IMPLEMENTASI APLIKASI PENGENALAN JENIS PAKAIAN ADAT SUMATERA UTARA "ULOS" BERBASIS ANDROID

Jhan Mayer[1]

[1e] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : mr.jhan_mayer@yahoo.com

ABSTRAK

Ulos adalah kain tenun tradisional Batak yang terdiri dari berbagai jenis, corak atau motif, fungsi serta ukuran. Diantara kain tenun ikat yang ada, hanya ulos yang memiliki arti mendalam dan sangat luas. Ulos dibedakan atas proses pembuatan *gorga* / motif beserta coraknya, dan menjadi ukuran penentu bagi nilai dan harganya. Seiring perkembangan jaman globalisasi dan teknologi masyarakat Indonesia terutama generasi muda, semakin mengandalkan teknologi dan menganggap hal itu lebih menarik dibandingkan dengan mengenal dan melestarikan kebudayaan daerahnya sendiri. Hal ini adalah penyebab dibutuhkannya perangkat lunak untuk memberikan pengenalan yang menarik dan sesuai perkembangan teknologi sebagai upaya untuk melestarikan ulos. Untuk mengenal gambar ulos digunakan fitur SURF (*Speed Up Robust Features*) untuk mencocokkan gambar objek ulos dengan menentukan *keypoint*, lalu mencari kesamaan *keypoint* pada tiap objek.
Kata Kunci : Ulos, Batak, SURF, *Keypoint.*

1. PENDAHULUAN

Ulos dalam pengertian umum adalah kain tenun tradisional Batak yang terdiri dari berbagai jenis, corak atau motif, fungsi serta ukuran. Diantara kain tenun ikat yang ada, hanya ulos yang memiliki arti mendalam dan sangat luas. Ulos dibedakan atas proses pembuatan *gorga* / motif beserta coraknya, dan menjadi ukuran penentu bagi nilai dan harganya. Desain yang dikembangkan pada kain ulos merupakan hasil karya penggalian ornamen budaya Batak yang diekspresikan dalam rancangan baru, dan masih terus digali dengan berbagai improvisasi. Suku Batak terbagi dalam 5 sub etnis secara geografis yaitu Batak Toba, Batak Karo, Batak Simalungun, Batak Pakpak, dan Batak Mandailing. Hal ini menyebabkan banyak ragam dan jenis ulos yang ada pada suku Batak. Sedangkan dalam suku Batak terdapat 12 jenis ulos, yaitu ulos *jugia(homitan)*, ulos *ragidup*, ulos *ragidup silinggom*, ulos *ragihotang*, ulos *godang* atau *ulos sadum angkola*, ulos *runjar*, ulos *sibolang*, ulos *suri-suri ganjang*, ulos *mangiring*, ulos bintang *maratur*, ulos *sitolu tuho*, ulos *bolean.*

Masyarakat Sumatera Utara pada umumnya orang Batak, hakikatnya memutuskan untuk merantau dan berkeluarga di tempat mereka merantau tersebut. Sehingga orang tua mulai mengabaikan dan tidak mengenalkan arti, jenis dan fungsi dari ulos yang sering mereka pakai disaat acara adat Batak. Berdasarkan pengujian kuesioner post-test yang dilakukan, maka dapat diambil kesimpulan bahwa 96,67% responden mengetahui ulos, tetapi 53,33% responden tidak mengetahui jenis-jenis ulos pada suku batak toba dan 63,33% responden tidak mengetahui kegunaan ulos pada suku batak toba, sehingga 63,67 % responden tidak mengetahui pola dan motif ulos pada suku batak toba dan 83,67% responden tertarik memiliki ulos, dan 53,33% ketertarikan terhadap ulos menurut responden adalah menurun.

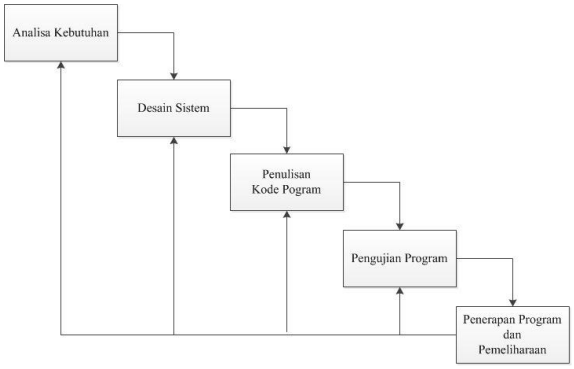
Karena banyaknya pengguna android saat ini di Indonesia, sehingga mempermudah memberikan informasi tentang ulos. Pada tahun 2014 pengguna android berkembang pesat mencapai 1,1 Miliar pengguna[4]. Dengan memanfaatkan Android yang memiliki sebuah sistem operasi Linux yang dioperasikan di dalam *smartphone*. Android juga menyediakan *platform* terbuka untuk pengembangan untuk menciptakan aplikasi terbaru dan menarik. Selain itu pengguna Android juga pada saat ini di Indonesia sudah berkembang pesat penggunaannya. Sehingga mempermudah memberikan informasi tentang ulos.

Adapun tujuan yang akan dicapai dari pembangunan aplikasi ini adalah untuk mempermudah masyarakat untuk memperoleh informasi tentang ulos.

2. ISI PENELITIAN

2.1 Model pembuatan perangkat lunak

Model pembangunan perangkat lunak menggunakan model waterfall, karena menghasilkan sistem yang terstruktur dengan baik ditiap prosesnya. Proses-proses yang terdapat dalam model waterfall dapat dilihat pada gambar berikut:



Gambar 1 Metode Waterfall

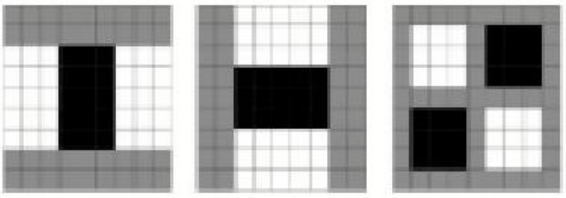
2.2 SURF (***Speed-Up Robust Features***)

Metoda *Speed-Up Robust Features* (SURF) merupakan sebuah metode deteksi fitur yang menggunakan *keypoint* dari sebuah gambar. *Keypoint* itu sendiri adalah bagian-bagian dari sebuah gambar yang nilainya kuat atau tetap ketika mengalami perubahan skala, rotasi, *blurring*, transformasi 3 dimensi, pencahayaan dan juga perubahan bentuk. Perubahan bentuk itu bisa terjadi karena bentuk gambar data awal yang tidak utuh atau tidak sempurna gambar yang ada di dalam gambar sampel tersebut. Gambar uji yang tidak utuh mungkin karena ada objek lain yang menutupi, atau pengambilan gambar yang tidak sempurna, atau keadaan objek itu sendiri yang telah mengalami perubahan.

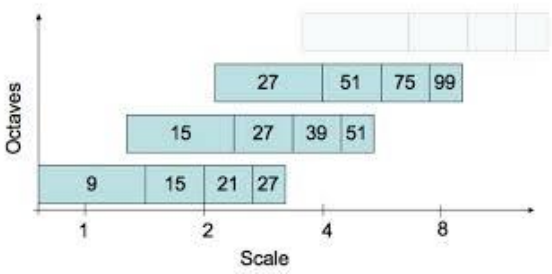
2.2.1 Scale Space Ekstrema Detection

Scale space dibagi kedalam bilangan octave, dimana satu octave merepresentasikan respon filter yang diperoleh dengan melakukan proses konvolusi gambar masukan dengan filter yang ukurannya bertambah tinggi. Dalam SURF, level paling rendah dalam *scale space* diperoleh dari output filter 9x9. Filter ini berhubungan dengan nilai Gaussian dengan σ = 1.2. Ketika membuat filter yang lebih besar, ada beberapa faktor yang harus diperhitungkan. Kenaikan ukuran dibatasi oleh panjang dari *lobe* yang positif atau negatif dari turunan *Gaussian* orde kedua. *Lobe* positif direpresentasikan dengan warna kotak putih pada Gambar 1, dan l*obe* negatif direpresentasikan dengan warna kotak hitam pada Gambar 1. Gambar 1 menunjukkan turunan orde dua Gaussian terhadap x,y, dan xy.

Karena kita membutuhkan posisi *central pixel* tetap, maka kenaikan ukuran *pixel* minimum 2 (satu *pixel* untuk setiap sisi), karena ada 3 *lobe* dimasing-masing filter, maka ukuran kenaikan filter adalah 6. Dengan demikian ukuran filter-filter untuk *octave* pertama adalah 9 x 9, 15 x 15, 21 x 21, 27 x 27. Untuk setiap *octave* baru, ukuran kenaikan filter adalah *double* (dari 6 menjadi 12, menjadi 24, dan menjadi 48), sehingga ukuran filter untuk *octave* kedua adalah 15, 27, 39, 51. *Octave* ketiga 27, 51, 75, 99, dan jika ukuran gambar asli masih lebih besar dari ukuran filter maka *octave* keempat dibuat yaitu dengan ukuran filter 51, 99, 147, dan 195. Gambar 2 menunjukkan contoh *scale space*.



Gambar 1. Turunan Orde Dua Gaussian



Gambar 2. Scale Space

2.2.2 Keypoint Localization

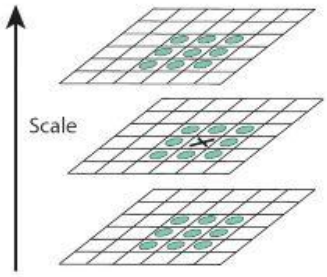
Sesudah proses penentuan ambang, selanjutnya proses *non-maxima suppression* dilakukan untuk mencari sekumpulan calon *keypoint*. Proses ini dilakukan dengan membandingkan masing-masing pixel gambar pada *scale space* dengan 26 tetangga, yang terdiri atas 8 titik di skala asli dan 9 di masing-masing skala diatas dan dibawahnya (total keseluruhan 9+8+9=26). Hasil dari *non-maxima suppression* adalah *keypoint* yang dihasilkan gambar. Proses ini adalah proses terakhir dimana dalam mencari *keypoint* yaitu dengan menggunakan interpolasi dengan data yang dekat dengan *keypoint*. Hal tersebut dilakukan dengan mencocokan quadratic 3 dimensi. H(x, y, σ) menyatakan determinan Hessian, yang didefinisikan sebagai persamaan berikut:

$$\boldsymbol{H(x) = H + \frac{\partial H^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 H}{\partial x^2} x} \qquad (0\text{-}1)$$

Lokasi ekstrim yang diinterpolasi, $x = (x, y, \sigma)$, ditemukan dengan mencari turunan dari fungsi ini dan diberi nilai nol, yang dapat digambarkan dengan persamaan berikut:

$$\boldsymbol{x = -\frac{\partial^2 H^{-1}}{\partial x^2} \frac{\partial H}{\partial x}} \qquad (0\text{-}2)$$

Jika lebih besar dari 0.5 pada x,y atau arah σ, lokasi perlu disesuaikan lagi dengan melakukan interpolasi lagi. Prosedur ini diulang sampai nilai kurang dari 0.5. Bila interpolasi telah dilakukan beberapa kali tetapi hasilnya tidak memenuhi nilai di bawah 0.5, maka *keypoint* tersebut dihapus. *Keypoint* yang tersisa adalah yang stabil.

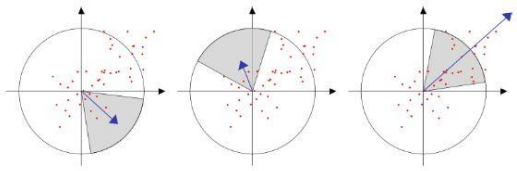


Gambar 3. Non-Maxima Suppression

2.2.3 Orientation Assignment

Agar supaya *keypoint* invarian terhadap rotasi gambar, maka kita harus melakukan identifikasi orientasi yang dihasilkan disekitar *keypoint*. Untuk tujuan tersebut, langkah pertama yang dilakukan adalah dengan menghitung respon Haar wavelet dalam arah x dan arah y pada daerah piksel tetangga disekitar *keypoint* yang berbentuk lingkaran yang dengan jarak 6*s*, dimana *s* adalah *scale* dari *keypoint* yang terdeteksi. Ukuran wavelet ditentukan menjadi 4s. Filtering dipercepat prosesnya dengan integral gambar, dan keluaran yang dihasilkan direpresentasikan sebagai titiktitik dalam ruang dengan respon horizontal sepanjang sumbu *axis* dan respon vertikal sepanjang sumbu koordinat.

Orientasi dominan diperkirakan dengan menghitung jumlah semua respon dalam pergeseran orientasi dengan ukuran window π /3 (seperti ditunjukkan pada Gambar 3), dimana respon horizontal dan vertikal dalam window tersebut dijumlahkan. Dari kedua respon yang dijumlahkan akan menghasilkan sebuah vektor orientasi lokal. Vektor terpanjang mendefinisikan orientasi *keypoint*. Ukuran pergeseran window merupakan parameter yang harus dipilih secara hati-hati.



Gambar 4. Pergeseran Orientasi

Untuk proses ekstraksi deskriptor, langkah pertama yang dilakukan adalah membuat daerah kotak sekitar *keypoint*, dimana *keypoint* sebagai pusat dari daerah kotak tersebut, dan orientasinya di sekitar orientasi yang ditentukan.



Gambar 5. Orientasi Perhitungan pada Skala yang Berbeda

Selanjutnya, respon wavelet *dx* dan *dy* dijumlahkan untuk masing-masing subregion. Selain memberikan informasi tentang polar dari perubahan intesitas, juga akan dihasilkan jumlah nilai absolut dari respon |*dx*| dan |*dy*|. Masing-masing sub-region mempunyai 4 dimensi deskriptor vektor *v*, yaitu *dx* ,*dy*, |*dx*| dan |*dy*|. Untuk 4 x 4 subregion, maka panjang vektor deskriptornya adalah 64 [3]. Vektor *v* dapat dilihat pada Persamaan (0-3).

$$\mathbf{v} = [\sum \boldsymbol{dx}, \sum \boldsymbol{dy}, \sum |\boldsymbol{dx}|, \sum |\boldsymbol{dy}|] \qquad (0\text{-}3)$$

Respon wavelet juga invarian terhadap pencahayaan, sedangkan sifat invarian terhadap kontras dicapai melalui pembentukan deskriptor kedalam satuan vektor.

2.3 Matriks Hessian Affine Region

Matrik adalah susunan bilangan yang diatur berdasarkan baris dan kolom. Bilangan – bilangan tersebut dinamakan entri dalam matrik atau disebut juga elemen (unsur). Matrik Hessian adalah matrik yang setiap elemennya dibentuk dari turunan parsial kedua dari suatu fungsi. Misalkan f(x) fungsi dengan n variabel yang memiliki turunan parsial kedua dan turunannya kontinu, matrik Hessian f(x) ditulis H adalah :

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial_n} \\ \frac{\partial^2 f}{\partial x_2 \partial_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial_n} \\ \cdots & \cdots & \ddots & \cdots \\ \frac{\partial^2 f}{\partial x_n \partial_1} & \frac{\partial^2 f}{\partial x_n \partial_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

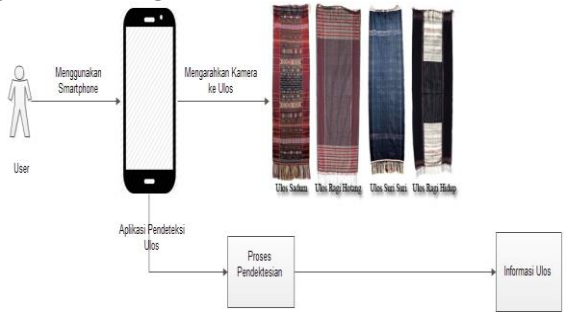
Matrik Hessian dapat digunakan untuk melakukan uji turunan kedua fungsi lebih dari satu variabel,

yaitu untuk mengidentifikasi optimum relatif dari nilai fungsi tersebut. Penggolongan titik stasioner fungsi dua variabel dengan menggunakan matriks Hessian misalkan f(x) = F(x1, …, xn) adalah fungsi bernilai real dimana semua turunan parsialnya kontinu. Misalnya x0 adalah titik stasioner dari F dan didefinisikan H = H(x0) dengan persamaan Hij = Fxi, yj (x0). H (x0) adalah Hessian dari F pada x0
Titik stasioner dapat digolongkan sebagai berikut :

a. x0. Adalah suatu minimum relatif dari F jika jika H(x0.) definit positif
b. x0. Adalah suatu maksimum relatif dari F jika H(x0.) definit negatif
c. x0. Adalah suatu titik pelana dari F jika H(x0.) indefinite

2.4 Analisis Arsitektur Sistem

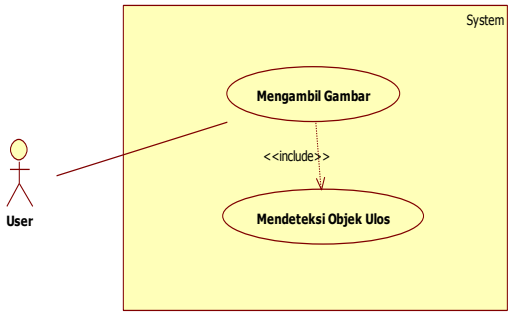
Arsitektur fisik sistem yang dibangun terdiri dari pengambilan gambar ulos, sistem ekstrasi gambar menjadi informasi teks dengan menggunakan algoritma SURF untuk menghasilkan jenis ulos tersebut. Aplikasi ini berfungsi sebagaiantarmuka agar pengguna dapat dengan mudah melakukan pengenalan jenis dan fungsi ulos tersebut.



Pada bagian ini, pengguna yang menggunakan aplikasi pengenalan ulos mengambil gambar dari motif ulos tersebut menggunakan kamera pada smartphone android. Setelah gambar diambil maka gambar tersebut akan diproses untuk menentukan jenis pada ulos tersebut. Fungsi utama aplikasi pengenalan jenis ulos adalah sebagai media informasi dalam bentuk aplikasi untuk penyampaian informasi mengenai jenis ulos secara mudah dan menguji algoritma yang digunakan apa berfungsi sesuai pengertian algoritma SURF itu sendiri yang berbasis *mobile Andorid*.

2.5 *Use Case Diagram*

Usecase Diagram merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antar aktor dengan aktivitas yang terdapat pada sistem. Sasaran pemodelan *use case* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan sistem yang akan dibangun.



Gambar 6. Use Case Diagram

3. PENUTUP

3.1 Kesimpulan

Berdasarkan hasil yang didapat dalam penelitian dan penyusunan skripsi ini serta disesuaikan dengan tujuan, maka diperoleh kesimpulan bahwa, aplikasi pengenalan ulos mempermudah masyarakat untuk mengenal ulos dan mengetahui arti, jenis dan fungsi ulos.

3.2 Saran

Berdasarkan hasil keseluruhan aplikasi, maka dapat diberikan saran yaitu penambahan fitur yang mungkin dapat dikembangkan. Bahkan pengembangan aplikasi untuk pendeteksian ulos pada suku batak selain batak toba dan lebih memepercepat pengambilan data.

DAFTAR PUSTAKA

[1] Sommerville, Software Enginering (Rekayasa Perangkat Lunak) edisi 6 jilid 1. Jakarta: Erlangga, 2003.

[2] Lowe, D.G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 2004.

[3] Bay, H, Ess A, Tuytelaars, T, Van Gool, L (2006). SURF: speeded up robust features. Proceedings of the 9th European Conference on Computer Vision, 3951 (1), 404–417. Springer LNCS.

[4] Gartner (2014): “Tabloid Pulsa dan berita teknologi”.

[5] Pudjo Widodo, Prabowo dan Herlawati “Menggunakan UML”, Informatika Bandung, oktober 2011.