat pada Form_Proses. Di dalam *form* ini juga terdapat dua tombol, yaitu tombol Kembali dan tombol Keluar. Tombol Kembali digunakan apabila pengguna ingin menutup *form* ini dan kembali ke Form_Utama. Tombol Keluar digunakan apabila pengguna ingin menutup semua *form* yang sedang dibuka dan keluar dari aplikasi. Tampilan dari implementasi Form_TentangPembuat dapat dilihat pada Gambar 4.4.

Gambar 4.4 Antarmuka Form_TentangPembuat

4.1.2 Implementasi *Input* dan *Output*

Implementasi *input* dan *output* menjelaskan proses yang harus dilakukan pada sistem ini. Proses pertama adalah pengguna memasukkan citra RGB KTM UKDW sebagai citra *input* hingga proses segmentasi NIM yang menghasilkan *output* berupa citra karakter hasil segmentasi NIM. Berikut ini akan dijelaskan masing-masing proses mulai dari *input* hingga diperoleh *output.*

Proses pertama yang dilakukan adalah pengguna memasukkan citra RGB KTM UKDW dengan format *bitmap* (BMP) dengan kedalaman warna 24 bit yang berukuran kurang dari 640 x 480 piksel sesuai dengan batasan masalah yang telah ditentukan pada Subbab 1.3. Jika citra yang dipilih melebihi ukuran tersebut, maka sistem akan menampilkan pesan di *message box* bahwa pengguna harus memasukkan kembali citra *input* dengan ukuran kurang dari 640 x 480 piksel.

Proses ini dapat dilakukan dengan menekan tombol proses Buka Citra atau dengan melakukan klik menu Berkas dan melakukan klik pada sub menu Buka Citra. Ketika proses memasukkan citra *input* selesai, maka secara otomatis tombol proses *Resize* menjadi aktif. Implementasi tampilan dari proses memasukkan citra *input* dapat dilihat pada Gambar 4.5.

Gambar 4.5 Proses Buka Citra

Gambar 4.6 Proses *Resize* Citra *Input*

Proses selanjutnya adalah proses *resize* ukuran citra yang bertujuan agar sistem tidak terlalu berat di dalam melakukan proses, selain itu waktu proses yang digunakan juga menjadi lebih cepat. Pada sistem ini, citra *input* dilakukan proses *resize* ukuran citra yaitu menjadi ukuran 240 x 160 piksel sesuai dengan batasan masalah yang telah ditentukan pada Subbab 1.3. Pengguna dapat menekan tombol Resize atau menekan menu Proses dan memilih sub menu Resize untuk melakukan proses ini. Tampilan dari proses *resize* ukuran citra *input* dapat dilihat pada Gambar 4.6. Ketika proses *resize* citra *input* selesai, maka secara otomatis tombol proses Grayscale menjadi aktif.

Setelah dilakukan proses *resize* ukuran pada citra *input*, maka proses selanjutnya adalah mengurangi *noise* yang terdapat pada citra *input.* Metode yang digunakan untuk mengurangi *noise* pada citra *input* yaitu dengan menggunakan metode *median filter.* Proses ini dapat dilakukan dengan menekan tombol proses Filtering atau dengan menekan menu Proses dan kemudian memilih sub menu Filtering. Gambar 4.7 merupakan tampilan implementasi proses *filtering* menggunakan *median filtering.*

Gambar 4.7 Proses *Filtering* menggunakan *Median Filter*

Gambar 4.7 merupakan tampilan dari hasil proses *filtering* dengan menggunakan *median filter.* Ketika proses *filtering* ini selesai maka tombol proses Grayscale secara otomatis akan aktif. Pengguna dapat melanjutkan ke proses selanjutnya yaitu konversi citra RGB hasil *median filter* ke citra *grayscale.*

Setelah proses *filtering* pada citra *input* selesai, maka proses selanjutnya adalah mengkonversi citra RGB menjadi citra *grayscale.* Pengguna dapat menekan tombol Grayscale atau menekan menu Proses dan memilih sub menu Grayscale untuk melakukan proses ini. Pada sistem ini, penulis menggunakan rumus rata-rata nilai dari R, G, dan B dari setiap piksel, kemudian nilai rata-rata tersebut disimpan dalam variabel ‘warna’ sesuai dengan persamaan [2.1] yang telah dijelaskan pada Sub subbab 2.2.2 sebelumnya. Ketika proses konversi citra RGB ke citra *grayscale* selesai, maka tombol proses Biner secara otomatis akan aktif. Gambar 4.8 merupakan tampilan dari hasil implementasi proses konversi citra RGB menjadi citra *grayscale.* Citra hasil konversi dari citra RGB menjadi citra *grayscale* tersebut yang akan digunakan untuk mengubah derajat keabuan pada citra *grayscale* menjadi citra biner (hitam-putih). Proses ini dilakukan dengan menggunakan *thresholding* biner.

Gambar 4.8 Proses Konversi Citra RGB ke Citra *Grayscale*

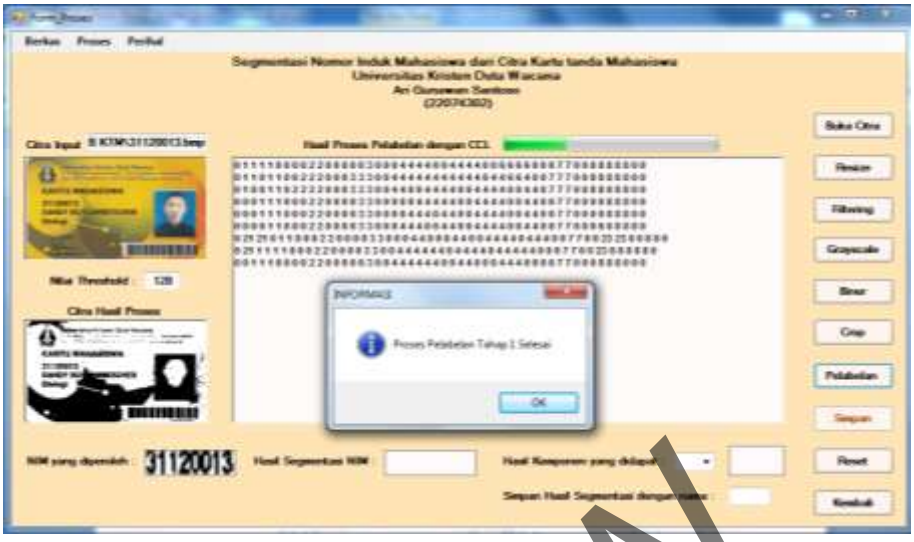
Proses selanjutnya adalah mengkonversi citra *grayscale* menjadi citra biner. Di dalam proses *thresholding* biner ini, penulis menggunakan aturan pada persamaan [2.3] yang telah dijelaskan pada Sub subbab 2.2.3. Pengguna dapat menekan tombol Biner atau menekan menu Proses dan memilih sub menu Biner untuk melakukan proses ini. Sebelum melakukan proses konversi citra *grayscale* ke citra biner, pengguna harus mengetikkan nilai *threshold* yang digunakan sebagai nilai ambang dalam *thresholding* biner.

Jika pengguna belum mengetikkan nilai *threshold,* maka sistem akan memberi pesan di *message box* bahwa pengguna harus mengetikkan nilai *threshold* terlebih dahulu. Pengguna hanya bisa mengetikkan nilai *threshold* antara 0 – 255, jika di luar rentang nilai tersebut maka sistem akan memberi pesan di *message box* bahwa nilai *threshold* yang harus pengguna masukkan harus di antara 0 – 255. Sistem ini juga mampu mendeteksi nilai *threshold* yang diketikkan pengguna, jika mengandung karakter huruf atau karakter lainnya maka sistem akan memberi pesan di *message box* bahwa nilai *threshold* yang dimasukkan harus berupa angka. Ketika proses konversi dari citra *grayscale* ke citra biner selesai, maka secara otomatis tombol proses Crop akan aktif. Tampilan dari implementasi proses konversi citra *grayscale* ke citra biner dapat dilihat pada Gambar 4.9.

Gambar 4.9 Proses Konversi Citra *Grayscale* ke Citra Biner

Proses selanjutnya adalah proses *cropping* citra yang bertujuan untuk memotong bagian citra KTM yang tidak akan disegmentasi sehingga dihasilkan citra NIM yang akan disegmentasi. Pengguna dapat menekan tombol Crop atau menekan menu Proses dan memilih sub menu Crop untuk melakukan proses ini. Gambar 4.10 merupakan tampilan implementasi citra proses hasil *cropping* yang menghasilkan citra berupa NIM yang akan disegmentasi. Ketika proses ini selesai, maka tombol proses Pelabelan secara otomatis akan aktif.

Gambar 4.10 Proses *Cropping* Citra NIM



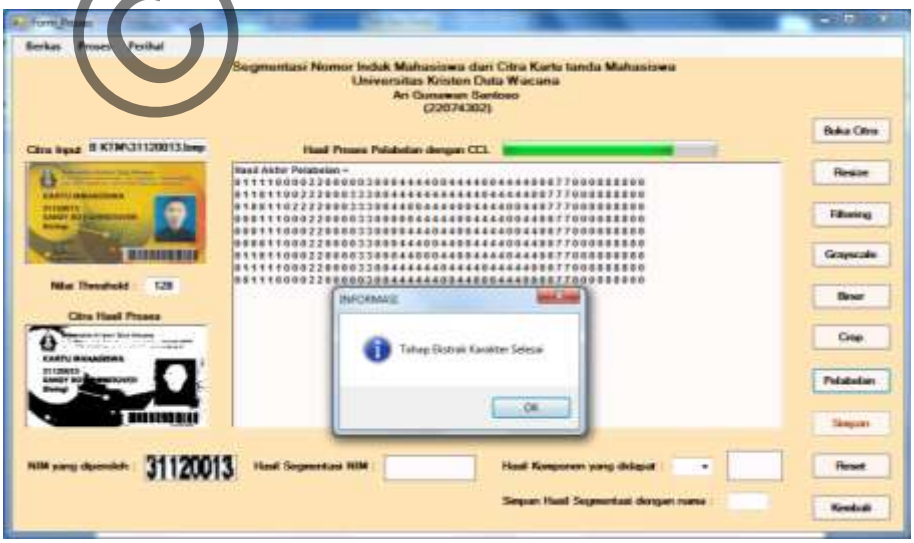
Gambar 4.11 Proses Pelabelan Piksel Tahap Pertama

Setelah proses *cropping*, maka dilakukan proses pelabelan piksel menggunakan *Connected Component Labeling* yang terdiri dari dua tahap, yaitu pelabelan piksel tahap pertama dan pelabelan piksel tahap kedua. Proses pelabelan piksel tahap pertama terdapat tiga kondisi yang menjadi ketentuan di dalam pemberian label pada piksel yang saling terhubung. Sistem akan menampilkan hasil dari proses pelabelan piksel pada *rich text box* sehingga pengguna dapat mengetahui hasil dari proses tersebut. Ketika proses pelabelan tahap pertama selesai akan muncul pesan di *message box* bahwa proses pelabelan tahap pertama sudah selesai. Pengguna harus menekan tombol OK sehingga sistem dapat melanjutkan ke proses yang selanjutnya yaitu proses pelabelan piksel tahap kedua. Tampilan implementasi hasil proses pelabelan piksel tahap pertama dapat dilihat pada Gambar 4.11.

Setelah proses pelabelan piksel tahap pertama selesai, maka dilanjutkan ke proses pelabelan piksel tahap kedua. Proses pelabelan piksel tahap kedua merupakan hasil dari penggantian nomor label sesuai dengan nomor label yang terdapat pada tabel ekuivalen Hasil dari proses pelabelan piksel tahap kedua ini ditampilkan pada *rich text box*. Tampilan dari implementasi proses pelabelan piksel tahap kedua dapat dilihat pada Gambar 4.12.

Gambar 4.12 Proses Pelabelan Piksel Tahap Kedua

Ketika proses pelabelan piksel tahap kedua selesai, maka sistem akan menampilkan pesan di *message box* bahwa proses pelabelan piksel tahap kedua telah selesai dan pengguna harus menekan tombol OK untuk melanjutkan proses selanjutnya. Proses terakhir di dalam sistem ini adalah proses ekstraksi dan segmentasi karakter untuk mendapatkan komponen yang saling terhubung pada citra Nomor Induk Mahasiswa (NIM). Gambar 4.13 merupakan tampilan implementasi dari proses ekstraksi karakter.



Gambar 4.13 Proses Ekstraksi Karakter

Gambar 4.14 Proses Segmentasi NIM

Proses ekstraksi karakter bertujuan untuk mencari titik x1, y1, x2, dan y2 yang merupakan posisi dari masing-masing komponen piksel yang saling terhubung yang memiliki nomor label yang sama. Hasil dari ekstraksi karakter ini akan dijadikan untuk melakukan proses segmentasi NIM berdasarkan piksel yang saling terhubung dan memiliki nomor label yang sama. Gambar 4.14 merupakan tampilan implementasi proses segmentasi NIM setelah proses ekstraksi karakter selesai. Setelah proses segmentasi NIM selesai, maka pengguna harus menekan tombol OK pada *message box* yang berisi bahwa segmentasi telah selesai. Citra hasil dari proses pelabelan piksel dan segmentasi NIM ditampilkan pula pada *picture box* pada hasil segmentasi NIM.

Pengguna dapat melihat komponen piksel yang saling terhubung yang didapat dari hasil proses pelabelan piksel pada *combo box* seperti yang ditunjukkan pada Gambar 4.15. *Combo box* tersebut berisi nomor-nomor label yang diperoleh dari proses pelabelan piksel dan segmentasi NIM. Pada Gambar 4.15 tersebut terdapat 6 komponen terhubung, yaitu dengan nomor label 1, 2, 3, 4, 7, dan 8. Ketika pengguna memilih salah satu nomor label pada *combo box* tersebut, maka pada *picture box* akan ditampilkan citra karakter angka pada NIM yang memiliki nomor label yang pengguna pilih sebelumnya pada *combo box.*

Gambar 4.15 Hasil Komponen dari Proses Pelabelan Piksel

Proses selanjutnya adalah melihat hasil akhir dari segmentasi NIM berupa citra karakter NIM yang memiliki piksel saling terhubung dan nomor label yang sama. Tampilan hasil *output* dari sistem ini berupa citra karakter angka pada NIM yang berukuran 30 x 50 piksel. Citra hasil segmentasi tersebut diperoleh dengan memilih nomor label pada komponen piksel terhubung yang terdapat pada *combo box.* Tampilan hasil *output* segmentasi ini akan ditampilkan tiap-tiap karakter yang memiliki piksel yang saling terhubung dan nomor label yang sama. Gambar 4.16 merupakan tampilan dari hasil segmentasi karakter NIM setelah pengguna memilih salah satu nomor label yang terdapat pada *combo box.*

Gambar 4.16 Hasil Segmentasi Karakter NIM

Misalkan pengguna memilih komponen dengan nomor label 1, maka akan ditampilkan citra karakter angka 3 seperti yang ditunjukkan pada Gambar 4.16. Hasil akhir dari segmentasi NIM tersebut dapat disimpan ke dalam bentuk citra berformat bitmap (.bmp). Pengguna dapat menyimpan masing-masing hasil segmentasi NIM tersebut. Sebelumnya, pengguna harus mengetikkan *file name* untuk citra karakter NIM terlebih dahulu pada *text box* "Simpan Hasil Segmentasi dengan nama", setelah itu tekan tombol Simpan atau menekan menu Proses dan memilih Simpan Hasil Segmentasi. Jika pengguna belum mengetikkan *file name* untuk menyimpannya, maka sistem akan memberikan pesan bahwa "Masukkan File Name untuk Karakter NIM yang akan Disimpan". Jika tampilan *picture box* hasil segmentasi NIM masih kosong, maka sistem akan memberikan pesan di *message box* bahwa "Pilih Komponen Hasil CCL yang akan Disimpan Terlebih Dahulu".

4.1.3 Implementasi Algoritma

Sub-bab ini akan menjelaskan implementasi algoritma-algoritma yang digunakan untuk membangun sistem ini. Algoritma yang dipakai untuk melakukan segmentasi Nomor Induk Mahasiswa pada citra Kartu Tanda Mahasiswa (KTM) UKDW adalah *Connected Component Labeling.* Algoritma ini menjadi algoritma utama untuk melakukan proses segmentasi NIM dengan cara pemberian label pada piksel yang saling terhubung. Selain algoritma tersebut, sistem ini juga mengimplementasikan algoritma-algoritma lain yang digunakan pada proses awal pengolahan sebelum dilakukan segmentasi *(image preprocessing),* antara lain melakukan *median filter* pada citra *input* yang telah dilakukan proses *resize* menjadi 240 x 160 piksel, mengubah citra RGB menjadi citra *grayscale,* mengubah citra *grayscale* menjadi citra biner, dan proses pelabelan piksel menggunakan *Connected Component Labeling.*

Pertama, penulis akan menjelaskan untuk algoritma untuk mengurangi *noise* pada citra *input,* yaitu menggunakan metode *median filter.* Pada proses ini

pertama yang dilakukan adalah mengambil matriks "jendela" yang berukuran 3 x 3, kemudian piksel-piksel pada matriks "jendela" tersebut diurutkan nilai pikselnya mulai dari terkecil hingga terbesar *(ascending).* Setelah diurutkan secara *ascending* diambil nilai tengahnya *(*nilai *median)* dan nilai tersebut digunakan untuk mengganti titik pusat pada matriks "jendela". Proses ini dilakukan pada semua piksel citra, tetapi untuk tepi citra tidak terkena proses *median filter.* Hal ini disebabkan titik pusat pada matriks "jendela" dimulai dari titik (1,1) bukan titik (0,0). Gambar 4.17 merupakan potongan program proses *median filtering.*

```
For y = 0 To citrafiltering.Height - 1
    For x = 0 To citrafiltering.Width - 1
        count = 0
        For i = 0 To 2
            For j = 0 To 2

              If ((i < 0) Or (i >= citrafiltering.Width) Or (j < 0) Or
(j >= citrafiltering.Height)) Then
                 arrTempR(count) = 0
                 arrTempG(count) = 0
                 arrTempB(count) = 0
              Else
                 arrTempR(count) = citrafiltering.GetPixel(i, j).R
                 arrTempG(count) = citrafiltering.GetPixel(i, j).G
                 arrTempB(count) = citrafiltering.GetPixel(i, j).B
              End If

            Next j
        Next i
    Next x
Next y

For i = 0 To count - 1
    For j = 0 To count - 1
        If (arrTempR(i) > arrTempR(j)) Then
           Temp = arrTempR(i)
           arrTempR(i) = arrTempR(j)
           arrTempR(j) = Temp
        End If

        If (arrTempG(i) > arrTempG(j)) Then
           Temp = arrTempR(i)
           arrTempG(i) = arrTempG(j)
           arrTempG(j) = Temp
        End If

        If (arrTempB(i) > arrTempB(j)) Then
           Temp = arrTempB(i)
           arrTempB(i) = arrTempB(j)
           arrTempB(j) = Temp
        End If

citrafiltering.SetPixel(x, y, Color.FromArgb((arrTempR(count / 2)),
(arrTempG(count / 2)), (arrTempB(count / 2))))
    Next j
Next i
```


Gambar 4.17 Potongan Program Proses *Median Filter*

```
For y = 0 To citragrayscale.Height - 1
  For x = 0 To citragrayscale.Width - 1
     warna = citragrayscale.GetPixel(x, y).R
     warna += citragrayscale.GetPixel(x, y).G
     warna += citragrayscale.GetPixel(x, y).B
     warna = Math.Round(warna / 3)
     citragrayscale.SetPixel(x, y, Color.FromArgb(warna,
     warna, warna))
   Next x
Next y
```

Gambar 4.18 Potongan Program Konversi Citra RGB menjadi Citra *Grayscale*

Selanjutnya penulis akan menjelaskan algoritma yang digunakan yaitu untuk mengubah citra RGB menjadi citra *grayscale* yang menggunakan rata-rata nilai R, G, dan B seperti yang telah dijelaskan pada persamaan [2.1]. Gambar 4.18 merupakan potongan program konversi citra RGB menjadi citra *grayscale.* Berdasarkan potongan program tersebut, bahwa proses konversi citra RGB menjadi citra *grayscale* pertama dilakukan proses *scanning* piksel mulai dari kiri atas ke kanan bawah. Setiap piksel pada citra RGB KTM UKDW diambil masing-masing nilai R, G, dan B kemudian dijumlahkan dan diambil rata-ratanya. Nilai rata-rata dari nilai R, G, B pada piksel tersebut menunjukkan derajat keabuan dan ditampung pada variabel warna yang akan menggantikan nilai pada tiap piksel.

Setelah proses konversi citra RGB ke *grayscale* selesai, maka selanjutnya adalah proses mengubah citra *grayscale* menjadi citra biner. Setiap piksel pada citra *grayscale* akan dilakukan *thresholding* biner dengan cara derajat keabuan yang terdapat pada piksel citra *grayscale* dibandingkan nilainya dengan nilai *threshold* (ambang batas) yang telah ditentukan pengguna sebelumnya sesuai dengan ketentuan [2.3]. Pada sistem ini, jika warna pada piksel tersebut kurang dari nilai *threshold,* maka warna pada piksel tersebut diubah menjadi hitam dan diberi nilai 1 *(foreground),* sedangkan jika lebih dari nilai *threshold*, maka warna pada piksel tersebut diubah menjadi putih dan diberi nilai 0 *(background).*

Gambar 4.19 merupakan potongan program dari konversi citra *grayscale* ke citra biner. Berdasarkan potongan program tersebut, sistem ini mampu untuk mendeteksi pengguna sudah memasukkan nilai *threshold* atau belum. Jika pengguna belum mengetikkan nilai *threshold,* maka sistem akan menampilkan

pesan di *message box* “Masukkan dulu nilai threshold”. Selain itu sistem juga mampu mendeteksi nilai *threshold* yang diketikkan pengguna mengandung bukan karakter angka, maka sistem akan menampilkan pesan di *message box* yang berisi bahwa “Nilai threshold yang Anda masukkan bukan berupa angka” dan nilai *threshold* yang telah diketikkan pengguna akan dihapus kemudian muncul pesan kembali agar pengguna mengetikkan kembali nilai *threshold.* Jika pengguna mengetikkan nilai *threshold* tidak berada di antara rentang nilai 0-255, maka sistem akan memunculkan pesan di *message box* yang berisi bahwa “Nilai threshold yang Anda ketikkan harus di antara 0-255”.

```
For y = 0 To citrabiner.Height - 1
  For x = 0 To citrabiner.Width - 1
    warna = citrabiner.GetPixel(x, y).R
    warna += citrabiner.GetPixel(x, y).G
    warna += citrabiner.GetPixel(x, y).B
    warna = Math.Round(warna / 3)
    citrabiner.SetPixel(x, y, Color.FromArgb(warna, warna, warna))

      If (TextBoxThreshold.Text = vbNullString) Then
        MessageBox.Show("Masukkan dulu nilai threshold",
        "WARNING!!!",
        MessageBoxButtons.OK, MessageBoxIcon.Warning)
        Return
      Else
          If (IsNumeric(TextBoxThreshold.Text) = False) Then
          MessageBox.Show("Nilai threshold yang Anda masukkan
          bukan berupa angka", "WARNING!!!",
          MessageBoxButtons.OK, MessageBoxIcon.Warning)
          TextBoxThreshold.Text = ""
        ElseIf ((Val(TextBoxThreshold.Text) >= 0) And
        (Val(TextBoxThreshold.Text) <= 255)) Then
          If warna < Val(TextBoxThreshold.Text) Then
            citrabiner.SetPixel(x, y, Color.Black)
            arrPixel_label(x, y) = 1
          Else
            citrabiner.SetPixel(x, y, Color.White)
            arrPixel_label(x, y) = 0
          End If
        Else
          TextBoxThreshold.Text = ""
          MessageBox.Show("Nilai threshold yang Anda ketikkan
          harus di antara 0 - 255", "WARNING!!!",
          MessageBoxButtons.OK, MessageBoxIcon.Warning)
        End If
      End If

  Next x
Next y
```

Gambar 4.19 Potongan Program Konversi Citra *Grayscale* ke Citra Biner

Proses selanjutnya adalah proses pelabelan piksel dan segmentasi NIM menggunakan algoritma *Connected Component Labeling* yang telah dimodifikasi oleh Stefano dan Bulgarelli (1999). Pada proses pelabelan ini terdiri dari dua tahap. Pelabelan piksel tahap pertama merupakan proses pelabelan tiap piksel dari kiri atas ke kanan bawah (mulai dari y ke x). Pada proses ini terdapat tiga kondisi atau ketentuan seperti pada metode klasik, hanya terdapat perbedaan pada pelabelan tahap pertama seperti yang telah dijelaskan pada Sub-bab 2.3.

Kondisi yang pertama adalah jika nilai tetangga kiri dan nilai tetangga atas adalah 0 *(background),* maka nomor label akan melakukan proses *increment* dan piksel tersebut diberi nomor label yang baru. Gambar 4.20 merupakan potongan program untuk kondisi pertama pada proses pelabelan tahap pertama.

Kondisi yang kedua adalah jika bertemu hanya salah satu tetangga yang memiliki nomor label baik tetangga kiri maupun tetangga atas, maka piksel tersebut diberi nomor label sesuai dengan nomor label yang dimiliki tetangga tersebut. Gambar 4.21 merupakan potongan program untuk kondisi kedua pada pelabelan piksel tahap pertama.

```
If ((tetangga_kiri = 0) And (tetangga_atas = 0)) Then
  nomorlabel += 1
  arrPixel_label(x, y) = nomorlabel
```

Gambar 4.20 Potongan Program Kondisi Pertama

```
ElseIf ((tetangga_kiri = 0) And (tetangga_atas <> 0)) Or
((tetangga_kiri <> 0) And (tetangga_atas = 0)) Then
  If (tetangga_kiri <> 0) Then
    If (tetangga_kiri < arrTabel_label(tetangga_kiri)) Then
      arrPixel_label(x, y) = tetangga_kiri
      arrPixel_label(x - 1, y) = tetangga_kiri
      arrTabel_label(tetangga_kiri) = tetangga_kiri
    Else
      arrPixel_label(x, y) = arrTabel_label(tetangga_kiri)
      arrPixel_label(x - 1, y) = arrTabel_label(tetangga_kiri)
    End If

  ElseIf (tetangga_atas < arrTabel_label(tetangga_atas)) Then
    arrPixel_label(x, y) = tetangga_atas
    arrPixel_label(x, y - 1) = tetangga_atas
    arrTabel_label(tetangga_atas) = tetangga_atas
  Else
    arrPixel_label(x, y) = arrTabel_label(tetangga_atas)
    arrPixel_label(x, y - 1) = arrTabel_label(tetangga_atas)
  End If
```

Gambar 4.21 Potongan Program Kondisi Kedua

```
ElseIf ((tetangga_kiri <> 0) And (tetangga_atas <> 0)) Then
  If ((tetangga_kiri < tetangga_atas)) Then
    arrPixel_label(x, y) = tetangga_kiri
    arrPixel_label(x - 1, y) = tetangga_kiri
    arrPixel_label(x, y - 1) = tetangga_kiri
    arrTabel_label(tetangga_atas) = tetangga_kiri
  Else
    arrPixel_label(x, y) = tetangga_atas
    arrPixel_label(x - 1, y) = tetangga_atas
    arrPixel_label(x, y - 1) = tetangga_atas
    arrTabel_label(tetangga_kiri) = tetangga_atas
  End If
End If
```

Gambar 4.22 Potongan Program Kondisi Ketiga

Kondisi yang ketiga adalah jika piksel bertemu tetangga kiri dan tetangga atas yang keduanya memiliki nomor label. Pada kondisi seperti ini, piksel diberi nomor label tetangga yang lebih kecil di antara kedua piksel tetangga tersebut sebagai nomor label piksel tersebut. Gambar 4.22 merupakan potongan program kondisi ketiga pada pelabelan piksel tahap pertama.

Potongan program untuk proses pelabelan piksel tahap kedua ditunjukkan pada Gambar 4.23. Proses pelabelan tahap kedua dilakukan setelah proses pelabelan piksel tahap pertama selesai. Proses ini merupakan proses penggantian nomor label sesuai dengan nomor label yang terdapat pada tabel ekuivalen dan dilakukan normalisasi terhadap nomor label untuk mendapatkan komponen yang saling terhubung.

```
selesai = False
While selesai = False
  berubah = False
  'mengubah label sesuai dengan data konversi labelnya
    For y = 0 To hasilpelabelan.Height - 1
      For x = 0 To hasilpelabelan.Width - 1
        If ((arrTabel_label(arrPixel_label(x, y)) <
        arrPixel_label(x, y)) And (arrTabel_label(arrPixel_label(x,
        y)) <> -1)) Then
        arrPixel_label(x, y) = arrTabel_label(arrPixel_label(x, y))
          berubah = True
        End If
      Next x
    Next y

  If berubah = False Then
    selesai = True
  End If
End While
```

Gambar 4.23 Potongan Program Proses Pelabelan Piksel Tahap Kedua

4.2 Uji Coba Sistem

Pada subbab ini akan dijelaskan hasil dari uji coba sistem segmentasi Nomor Induk Mahasiswa (NIM) dari beberapa contoh citra Kartu Tanda Mahasiswa (KTM) UKDW yang dijadikan sebagai *sample* dalam uji coba sistem segmentasi NIM yang telah dibangun.

Pengujian yang pertama akan ditampilkan uji coba terhadap sistem berdasarkan nilai *threshold* yang digunakan oleh pengguna yang akan mempengaruhi hasil dari proses *cropping,* jumlah komponen yang didapatkan dari proses pelabelan piksel, dan hasil akhir segmentasi NIM. Tabel 4.1 merupakan hasil akhir proses segmentasi NIM berdasarkan nilai *threshold* yang digunakan oleh pengguna di dalam uji coba sistem ini, yaitu dengan menggunakan nilai *threshold* 100, 115, 128, dan 140.

Tabel 4.1

Hasil Segmentasi NIM berdasarkan Nilai *Threshold*

Citra RGB KTM	**Nilai *Threshold***	**Citra NIM**	**Jumlah Komponen yang didapat**	**Hasil Segmentasi NIM**
	100		6	1 :  2 :  3 :  4 :  5 :  8 : 

Tabel 4.1 (Sambungan)

Citra RGB KTM	Nilai *Threshold*	Citra NIM	Jumlah Komponen yang didapat	Hasil Segmentasi NIM
	115		6	1 :  2 :  3 :  4 :  5 :  8 : 
	128		5	1 :  2 :  3 :  4 :  8 : 
	140		4	1 :  2 :  3 :  4 : 
	100		7	1 :  2 :  3 : 

Tabel 4.1 (Sambungan)

Citra RGB KTM	Nilai *Threshold*	Citra NIM	Jumlah Komponen yang didapat	Hasil Segmentasi NIM
				4 : 
				6 : 
				7 : 
				9 : 
	115		5	1 : 
				2 : 
				3 : 
				4 : 
				5 : 
	128		4	1 : 
				2 : 
				3 : 
				4 : 
	140		4	1 : 
				2 : 
				3 : 
				4 : 

Tabel 4.1 (Sambungan)

Citra RGB KTM	Nilai *Threshold*	Citra NIM	Jumlah Komponen yangdidapat	Hasil Segmentasi NIM
	100		7	1 :  2 :  3 :  4 :  5 :  7 :  8 : 
	115		6	1 :  2 :  3 :  4 :  5 :  8 : 
	128		5	1 : 2 :  6 :  8 :  9 : 

Tabel 4.1 (Sambungan)

Citra RGB KTM	Nilai *Threshold*	Citra NIM	Jumlah Komponen yangdidapat	Hasil Segmentasi NIM
	140		4	1 :  2 :  3 :  4 : 

Berdasarkan Tabel 4.1 dapat disimpulkan bahwa nilai *threshold* yang digunakan akan berpengaruh pada hasil *cropping* citra yang menghasilkan citra NIM, jumlah komponen yang didapat pada proses pelabelan piksel dan tampilan hasil akhir dari proses segmentasi NIM yang diuji pada citra KTM yang sama. Selain itu, hasil dari pelabelan piksel dan segmentasi NIM menggunakan *Connected Component Labeling* menyebabkan hasil segmentasi NIM tidak bisa urut sesuai dengan karakter angka pada citra NIM. Hal ini disebabkan tinggi karakter angka pada citra NIM tersebut lebih tinggi dibanding tinggi karakter angka yang lain (tinggi karakter angka tidak sejajar). Kondisi seperti ini terjadi seperti pada citra dengan NIM 12120007 yang menggunakan nilai *threshold* 100, citra dengan NIM 31120006 yang menggunakan nilai *threshold* 100, dan citra dengan NIM 41120033 yang menggunakan nilai *threshold* 128. Proses pelabelan piksel menggunakan *Connected Component Labeling* dilakukan pemindaian dan pemberian piksel dari kiri atas ke kanan bawah sehingga piksel dengan tinggi karakter yang lebih tinggi akan diberi nomor label terlebih dahulu.

Pengujian yang kedua, penulis melakukan uji coba sistem berdasarkan kondisi warna KTM yang di-*scan.* Beberapa di antara citra KTM tersebut memiliki warna yang terlalu cerah dan terlalu gelap. Kondisi seperti ini juga akan mempengaruhi penggunaan dan pemilihan nilai *threshold* yang optimal. Hal ini dilakukan untuk mendapatkan hasil *cropping* citra dan segmentasi NIM yang

sempurna. Berdasarkan uji coba terhadap citra KTM dengan kondisi warna yang terlalu cerah, nilai *threshold* yang digunakan harus berada di antara 140-155 untuk mendapatkan hasil segmentasi NIM yang sempurna. Sedangkan untuk kondisi citra KTM dengan warna yang terlalu gelap, nilai *threshold* yang digunakan harus berada di antara 110-120 untuk mendapatkan hasil segmentasi NIM yang sempurna. Tabel 4.2 merupakan hasil dari uji coba sistem terhadap citra KTM dengan kondisi warna yang terlalu cerah dan terlalu gelap.

Tabel 4.2
Hasil Segmentasi NIM berdasarkan Kondisi Warna Citra KTM

Citra RGB KTM	**Nilai *Threshold***	**Citra NIM**	**Jumlah Komponen yang didapat**	**Hasil Segmentasi NIM**
	140		6	1 :  3 :  4 :  5 :  6 :  8 : 
	155		3	1 :  6 :  8 : 
	110		5	1 :  3 : 

Tabel 4.2 (Sambungan)

Citra RGB KTM	Nilai *Threshold*	Citra NIM	Jumlah Komponen yang didapat	Hasil Segmentasi NIM
				6 : 
				7 : 
				8 : 
	120		4	1 : 
				6 : 
				7 : 
				8 : 

4.3 Analisis Sistem

Bagian ini akan menjelaskan analisis sistem berdasarkan hasil uji coba sistem yang telah dibangun. Hal ini dilakukan untuk mengetahui kemampuan sistem dalam melakukan segmentasi Nomor Induk Mahasiswa (NIM) dari citra Kartu Tanda Mahasiswa (KTM) UKDW. Sistem segmentasi NIM ini menggunakan beberapa proses seperti yang sudah dijelaskan pada Subbab 4.1.2 di atas, mulai dari *input* berupa citra RGB KTM UKDW, *image preprocessing* hingga proses pelabelan piksel dan segmentasi NIM menggunakan *Connected Component Labeling*. Setelah dilakukan uji coba sistem maka terdapat beberapa faktor yang mempengaruhi di dalam proses segmentasi NIM sehingga menyebabkan proses segmentasi NIM tidak sempurna.

Berikut ini akan dijelaskan faktor-faktor yang mempengaruhi di dalam proses segmentasi NIM dan hasil akhir segmentasi NIM. Faktor pertama, nilai *threshold* yang digunakan oleh pengguna untuk melakukan konversi citra

grayscale ke citra biner akan memberikan hasil yang berbeda-beda pada hasil proses pelabelan piksel dan hasil akhir dari proses segmentasi NIM. Berdasarkan uji coba yang dilakukan dengan menggunakan nilai *threshold* 100, 115, 128, dan 140 citra KTM dapat disegmentasi dengan baik, meskipun hasil dari segmentasi NIM tersebut berbeda untuk masing-masing nilai *threshold.*

Faktor kedua, KTM yang di-*scan* beberapa memiliki warna terlalu cerah atau terlalu gelap. Berdasarkan uji coba yang telah dilakukan terhadap KTM dengan kondisi warna yang terlalu cerah, nilai *threshold* yang digunakan harus berada di antara 140-155 untuk mendapatkan hasil segmentasi NIM yang sempurna. Sedangkan untuk kondisi KTM dengan warna yang terlalu gelap, nilai *threshold* yang digunakan harus berada di antara 110-120 untuk mendapatkan hasil segmentasi NIM yang sempurna. Dengan demikian, sistem ini tidak dapat menentukan nilai *threshold* standar terbaik untuk menghasilkan segmentasi NIM yang sempurna, karena nilai *threshold* bersifat kondisional atau bergantung pada kondisi citra KTM yang dimasukkan pengguna.

Faktor ketiga, hasil dari proses pelabelan piksel dan segmentasi NIM menggunakan *Connected Component Labeling* tidak dapat disegmentasi secara urut. Hal ini disebabkan karena proses pelabelan piksel menggunakan *Connected Component Labeling* melakukan pemindaian dan pemberian piksel dari kiri atas ke kanan bawah. Pada proses pelabelan piksel ini, nomor label akan dilabeli terlebih dahulu pada piksel dengan tinggi karakter angka yang lebih tinggi dibandingkan tinggi karakter angka lainnya.

Faktor keempat, sistem di dalam melakukan segmentasi NIM masih belum dapat mensegmentasi masing-masing karakter, hal ini disebabkan jarak antar karakter yang saling berdekatan sehingga piksel yang saling berdekatan tersebut akan saling terhubung dan diberi nomor label yang sama. Hal ini ditunjukkan pada tabel hasil uji coba sistem, beberapa karakter dianggap objek yang sama dan disegmentasi menjadi satu bagian.

4.3.1 Kelebihan dan Kekurangan Sistem

Setelah dilakukan uji coba dan analisis terhadap sistem segmentasi NIM ini, penulis menemukan beberapa kelebihan dan kekurangan dari sistem yang telah dibangun. Kelebihan yang dimiliki sistem ini adalah mampu mensegmentasi NIM dengan waktu proses yang cepat. Selain itu, sistem segmentasi NIM memiliki beberapa kekurangan setelah dilakukan uji coba sistem. Beberapa kekurangan tersebut yaitu sistem tidak mampu menentukan nilai *threshold* yang optimal dan sesuai dengan kondisi citra KTM yang dimasukkan pengguna untuk menghasilkan segmentasi NIM yang baik. Segmentasi menggunakan *Connected Component Labeling* akan melakukan proses pelabelan dari kiri atas sampai kanan bawah, hal ini menyebabkan hasil segmentasi karakter tidak dapat urut sesuai dengan urutan karakter angka pada NIM jika tinggi karakter berbeda satu dengan lainnya. Sistem di dalam melakukan segmentasi NIM masih belum dapat mensegmentasi masing-masing karakter, hal ini disebabkan jarak antar karakter yang saling berdekatan sehingga piksel yang saling berdekatan tersebut akan saling terhubung dan diberi nomor label yang sama.

LAMPIRAN

LAMPIRAN A
LISTING PROGRAM

Form_Utama.vb

```
Public Class Form_Utama

    Private Sub Button_Buka_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button_Buka.Click
        Form_Proses.Show()
        Me.SuspendLayout()
    End Sub

    Private Sub Button_Keluar_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button_Keluar.Click
        Application.Exit()
    End Sub

    Private Sub TentangProgramToolStripMenuItem_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
TentangProgramToolStripMenuItem.Click
        Form_TentangProgram.Show()
        Me.SuspendLayout()
    End Sub

    Private Sub TentangPembuatToolStripMenuItem_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
TentangPembuatToolStripMenuItem.Click
        Form_TentangPembuat.Show()
        Me.SuspendLayout()
    End Sub
End Class
```


Form_Proses.vb

```
Imports System.Drawing
Imports System.Drawing.Imaging
Imports System.Drawing.Drawing2D

Public Class Form_Proses
    Dim arrPixel_label(300, 200) As Integer
    Dim arrHasil_ccl(1000) As Bitmap

    Private Sub Button_BukaCitra_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button_BukaCitra.Click
        Dim bukacitra As New OpenFileDialog
        bukacitra.Filter = "|*.*|Bitmap Files (*)|*.bmp"
        bukacitra.ShowDialog()
        TextBox_FileName.Text = bukacitra.FileName
```

```
        TextBox_FileName.SelectionStart =
TextBox_FileName.Text.Length

        If Trim(TextBox_FileName.Text) <> "" Then
            PictureBox1.Image =
Image.FromFile(TextBox_FileName.Text)
            If (PictureBox1.Image.Width > 640) Or
(PictureBox1.Image.Height > 480) Then
                MessageBox.Show("Ukuran Citra KTM terlalu besar.
Pilih Citra KTM lagi..")
                PictureBox1.Image = Nothing
                TextBox_FileName.Text = ""
            Else
                PictureBox1.SizeMode =
PictureBoxSizeMode.StretchImage
                Button_Resize.Enabled = True
            End If
        End If

    End Sub

    Private Sub Submenu_BukaCitra_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Submenu_BukaCitra.Click
        Dim bukacitra As New OpenFileDialog
        bukacitra.Filter = "|*.*|Bitmap Files (*)|*.bmp"
        bukacitra.ShowDialog()
        TextBox_FileName.Text = bukacitra.FileName
        TextBox_FileName.SelectionStart =
TextBox_FileName.Text.Length

        If Trim(TextBox_FileName.Text) <> "" Then
            PictureBox1.Image =
Image.FromFile(TextBox_FileName.Text)
            If (PictureBox1.Image.Width > 640) Or
(PictureBox1.Image.Height > 480) Then
                MessageBox.Show("Ukuran Citra KTM terlalu besar.
Pilih Citra KTM lagi..")
                PictureBox1.Image = Nothing
                TextBox_FileName.Text = ""
            Else
                PictureBox1.SizeMode =
PictureBoxSizeMode.StretchImage
                Button_Resize.Enabled = True
            End If
        End If
    End Sub

    Private Sub Button_Resize_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button_Resize.Click
        Dim citraresize As Bitmap
        citraresize = New Bitmap(PictureBox1.Image, 240, 160)

        PictureBox2.Image = citraresize
        PictureBox2.SizeMode = PictureBoxSizeMode.StretchImage
        Button_MedianFilter.Enabled = True
```

```
    End Sub

    Private Sub Submenu_Resize_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Submenu_Resize.Click
        Dim citraresize As Bitmap
        citraresize = New Bitmap(PictureBox1.Image, 240, 160)

        PictureBox2.Image = citraresize
        PictureBox2.SizeMode = PictureBoxSizeMode.StretchImage
        Button_MedianFilter.Enabled = True
    End Sub

    Private Sub Button_MedianFilter_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button_MedianFilter.Click
        Dim citrafiltering As New Bitmap(PictureBox2.Image)
        Dim arrTempR(3) As Integer
        Dim arrTempG(3) As Integer
        Dim arrTempB(3) As Integer
        Dim x, y, i, j, count, Temp As Integer

        For y = 0 To citrafiltering.Height - 1
            For x = 0 To citrafiltering.Width - 1
                count = 0
                For i = 0 To 2
                    For j = 0 To 2

                        If ((i < 0) Or (i >= citrafiltering.Width)
Or (j < 0) Or (j >= citrafiltering.Height)) Then
                            arrTempR(count) = 0
                            arrTempG(count) = 0
                            arrTempB(count) = 0
                        Else
                            arrTempR(count) =
citrafiltering.GetPixel(i, j).R
                            arrTempG(count) =
citrafiltering.GetPixel(i, j).G
                            arrTempB(count) =
citrafiltering.GetPixel(i, j).B
                        End If

                    Next j
                Next i
            Next x
        Next y

        For i = 0 To count - 1
            For j = 0 To count - 1
                If (arrTempR(i) > arrTempR(j)) Then
                    Temp = arrTempR(i)
                    arrTempR(i) = arrTempR(j)
                    arrTempR(j) = Temp
                End If

                If (arrTempG(i) > arrTempG(j)) Then
```

```
                    Temp = arrTempR(i)
                    arrTempG(i) = arrTempG(j)
                    arrTempG(j) = Temp
                End If

                If (arrTempB(i) > arrTempB(j)) Then
                    Temp = arrTempB(i)
                    arrTempB(i) = arrTempB(j)
                    arrTempB(j) = Temp
                End If

                citrafiltering.SetPixel(x, y,
Color.FromArgb((arrTempR(count / 2)), (arrTempG(count / 2)),
(arrTempB(count / 2))))
            Next j
        Next i

        PictureBox2.Image = citrafiltering
        PictureBox2.SizeMode = PictureBoxSizeMode.StretchImage
        Button_Grayscale.Enabled = True

    End Sub

    Private Sub Submenu_Filtering_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Submenu_Filtering.Click
        Dim citrafiltering As New Bitmap(PictureBox2.Image)
        Dim arrTempR(3) As Integer
        Dim arrTempG(3) As Integer
        Dim arrTempB(3) As Integer
        Dim x, y, i, j, count, Temp As Integer

        For y = 0 To citrafiltering.Height - 1
            For x = 0 To citrafiltering.Width - 1
                count = 0
                For i = 0 To 2
                    For j = 0 To 2

                        If ((i < 0) Or (i >= citrafiltering.Width)
Or (j < 0) Or (j >= citrafiltering.Height)) Then
                            arrTempR(count) = 0
                            arrTempG(count) = 0
                            arrTempB(count) = 0
                        Else
                            arrTempR(count) =
citrafiltering.GetPixel(i, j).R
                            arrTempG(count) =
citrafiltering.GetPixel(i, j).G
                            arrTempB(count) =
citrafiltering.GetPixel(i, j).B
                        End If

                    Next j
                Next i
            Next x
        Next y
```

```
        For i = 0 To count - 1
            For j = 0 To count - 1
                If (arrTempR(i) > arrTempR(j)) Then
                    Temp = arrTempR(i)
                    arrTempR(i) = arrTempR(j)
                    arrTempR(j) = Temp
                End If

                If (arrTempG(i) > arrTempG(j)) Then
                    Temp = arrTempR(i)
                    arrTempG(i) = arrTempG(j)
                    arrTempG(j) = Temp
                End If

                If (arrTempB(i) > arrTempB(j)) Then
                    Temp = arrTempB(i)
                    arrTempB(i) = arrTempB(j)
                    arrTempB(j) = Temp
                End If

                citrafiltering.SetPixel(x, y,
Color.FromArgb((arrTempR(count / 2)), (arrTempG(count / 2)),
(arrTempB(count / 2))))
            Next j
        Next i

        PictureBox2.Image = citrafiltering
        PictureBox2.SizeMode = PictureBoxSizeMode.StretchImage
        Button_Grayscale.Enabled = True
    End Sub

    Private Sub Button_Grayscale_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button_Grayscale.Click
        Dim x, y As Integer
        Dim warna As Long
        Dim citragrayscale As New Bitmap(PictureBox2.Image)
        x = 0
        y = 0
        For y = 0 To citragrayscale.Height - 1
            For x = 0 To citragrayscale.Width - 1
                warna = citragrayscale.GetPixel(x, y).R
                warna += citragrayscale.GetPixel(x, y).G
                warna += citragrayscale.GetPixel(x, y).B
                warna = Math.Round(warna / 3)
                citragrayscale.SetPixel(x, y,
Color.FromArgb(warna, warna, warna))
            Next x
        Next y

        PictureBox2.Image = citragrayscale
        PictureBox2.SizeMode = PictureBoxSizeMode.StretchImage
        Button_Biner.Enabled = True
    End Sub
```

```
    Private Sub Submenu_Grayscale_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Submenu_Grayscale.Click
        Dim x, y As Integer
        Dim warna As Long
        Dim citragrayscale As New Bitmap(PictureBox2.Image)
        x = 0
        y = 0
        For y = 0 To citragrayscale.Height - 1
            For x = 0 To citragrayscale.Width - 1
                warna = citragrayscale.GetPixel(x, y).R
                warna += citragrayscale.GetPixel(x, y).G
                warna += citragrayscale.GetPixel(x, y).B
                warna = Math.Round(warna / 3)
                citragrayscale.SetPixel(x, y,
Color.FromArgb(warna, warna, warna))
            Next x
        Next y

        PictureBox2.Image = citragrayscale
        PictureBox2.SizeMode = PictureBoxSizeMode.StretchImage
        Button_Biner.Enabled = True
    End Sub

    Private Sub Button_Biner_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button_Biner.Click
        Dim x, y As Integer
        Dim warna As Long
        Dim citrabiner As New Bitmap(PictureBox2.Image)

        x = 0
        y = 0
        For y = 0 To citrabiner.Height - 1
            For x = 0 To citrabiner.Width - 1
                warna = citrabiner.GetPixel(x, y).R
                warna += citrabiner.GetPixel(x, y).G
                warna += citrabiner.GetPixel(x, y).B
                warna = Math.Round(warna / 3)
                citrabiner.SetPixel(x, y, Color.FromArgb(warna,
warna, warna))

                If (TextBox_Threshold.Text = vbNullString) Then
                    MessageBox.Show("Masukkan dulu nilai
threshold", "WARNING!!!", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
                    Return
                Else
                    If (IsNumeric(TextBox_Threshold.Text) = False)
Then
                        MessageBox.Show("Nilai threshold yang Anda
ketikkan bukan berupa angka", "WARNING!!!", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
                        TextBox_Threshold.Text = ""
                        TextBox_Threshold.Focus()
                    ElseIf ((Val(TextBox_Threshold.Text) >= 0) And
(Val(TextBox_Threshold.Text) <= 255)) Then
```

```
                        If warna < Val(TextBox_Threshold.Text)
Then
                            citrabiner.SetPixel(x, y, Color.Black)
                            arrPixel_label(x, y) = 1
                        Else
                            citrabiner.SetPixel(x, y, Color.White)
                            arrPixel_label(x, y) = 0
                        End If
                    Else
                        TextBox_Threshold.Text = ""
                        MessageBox.Show("Nilai threshold yang Anda
ketikkan harus di antara 0 - 255", "WARNING!!!",
MessageBoxButtons.OK, MessageBoxIcon.Warning)
                        TextBox_Threshold.Focus()
                    End If
                End If

            Next x
        Next y

        PictureBox2.Image = citrabiner
        PictureBox2.SizeMode = PictureBoxSizeMode.StretchImage
        Button_Crop.Enabled = True
    End Sub

    Private Sub Submenu_Biner_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Submenu_Biner.Click
        Dim x, y As Integer
        Dim warna As Long
        Dim citrabiner As New Bitmap(PictureBox2.Image)

        x = 0
        y = 0
        For y = 0 To citrabiner.Height - 1
            For x = 0 To citrabiner.Width - 1
                warna = citrabiner.GetPixel(x, y).R
                warna += citrabiner.GetPixel(x, y).G
                warna += citrabiner.GetPixel(x, y).B
                warna = Math.Round(warna / 3)
                citrabiner.SetPixel(x, y, Color.FromArgb(warna,
warna, warna))

                If (TextBox_Threshold.Text = vbNullString) Then
                    MessageBox.Show("Masukkan dulu nilai
threshold", "WARNING!!!", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
                    Return
                Else
                    If (IsNumeric(TextBox_Threshold.Text) = False)
Then
                        MessageBox.Show("Nilai threshold yang Anda
ketikkan bukan berupa angka", "WARNING!!!", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
                        TextBox_Threshold.Text = ""
                        TextBox_Threshold.Focus()
```

```
                    ElseIf ((Val(TextBox_Threshold.Text) >= 0) And
(Val(TextBox_Threshold.Text) <= 255)) Then
                        If warna < Val(TextBox_Threshold.Text)
Then
                            citrabiner.SetPixel(x, y, Color.Black)
                            arrPixel_label(x, y) = 1
                        Else
                            citrabiner.SetPixel(x, y, Color.White)
                            arrPixel_label(x, y) = 0
                        End If
                    Else
                        TextBox_Threshold.Text = ""
                        MessageBox.Show("Nilai threshold yang Anda
ketikkan harus di antara 0 - 255", "WARNING!!!",
MessageBoxButtons.OK, MessageBoxIcon.Warning)
                        TextBox_Threshold.Focus()
                    End If
                End If

            Next x
        Next y

        PictureBox2.Image = citrabiner
        PictureBox2.SizeMode = PictureBoxSizeMode.StretchImage
        Button_Crop.Enabled = True
    End Sub

    Private Sub Button_Crop_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button_Crop.Click
        Dim citranim As New Bitmap(PictureBox2.Image)
        Dim hasilakhir As New Bitmap(PictureBox2.Image, 100, 30)
        Dim x, y As Integer
        Dim jml_pikselhitam As Integer
        Dim x1crop As Integer = -1
        Dim y1crop As Integer = -1
        Dim x2crop As Integer = -1
        Dim y2crop As Integer = -1
        Dim xset As Integer
        Dim yset As Integer

        'cari j1crop (y1) dan j2crop (y2)
        For y = 58 To citranim.Height - 85 'y = 58 sampai y = 75
            jml_pikselhitam = 0
            For x = 9 To citranim.Width - 168 'x = 9 sampai x = 72
                If ((citranim.GetPixel(x, y).R = 0) And
(citranim.GetPixel(x, y).G = 0) And (citranim.GetPixel(x, y).B =
0)) Then
                    jml_pikselhitam += 1
                End If
            Next x

            If (jml_pikselhitam > 25 And jml_pikselhitam < 52)
Then
                'MessageBox.Show("Koordinat y " + y.ToString + "
ada polanya = " + jml_pikselhitam.ToString)
                If y1crop = -1 Then
```

```
                    y1crop = y - 1
                Else
                    y2crop = y + 1
                End If
            End If
        Next y

        'cari i1crop (x1) dan i2crop (x2)
        For x = 9 To citranim.Width - 168 'x = 9 sampai x = 72
            jml_pikselhitam = 0
            For y = 58 To citranim.Height - 85 'y = 58 sampai y =
75
                If ((citranim.GetPixel(x, y).R = 0) And
(citranim.GetPixel(x, y).G = 0) And (citranim.GetPixel(x, y).B =
0)) Then
                    jml_pikselhitam += 1
                End If
            Next y

            If (jml_pikselhitam > 1 And jml_pikselhitam < 9) Then
                'MessageBox.Show("Koordinat x " + x.ToString + "
ada polanya = " + jml_pikselhitam.ToString)
                If x1crop = -1 Then
                    x1crop = x - 1
                Else
                    x2crop = x + 1
                End If
            End If
        Next x

        citranim = New Bitmap(Math.Abs(x2crop - x1crop) + 1,
(Math.Abs(y2crop - y1crop) + 1))
        'copy citra pada PictureBox3 ke citranim
        yset = -1
        hasilakhir = New Bitmap(PictureBox2.Image)

        For y = y1crop To y2crop
            xset = -1
            yset += 1
            For x = x1crop To x2crop
                xset += 1
                citranim.SetPixel(xset, yset,
hasilakhir.GetPixel(x, y))
            Next x
        Next y

        PictureBox3.Image = citranim
        Button_CCL.Enabled = True

    End Sub

    Private Sub Submenu_Crop_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Submenu_Crop.Click
        Dim citranim As New Bitmap(PictureBox2.Image)
        Dim hasilakhir As New Bitmap(PictureBox2.Image, 100, 30)
        Dim x, y As Integer
```

```
        Dim jml_pikselhitam As Integer
        Dim x1crop As Integer = -1
        Dim y1crop As Integer = -1
        Dim x2crop As Integer = -1
        Dim y2crop As Integer = -1
        Dim xset As Integer
        Dim yset As Integer

        'cari j1crop (y1) dan j2crop (y2)
        For y = 58 To citranim.Height - 85
            jml_pikselhitam = 0
            For x = 9 To citranim.Width - 168
                If ((citranim.GetPixel(x, y).R = 0) And
(citranim.GetPixel(x, y).G = 0) And (citranim.GetPixel(x, y).B =
0)) Then
                    jml_pikselhitam += 1
                End If
            Next x

            If (jml_pikselhitam > 25 And jml_pikselhitam < 52)
Then
                'MessageBox.Show("Koordinat y " + y.ToString + "
ada polanya = " + jml_pikselhitam.ToString)
                If y1crop = -1 Then
                    y1crop = y - 1
                Else
                    y2crop = y + 1
                End If
            End If
        Next y

        'cari i1crop (x1) dan i2crop (x2)
        For x = 9 To citranim.Width - 168
            jml_pikselhitam = 0
            For y = 58 To citranim.Height - 85
                If ((citranim.GetPixel(x, y).R = 0) And
(citranim.GetPixel(x, y).G = 0) And (citranim.GetPixel(x, y).B =
0)) Then
                    jml_pikselhitam += 1
                End If
            Next y

            If (jml_pikselhitam > 1 And jml_pikselhitam < 9) Then
                'MessageBox.Show("Koordinat x " + x.ToString + "
ada polanya = " + jml_pikselhitam.ToString)
                If x1crop = -1 Then
                    x1crop = x - 1
                Else
                    x2crop = x + 1
                End If
            End If
        Next x

        citranim = New Bitmap(Math.Abs(x2crop - x1crop) + 1,
(Math.Abs(y2crop - y1crop) + 1))
        'copy citra pada PictureBox3 ke citranim
```

```
        yset = -1
        hasilakhir = New Bitmap(PictureBox2.Image)

        For y = y1crop To y2crop
            xset = -1
            yset += 1
            For x = x1crop To x2crop
                xset += 1
                citranim.SetPixel(xset, yset,
hasilakhir.GetPixel(x, y))
            Next x
        Next y

        PictureBox3.Image = citranim
        Button_CCL.Enabled = True
    End Sub

    Private Sub Button_CCL_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button_CCL.Click
        Dim tetangga_kiri As Integer
        Dim tetangga_atas As Integer
        Dim nomorlabel As Integer
        Dim x, y As Integer
        Dim min As Integer
        Dim arrPixel_label(300, 200) As Integer
        Dim arrTabel_label(1000) As Integer
        Dim berubah As Boolean
        Dim selesai As Boolean
        Dim hasilpelabelan As New Bitmap(PictureBox3.Image)
        ProgressBar1.Maximum = 100
        ProgressBar1.Minimum = 0

        'Program Connected Component Labeling menggunakan
algoritma yang dikembangkan oleh Stefano dan Bulgarelli (1999)
        'Source code dikutip dari Hartono, H. C. (2013). Konversi
Citra Papan Penunjuk Jalan menjadi Karakter ASCII dengan Optical
Character Recogniton. (Undergraduate thesis, Duta Wacana Christian
University, 2013). Retrieved from http://sinta.ukdw.ac.id dengan
perubahan dan penyesuaian
        'Proses Pelabelan dengan Connected Component Labeling:
        'PROSES PELABELAN - TAHAP PERTAMA
        'Scan dari atas ke bawah, mulai dari kiri ke kanan (dari y
ke x)
        'Jika ketemu angka 1 (foreground), cek tetangganya atas
dan kiri
        'Ada 3 kondisi pada proses pelabelan tahap satu:
        '1. Jika tetangga atas dan tetangga kiri adalah
background, maka inc nilai label
        '2. Jika bertemu dengan tetangganya hanya atas atau kiri,
maka beri dengan nilai tetangganya
        '3. Jika bertemu dengan tetangga atas dan kiri, maka beri
nilai tetangganya yang lebih kecil


        ProgressBar1.Value = 0
        For y = 1 To 1000    'inisialisasi nilai maksimum label
```

```
            arrTabel_label(y) = 1000  'C[y] = y
        Next y

        nomorlabel = 0
        For y = 0 To hasilpelabelan.Height - 1
            For x = 0 To hasilpelabelan.Width - 1
                If (hasilpelabelan.GetPixel(x, y).R = 0) And
(hasilpelabelan.GetPixel(x, y).G = 0) And
(hasilpelabelan.GetPixel(x, y).B = 0) Then
                    tetangga_kiri = 0
                    If x - 1 >= 0 Then
                        If ((hasilpelabelan.GetPixel(x - 1, y).R =
0) And (hasilpelabelan.GetPixel(x - 1, y).G = 0) And
(hasilpelabelan.GetPixel(x - 1, y).B = 0)) Then
                            tetangga_kiri = arrPixel_label(x - 1,
y)
                        End If
                    End If

                    tetangga_atas = 0
                    If y - 1 >= 0 Then
                        If ((hasilpelabelan.GetPixel(x, y - 1).R =
0) And (hasilpelabelan.GetPixel(x, y - 1).G = 0) And
(hasilpelabelan.GetPixel(x, y - 1).B = 0)) Then
                            tetangga_atas = arrPixel_label(x, y -
1)
                        End If
                    End If
                    ProgressBar1.Value = 5

                    If ((tetangga_kiri = 0) And (tetangga_atas =
0)) Then
                        'kondisi 1
                        nomorlabel += 1
                        arrPixel_label(x, y) = nomorlabel

                    ElseIf ((tetangga_kiri = 0) And (tetangga_atas
<> 0)) Or ((tetangga_kiri <> 0) And (tetangga_atas = 0)) Then
                        'kondisi 2
                        If (tetangga_kiri <> 0) Then
                            If (tetangga_kiri <
arrTabel_label(tetangga_kiri)) Then
                                arrPixel_label(x, y) =
tetangga_kiri
                                arrPixel_label(x - 1, y) =
tetangga_kiri
                                arrTabel_label(tetangga_kiri) =
tetangga_kiri
                            Else
                                arrPixel_label(x, y) =
arrTabel_label(tetangga_kiri)
                                arrPixel_label(x - 1, y) =
arrTabel_label(tetangga_kiri)
                            End If

                        Else
```

```
                            If (tetangga_atas <
arrTabel_label(tetangga_atas)) Then
                                arrPixel_label(x, y) =
tetangga_atas
                                arrPixel_label(x, y - 1) =
tetangga_atas
                                arrTabel_label(tetangga_atas) =
tetangga_atas
                            Else
                                arrPixel_label(x, y) =
arrTabel_label(tetangga_atas)
                                arrPixel_label(x, y - 1) =
arrTabel_label(tetangga_atas)
                            End If
                            ProgressBar1.Value = 10
                        End If
                    ElseIf ((tetangga_kiri <> 0) And
(tetangga_atas <> 0)) Then     'Kondisi 3
                            If ((tetangga_kiri < tetangga_atas))
Then
                                arrPixel_label(x, y) =
tetangga_kiri
                                arrPixel_label(x - 1, y) =
tetangga_kiri
                                arrPixel_label(x, y - 1) =
tetangga_kiri
                                arrTabel_label(tetangga_atas) =
tetangga_kiri
                            Else
                                arrPixel_label(x, y) =
tetangga_atas
                                arrPixel_label(x - 1, y) =
tetangga_atas
                                arrPixel_label(x, y - 1) =
tetangga_atas
                                arrTabel_label(tetangga_kiri) =
tetangga_atas
                            End If
                        End If
                End If
            Next
        Next
        ProgressBar1.Value = 15

        'Proses ini digunakan untuk mengubah nilai konversi label
yang masih belum optimal menjadi optimal
        'Caranya dengan mencari nilai min masing-masing
konversinya
        min = 1999    'min = C[x],C[y]
        For x = 1 To nomorlabel
            For y = 1 To nomorlabel
                If ((arrTabel_label(x) < arrTabel_label(y)) And (y
= arrTabel_label(x))) Then
                    min = arrTabel_label(x)
                End If
            Next y
```

```
            If (min < 1999) And (min <> 0) Then
                arrTabel_label(y) = min
            End If
        Next x

        ProgressBar1.Value = 20
        'Lihat Hasil Pelabelan Tahap 1

        For y = 0 To hasilpelabelan.Height - 1
            For x = 0 To hasilpelabelan.Width - 1
                RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + arrPixel_label(x, y).ToString() +
Chr(32) + ""
                ProgressBar1.Value = 25
            Next x
            RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + Chr(13)
            ProgressBar1.Value = 30
        Next y

        MessageBox.Show("Proses Pelabelan Tahap 1 Selesai",
"INFORMASI", MessageBoxButtons.OK, MessageBoxIcon.Information)
        'Pelabelan Tahap 1 selesai

        'PROSES PELABELAN - TAHAP KEDUA:
        selesai = False
        While selesai = False
            berubah = False
            'mengubah label sesuai dengan data konversi labelnya
            For y = 0 To hasilpelabelan.Height - 1
                For x = 0 To hasilpelabelan.Width - 1
                    If ((arrTabel_label(arrPixel_label(x, y)) <
arrPixel_label(x, y)) And (arrTabel_label(arrPixel_label(x, y)) <>
-1)) Then
                        arrPixel_label(x, y) =
arrTabel_label(arrPixel_label(x, y))
                        berubah = True
                    End If
                Next x
            Next y

            If berubah = False Then
                selesai = True
            End If
        End While
        ProgressBar1.Value = 35

        'Cari nomor label sekarang setelah dilakukan normalisasi
        nomorlabel = 0
        For y = 0 To hasilpelabelan.Height - 1
            For x = 0 To hasilpelabelan.Width - 1
                If (arrPixel_label(x, y) > nomorlabel) Then
                    nomorlabel = arrPixel_label(x, y)
                End If
            Next x
        Next y
```

```
        ProgressBar1.Value = 40
        'Pelabelan Tahap 2 selesai

        'Lihat Hasil Akhir Proses Pelabelan
        RichTextBoxHasilPelabelan.Clear()
        RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + "Hasil Akhir Pelabelan = " +
Chr(13)
        For y = 0 To hasilpelabelan.Height - 1
            For x = 0 To hasilpelabelan.Width - 1
                RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + arrPixel_label(x, y).ToString() +
Chr(32) + ""
                ProgressBar1.Value = 45
            Next x
            RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + Chr(13)
            ProgressBar1.Value = 50
        Next y

        PictureBox3.Image = hasilpelabelan
        MessageBox.Show("Proses Pelabelan Tahap 2 Selesai",
"INFORMASI", MessageBoxButtons.OK, MessageBoxIcon.Information)


        'TAHAP EKSTRAK KARAKTER
        Dim iterasi As Integer
        Dim x1 As Integer
        Dim y1 As Integer
        Dim x2 As Integer
        Dim y2 As Integer
        Dim arrLabel_data_x1(100) As Integer
        Dim arrLabel_data_x2(100) As Integer
        Dim arrLabel_data_y1(100) As Integer
        Dim arrLabel_data_y2(100) As Integer

        For iterasi = 1 To nomorlabel
            x1 = 1000
            y1 = 1000
            x2 = -1
            y2 = -1

            For y = 0 To hasilpelabelan.Height - 1
                For x = 0 To hasilpelabelan.Width - 1
                    If arrPixel_label(x, y) = iterasi Then
                        If x1 > x Then
                            x1 = x
                        End If
                        If y1 > y Then
                            y1 = y
                        End If
                        If x2 < x Then
                            x2 = x
                        End If
                        If y2 < y Then
                            y2 = y
```

```
                    End If
                End If   'end if untuk iterasi
            Next x
        Next y

        ProgressBar1.Value = 60

        'catat nomor label sekarang untuk x1, x2, y1, dan y2
        If x2 = -1 Then
            arrLabel_data_x1(iterasi) = 0
            arrLabel_data_y1(iterasi) = 0
            arrLabel_data_x2(iterasi) = 0
            arrLabel_data_y2(iterasi) = 0
        Else
            arrLabel_data_x1(iterasi) = x1
            arrLabel_data_y1(iterasi) = y1

            arrLabel_data_x2(iterasi) = x2
            If (arrLabel_data_x2(iterasi) >
hasilpelabelan.Width - 1) Then
                arrLabel_data_x2(iterasi) =
hasilpelabelan.Width - 1
            End If

            arrLabel_data_y2(iterasi) = y2
            If (arrLabel_data_y2(iterasi) >
hasilpelabelan.Height - 1) Then
                arrLabel_data_y2(iterasi) =
hasilpelabelan.Height - 1
            End If
        End If
        'MessageBox.Show("x1 : " +
arrLabel_data_x1(iterasi).ToString + Chr(9) + "y1 : " +
arrLabel_data_y1(iterasi).ToString + Chr(9) + "x2 : " +
arrLabel_data_x2(iterasi).ToString + Chr(9) + "y2 : " +
arrLabel_data_y2(iterasi).ToString + " ")
    Next iterasi

    RichTextBoxHasilPelabelan.Clear()
    RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + "Hasil Akhir Pelabelan = " +
Chr(13)
    For y = 0 To hasilpelabelan.Height - 1
        For x = 0 To hasilpelabelan.Width - 1
            RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + arrPixel_label(x, y).ToString() +
Chr(32) + ""
            ProgressBar1.Value = 70
        Next x
        RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + Chr(13)
        ProgressBar1.Value = 80
    Next y
    MessageBox.Show("Tahap Ekstrak Karakter Selesai",
"INFORMASI", MessageBoxButtons.OK, MessageBoxIcon.Information)
```

```
        'TAHAP SEGMENTASI
        Dim lebar As Integer
        Dim tinggi As Integer
        Dim counter As Integer
        ComboBoxPelabelan.Items.Clear()
        counter = 0

        For iterasi = 1 To nomorlabel
            If ((arrLabel_data_x1(iterasi) >= 0) And
(arrLabel_data_y1(iterasi) >= 0) And (arrLabel_data_x2(iterasi) >
0) And (arrLabel_data_y2(iterasi) > 0)) Then
                counter += 1

                lebar =
Math.Round(Math.Abs(arrLabel_data_x1(iterasi) -
arrLabel_data_x2(iterasi))) + 1
                tinggi =
Math.Round(Math.Abs(arrLabel_data_y1(iterasi) -
arrLabel_data_y2(iterasi))) + 1

                If lebar <= 1 Then
                    lebar = 2
                End If

                If tinggi <= 1 Then
                    tinggi = 2
                End If

                arrHasil_ccl(iterasi) = New Bitmap(lebar, tinggi)


                ProgressBar1.Value = 90

                x1 = -1
                For x = arrLabel_data_x1(iterasi) To
arrLabel_data_x2(iterasi)
                    x1 = x1 + 1
                    y1 = -1
                    For y = arrLabel_data_y1(iterasi) To
arrLabel_data_y2(iterasi)
                        y1 = y1 + 1
                        arrHasil_ccl(iterasi).SetPixel(x1, y1,
hasilpelabelan.GetPixel(x, y))
                    Next y
                Next x

                arrHasil_ccl(iterasi) = New
Bitmap(arrHasil_ccl(iterasi.ToString()), 30, 50)
                PictureBox4.Image = hasilpelabelan
                ComboBoxPelabelan.Items.Add(iterasi.ToString())
            End If
        Next iterasi

        ProgressBar1.Value = 100
        MessageBox.Show("Segmentasi NIM Selesai", "INFORMASI",
MessageBoxButtons.OK, MessageBoxIcon.Information)
```

```
        Button_SimpanHasilSegmentasi.Enabled = True
        ProgressBar1.Value = 0
    End Sub

    Private Sub Submenu_CCL_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Submenu_CCL.Click
        Dim tetangga_kiri As Integer
        Dim tetangga_atas As Integer
        Dim nomorlabel As Integer
        Dim x, y As Integer
        Dim min As Integer
        Dim arrPixel_label(300, 200) As Integer
        Dim arrTabel_label(1000) As Integer
        Dim berubah As Boolean
        Dim selesai As Boolean
        Dim hasilpelabelan As New Bitmap(PictureBox3.Image)
        ProgressBar1.Maximum = 100
        ProgressBar1.Minimum = 0

        'Program Connected Component Labeling menggunakan
algoritma yang dikembangkan oleh Stefano dan Bulgarelli (1999)
        'Source code dikutip dari Hartono, H. C. (2013). Konversi
Citra Papan Penunjuk Jalan menjadi Karakter ASCII dengan Optical
Character Recogniton. (Undergraduate thesis, Duta Wacana Christian
University, 2013). Retrieved from http://sinta.ukdw.ac.id dengan
perubahan dan penyesuaian
        'Proses Pelabelan dengan Connected Component Labeling:
        'TAHAP PERTAMA:
        'Scan dari atas ke bawah, mulai dari kiri ke kanan (dari y
ke x)
        'Jika ketemu angka 1 (foreground), cek tetangganya atas
dan kiri
        'Ada 3 kondisi pada proses pelabelan tahap satu:
        '1. Jika tetangga atas dan tetangga kiri adalah
background, maka inc nilai label
        '2. Jika bertemu dengan tetangganya hanya atas atau kiri,
maka beri dengan nilai tetangganya
        '3. Jika bertemu dengan tetangga atas dan kiri, maka beri
nilai tetangganya yang lebih kecil


        ProgressBar1.Value = 0
        For y = 1 To 1000    'inisialisasi nilai maksimum label
            arrTabel_label(y) = 1000  'C[y] = y
        Next y

        nomorlabel = 0
        For y = 0 To hasilpelabelan.Height - 1
            For x = 0 To hasilpelabelan.Width - 1
                If (hasilpelabelan.GetPixel(x, y).R = 0) And
(hasilpelabelan.GetPixel(x, y).G = 0) And
(hasilpelabelan.GetPixel(x, y).B = 0) Then
                    tetangga_kiri = 0
                    If x - 1 >= 0 Then
```

```
                        If ((hasilpelabelan.GetPixel(x - 1, y).R =
0) And (hasilpelabelan.GetPixel(x - 1, y).G = 0) And
(hasilpelabelan.GetPixel(x - 1, y).B = 0)) Then
                            tetangga_kiri = arrPixel_label(x - 1,
y)
                        End If
                    End If

                    tetangga_atas = 0
                    If y - 1 >= 0 Then
                        If ((hasilpelabelan.GetPixel(x, y - 1).R =
0) And (hasilpelabelan.GetPixel(x, y - 1).G = 0) And
(hasilpelabelan.GetPixel(x, y - 1).B = 0)) Then
                            tetangga_atas = arrPixel_label(x, y -
1)
                        End If
                    End If
                    ProgressBar1.Value = 5

                    If ((tetangga_kiri = 0) And (tetangga_atas =
0)) Then
                        'Kondisi 1
                        nomorlabel += 1
                        arrPixel_label(x, y) = nomorlabel

                    ElseIf ((tetangga_kiri = 0) And (tetangga_atas
<> 0)) Or ((tetangga_kiri <> 0) And (tetangga_atas = 0)) Then
                        'Kondisi 2
                        If (tetangga_kiri <> 0) Then
                            If (tetangga_kiri <
arrTabel_label(tetangga_kiri)) Then
                                arrPixel_label(x, y) =
tetangga_kiri
                                arrPixel_label(x - 1, y) =
tetangga_kiri
                                arrTabel_label(tetangga_kiri) =
tetangga_kiri
                            Else
                                arrPixel_label(x, y) =
arrTabel_label(tetangga_kiri)
                                arrPixel_label(x - 1, y) =
arrTabel_label(tetangga_kiri)
                            End If

                        Else
                            If (tetangga_atas <
arrTabel_label(tetangga_atas)) Then
                                arrPixel_label(x, y) =
tetangga_atas
                                arrPixel_label(x, y - 1) =
tetangga_atas
                                arrTabel_label(tetangga_atas) =
tetangga_atas
                            Else
                                arrPixel_label(x, y) =
arrTabel_label(tetangga_atas)
```

```
                                arrPixel_label(x, y - 1) =
arrTabel_label(tetangga_atas)
                            End If
                        End If
                        ProgressBar1.Value = 10

                    ElseIf ((tetangga_kiri <> 0) And
(tetangga_atas <> 0)) Then     'Kondisi 3
                            If ((tetangga_kiri < tetangga_atas))
Then
                                arrPixel_label(x, y) =
tetangga_kiri
                                arrPixel_label(x - 1, y) =
tetangga_kiri
                                arrPixel_label(x, y - 1) =
tetangga_kiri
                                arrTabel_label(tetangga_atas) =
tetangga_kiri
                            Else
                                arrPixel_label(x, y) =
tetangga_atas
                                arrPixel_label(x - 1, y) =
tetangga_atas
                                arrPixel_label(x, y - 1) =
tetangga_atas
                                arrTabel_label(tetangga_kiri) =
tetangga_atas
                            End If
                        End If
                End If
            Next
        Next
        ProgressBar1.Value = 15

        'Proses ini digunakan untuk mengubah nilai konversi label
yang masih belum optimal menjadi optimal
        'Caranya dengan mencari nilai min masing-masing
konversinya
        min = 1999    'min = C[x],C[y]
        For x = 1 To nomorlabel
            For y = 1 To nomorlabel
                If ((arrTabel_label(x) < arrTabel_label(y)) And (y
= arrTabel_label(x))) Then
                    min = arrTabel_label(x)
                End If
            Next y
            If (min < 1999) And (min <> 0) Then
                arrTabel_label(y) = min
            End If
        Next x

        ProgressBar1.Value = 20
        'Lihat Hasil Pelabelan Tahap 1

        For y = 0 To hasilpelabelan.Height - 1
            For x = 0 To hasilpelabelan.Width - 1
```

```
                RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + arrPixel_label(x, y).ToString() +
Chr(32) + ""
                ProgressBar1.Value = 25
            Next x
            RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + Chr(13)
            ProgressBar1.Value = 30
        Next y

        MessageBox.Show("Proses Pelabelan Tahap 1 Selesai",
"INFORMASI", MessageBoxButtons.OK, MessageBoxIcon.Information)
        'Pelabelan Tahap 1 selesai

        'PROSES PELABELAN - TAHAP KEDUA:
        selesai = False
        While selesai = False
            berubah = False
            'mengubah label sesuai dengan data konversi labelnya
            For y = 0 To hasilpelabelan.Height - 1
                For x = 0 To hasilpelabelan.Width - 1
                    If ((arrTabel_label(arrPixel_label(x, y)) <
arrPixel_label(x, y)) And (arrTabel_label(arrPixel_label(x, y)) <>
-1)) Then
                        arrPixel_label(x, y) =
arrTabel_label(arrPixel_label(x, y))
                        berubah = True
                    End If
                Next x
            Next y

            If berubah = False Then
                selesai = True
            End If
        End While
        ProgressBar1.Value = 35

        'Cari nomor label sekarang setelah dilakukan normalisasi
        nomorlabel = 0
        For y = 0 To hasilpelabelan.Height - 1
            For x = 0 To hasilpelabelan.Width - 1
                If (arrPixel_label(x, y) > nomorlabel) Then
                    nomorlabel = arrPixel_label(x, y)
                End If
            Next x
        Next y
        ProgressBar1.Value = 40
        'Pelabelan Tahap 2 selesai

        'Lihat Hasil Akhir Proses Pelabelan
        RichTextBoxHasilPelabelan.Clear()
        RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + "Hasil Akhir Pelabelan = " +
Chr(13)
        For y = 0 To hasilpelabelan.Height - 1
            For x = 0 To hasilpelabelan.Width - 1
```

```
                RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + arrPixel_label(x, y).ToString() +
Chr(32) + ""
                ProgressBar1.Value = 45
            Next x
            RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + Chr(13)
            ProgressBar1.Value = 50
        Next y

        PictureBox3.Image = hasilpelabelan
        MessageBox.Show("Proses Pelabelan Tahap 2 Selesai",
"INFORMASI", MessageBoxButtons.OK, MessageBoxIcon.Information)

        'TAHAP EKSTRAK KARAKTER
        Dim iterasi As Integer
        Dim x1 As Integer
        Dim y1 As Integer
        Dim x2 As Integer
        Dim y2 As Integer
        Dim arrLabel_data_x1(100) As Integer
        Dim arrLabel_data_x2(100) As Integer
        Dim arrLabel_data_y1(100) As Integer
        Dim arrLabel_data_y2(100) As Integer

        For iterasi = 1 To nomorlabel
            x1 = 1000
            y1 = 1000
            x2 = -1
            y2 = -1

            For y = 0 To hasilpelabelan.Height - 1
                For x = 0 To hasilpelabelan.Width - 1
                    If arrPixel_label(x, y) = iterasi Then
                        If x1 > x Then
                            x1 = x
                        End If
                        If y1 > y Then
                            y1 = y
                        End If
                        If x2 < x Then
                            x2 = x
                        End If
                        If y2 < y Then
                            y2 = y
                        End If
                    End If   'end if untuk iterasi
                Next x
            Next y

            ProgressBar1.Value = 60

            'catat nomor label sekarang untuk x1, x2, y1, dan y2
            If x2 = -1 Then
                arrLabel_data_x1(iterasi) = 0
                arrLabel_data_y1(iterasi) = 0
```

```
                arrLabel_data_x2(iterasi) = 0
                arrLabel_data_y2(iterasi) = 0
            Else
                arrLabel_data_x1(iterasi) = x1
                arrLabel_data_y1(iterasi) = y1

                arrLabel_data_x2(iterasi) = x2
                If (arrLabel_data_x2(iterasi) >
hasilpelabelan.Width - 1) Then
                    arrLabel_data_x2(iterasi) =
hasilpelabelan.Width - 1
                End If

                arrLabel_data_y2(iterasi) = y2
                If (arrLabel_data_y2(iterasi) >
hasilpelabelan.Height - 1) Then
                    arrLabel_data_y2(iterasi) =
hasilpelabelan.Height - 1
                End If
            End If
            'MessageBox.Show("x1 : " +
arrLabel_data_x1(iterasi).ToString + Chr(9) + "y1 : " +
arrLabel_data_y1(iterasi).ToString + Chr(9) + "x2 : " +
arrLabel_data_x2(iterasi).ToString + Chr(9) + "y2 : " +
arrLabel_data_y2(iterasi).ToString + " ")
        Next iterasi

        RichTextBoxHasilPelabelan.Clear()
        RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + "Hasil Akhir Pelabelan = " +
Chr(13)
        For y = 0 To hasilpelabelan.Height - 1
            For x = 0 To hasilpelabelan.Width - 1
                RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + arrPixel_label(x, y).ToString() +
Chr(32) + ""
                ProgressBar1.Value = 70
            Next x
            RichTextBoxHasilPelabelan.Text =
RichTextBoxHasilPelabelan.Text + Chr(13)
            ProgressBar1.Value = 80
        Next y
        MessageBox.Show("Tahap Ekstrak Karakter Selesai",
"INFORMASI", MessageBoxButtons.OK, MessageBoxIcon.Information)

        'TAHAP SEGMENTASI
        Dim lebar As Integer
        Dim tinggi As Integer
        'Dim counter As Integer
        ComboBoxPelabelan.Items.Clear()
        'counter = 0

        For iterasi = 1 To nomorlabel
            If ((arrLabel_data_x1(iterasi) >= 0) And
(arrLabel_data_y1(iterasi) >= 0) And (arrLabel_data_x2(iterasi) >
0) And (arrLabel_data_y2(iterasi) > 0)) Then
```

```
                'counter += 1

                lebar =
Math.Round(Math.Abs(arrLabel_data_x1(iterasi) -
arrLabel_data_x2(iterasi))) + 1
                tinggi =
Math.Round(Math.Abs(arrLabel_data_y1(iterasi) -
arrLabel_data_y2(iterasi))) + 1

                If lebar <= 1 Then
                    lebar = 2
                End If

                If tinggi <= 1 Then
                    tinggi = 2
                End If

                arrHasil_ccl(iterasi) = New Bitmap(lebar, tinggi)

                ProgressBar1.Value = 90

                x1 = -1
                For x = arrLabel_data_x1(iterasi) To
arrLabel_data_x2(iterasi)
                    x1 = x1 + 1
                    y1 = -1
                    For y = arrLabel_data_y1(iterasi) To
arrLabel_data_y2(iterasi)
                        y1 = y1 + 1

arrHasil_ccl(iterasi.ToString()).SetPixel(x1, y1,
hasilpelabelan.GetPixel(x, y))
                    Next y
                Next x

                arrHasil_ccl(iterasi) = New
Bitmap(arrHasil_ccl(iterasi.ToString()), 30, 50)
                PictureBox4.Image = hasilpelabelan
                ComboBoxPelabelan.Items.Add(iterasi.ToString())
            End If
        Next iterasi

        ProgressBar1.Value = 100
        MessageBox.Show("Segmentasi selesai", "INFORMASI",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        Button_SimpanHasilSegmentasi.Enabled = True
        ProgressBar1.Value = 0
    End Sub

    Private Sub ComboBoxPelabelan_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As System.EventArgs) Handles
ComboBoxPelabelan.SelectedIndexChanged
        PictureBox5.Image = arrHasil_ccl(ComboBoxPelabelan.Text)
        PictureBox5.SizeMode = PictureBoxSizeMode.StretchImage
    End Sub
```

```
    Private Sub Button_SimpanHasilSegmentasi_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button_SimpanHasilSegmentasi.Click
        If (PictureBox5.Image Is Nothing) Then
            MessageBox.Show("Pilih Komponen Hasil CCL yang akan
Disimpan Terlebih Dahulu", "WARNING!!!", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
        ElseIf (String.IsNullOrEmpty(TextBox_SimpanAngka.Text))
Then
            MessageBox.Show("Masukkan File Name untuk Karakter NIM
yang akan Disimpan", "WARNING!!!", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
            TextBox_SimpanAngka.Text = ""
            TextBox_SimpanAngka.Focus()
        Else
            Dim angka As String = TextBox_SimpanAngka.Text
            Dim simpan As Bitmap
            simpan = New Bitmap(PictureBox5.Image, 30, 50)
            simpan.Save("E:\COLLEGE\SKRIPSI\CITRA HASIL SEGMENTASI
NIM\" + angka + ".bmp", System.Drawing.Imaging.ImageFormat.Bmp)
            MessageBox.Show("Karakter NIM Berhasil Disimpan",
"INFORMASI", MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If

    End Sub

    Private Sub Submenu_SimpanHasilSegmentasi_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
Submenu_SimpanHasilSegmentasi.Click
        If (PictureBox5.Image Is Nothing) Then
            MessageBox.Show("Pilih Komponen Hasil CCL yang akan
Disimpan Terlebih Dahulu", "WARNING!!!", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
        ElseIf (String.IsNullOrEmpty(TextBox_SimpanAngka.Text))
Then
            MessageBox.Show("Masukkan File Name untuk Karakter NIM
yang akan Disimpan", "WARNING!!!", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
            TextBox_SimpanAngka.Text = ""
            TextBox_SimpanAngka.Focus()
        Else
            Dim angka As String = TextBox_SimpanAngka.Text
            Dim simpan As Bitmap
            simpan = New Bitmap(PictureBox5.Image, 30, 50)
            simpan.Save("E:\COLLEGE\SKRIPSI\CITRA HASIL SEGMENTASI
NIM\" + angka + ".bmp", System.Drawing.Imaging.ImageFormat.Bmp)
            MessageBox.Show("Karakter NIM Berhasil Disimpan",
"INFORMASI", MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If
    End Sub

    Private Sub Button_Reset_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button_Reset.Click
        PictureBox1.Image = Nothing
        PictureBox2.Image = Nothing
        PictureBox3.Image = Nothing
```

```
        PictureBox4.Image = Nothing
        PictureBox5.Image = Nothing
        RichTextBoxHasilPelabelan.Text = Nothing
        TextBox_FileName.Text = ""
        TextBox_Threshold.Text = ""
        TextBox_SimpanAngka.Text = ""
        ComboBoxPelabelan.Text = ""
        Button_Reset.Enabled = True
        Button_BukaCitra.Enabled = True
        Button_Resize.Enabled = False
        Button_MedianFilter.Enabled = False
        Button_Grayscale.Enabled = False
        Button_Biner.Enabled = False
        Button_Crop.Enabled = False
        Button_CCL.Enabled = False
        Button_SimpanHasilSegmentasi.Enabled = False
        ProgressBar1.Value = 0
    End Sub

    Private Sub Submenu_Reset_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Submenu_Reset.Click
        PictureBox1.Image = Nothing
        PictureBox2.Image = Nothing
        PictureBox3.Image = Nothing
        PictureBox4.Image = Nothing
        PictureBox5.Image = Nothing
        RichTextBoxHasilPelabelan.Text = Nothing
        TextBox_FileName.Text = ""
        TextBox_Threshold.Text = ""
        TextBox_SimpanAngka.Text = ""
        ComboBoxPelabelan.Text = ""
        Button_Reset.Enabled = True
        Button_BukaCitra.Enabled = True
        Button_Resize.Enabled = False
        Button_MedianFilter.Enabled = False
        Button_Grayscale.Enabled = False
        Button_Biner.Enabled = False
        Button_Crop.Enabled = False
        Button_CCL.Enabled = False
        Button_SimpanHasilSegmentasi.Enabled = False
        ProgressBar1.Value = 0
    End Sub

    Private Sub Button_Kembali_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button_Kembali.Click
        Form_Utama.Show()
        Me.Close()
    End Sub

    Private Sub Submenu_Kembali_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Submenu_Kembali.Click
        Form_Utama.Show()
        Me.Close()
    End Sub
```

```
    Private Sub Submenu_Program_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Submenu_Program.Click
        Form_TentangProgram.Show()
    End Sub

    Private Sub Submenu_Pembuat_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Submenu_Pembuat.Click
        Form_TentangPembuat.Show()
    End Sub

End Class
```

Form_TentangProgram.vb

```
Public Class Form_TentangProgram

    Private Sub Button_Kembali_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button_Kembali.Click
        Form_Utama.Show()
        Form_Proses.Close()
        Me.Close()
    End Sub

    Private Sub Button_Keluar_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button_Keluar.Click
        Application.Exit()
    End Sub
End Class
```

Form_TentangPembuat.vb

```
Public Class Form_TentangPembuat

    Private Sub Button_Kembali_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button_Kembali.Click
        Form_Utama.Show()
        Form_Proses.Close()
        Me.Close()
    End Sub

    Private Sub Button_Keluar_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button_Keluar.Click
        Application.Exit()
    End Sub
End Class
```