

Navigasi Perangkat Lunak eBook Reader Berbasis Mobile Menggunakan Fitur Face Tracking

Rikza Azriyan, Saparudin, dan Reza Firsandaya Malik
Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya
Email: rezafm@unsri.ac.id

Abstraksi— eBook Reader adalah perangkat lunak untuk membaca buku elektronik pada komputer, dimana navigasi perpindahan halaman biasanya dilakukan dengan menggunakan tombol. Pada makalah ini akan dilakukan pembahasan mengenai navigasi perangkat lunak eBook Reader berbasis mobile berkamera depan menggunakan teknik *Face Tracking*. Dengan memori yang terbatas maka pada teknik *Face Tracking* digunakan metode perbandingan warna citra dalam *OpenCV Frameworks* untuk mengeliminasi proses perhitungan yang kompleks. Dari teknik *Face Tracking* ini didapatkan perkiraan letak kordinat wajah pada sumbu x dapat dimanfaatkan untuk navigasi perpindahan halaman horizontal pada perangkat lunak eBook Reader.

Kata Kunci — Face Tracking, eBook Reader, Navigasi, OpenCV Frameworks

I. PENDAHULUAN

eBook Reader adalah perangkat lunak untuk membaca buku elektronik pada media elektronik seperti komputer, *smart phone*, PDA (*Personal Digital Assistent*), *Tablet PC*, dan media - media lain yang mendukung. Sistem navigasi eBook Reader selama ini dilakukan dengan memberi aksi pada tombol navigasi, adapun lainnya menggunakan *Gesture Recognition* yang menerima masukan dari perangkat yang dilengkapi fasilitas *touch screen*. Dalam makalah ini, akan dibahas mengenai pengembangan sistem navigasi perpindahan halaman eBook Reader pada perangkat mobile berkamera depan menggunakan wajah melalui fitur *Face Tracking*.

Struktur tulisan ini pada bagian I mengenai pendahuluan yang membahas kegunaan eBook Reader dan pengembangan sistem navigasi pada eBook Reader, bagian II membahas metodologi, bagian III membahas hasil dan pembahasan dan bagian IV kesimpulan.

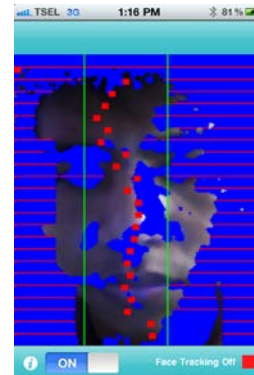
II. METODOLOGI

A. Navigasi Perpindahan Halaman eBook Reader

Sistem navigasi perpindahan halaman eBook Reader pada perangkat mobile dapat beraneka ragam, diantaranya dengan melakukan interaksi *touch* pada *touch screen*, interaksi pada tombol navigasi, dan pemanfaatan *Gesture Recognition*. Pada makalah ini akan dilakukan pembahasan mengenai pengembangan sistem navigasi eBook Reader berbasis mobile berkamera depan menggunakan fitur *Face Tracking*.

Pada Gambar 1, diasumsikan posisi wajah sebagai inisialisasi awal diletakkan berada di tengah kamera depan. Dengan pergerakan wajah ke arah kanan atau kiri sebagai letak kordinat wajah pada sumbu horinzontal (sumbu x). Dimana kordinat tersebut akan didefinisikan sebagai navigasi perpindahan halaman pada eBook Reader. Pergerakan wajah

ke arah kiri, mengindikasikan navigasi perpindahan ke halaman selanjutnya. Pergerakan wajah ke arah kanan, mengindikasikan perpindahan ke halaman sebelumnya. Seperti yang digambarkan pada Gambar 2 dan 3.



Gambar 1. Posisi Wajah Berada di Tengah Kamera Depan.



Gambar 2. Pergerakan Wajah ke Arah Kanan



Gambar 3. Pergerakan Wajah ke Arah Kiri

B. Teknik Face Tracking

Metode yang digunakan pada teknik *Face Tracking* ini adalah menggunakan metode perbandingan warna citra. Metode ini terbagi menjadi beberapa tahap, yaitu penyesuaian rata – rata cahaya, eliminasi daerah flat, eliminasi daerah yang memiliki fluktuasi, pencarian warna kulit, pencarian hue, eliminasi keluaran noise, dan proses *clipping*.

a. Proses penyesuaian rata – rata cahaya

Dalam melakukan proses ini dibutuhkan nilai *Light* dari citra, oleh sebab itu citra harus berada pada *HLS Color Space*. Setelah dilakukan *adjust* pada nilai *Light*, nilai tersebut diletakkan kembali pada citra menggantikan nilai *Light* yang lama.

$$L_{p'} = \begin{cases} L_p \times \frac{Aim}{Avg'} & , L_p < \\ 255 - (255 - L_p) \times \frac{255-Aim}{255-Avg'} & , L_p \geq \end{cases}$$

dimana $L_{p'}$ adalah nilai *Light* yang baru, L_p adalah nilai *Light* pada piksel suatu citra, Avg adalah nilai *Light* rata - rata pada suatu citra, dan Aim adalah target *Average Light* pada citra dan nilai berskala antara 0 - 250.

b. Proses eliminasi daerah flat

Dalam melakukan proses ini dibutuhkan nilai *Light* dari citra, oleh sebab itu citra harus berada pada *HLS Color Space*. Proses ini melakukan penjumlahan selisih nilai *Light* pada piksel - piksel citra yang terletak berurut dalam jumlah yang ditentukan. Langkahnya sebagai berikut :

$$\sum_{i \in n(i)} |L'_i| < th \Rightarrow eliminate$$

dimana $n(i)$ adalah piksel tetangga dari piksel i , dan th adalah nilai *threshold*. Nilai *threshold* ditentukan sebesar 25 untuk 30 piksel yang berurut, dan L_i' adalah hasil penjumlahan selisih nilai *Light* citra pada citra yang berurutan.

c. Proses eliminasi daerah yang memiliki fluktuasi

Untuk melakukan proses ini citra harus berada pada *RGB Color Space*, dan selisih nilai R, G, B masing - masing diperhitungkan. Selanjutnya nilai dari selisih masing - masing R, G, B tersebut dijumlahkan. Jika jumlahnya lebih besar dari *threshold* dalam jumlah 30 piksel yang berurut, maka bagian ini tereliminasi. *Threshold* ditentukan sebesar 750. Langkahnya sebagai berikut :

$$\sum_{i \in n(i)} |R''_i| + |G''_i| + |B''_i| > th \Rightarrow eliminate$$

dimana $n(i)$ adalah piksel tetangga dari i dan th adalah nilai *threshold*. $|R''_i|, |G''_i|, dan |B''_i|$ adalah nilai hasil dari selisih piksel i terhadap piksel $n(i)$.

d. Proses pencarian warna kulit

Untuk melakukan proses ini citra harus berada pada $R, G, B Color Space$. Dalam pencarian warna kulit, perbandingan rasio warna merah terhadap biru haruslah berada pada rentang 1:1 hingga 3:1. Berdasarkan pada penafsiran tersebut, warna diluar rasio tersebut dieliminasi karena tidak dapat dikatakan sebagai warna kulit. Akan tetapi proses ini akan berjalan tidak optimal jika latar belakang berwarna kayu, furniture, dan coklat.

e. Proses pencarian hue

Dalam melakukan proses ini citra harus berada pada $H, L, S Color Space$. Dalam proses ini dilakukan pembagian nilai $H Color Space$ ke dalam beberapa *cluster*. Dengan begitu, pada masing - masing *cluster*, dilakukan pencarian jumlah maksimum yang diberikan nilai $H Color Space$ yang paling sering digunakan dan diasumsikan sebagai *cluster* nilai *Hue* pada wajah. Selanjutnya dilakukan eliminasi terhadap bagian - bagian lain yang tidak termasuk ke dalam *cluster* tersebut. Jumlah *cluster* ditentukan sebanyak 40 *cluster*. Langkahnya sebagai berikut :

f. Proses eliminasi keluaran noise

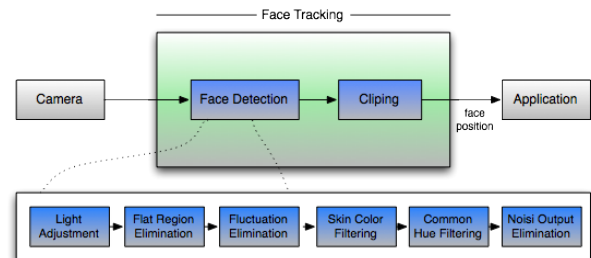
Setelah seluruh proses eliminasi dilakukan, proses eliminasi terakhir adalah mengeliminasi *noisy region* pada citra dengan menggunakan *median filter*.

e. Proses clipping

Proses terakhir adalah *Clipping*, yaitu melakukan perhitungan letak posisi wajah pada garis *scan* yang telah ditentukan. Pada makalah ini garis *scan* yang akan digunakan adalah garis *scan horizontal* untuk mendapatkan perkiraan posisi wajah pada sumbu x. posisi wajah ini akan digunakan untuk proses navigasi pada eBook Reader.

III. HASIL DAN PEMBAHASAN

Tiap - tiap proses dibawah ini dilakukan secara berurutan, seperti yang ditunjukkan pada Gambar 4.



Gambar 4. Diagram Urutan Proses *Face Tracking*

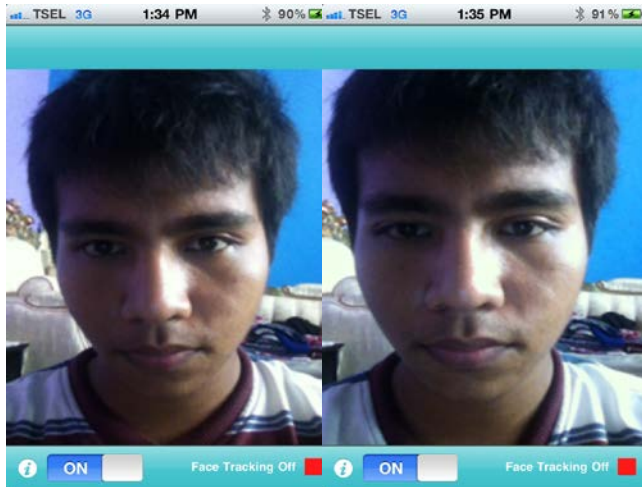
A. Proses penyesuaian rata – rata cahaya

Proses penyesuaian rata – rata cahaya dilakukan untuk membantu penyesuaian pencahayaan pada citra di berbagai lingkungan. Langkahnya sebagai berikut :

$$L_{p'} = \begin{cases} L_p \times \frac{Aim}{Avg}, & L_p < Avg \\ 255 - (255 - L_p) \times \frac{255 - Aim}{255 - Avg}, & L_p \geq Avg \end{cases}$$

dimana $L_{p'}$ adalah nilai *Light* yang baru, L_p adalah nilai *Light* pada piksel suatu citra, Avg adalah nilai *Light* rata - rata pada suatu citra, dan Aim adalah target *Average Light* pada citra dan nilai berskala antara 0 - 250, seperti yang ditunjukkan pada Gambar 5.

1. Ubah citra ke dalam bentuk *HLS Color Space*.
2. Dapatkan Avg pada citra
3. Dapatkan L_p yang terletak di kiri atas citra
4. Jika $L_p < Avg$ maka $L_{p'} = L_p * (Aim/Avg)$
5. Jika $L_p \geq Avg$ maka $L_{p'} = (255 - (255 - L_p)) * ((255 - Aim)/(255 - Avg))$
6. Letakkan nilai $L_{p'}$ menggantikan nilai L_p yang lama.



Gambar 5. Adjusting The Average Light

B. Proses eliminasi daerah flat

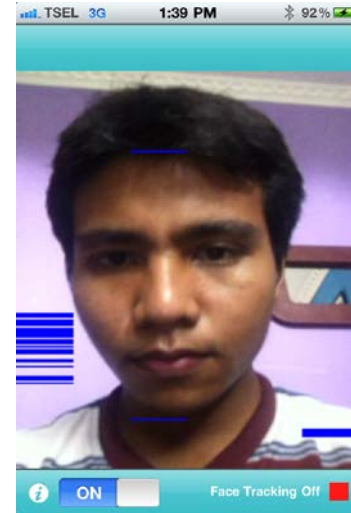
Pada proses ini dilakukan eliminasi pada citra yang memiliki nilai *Light* tidak berubah sepanjang jumlah urutan piksel dikarenakan lekukan dari wajah manusia. Langkahnya sebagai berikut :

$$\sum_{i \in n(i)} |L'_i| < th \Rightarrow \text{eliminate}$$

dimana $n(i)$ adalah piksel tetangga dari piksel i , dan th adalah nilai *threshold*. Nilai *threshold* ditentukan sebesar 25 untuk 30 piksel yang berurut, dan L_i' adalah hasil penjumlahan selisih nilai *Light* citra pada citra yang berurutan. Seperti yang ditunjukkan pada Gambar 6.

1. Ubah citra ke dalam bentuk *HLS Color Space*.

2. Tentukan selisih nilai *Light* antara piksel i dengan pikse $n(i)$, begitu pula piksel $n(i)$ dengan pikse tetangganya hingga 30 piksel yang berurutan.
3. Tentukan nilai L_i' pada 30 piksel yang berurutan
4. Jika $L_i' < th$ maka eliminasi piksel tersebut.



Gambar 6. Eliminating Flat Region

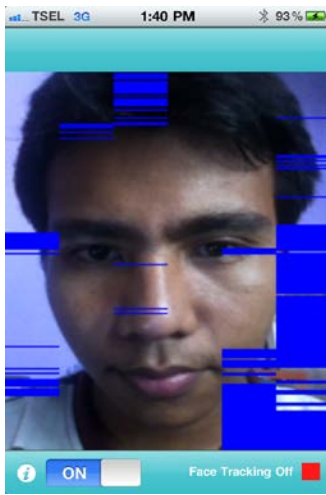
C. Proses eliminasi daerah yang memiliki fluktuasi

Pada proses ini dilakukan eliminasi pada piksel yang memiliki warna yang berulang sepanjang urutan jumlah piksel. Bagian ini tidak dapat dikatakan sebagai warna wajah karena warna wajah selalu berubah sepanjang arah tertentu.

$$\sum_{i \in n(i)} |R_i''| + |G_i''| + |B_i''| > th \Rightarrow \text{eliminate}$$

dimana $n(i)$ adalah piksel tetangga dari i dan th adalah nilai *threshold*. $|R_i''|$, $|G_i''|$, dan $|B_i''|$ adalah nilai hasil dari selisih piksel i terhadap piksel $n(i)$. Seperti yang ditunjukkan pada Gambar 7.

1. Ubah citra ke dalam bentuk *RGB Color Space*
2. Tentukan $|R_i''|$, $|G_i''|$, dan $|B_i''|$.
3. Jumlahkan $|R_i''|$, $|G_i''|$, dan $|B_i''|$.
4. Jika $|R_i''| + |G_i''| + |B_i''| > th$ maka eliminasi 30 piksel yang berurut tersebut.



Gambar 7. Eliminating Region That Have Fluctuation

D. Proses Pencarian Warna Kulit

Pada Proses ini dilakukan eliminasi pada piksel yang tidak termasuk warna kulit. Berdasarkan penelitian sebelumnya [2], warna biru pada kulit bervariasi dalam rentang yang begitu luas. Oleh sebab itu pada warna biru tidak efektif pada seluruh warna dibandingkan warna merah. Sesuai dengan penelitian [2], perbandingan warna merah-hijau pada warna kulit berubah antara 1:1 hingga 3:1. Berdasarkan penafsiran tersebut, bagian selain warna tersebut harus dieliminasi karena bukan merupakan warna kulit. Seperti yang ditunjukkan Gambar 8.



Gambar 8. Elminasi Selain Warna Kulit

E. Proses pencarian hue

Pada proses ini diasumsikan porsi wajah pada citra diatas 1/2 atau minimal 1/3 dari citra. Berdasarkan asumsi ini, dilakukan pembagian nilai Hue kedalam 40 cluster. Seperti yang ditunjukkan pada Gambar 9.

1. Lakukan pembagian 40 cluster, yaitu;

cluster 1 : 0 - 8

cluster 2 : 9 - 17

cluster 3 : 18 - 26

hingga cluster 40 : 351 - 360

2. Lakukan iterasi nilai *H Color Space* pada seluruh piksel, dan tetapkan masing - masing piksel pada rentang cluster yang sesuai
3. Lakukan iterasi pencarian cluster terbanyak yang digunakan pada 40 cluster dalam penentuan rentang nilai Hue terbanyak yang digunakan.
4. Eliminasi seluruh piksel yang tidak termasuk dalam rentang cluster tersebut.



Gambar 9. Finding Mostly Used Hue

F. Proses eliminasi keluaran noise

Proses ini merupakan proses eliminasi piksel yang terkahir, karena masih ada bagian kecil piksel yang bukan warna kulit yang tidak ikut tereliminasi. Diaplikasikan median filter untuk mengeliminasi bagian piksel ini. Langkahnya diantaranya adalah :

5	5	4
7	2	6
1	4	1

median

Pengurutan: 1 1 2 4 4 5 5 6 7

Jadi $f(x,y) = 4$ sehingga bagian dari citra ini berubah menjadi:

5	5	4
7	4	6
1	4	1

1. Urutkan nilai piksel tetangga dari piksel yang akan dilakukan median filter.
2. Cari nilai yang terletak di tengah dari piksel yang telah diurutkan.

3. Letakkan nilai piksel *median* pada piksel yang dilakukan *median filter*.

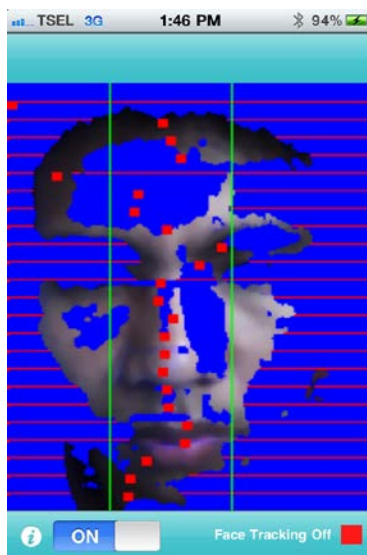
Hasil dari proses *median filter* ditunjukkan pada Gambar 10.



Gambar 10. Median Filter

G. Proses Clipping

Dalam proses ini dilakukanlah seluruh perhitungan perkiraan kordinat wajah pada sumbu x. kordinat wajah terdeteksi dengan *scanning* sepanjang *clip* garis horizontal yang telah ditentukan. Semakin banyak *clip* garis yang diberikan, semakin akurat pula perkiraan kordinat wajah.



Gambar 11. Proses Clipping

Seperti yang ditunjukkan pada Gambar 11, dilakukan *scanning* sepanjang garis - garis horizontal yang diberikan untuk mendapatkan titik perkiraan posisi wajah. Setiap titik memiliki 3 wilayah kordinat, yaitu kordinat kiri, tengah, dan kanan. Dilakukan perhitungan jumlah titik yang menempati tiap - tiap wilayah kordinat. Jumlah titik terbanyak pada suatu wilayah kordinat menjelaskan navigasi perpindahan halaman yang akan dilakukan.

Untuk mengoptimalkan kebutuhan *memory*, proses *rendering* perpindahan halaman eBook dilakukan hanya

disaat titik - titik kordinat wajah terbanyak kembali berada pada wilayah kordinat tengah.

IV. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut :

1. Perangkat *mobile* memiliki keterbatasan *memory* untuk melakukan perhitungan yang kompleks. Oleh sebab itu dibutuhkan suatu metode yang ringan dan efisien dalam teknik *Face Tracking*.
2. Teknik pengurutan proses yang baik, dapat membantu mengurangi penggunaan *memory* yang kurang efisien.
3. Metode ini masih tetap bergantung pada kondisi latar belakang lingkungan sekitar yang tertangkap oleh kamera depan perangkat *mobile*.

DAFTAR PUSTAKA

- [1] A. Bulbul. 2009. A Face Tracking Algorithm for User Interaction in Mobile Devices. Proceeding of International Conference on CyberWorlds: IEEE, 2009.
- [2] J. Brdan dan J. Mason. 2000. A Comparative Assessment of Three Approaches to Pixel-Level Human Skin Detection. Proceeding of IEEE International Conference Pattern Recognition, vol. 1, pp. 1056-1059, Sept. 2000.
- [3] M. Hunke dan A. Waibel. 1994. Face locating and tracking for human-computer interaction. Proceeding of the 28th Asilomar Conf. on Signals, System dan Computers, pages 1277-1281, 1994.
- [4] V. Vezhnevets, V. Sazanov, dan A. Danereeva. 2003. A Survey on pixel-based skin color detection techniques. Proceeding of GRAPHICON03, pp. 85-92, 2003
- [5] G. R. Bradski. 1998. Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal, 1998.
- [6] Jacobson, I, G.Booch, dan J. Rumbaugh. 1998. The Unified Software Development Process. Adisson Wesley, USA.
- [7] Kruchten, Philippe. 2000. The Rational Unified Process: An Introduction (Second Edition), E-book.
- [8] T. Sutoyo, Edy Mulyanto, Vincent Suhartono, Oky Nuhayati dan Wijanartono. 2009, Teori Pengolahan Citra Digital, Andi.
- [9] G. R Bradski, dan Andrian Keahler. 2008. Learning OpenCV Computer Vision With The OpenCV Library, O-Reilly.