APLIKASI PENGGANTI *BACKGROUND* PASFOTO DENGAN METODE L1-*METRIC* DAN *INPUT CHANNEL* RGB

Wilson Suryajaya Leoputra
wleoputra@bundamulia.ac.id
Program Studi Teknik Informatika, Universitas Bunda Mulia

ABSTRAK

Penggantian warna background pada pasfoto dengan menggunakan aplikasi pengolahan citra seperti photoshop merupakan proses yang sederhana, tapi memerlukan ketelitian dan ketrampilan yang tinggi. Beberapa pemakai aplikasi tidak mempunyai ketelitian dan ketrampilan dalam mengoperasikan pengolahan citra yang ada.

Setiap piksel citra memiliki warna. Model warna RGB memiliki warna yang merupakan variasi nilai intensitas dari ketiga channel RGB. L1-metric merupakan suatu pengukuran jarak memerlukan lebih dari satu input dari masing-masing obyek dan pembandingnya.

Dengan mempergunakan input nilai intensitas dari ketiga channel RGB yang menjadi input pada pengukuran jarak L1-metric, disusun program aplikasi pengganti warna background pas foto yang sederhana pengoperasiannya. Proses didalamnya adalah menggunakan L1-metric untuk menentukan suatu piksel termasuk background atau foreground dengan membandingkan hasil pengukuran jarak L1-metric dengan nilai input threshold.

Pengukuran jarak nilai input intensitas masing-masing channel R, G dan B. antara piksel input dengan nilai pembanding dapat diterapkan pada aplikasi pengubah warna background. Background dengan berbagai variasi kondisi dapat diganti dengan perlakuan yang sama. Serta pengoperasian yang mudah, karena tidak memerlukan keterampilan dan ketelitian yang tinggi.

Kata Kunci: *L1-metric, piksel, background, RGB*

PENDAHULUAN

Proses penggantian warna *background* dari suatu pas foto merupakan suatu proses yang sangat sering dilakukan, karena seringkali beberapa institusi mensyaratkan suatu pas foto dengan warna background tertentu.

Jika harus foto berulang kali di studio foto untuk beberapa persyaratan yang berbeda tentu akan membuat banyak biaya terbuang. Belum lagi jika persyaratan pas foto dengan warna tertentu diperlukan sangat mendesak.

Satu-satunya cara adalah dengan mengedit foto tersebut untuk mengganti backgroundnya. Sebenarnya tersedia banyak aplikasi pengolahan citra ayng mampu mengganti *background* suatu citra, sayangnya aplikasi pengolah citra yang ada seperti photoshop memerlukan keterampilan tangan yang tidak semua orang mampu melakukannya.

Berdasarkan latar belakang itulah dibuat suatu aplikasi pengganti *background* pas foto yang proses pengoperasianya sederhana tidak, memerlukan menu yang berbelit-belit, serta tidak memerlukan keterampilan tangan dalam mengolah citra yang pada kasus ini sebenarnya hanya perlu mengganti warna *bacground*nya saja.

Permasalahannya tidak semua *background* pas foto sudah dalam bentuk warna polos atau warna tunggal, tetapi seringkali ada gradasi warna dari terang ke gelap akibat adanya pencahayaan dari studio foto. Tetapi karena studio foto sekarang sudah sangat baik pencahayaannya gradasi warna itu tidak terlalu bervariasi yang dapat menimbulkan warna *background* yang sangat berbeda.

Piksel merupakan dasar penyusun citra digital. [5][6]. Pada model warna RGB, piksel citra tersusun dari tiga warna dasar, merah, hijau dan biru. [1][2]

L1-*metric* melakukan pengukuran jarak antara fitur-fitur yang dimiliki dua buah citra. Dimana jarak kedua buah citra ini yang nantinya akan dipertimbangkan sebagai kemiripan antara dua buah citra. Semakin kecil nilai jarak yang dihasilkan maka kedua citra akan dianggap semakin mirip. semakin besar nilai jarak yang dihasilkan maka kedua citra akan dianggap semakin berbeda. [3][4].

Pengukuran jarak antara dua buah citra dapat dipergunakan dengan menggunakan rumus L1-*Metric* . Rumus dari L1-*Metric* dapat dilihat pada persamaan (1)

$$d(I,H) = \sum_{l=1}^{n} | i_l - h_l | \quad \text{.....(1)}$$

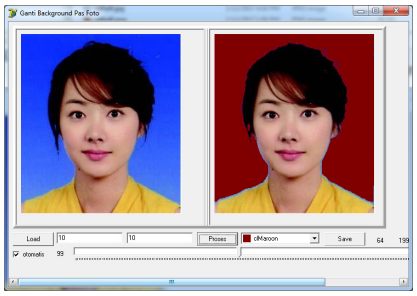
Notasi-notasi yang dipergunakan pada persamaan (1) tersebut adalah :

l Pencacah fitur
n Jumlah fitur
I Himpunan fitur citra disimpan
i fitur citra yang diuji
H Himpunan feature citra yang akan diuji
h fitur citra yang akan diuji
D(I,H) Jarak citra I terhadap citra H.

Variasi *channel* warna RGB memungkinkan diukur perbedaannya dengan menggunakan L1-*Metric*. Dimana adanya tiap-tiap input pada masing-masing piksel berupa *channel* warna RGB dengan pembandingnya berupa *input channel* RGB yang diambil dari sampel. Sehingga berdasarkan pengetahuan itu akan disusun aplikasi pengubah warna *background* ini.

PERANCANGAN PROSES

Proses penggantian *background* terdiri dari beberapa proses. Proses-proses tersebut terdiri dari beberapa proses yaitu proses pemilihan piksel, proses *scanline* dari piksel pertama dari baris pertama sampai piksel terakhir dari baris terakhir, proses penentuan piksel sebagai *background* atau sebagai *foreground* dengan menggunakan L1-*metric*, dan terakhir penggantian piksel dengan warna yang disyaratkan. Tampilan panel program dapat dilihat pada gambar 1.



Gambar 1 Tampilan Program

Secara lengkap, tampilan program terdiri dari dua panel citra, panel sebelah kiri untuk menampilkan citra asli, panel sebelah kanan untuk menampilkan pemilihan sampel *background* yang akan menggunakan *click & drag*, serta hasil penggantian *background* nantinya.

Dibagian bawah terdapat *button* ”*Load*” untuk mengambil citra, *button* ”Proses” untuk memproses serta *button* ”*Save*” untuk menyimpan citra hasil penggantian *background*.

Ke bagian bawah lagi ada *checkbox* otomatis untuk menentukan nilai *threshold* secara otomatis ketika pemilihan daerah segiempat sampel

ISSN: 1979-1496

background dilakukan, serta sebuat *trackbar threshold* untuk input nilai *threshold* jarak.
Di sebelah *button* proses terdapat sebuah comboboxColor untuk menentukan warna pengganti background. Nanti setiap piksel yang dianggap *background* akan diganti dengan warna yang dipilih dari comboBoxColor tersebut.

Input terdiri dari input citra yang akan diganti backgroundnya. Input berikutnya adalah input nilai *threshold* untuk menentukan jarak antara piksel dengan nilai rata-rata sampel terpilih. Input ini merupakan proses alternatif dari penentuan nilai threshold.
Sebagai input warna pengganti, di sebelah *button* proses terdapat sebuah comboboxColor untuk menentukan warna pengganti *background*. Nanti setiap piksel yang dianggap background akan diganti dengan warna yang dipilih dari comboBoxColor tersebut.

Pemilihan Sampel *Background*
Pemilihan sampel *background* dilakukan dengan melakukan *click & drag* pada daerah *background*. Contoh pemilihan sampel diperagakan pada gambar 2. Proses ini memanfaatkan *event-event* dari panel gambar. *Event* yang dipergunakan antara lain *event* MouseDown, *event* MouseMove dan *event* MouseUp.

Gambar 2. Pemilihan Sampel Background

Event MouseDown seperti yang diperagakan pada gambar 3. merupakan proses penentuan awal click *mouse*. Proses ini menentukan nilai awalX nila awalY yang merupakan koordinat awal click & drag. Nilai awalX dan nilai awalY diambil dari parameter x dan parameter y yang di passing ke parameter *method* dari event MouseDown. Pada saat mouse bergerak di atas suatu komponen nilai posisi koordinat x dan y dari *mouse* terhadap *desktop* ataupun komponen GUI selalu diteruskan pada atribut yang dimiliki komponen tersebut. Inilah yang dimanfaatkan oleh *event-vent mouse* baik mouseDown, mouseMove maupun mouseUp.

```
procedure TForm1.Image3MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  awalx := x;
  awaly := y;
  label1.Caption := IntToStr(awalx);
  label2.Caption := IntToStr(awaly);
  didrag := true;
end;
```

Gambar 3. *Event Mouse Up*

Event MouseMove seperti yang diperagakan pada gambar 4, merupakan proses pemberian nilai akhirX dan akhirY sebagai penyimpan posisi koordinat x dan y dari *mouse* saat tersebut. Tetapi ini bukanlah merupakan nilai posisi koordinat akhir x dan y dari proses *drag* yang sebenarnya. Ini hanya merupakan posisi koordinat akhir x dan y sementara saja sampai nanti *event* mouseUp di eksekusi. Proses ini memperagakan pula daerah *background* terpilih yang diperagakan dalam bangun persegi empat yang dibatasi oleh visualisasi garis putus-putus.

```
procedure TForm1.Image3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  if didrag then
  begin
    akhirx := x;
    akhiry := y;
    label3.Caption := IntToStr(akhirx);
    label4.Caption := IntToStr(akhiry);
    image3.Picture := image2.Picture;
    image3.Canvas.Brush.Style := bsClear;
    image3.Canvas.Pen.Style := psdot;
    image3.Canvas.Pen.Color := clblack;
    image3.Canvas.Rectangle(awalx,awaly,akhirx,akhiry);
  end;
end;
```

Gambar 4. *Event Mouse Move*

Event mouseUp merupakan proses utama dari pemilihan sampel background. *Event* ini tidak memanfaatkan passing parameter x dan y sebagai koordinat akhir dari proses drag, tetapi mempegunakan nilai akhirX dan akhirY yang didapat dari *event mouseMove* terakhir sebelum tombol click mouse diangkat. Proses ini dapat dilihat pada gambar 5.
Pada proses penentuan *threshold* otomatis ini. Nilai jarak antara nilai terendah dan tertinggi masing-masing *channel* dijumlahkan dan di bagi 3 untuk mendapatkan rata-rata jarak nilai

tertinggi dan terendah dari masing-masing *channel* R, G dan B.

Gambar 5. *Event Mouse Up*

Penentuan Nilai Rata-Rata Sampel *Background*

Proses utama diterapkan pada *event timer*, seperti yang diperagakan pada gambar 6. *event* ini dipicu oleh penekanan tombol ”proses”.

Gambar 6. *Event Timer*

Penentuan nilai rata-rata sampel *background* memanfaatkan nilai awalX, nilai awalY, nilai akhirX, dan nilai akhirY sebagai batas-batas koordinat daerah segi empat dari sampel *background*. Nilai rata-rata ini berbeda dengan nilai *threshold* secara otomatis ketika *event mouseUp* dieksekusi. Pada program yang diperagakan pada gambar 6. proses ini dapat dilihat pada baris 142 sampai baris 164.

Nilai ini merupakan nilai rata-rata intensitas dari masing-masing *channel* R, G dan B. Sehingga setiap *channel* R, G dan B memiliki nilai rata-rata intensitasnya masing-masing. Sedangkan pada proses penentuan *threshold* otomatis, yang diambil adalah rata-rata jarak dari nilai intensitas tertinggi dan terendah pada tiap-tiap *channel* R G dan B.

Nantinya nilai rata-rata ini akan digunakan pada pengukuran jarak untuk menentukan suatu piksel dianggap *foreground* atau dianggap *background.*

Pada baris 142 sampai baris 143 merupakan inisialisasi tinggi dan lebar bidang segi empat yang membingkai sampel bakground. Pada baris 145 diinisialisasi nilai TR, TG, dan TB yang nantinya akan diisi dengan nilai jarak intensitas tertinggi dan nilai intensitas terendah dari masing-masing *channel* R, G dan B.

Variabel TR merupakan penyimpan nilai jarak intensitas terendah dan nilai intensitas tertinggi dari *channel* R piksel yang terdapat pada daerah segiempat *background* yang terpilih sebagai sampel.

Variabel TG merupakan penyimpan nilai jarak intensitas terendah dan nilai intensitas tertinggi dari *channel* G piksel yang terdapat pada daerah segiempat *background* yang terpilih sebagai sampel.

Variabel TB merupakan penyimpan nilai jarak intensitas terendah dan nilai intensitas tertinggi dari *channel* B piksel yang terdapat pada daerah segiempat *background* yang terpilih sebagai sampel.

Proses pengumpulan nilai jarak ini dimulai dari baris 147 sampai baris 158. Proses ini menumpulkan nilai R, G dan B dari masing-masing piksel pada daerah segiempat yang dipilih, seniap nilai dijumlahkan ke variabel TR untuk intensitas *channel* R, TG untuk intensitas *channel* G dan TB untuk intensitas *channel* B.

Setelah didapat jumlah total variabel masing-masing variabel TR, TG dan TB, kemudian hasil penjumlahan tersebut dibagi dengan luas daerah segiempat yang didapat dari hasil perkalian tinggi x lebar segiempat. Maka didapatlah nilai tengah masing-masing sampel *channel* RGB TR, TG, dan TB. Proses ini ditunjukkan pada baris 159 sampai baris 164 dari program pada gambar 5.9.

Pada baris 159 terlihat ada fungsi percabangan IF yang dipergunakan untuk membatasi agar jika nilai nya 0 tidak mengakibatkan error program ”*divide by zero*”. Mengingat pembagian dengan nol tidak diperbolehkan pada pemrograman delphi.

Pengukuran Jarak

Proses pengukuran jarak, penentuan piksel berperan sebagai *foreground* atau *background*, serta penggantian nilai piksel dengan warna terpilih diperagakan pada program gambar 6 pada baris 165 sampai baris 176.

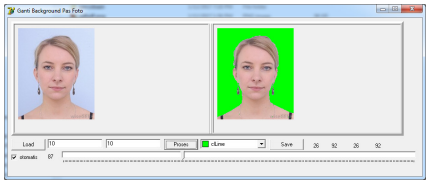
Semua piksel dari citra mulai dari piksel awal sampai piksel terakhir dari citra diambil nilai R, G dan B nya, kemudian diukur perbedaannya masing-masing intensitas *channel-channel* tersebut terhadap nilai TR, TG dan TB. Nilai jarak merupakan penjumlahan dari perbedaan nilai intensitas masing-masing *channel* R, G dan B terhadap TR, TG dan TB seperti yang ditunjukkan pada baris 172.

Penggantian Nilai piksel

Penggantian nilai piksel tidak terlepas dari proses penentuan piksel berperan sebagai *foreground* atau sebagai *background*. Jika berperan sebagai *background*, maka piksel akan diganti warnanya, tetapi jika berperan sebagai *foreground*, maka piksel akan dibiarkan apa adanya, tidak diganti warnanya.

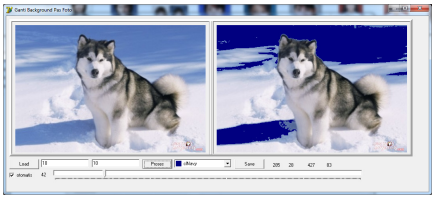
Proses penentuan piksel sebagai *foreground* atau sebagai *background* ditentukan berdasarkan nilai jarak yang didapat pada baris 172 dari program pada gambar 6. Nilai jarak ini dibandingkan dengan nilai variabel patok, yang menyimpan nilai *threshold*. Dapat dilihat pada baris 144.

Jika nilai jarak kurang dari sama dengan nilai *threshold*, maka piksel dianggap background dan diganti warnanya dengan nilai pengganti yang tercantum pada pilihan comboBoxColor. Jika nilai jarak lebih besar dari nilai threshold, maka piksel dianggap *foreground*, dan tidak akan diubah nilai warnanya.



Gambar 7, Ujicoba dengan *background* polos

Gambar 8, Uji Coba Dengan *Background* Bergradasi



Gambar 9. Uji Coba Pada *Background* Bervariasi

Gambar 10. Uji Coba Pada *Background* Bercorak Monokrom

HASIL DAN PEMBAHASAN

Ujicoba dilakukan terhadap program dengan cara menginputkan beberapa foto yang akan diubah *background*nya. Kondisi *background* foto ada yang *background*nya polos seperti pada gambar 7, bergradasi dari gelap ke terang seperti pada 8, gambar dengan *background* bervariasi lebih dari satu warna seperti pada gambar 9, serta *background* dengan corak yang monokrom seperti pada gambar 10.

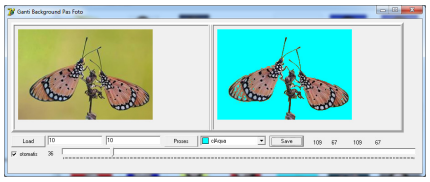
Dari hasil ujicoba hampir semua foto dapat diubah *background*nya dengan baik. Bahkan foto dengan *background* bercorak monokrom dapat diubah warnanya dengan baik. Kesulitan terjadi pada sebuah foto yang diambil dari jauh, sehingga *foreground* hampir menyatu

dengan *background* seperti yang diperagakan pada gambar 11. Pada ujicoba ini menghasilkan foto dengan *foreground* yang seakan-akan dikelilingi warna background asalnya.

Semua proses melakukan perubahan nilai *treeshold* yang sudah didapat menjadi lebih besar agar dapat menjangkau semua *background*. Serta pengambilan sampel *background* dilakukan beberapa kali karena adanya perbedaan nilai *treeshold* yang terlalu besar antara piksel-piksel pada *background*. Tetapi secara kesuluruhan proses dapat dilakukan dengan mudah karena hanya perlu mengambil daerah segiempat pada *background* yang belum terubah warnanya.

Gambar 11. Ujicoba Pada Foto Yang Diambil dari Jauh

Pengujian juga dilakukan pada sebuah citra dengan *background* yang terdiri lebih dari dua warna seperti pada gambar 12 dan berhasil. Pengujian ini dilakuakn dengan mengambil sampel *background* beberapa kali dan akhirnya semua background dapat diubah warnanya meskipun warna *background* pada beberapa daerah berbeda.



Gambar 12. Pengujian Pada *Background* Yang Lebih Dari Satu Warna

SIMPULAN DAN SARAN

Simpulan

Kesimpulan dari penelitian penggantian warna *bacground* dengan L1-*metric* ini adalah

1. L1-*metric* selain digunakan pada *computer vision*, dapat juga digunakan untuk membedakan *foreground* dengan *background*.
2. L1-*metric* dapat diterapkan pada penggantian warna *background* pada pas foto yang memerlukan proses membedakan *foreground* dengan *background*.

Saran

Berdasarkan penelitian yang telah dilakukan, maka disarankan:

1. Beberapa *background* memerlukan proses pemilihan piksel beberapa kali, disarankan ada penelitian lebih lanjut dengan menggunakan klastering untuk membagi *background* dalam beberapa klaster, sehingga pemilihan sampel *background* dapat dilakukan sesedikit mungkin.

DAFTAR PUSTAKA

[1] J, Simarmata dan T. Chandra, “Grafika Komputer”, Andi Offset, Yogyakarta, 2007.

[2] Low, Adrian., Introductory Computer Vision and Image Processing, McGraw-Hill, Berkshire, UK, 1991.

[3] Lu, Guojun., Multimedia Database Management Systems, Artech House, London, 1999.

[4] Mulyana, Teady Matius Surya., & Harjoko, Agus., “A Chinese Character Recognition Method Based On Population Matrix and Relational Database”, Proceeding of Information & Communication Technology Seminar, IEEE (ISSN 1858-1633; P518-523), 2006.

[5] Mulyana, Teady M.S., “Modul Praktikum Pengolahan Citra”, Universitas Bunda Mulia, 2016.

[6] Sutoyo, T., mulyanto, E., Suhatono V., Nurhayanti OD., Wijanarto., Teori Pengolahan Digital, Andi Offset, Yogyakarta, 2009.