

Pembuatan Pola Tulisan Tangan Melalui Webcam Dengan Metode L*a*b* Color Space

Sahwari¹, Untung Jasuli²

¹ Pendidikan Teknologi Informasi, STKIP PGRI SITUBONDO, Situbondo, Indonesia

² Teknik Informatika, STIKOM PGRI, Banyuwangi, Indonesia

Info Artikel	ABSTRAK
<p>Riwayat Artikel: Diterima : 14-Juni-2022 Direvisi : 30-Juni-2022 Disetujui : 12-Juli-2022</p> <hr/> <p>Kata Kunci:</p> <p><i>L*a*b* Color Space,</i> <i>Pola Tulisan Tangan,</i> <i>Webcam.</i></p> <hr/> <p>Keywords:</p> <p><i>L*a*b* Color Space,</i> <i>Handwriting Pattern,</i> <i>Webcam.</i></p>	<p>Kemajuan teknologi membuat sebuah perangkat komputer memiliki kemampuan yang tinggi untuk meningkatkan kinerja dalam pengolahan data menjadi informasi. Salah satu hal yang dapat dilakukan adalah memanfaatkan teknologi tersebut untuk memasukkan data ke dalam komputer. Banyak sekali teknik yang dapat memudahkan pengguna untuk memasukkan data ke dalam komputer salah satunya adalah menggunakan webcam untuk pengambilan gambar yang selanjutnya bisa diproses atau disimpan dalam komputer. Tidak menutup kemungkinan bagi seorang pengguna yang awalnya melakukan pembuatan pola tulisan tangan menggunakan touch screen atau menggunakan mouse, sekarang menggunakan webcam sebagai alat input pembuatan pola tulisan tangan tersebut.</p> <p>Untuk itulah perlu diadakan penelitian untuk membuktikan kemungkinan tersebut yaitu pengenalan pola dari isyarat yang diberikan pengguna melalui webcam dengan warna sebagai acuan untuk penulisan pola. Pola tersebut ditulis melalui webcam menggunakan acuan warna yang diklasifikasikan dengan metode L*a*b* color space berdasarkan karakteristik warna yang telah dipilih.</p> <p>Dalam penelitian ini telah dihasilkan aplikasi yang mampu melakukan pembuatan pola tulisan tangan yang diisyaratkan melalui webcam berdasarkan warna sebagai acuan untuk melakukan pembuatan pola tulisan tangan.</p> <hr/> <p>ABSTRACT</p> <p><i>Advances in technology make a computer device has a high ability to improve performance in processing information data. One thing that can be done is to use this technology to enter data into a computer. Many techniques can make it easier for users to enter data into a computer, one of which is using a webcam to take pictures which can then be stored or stored on a computer. It is possible for a user who started making handwriting patterns using a touch screen or using a mouse, now using a webcam as an input tool for making the handwriting pattern.</i></p> <p><i>For this reason, it is necessary to conduct research to prove this possibility, namely pattern recognition from the use provided by the user via a webcam as a reference for writing patterns. The pattern is written via webcam using color references which are classified by the L*a*b* color space method based on the selected characteristics.</i></p> <p><i>In this research, an application has been produced that is able to make the required handwriting patterns via a webcam based on color as a reference for making handwriting patterns.</i></p>
<p>Penulis Korespondensi: Sahwari, Program Studi Pendidikan Teknologi Informasi, STKIP PGRI SITUBONDO, Email: sahwari52@gmail.com</p>	

1. PENDAHULUAN

Pada dasarnya webcam digunakan untuk mengambil gambar ketika seorang pengguna ingin melakukan pengambilan gambar, chating atau video-call dengan komputer. Namun seiring dengan semakin pesatnya perkembangan teknologi, telah banyak ditemukan metode-metode untuk melakukan pemrosesan data khususnya data yang berupa gambar. Tidak menutup kemungkinan bagi seorang pengguna yang awalnya melakukan pembuatan pola tulisan tangan menggunakan laya sentuh atau menggunakan mouse, sekarang menggunakan webcam sebagai alat input pembuatan pola tulisan tangan. Pola tersebut merupakan hasil dari gambar yang ditangkap webcam, yang dibentuk melalui isyarat tangan dengan menggunakan warna sebagai acuan.

Untuk itulah perlu diadakan penelitian untuk membuktikan kemungkinan tersebut yaitu pembuatan pola tulisan tangan melalui webcam dengan warna sebagai acuan. Metode yang digunakan adalah metode untuk mengklasifikasikan warna sesuai dengan warna yang dipilih dan akan dijadikan sebagai acuan untuk melakukan proses pembuatan pola tulisan tangan.

Metode yang akan digunakan untuk mengklasifikasikan warna berdasar karakteristik warna tertentu adalah menggunakan metode L*a*b* Color Space.

2. METODE PENELITIAN

2.1 L*A*B* COLOR SPACE

Segmentasi warna kulit tidak tergantung hanya pada pendekatan segmentasi, tetapi juga distribusi warna dalam ruang warna yang berbeda. Dalam kebanyakan non-parametrik metode pelacakan warna kulit, ruang warna HSI digunakan, sebagai distribusi warna dalam ruang warna HSI lebih konsentratif dari pada ruang warna RGB. Bentuk yang berbeda RGB dan ruang warna HSI, L*a*b* ruang warna yang digunakan dalam metode yang diusulkan, di mana L* mewakili lightness dan a* dan b* adalah koordinat Kromatisitas. Hal ini hampir sebanding dengan persepsi visual, yang berarti bahwa jarak yang sama dalam ruang warna sesuai dengan perbedaan warna yang dirasakan sama. Nilai-nilai L*, a* dan b* dihitung dengan menggunakan persamaan berikut.[1]

$$L^* = \begin{cases} 116 \left(\frac{Y}{Y_n}\right)^{1/3} - 16 & \text{if } \frac{Y}{Y_n} > 0.008856 \\ 903.3 \left(\frac{Y}{Y_n}\right)^{3/5} - 16 & \text{if } \frac{Y}{Y_n} > 0.008856 \end{cases}$$

$$a^* = 500 \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right]$$

$$b^* = 500 \left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right]$$

Dimana

$$f(t) = \begin{cases} t^{1/3} & t > 0.008856 \\ 7.787 * t + 16/116 & \text{otherwise} \end{cases}$$

Dimana X_n, Y_n, dan Z_n adalah CIE XYZ nilai tristimulus dari pencerminan sempurna diffuser. Dalam pendekatan kami, mereka ditetapkan sebagai 250,155, 255.000 dan 301,410, masing-masing. X, Y, dan Z adalah nilai-nilai tristimulus yang dihitung dari informasi R, G, B setiap pixel berdasarkan Persamaan berikut:

$$X = 0.607 * R + 0.174 * G + 0.2 * B$$

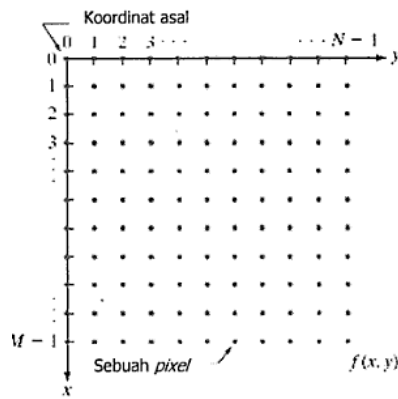
$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

$$Z = 0.066 * G + 1.116 * B$$

2.2 PENGOLAH CITRA DIGITAL

Secara umum, pengolahan citra digital menunjuk pada pemrosesan gambar 2 dimensi menggunakan komputer. Dalam konteks yang lebih luas, pengolah citra digital mengacu pada pemrosesan setiap data 2 dimensi. Citra digital merupakan sebuah lari (array) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu.[2]

Suatu citra dapat didefinisikan sebagi fungsi f(x,y) berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan aplitudo f di titik koordinat (x, y) dinamakan intesnsitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x, y dan nilai aplitudo f secara keseluruhan berhingga (finite) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital. Gambar di bawah menunjukkan posisi koordinat citra digital.[2]



Gambar 1: Koordinat Citra Digital

Citra digital dapat dituli dalam bentuk matrik sebagai berikut:

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

Nilai pada suatu irisan antara baris dan kolom (pada posisi x, y) disebut dengan picture element, image element, pels atau pixels. Istilah terakhir (pixel) paling sering digunakan pada citra digital. Gambar 2.2 menunjukkan ilustrasi digitalisasi citra dengan M=16 baris dan N = 16 kolom.[2]

Pembentukan citra digital (diskrit) melalui beberapatahapan, yaitu akuisisi citra, sampling dan kauntisasi. Proses akuisisi citra adalah pemetaan suatu pandangan (scence) menjadi citra kontinu dengan menggunakan sensor. Ada beberapa macam sensor untuk akuisisi citra, yaitu sensor tunggal (single sensor), sensor garis (sensor strip) dan sensor lari (sensor array).[2]

Tahap berikutnya untuk pembentukan citra digital setelah kontinu terbentuk adalah proses sampling. Proses sampling adalah proses digitasi pada koordinat x, y. Seperti yang disebutkan di atas, hasil dari sensor masih berupa citra kontinu yang merupakan fungsi kontinu f(x,y). Fungsi tersebut merupakan sinyal kontinu pada nilai x, y dan juga amplitudonya (intensitas). Nilai x dan y yang kontinu akan diubah menjadi bentuk diskrit.[2]

Proses selanjutnya adalah proses kuantisasi. Proses kuantisasi adalah proses perubahan nilai emplitudo kontinu menjadi nilai baru yang berupa nilai diskrit. Nilai amplitudo yang dikuantisasi adalah nilai-nilai pada koordinat diskrit hasil proses sampling.[2]

2.3 BAHASA PEMROGRAMAN C-SHARP

C# (dibaca: C sharp) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET Framework. Bahasa pemrograman ini dibuat berbasis bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic, dan lain-lain) dengan beberapa penyederhanaan.[3]

C# ditujukan agar cocok digunakan untuk menulis program aplikasi baik dalam sistem client server (hosted system) maupun sistem embedded (embedded system), selain itu C# juga ditujukan sebagai bahasa yang sederhana, moderen, bertujuan umum dan berorientasi objek.[3]

C# menghilangkan beberapa hal yang bersifat kompleks yang terdapat dalam beberapa macam bahasa perograman seperti JAVA dan C++, termasuk diantaranya menghilangkan *macro*, *templates*, *multiple inheritance* dan *virtual base classes*. Hal-hal tersebut yang dapat menyebabkan kebingungan pada saat menggunakannya, dan juga berpotensi dapat menjadi masalah bagi para progremmer C++. Jika pertama kali belajar C# sebagai bahasa pemrograman, maka hal-hal tersebut diatas tidak akan membuang waktu terlalu banyak untuk mempelajarinya.[3]

C# bersifat sederhana , karena bahasa ini didasarkan kepada bahasa C dan C++. Jika familiar dengan C dan C++ atau bahkan java, akan menemukan aspek –aspek yang begitu familiar, seperti *statements*, *expression*, *operators*, dan beberapa fungsi yang diadopsi langsung dari C dan C++, tetapi dengan berbagai perbaikan yang membuat bahasanya menjadi lebih sederhana.[3]

Dikatakan C# bersifat modern, karena adanya beberapa fitur seperti *exception handling*, *garbage collection*, *extensible data types*, dan *code security* (keamanan kode/ bahasa pemrograman). Dengan adanya fitur-fitur tersebut, menjadikan bahasa C# sebagai bahasa pemrograman yang modern.[3]

2.4 .NET FRAMEWORK

.Net adalah “platform perangkat lunak”, suatu lingkup netral bahasa yang digunakan untuk mengembangkan dan membangun aplikasi-aplikasi yang dengan mudah dan aman akan beroperasi di dalamnya. Pada saat aplikasi-aplikasi di-deploy ke komputer tujuan, aplikasi akan dieksekusi dimana saja .Net diimplementasikan. Adapun komponen-komponen yang menyusun platform .Net secara bersamaan disebut .Net Framework.[4]

.Net Framework adalah model pemrograman platform .Net yang digunakan untuk membangun, men-deploy dan menjalankan XML Web service serta aplikasi. .Net framework berisi library-library class untuk menyediakan layanan standar yang dapat diintegrasikan ke bagian sistem komputer. Framework ini mengatur semua aspek eksekusi program, seperti alokasi memori untuk penyimpanan data dan intruksi, eksekusi aplikasi, izin akses aplikasi dan re-alokasi memori pada resource yang tidak diperlukan lagi.[4]

.Net framework memiliki dua komponen utama, yaitu Common Language Runtime (CLR) dan Base Class Library (BCL). CLR di .Net framework bertugas untuk menglola memori, mengeksekusi thread, mengeksekusi kode, verifikasi keamanan kode, melakukan kompilasi, menangani aksepsi lintas bahasa, binding dinamik dan menyediakan layanan-layanan sistem lainnya sehingga menjadikan proses pengembangan aplikasi semakin mudah. BCL .Net framework merupakan koleksi tipe-tipe reusable yang terintegrasi dengan CLR, BCL menyediakan kumpulan class, interface, dan tipe-tipe nilai yang mampu memberi akses ke utilitas spesifik dan berbagai fungsi sistem. BCL memungkinkan untuk pengembangan bermacam aplikasi, mulai dari aplikasi command-line, Graphical User Interface (GUI) hingga aplikasi Web Form dan XML Web Services.[4]

2.5 L*A*B* COLOR SPACE

Segmentasi warna kulit tidak tergantung hanya pada pendekatan segmentasi, tetapi juga distribusi warna dalam ruang warna yang berbeda. Dalam kebanyakan non-parametrik metode pelacakan warna kulit, ruang warna HSI digunakan, sebagai distribusi warna dalam ruang warna HSI lebih konsentratif dari pada ruang warna RGB. Bentuk yang berbeda RGB dan ruang warna HSI, L*a*b* ruang warna yang digunakan dalam metode yang diusulkan, di mana L* mewakili lightness dan a* dan b* adalah koordinat Kromatisitas. Hal ini hampir sebanding dengan persepsi visual, yang berarti bahwa jarak yang sama dalam ruang warna sesuai dengan perbedaan warna yang dirasakan sama. Nilai-nilai L*, a* dan b* dihitung dengan menggunakan persamaan berikut.[1]

$$L^* = \begin{cases} 116 \left(\frac{Y}{Y_n}\right)^{1/3} - 16 & \text{if } \frac{Y}{Y_n} > 0.008856 \\ 903.3 \left(\frac{Y}{Y_n}\right)^{1/3} - 16 & \text{if } \frac{Y}{Y_n} > 0.008856 \end{cases}$$

$$a^* = 500 \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right]$$

$$b^* = 500 \left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right]$$

Dimana

$$f(t) = \begin{cases} t^{1/3} & t > 0.008856 \\ 7.787 * t + 16/116 & \text{otherwise} \end{cases}$$

Dimana X_n, Y_n, dan Z_n adalah CIE XYZ nilai tristimulus dari pencerminan sempurna diffuser. Dalam pendekatan kami, mereka ditetapkan sebagai 250,155, 255.000 dan 301,410, masing-masing. X, Y, dan Z adalah nilai-nilai tristimulus yang dihitung dari informasi R, G, B setiap pixel berdasarkan Persamaan berikut:

$$X = 0.607 * R + 0.174 * G + 0.2 * B$$

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

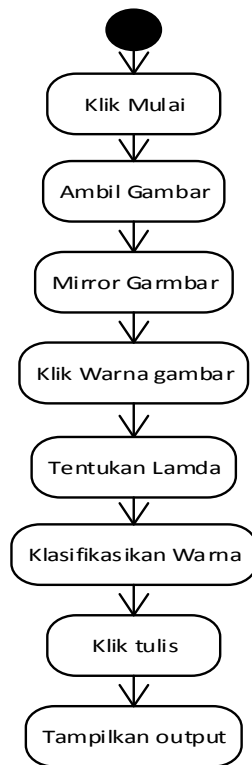
$$Z = 0.066 * G + 1.116 * B$$

2.6 DESAIN APLIKASI

Desain sistem aplikasi ini dibagi menjadi 2 bagian, yaitu activity diagram dan desain form. Masing-masing untuk desain tersebut akan dijelaskan dibawah ini:

2.7 ACTIVITY DIAGRAM

Activity diagram aplikasi digambarkan seperti dibawah ini:

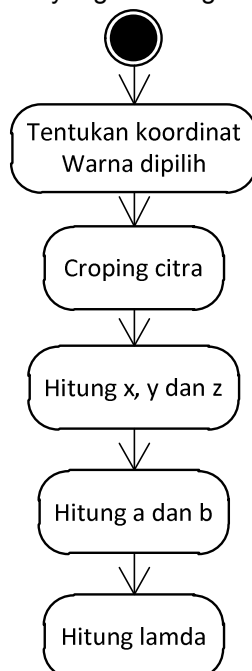


Gambar 2: Activity Diagram Aplikasi

Pada activity diagram diatas dapat dijelaskan bahwa untuk memulai proses diharuskan untuk mengklik tombol mulai dan aplikasi akan mengambil gambar melalui webcam. Selanjutnya klik warna pada gambar yang ditangkap sebagai acuan untuk menuliskan pola sehingga aplikasi bisa mengklasifikasikan warna yang telah dipilih berdasarkan karakteristik warna tersebut. Untuk memulai menulis pola bisa mengklik tombol tulis dan pola tulisan tangan akan ditampilkan pada tampilan output yang telah disediakan.

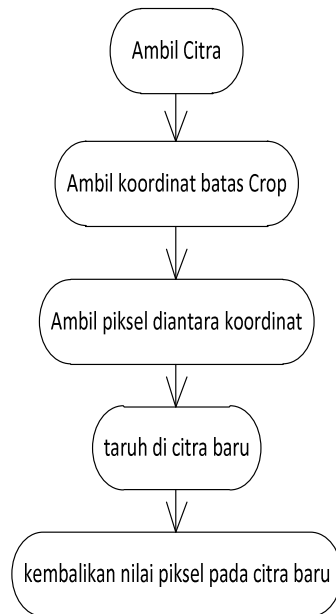
a. Activity Diagram Menentukan Lamda

Proses ini adalah untuk menghitung lamda yang akan digunakan dalam pengklasifikasian warna.



Gambar 3: Activity Diagram Menentukan Lamda

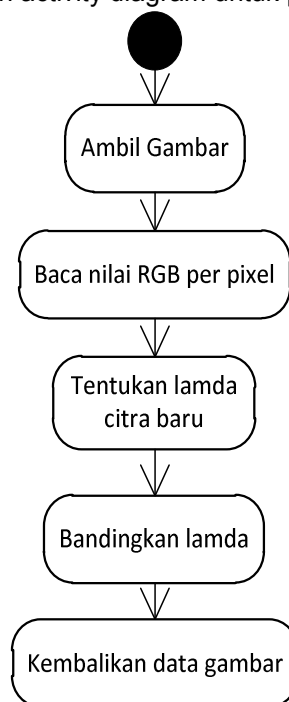
b. Activity Diagram Cropping Citra



Gambar 4: Activity Diagram Cropping Citra

c. Activity Diagram Klasifikasi Warna

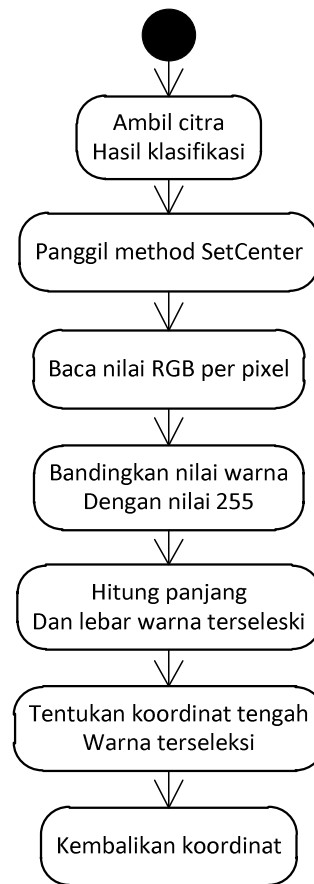
Untuk mengklasifikasikan warna, seperti yang dijelaskan sebelumnya aplikasi ini menggunakan metode $L^*a^*b^*$ color space. Berikut adalah activity diagram untuk proses tersebut.



Gambar 5: Activity Diagram Klasifikasi Warna

d. Activity Diagram Tulis

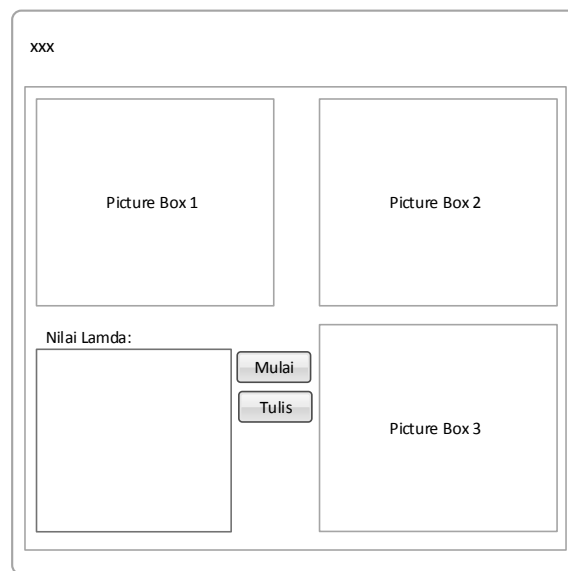
Proses ini untuk melakukan penulisan berdasarkan acuan warna yang telah dipilih sebelumnya. Berikut adalah activity diagram untuk proses tulis tersebut.



Gambar 6: Activity Diagram Tulis

2.8 DESAIN FORM

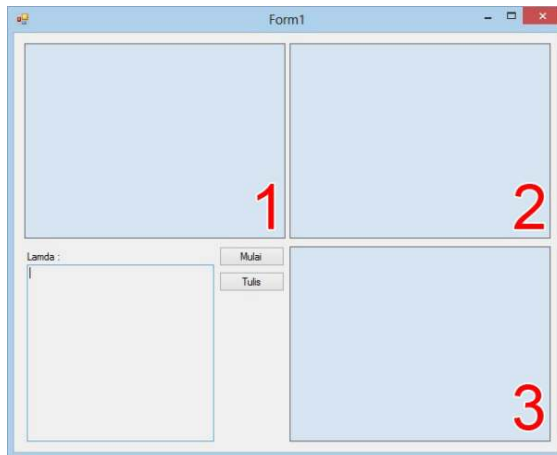
Adapun desain form pada aplikasi ini adalah seperti gambar dibawah ini:



Gambar 7: Desain Form Aplikasi

3. HASIL DAN ANALISIS

Dari hasil penelitian berdasarkan teori dan desain sistem yang dijelaskan sebelumnya, dapat dihasilkan sebuah aplikasi yang mampu melakukan pembuatan pola tulisan tangan melalui webcam. Dibawah ini adalah tampilan dan penjelasan hasil uji coba dari aplikasi tersebut.

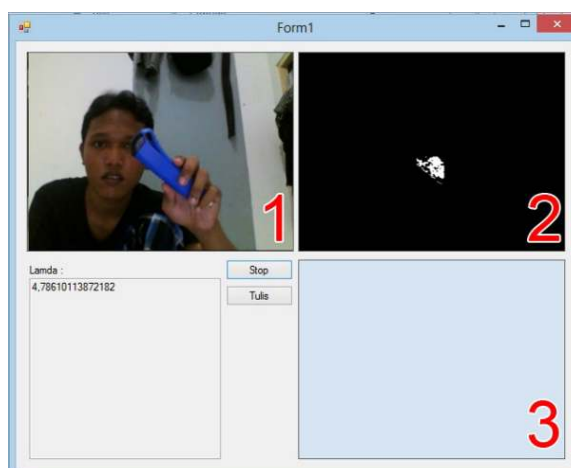


Gambar 8: Tampilan Awal

Gambar diatas adalah tampilan ketika aplikasi dijalankan pertama kali. PictureBox yang disediakan tidak akan menampilkan gambar, karena webcam belum memulai untuk melakukan capture. Untuk memulai proses capture pada aplikasi harus mengklik tombol mulai dan citra akan ditampilkan pada picturebox 1. Tampilan ketika webcame melakukan capture adalah seperti pada gambar 8 dan selanjutnya memilih warna objek pada citra yang ditampilkan pada picturebox 1 untuk dijadikan acuan dalam pembuatan pola tulisan tangan.



Gambar 9: Tampilan Ketika Ambil Gambar dari Webcam

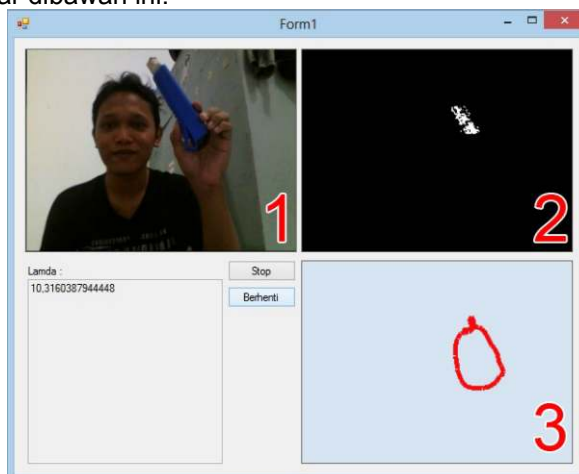


Gambar 10: Tampilan Ketika Memilih Objek

Gambar 9 adalah tampilan ketika memilih warna objek pada citra picturebox 1 untuk dijadikan acuan. Warna objek yang dipilih akan diklasifikasikan menurut karakteristik warna tersebut kemudian akan dilakukan perbandingan terhadap warna pada setiap piksel citra yang ditangkap. Hasil perbandingan tersebut adalah citra biner yang akan ditampilkan pada picturebox 2. Pada citra tersebut, nilai warna pada

piksel yang sesuai dengan karakteristik yang telah dipilih akan diberikan warna putih dan yang tidak sesuai akan diberikan warna hitam. Sedangkan nilai lamda akan ditampilkan pada textbox yang telah disediakan.

Untuk memulai membuat pola tulisan tangan, dapat dilakukan dengan menekan tombol tulis dan menggerakkan objek dengan warna yang sebelumnya telah dipilih. Pola yang telah dibuat akan ditampilkan pada picturebox 3 yang akan berbentuk garis tebal hasil pergerakan objek tersebut. Proses tersebut seperti yang ditampilkan pada gambar dibawah ini.



Gambar 10: Tampilan Ketika Menulis Pola

Proses untuk menulis pola akan berhenti jika tombol berhenti diklik, dan akan menghentikan proses capture jika tombol stop diklik.

4. KESIMPULAN

Ada beberapa kesimpulan yang dapat diambil dari peneliatian ini, diantaranya adalah:

1. Metode $L^*a^*b^*$ Color Space mampu untuk mengklasifikasina warna sesuai karakteristik yang telah dipilih.
2. Pembuatan pola tulisan tangan menggunakan webcam bisa dilakukan dengan penerapan algoritma $L^*a^*b^*$ Color Space.
3. Bahasa pemrograman C-Sharp merupakan bahasa pemrograman yang power full untuk malakukan image processing terhadap citra yang ditangkap oleh webcam secara real time.

5. UCAPAN TERIMAKASIH

Bagian Ucapan Terima Kasih adalah opsional. Sumber penelitian dapat dimasukkan dalam bagian ini.

6. REFERENSI

- [1] Yuan, Miaolong, dkk. 2009, *Robust Hand Tracking Using a Simple Color Classification Tehnique*, Singapore.
- [2] Putra, Darma, 2010, *Pengolahan Citra Digital*, Yogyakarta.
- [3] Heryana Ali Ahmad, *Pengenalan Bahasa C#*. Jakarta. 2004.
- [4] Prasetyo, Didik Dwi, 2006, *Pemrograman Aplikasi Database dengan Visual Basic .Net 2005 dan MS Access*, Jakarta.