

PENDETEKSIAN KELILING LUKA MENGUNAKAN METODE BORDER FOLLOWING DENGAN BANTUAN INTERPOLASI SPLINE

Pramudio¹, Muhammad Eka Suryana², Mulyono³

Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Jakarta, Jakarta Timur, Indonesia
pramudiohanggoro@gmail.com¹, eka-suryana@unj.ac.id², mulyono@unj.ac.id³

Abstrak— Luka kronis merupakan salah satu penyakit yang kompleks, khususnya bagi penyandang penyakit Diabetes Melitus (DM). Proses penyembuhan luka diawasi oleh pekerja medis dengan asesmen luka untuk memberi keputusan dalam penanganan luka. Proses penyembuhan luka perlu melakukan asesmen yang tepat dan pengelolaan yang efektif, hanya saja asesmen manual dalam pengukuran luka sangat memakan waktu dan berpotensi mengganggu penyembuhan. Penelitian ini bertujuan untuk membantu pekerja medis dalam asesmen luka kronis penggunaan metode border following dibantu dengan interpolasi spline dalam pemrosesan citra untuk mengurangi asesmen luka manual.

Kata kunci: luka kronis, border following, interpolasi spline, asesmen luka, pemrosesan citra, asesmen luka, penyembuhan luka

I. PENDAHULUAN

Kulit merupakan perlindungan pertama manusia dalam menjaga tubuhnya dari berbagai macam substansi. Apabila kulit terluka, maka diperlukan penanganan yang baik agar tidak terjadi infeksi. Luka adalah keadaan di mana fungsi anatomis kulit normal mengalami kerusakan akibat proses patologi yang berasal dari internal maupun eksternal dan mengenai organ tertentu. Ketika luka tidak terinfeksi, maka pada normalnya luka tersebut akan melakukan penyembuhan [1].

Dari beberapa kondisi luka, terdapat luka yang proses penyembuhannya tidak normal dengan durasi fase-fase yang sesuai. Kondisi tersebut disebut dengan luka kronis [2], kondisi ini dapat memiliki kaitan dengan berbagai faktor yang memperlambat penyembuhan luka seperti adanya penyakit kronis, insufisiensi vaskuler, diabetes, gangguan nutrisi, penuaan, dan berbagai faktor lokal pada luka (tekanan, infeksi, dan edema). Secara umum, luka kronis dapat terjadi akibat ulkus vena, ulkus arteri, ulkus dekubitus, dan ulkus diabetik [3].

Proses pertama dalam penanganan luka adalah melakukan pengidentifikasian terhadap luka. pengidentifikasian ini dilakukan dengan meninjau kondisi luka (dilihat dari luas, keliling, kedalaman, dan lain-lain) yang pada umumnya dilakukan secara manual, hal ini terdapat resiko menyebabkan infeksi terhadap luka [4]. Salah satu metode yang diusulkan untuk mengatasi masalah ini adalah dengan menggunakan metode yang berbasis digital image processing.

Metode berbasis *digital image processing* merupakan alternatif untuk penilaian luka karena dapat memberikan langkah-langkah yang objektif, lebih akurat, dan dapat direproduksi. Salah satu keuntungan besarnya adalah metode ini memiliki resiko yang lebih rendah karena tidak ada kontak antara luka dan sistem pengukuran [5].

Salah satu metode untuk mendeteksi luka adalah *active contour* yang sudah pernah diteliti oleh Muhammad Rizki. Namun, hasilnya masih kurang memuaskan sehingga Muhammad Rizki menyarankan untuk mengembangkan atau merubah metode yang dipakai [6].

Berdasarkan permasalahan yang telah dijelaskan, penulis akan menggunakan metode *border following* yang dibantu interpolasi spline untuk mendeteksi keliling luka. Hasil yang diharapkan adalah sistem yang mampu memisahkan antara keliling luka dengan kulit.

II. KAJIAN PUSTAKA

A. Border Following

Salah satu teknik yang sering dipakai dalam pengolahan citra adalah *Border Following*, terutama citra biner. *Border following* telah diteliti oleh beberapa orang karena teknik ini memiliki pengaplikasian yang cukup luas, seperti pengenalan objek, analisis citra, deteksi objek, dan kompresi data citra [7].

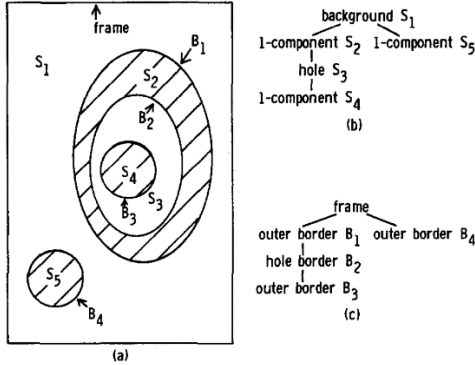
Dalam penelitian yang dibuat oleh Suzuki, ia menawarkan dua algoritma border following yang sampai saat ini masih digunakan oleh library pengolahan citra yang umum diketahui yaitu **OpenCV**. Algoritma ini merupakan salah satu algoritma yang mengenalkan pendefinisian hubungan hierarki antar tepi (border). Tidak hanya itu, algoritma ini juga membedakan tepi lubang (hole) dan tepi luar (outer border) di sebuah objek. Suzuki memberikan definisi pada algoritma ini, yaitu:

- Definisi 1 (*border point*). Sebuah piksel bernilai 1 yang dikelilingi oleh piksel bernilai 0 dapat disebut sebagai border point.
- Definisi 2 (komponen yang mengelilingi komponen lain.). Dari gambar (Gambar 2.1) dapat dikatakan bahwa komponen S_2 mengelilingi komponen S_4
- Definisi 3 (*outer border* dan *hole border*). Sebuah rangkaian border point antar komponen yang mempunyai nilai

piksel 1 (objek) dengan komponen yang mempunyai nilai piksel 0 (latar belakang)

Definisi 4 (*parent border*). Berdasarkan gambar (Gambar 2.1) dapat dikatakan bahwa tepi B_2 adalah parent border dari tepi B_3

Definisi 5 (tepiian yang mengelilingi tepiian lain). Dari gambar (Gambar 2.1) dapat dikatakan bahwa tepi B_2 mengelilingi tepi B_3 .



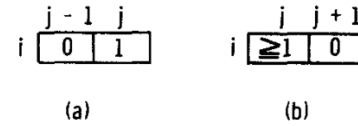
Gambar 2.1 surroundness antar komponen yang terhubung (b) dan antar tepi(borders) (c)

Diasumsikan citra yang dimasukan algoritma ini berupa citra biner. Piksel dengan nilai 0 dan 1 disebut sebagai 0-piksel dan 1-piksel. Nilai sebuah piksel dalam koordinat (i, j) yang masing-masing melambangkan baris dan kolom sebuah citra digital didefinisikan sebagai $f_{i,j}$. Baris paling pinggir yang berada di atas, kanan, bawah, dan kiri adalah frame(bingkai) dari citra tersebut. Dalam hal ini, setiap tepi baru yang ditemukan akan diberi angka unik dan disebut sebagai NBD. Asumsikan **NBD** dari *frame* adalah 1, sementara tepi lain diberi angka **NBD** secara berurutan. Nilai NBD dari parent setiap tepi di dalam variabel **LNBD** kemudian disimpan. Berikut merupakan langkah-langkah algoritma pertama border following:

Dimulai dengan pemindaian piksel citra dari kiri ke kanan hingga scanner menemukan piksel objek ($f_{i,j} \neq 0$). Dilanjutkan dengan menentukan apakah piksel tersebut merupakan tepi luar atau tepi lubang. Untuk setiap baris baru yang dipindai, kembalikan nilai **LNBD** ke 1.

Langkah 1

- jika $f_{i,j} = 1$ dan $f_{i,j-1} = 0$, maka piksel (i, j) adalah *starting point* dari operasi *border following* untuk tepi luar(*outer border*). inkremen **NBD** dan set $(i_2, j_2) \leftarrow (i, j - 1)$
- jika $f_{i,j} \geq 1$ dan $f_{i,j+1} = 0$, maka piksel (i, j) adalah *starting point* dari operasi *border following* untuk tepi lubang(*hole border*). inkremen **NBD**, set $(i_2, j_2) \leftarrow (i, j + 1)$, dan **LNBD** $\leftarrow f_{i,j}$ jika $f_{i,j} > 1$
- jika kedua kondisi tidak terpenuhi, maka lanjut ke langkah 4.



Gambar 2.2 kondisi *starting point* dari algoritma *border following* untuk tepi luar (a) dan tepi lubang (b)

Langkah 2

Tentukan parent border untuk tepi piksel (i, j) berdasarkan jenis tepi piksel dari piksel (i, j) dan **LNBD** (tepi terakhir yang ditemukan), mengikuti aturan pada tabel (Tabel 2.1)

Tabel 2.1 Aturan Penetapan *parent border*

| | Type of border B' with the sequential number of LNBD | |
|--------------|---|--------------------------------------|
| Type of B | Outer Border | Hole Border |
| Outer border | The parent border of the border B' | The border B' |
| Hole Border | The border B' | The parent border of the border B' |

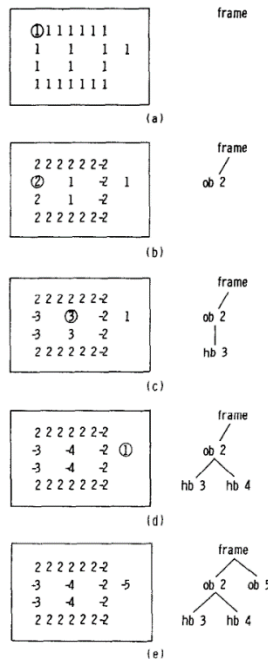
Langkah 3

Proses border following dimulai dari *starting point* piksel (i, j) pada langkah ini

- Dimulai dari piksel (i_2, j_2) , periksa piksel tetangga searah jarum jam dari piksel (i, j) sampai menemukan piksel *non-zero*. Piksel *non-zero* yang pertama ditemukan didefinisikan sebagai (i_1, j_1) . jika piksel *non-zero* tidak ditemukan, set $f_{i,j} = -\mathbf{NBD}$ dan lanjutkan ke langkah keempat
- $(i_2, j_2) \leftarrow (i_1, j_1)$ dan $(i_3, j_3) \leftarrow (i, j)$
- Mulai dari elemen setelah piksel (i_2, j_2) , mencari piksel *non-zero* di sekitar piksel (i_3, j_3) dengan arah berlawanan jarum jam dan set piksel *non-zero* pertama yang ditemukan ke dalam variabel (i_4, j_4) .
- Ubah nilai f_{i_3, j_3} dari piksel (i_3, j_3) dengan aturan menandai(*marking policy*):
 - Jika piksel (i_3, j_3+1) adalah 0-piksel (piksel bernilai 0), set $(i_3, j_3+1) \leftarrow -\mathbf{NBD}$
 - Jika piksel (i_3, j_3+1) bukan 0-piksel (piksel bernilai 0), set $(i_3, j_3+1) \leftarrow \mathbf{NBD}$
 - Jika kedua kondisi di atas tidak terpenuhi, jangan ubah nilai f_{i_3, j_3}
- Jika $(i_4, j_4) = (i, j)$ dan $(i_3, j_3) = (i_1, j_1)$ (kembali ke *starting point*), maka lanjut ke langkah 4, jika tidak set $(i_2, j_2) \leftarrow (i_3, j_3)$, $(i_3, j_3) \leftarrow (i_4, j_4)$ dan kembali ke langkah (3.c).

Langkah 4 1232

jika $f_{i,j} \neq 1$, maka **LNBD** $\leftarrow |f_{i,j}|$ dan lanjutkan pemindaian dari mulai piksel $(i, j + 1)$ untuk kembali mencari nilai piksel objek $f_{i,j} \neq 0$. Algoritma dinyatakan selesai jika pemindai sudah sampai sudut kanan bawah dari citra(piksel terakhir)



Gambar 2.3 Ilustrasi proses pemberian hierarki terhadap piksel citra; lingkaran pada piksel menunjukkan titik mula pindai *border following*

B. Splines interpolation imaging

Pada konteks pengolahan citra, Interpolasi merupakan proses pembangunan kurva atau *spline*, atau permukaan(*surface*) terdefinisi bebas yang melalui sekumpulan titik data(*data points*) secara tepat. Terkadang diberikan pembatas tambahan untuk mendapatkan interpolasi yang dibatasi agar mendapatkan hasil kurva atau *spline* yang diinginkan[8]. *Spline* yang sering dirujuk pada pengolahan citra adalah kurva yang dibentuk dengan gabungan polinomial.

1. Bézier Spline

Bézier Spline merupakan salah satu kurva polinomial menggunakan kumpulan diskrit titik kontrol untuk membuat kurva yang halus dan kontinu. *Bézier spline* dengan derajat k terdefinisi sebagai berikut

$$\mathbf{C}(u) = \sum_{i=0}^n B_{i,n}(u) \mathbf{P}_i \quad 0 \leq u \leq 1 \quad (1)$$

Fungsi dasar (*blending*), $\{B_{i,n}(u)\}$, merupakan polinomial *Bernstein* yang didefinisikan sebagai

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \quad (2)$$

2. B-Spline

Pada analisis numerik dalam matematika, B-Spline atau bisa disebut dengan basis spline merupakan fungsi spline dengan tunjangan yang sedikit dengan diberikan derajat, kehalusan, dan partisi domain. B-Spline memiliki rumus dasar yang bisa digambarkan sebagai berikut

$$\mathbf{C}(u) = \sum_{i=0}^n f_i(u) \mathbf{P}_i \quad (3)$$

\mathbf{C} menggambarkan fungsi yang membentuk kurva pada titik $\text{knot}(u)$, n titik yang dimiliki kurva, \mathbf{P}_i merupakan titik kontrol yang menentukan bentuk kurva, dan f_i adalah fungsi pembentuk kurva. f_i bisa dibentuk dengan fungsi rekursif $N_{i,p}(u)$ yang merupakan step function digambarkan sebagai berikut

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (4)$$

3. Global Interpolation

Interpolasi global merupakan interpolasi yang mengubah suku data dari *spline* secara tergeneralisir sehingga setiap pemindahan titik akan merubah keseluruhan *spline*. Interpolasi global dapat digambarkan rumusnya sebagai persamaan berikut:

$$\mathbf{Q}_k = \mathbf{C}(\bar{u}_k) = \sum_{i=0}^n N_{i,p}(\bar{u}_k) \mathbf{P}_i \quad (5)$$

$\{\mathbf{Q}_k\}$, $k = 0, \dots, n$ merupakan sekumpulan titik yang akan dibangun dengan p -derajat, knot vector (\bar{u}) merupakan nilai parameter yang akan dibangun bersama dengan knot (U).

Nilai knot vector \bar{u} akan dibangun dengan metode *chord length* dengan rumus dilakukan sebagai berikut.

$$d = \sum_{k=1}^n |\mathbf{Q}_k - \mathbf{Q}_{k-1}|$$

$$\bar{u}_0 = 0 \quad \bar{u}_n = 1 \quad (6)$$

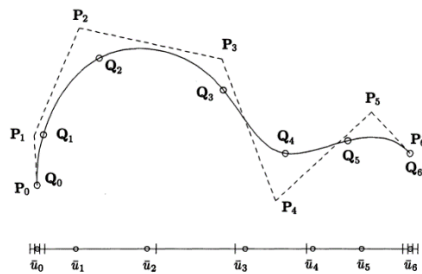
$$\bar{u}_k = \bar{u}_{k-1} + \frac{|\mathbf{Q}_k - \mathbf{Q}_{k-1}|}{d} \quad k = 1, \dots, n-1$$

Untuk menentukan U digunakan teknik rata rata sebagai berikut

$$u_0 = \dots = u_p = 0 \quad u_{m-p} = \dots = u_m = 1$$

$$u_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} \bar{u}_i \quad j = 1, \dots, n-p \quad (7)$$

Dari kumpulan rumus diatas dapat diperoleh kurva sebagai berikut



Gambar 2.4 Contoh interpolasi kurva menggunakan parameterisasi chord length dan knot vector yang diperoleh dengan rata-rata parameter.

III. METODE PENELITIAN

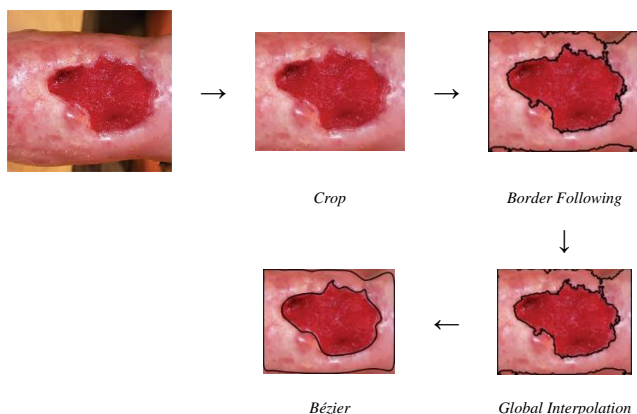
Metode penelitian akan dilakukan dengan dua tahap, yaitu menjalankan metode pengolahan citra lalu akan divalidasi hasil pengolahan citra. Penelitian ini akan menggunakan luka kronis yang dibagi menjadi 3 kategori, yaitu: luka merah, luka kuning, dan luka hitam sebagai datanya yang diambil dari penelitian luka Ns. Ratna Aryani, M,Kep[9] diambil dari data klinik yang tersedia di <https://github.com/mekas/InjuryDetection>.

Pengolahan citra dimulai dengan pengolahan data citra input Dimana peneliti akan melakukan pemotongan gambar menggunakan *GNU Image Manipulation Program*(GIMP), lalu mengolah citra tersebut dengan metode *border following* untuk mendapatkan tepi luka, apabila berhasil mendapatkan tepi luka maka akan dilanjut dengan menginterpolasi tepi luka dengan interpolasi global.

Setelah hasil didapat, data akan diolah dengan cara menghitung similiaritas luas piksel dari hasil pengolahan citra ke ground truth yang didapat dari menggambar manual tepi luka, lalu dibandingkan hasilnya dengan penelitian Rizki[6].

IV. HASIL DAN PEMBAHASAN

Pada tahap awal, peneliti mulai mengolah 69 data citra luka dengan pemotongan gambar, setelah memotong citra bisa dilanjutkan dengan menjalankan *border following*. Didapatkan 14 citra yang berhasil tergambar tepi lukanya. Dari 14 data tersebut dilakukanlah interpolasi. Proses ini digambarkan sebagai berikut



Gambar 4.1 Proses Pengolahan Citra

1. Hasil Pengolahan Citra

Training dilakukan dengan melatih tiga *classifier* pada posisi spesifik yang ditentukan pada setiap kelas. Hal ini dikarenakan untuk ketiga kelas ikan, lokasi mulut, sirip dan ekor mungkin berbeda satu dengan yang lainnya, atau karena fitur yang lebih kritis bisa didapatkan pada tempat yang berbeda. Gambar 2 menunjukkan lokasi *offset* yang digunakan pada pelatihan *classifier* ditandai oleh persegi merah.

Pengolahan citra dimulai dengan pemotongan citra lalu dideteksi dengan *border following*. Seperti dipenelitian Rizki, data citra luka hitam lebih banyak terdeteksi dibanding dengan data citra lain; *border following* mendapatkan 33,3% terdeteksi dan *active contour* 87,5%. tetapi *border following* mendapatkan hasil yang sama terhadap citra luka kuning dan luka merah dengan hasil 13,3% *active contour* mendapatkan hasil yang berbeda; luka merah 40% dan luka kuning 73,3%.

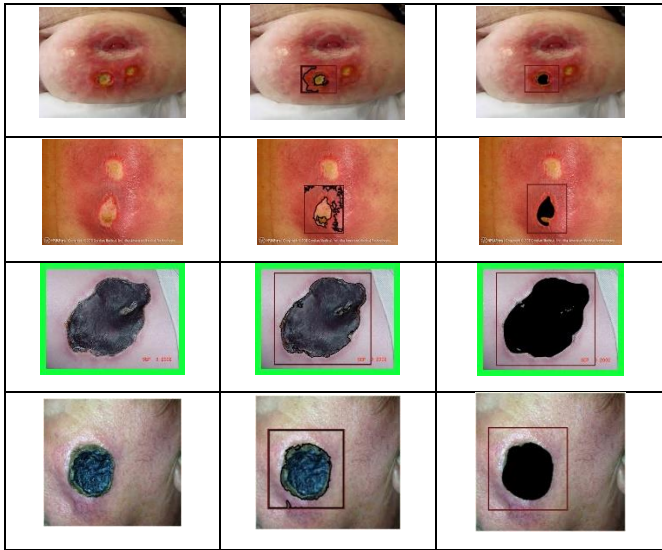
Tabel 4.1 Hasil Deteksi luka

| Data | Active Contour | Border Following |
|------|----------------|------------------|
| | | |
| | | |
| | | |
| | | |

Dari data yang berhasil terdeteksi lalu dilakukan Interpolasi terhadap garis batas sekeliling luka,

Tabel 4.2 Hasil Interpolasi Global

| Data | Active Contour | Border Following |
|------|----------------|------------------|
| | | |
| | | |



- Int = Interpolation
- Avg = Average (rata-rata)

Dilanjutkan dengan validasi luka kuning, di mana active contour mendapatkan hasil yang lebih baik tetapi border following masih mendapat hasil yang sama.

Tabel 4. 4 Hasil Validasi Luka Kuning

| Id | GT(px) | AC(px) | BF(px) | Int(px) |
|----|--------|--------|--------|---------|
| 17 | 319 | 219 | 317 | 282 |
| 25 | 2831 | 1409 | 1345 | 1291 |

| Id | AC(%) | BF(%) | Int(%) |
|-----|-------------|-------------|-------------|
| 12 | 68.65203762 | 99.37304075 | 88.40125392 |
| 18 | 49.77039915 | 47.50971388 | 45.60226069 |
| Avg | 59.21121838 | 73.44137732 | 67.0017573 |

Terakhir, dilakukan validasi terhadap luka hitam di mana active contour berhasil mendeteksi 21 dan border following mendeteksi 8 dari 24 citra.

Tabel 4.5 Hasil Validasi Luka Merah

| Id | GT(px) | AC(px) | BF(px) | Int(px) |
|----|--------|--------|--------|---------|
| 7 | 2146 | 1885 | 2061 | 1928 |
| 18 | 7328 | 6351 | 6430 | 6009 |
| 26 | 60411 | 59627 | 60923 | 59143 |
| 28 | 5408 | 5724 | 5608 | 5902 |
| 29 | 43763 | 32833 | 44984 | 43368 |
| 33 | 12899 | 10251 | 9792 | 9417 |
| 40 | 3040 | 3869 | 3069 | 2948 |
| 41 | 9644 | 10073 | 9466 | 9365 |

| Id | AC(%) | BF(%) | Int(%) |
|-----|-------------|-------------|-------------|
| 7 | 87.83783784 | 96.03914259 | 89.8415657 |
| 18 | 86.66757642 | 87.74563319 | 82.00054585 |
| 26 | 98.70222311 | 99.15247223 | 97.90104451 |
| 28 | 94.15680473 | 96.30177515 | 90.86538462 |
| 29 | 75.02456413 | 97.20997189 | 99.09741106 |
| 33 | 79.47127684 | 75.91286146 | 73.00565935 |
| 40 | 72.73026316 | 99.04605263 | 96.97368421 |
| 41 | 95.55163832 | 98.15429282 | 97.10700954 |
| Avg | 80.69443561 | 92.5807947 | 91.54594104 |

2. Hasil Validasi

Validasi dilakukan menggunakan GIMP dengan menghitung total piksel pada area deteksi dibanding dengan ground truth yang sudah disediakan di penelitian Rizki[6]. Pada proses pendeteksian, penulis curiga dengan hasil interpolasi yang mungkin similiaritas-nya lebih rendah dibanding deteksi awal, maka akan dibandingkan juga hasil deteksi sebelum dan sesudah di-interpolasikan.

Dimulai dengan data luka merah, didapatkan 4 citra yang terdeteksi, hasil nya tidak memuaskan tetapi ada 2 citra yang berhasil dideteksi oleh border following tapi tidak terdeteksi oleh active contour.

Tabel 4.3 Hasil Validasi Luka Merah

| Id | GT(px) | AC(px) | BF(px) | Int(px) |
|----|--------|--------|--------|---------|
| 12 | 29834 | - | 26565 | 26182 |
| 18 | 2788 | - | 3596 | 3337 |
| 33 | 10951 | 11038 | 11176 | 10732 |
| 44 | 2395 | 3123 | 4172 | 3966 |

| Id | AC(%) | BF(%) | Int(%) |
|-----|-------------|-------------|-------------|
| 12 | - | 89.04270296 | 87.75893276 |
| 18 | - | 71.01865136 | 80.30846485 |
| 33 | 99.205552 | 97.94539311 | 98.00018263 |
| 44 | 69.60334029 | 25.80375783 | 34.40501044 |
| Avg | 84.40444615 | 70.95262632 | 75.11814767 |

- Id = Index
- GT = Ground Truth
- AC = Active Contour (milik Rizki)
- BF = Border Following

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil eksperimen yang telah dilakukan penulis, maka didapatkan kesimpulan sebagai berikut:

1. Hasil deteksi luka dengan menggunakan border Following berhasil memindai 14 luka dari 69 luka di mana luka hitam yang paling banyak terpindai. Hasil luka yang berhasil terpindai menggunakan border following menghasilkan rata-rata akurasi 84.3%, dan setelah menggunakan interpolasi rata-rata akurasi menjadi 82.9%.
2. Apabila dibandingkan dengan active contour yang digunakan Rizki, hasil pindai border Following lebih banyak gagal dibanding berhasil tetapi memiliki akurasi yang lebih tinggi dengan rata-rata perbedaan 3.6% untuk yang hanya menggunakan border Following dan 1.3% untuk yang dibantu dengan interpolasi..

B. Saran

Adapun saran untuk penelitian selanjutnya adalah mengembangkan atau mengubah metode pendeteksian luka untuk meningkatkan keberhasilan deteksi luka dan meningkatkan akurasi deteksi luka.

DAFTAR PUSTAKA

- [1] Simon, P. E. (2018). Skin Wound Healing: Overview, Hemostasis, Inflammatory Phase.

- <https://emedicine.medscape.com/article/884594-overview>.
- [2] Landén, N. X., D. Li, and M. Ståhle (2016). "Transition from inflammation to proliferation: a critical step during wound healing". *Cellular and Molecular Life Sciences* 73.20, pp. 3861–3885.
- [3] Zhao, R. et al., (2016). "Inflammation in chronic wounds". *International journal of molecular sciences* 17.12, p. 2085.
- [4] Gupta, A. (2017). "Real time wound segmentation/management using image processing on handheld devices". *Journal of Computational Methods in Sciences and Engineering* 12.2, pp. 321–329.
- [5] Faten Abu Shmmala, W. A. (2013). "Color Based Image Segmentation using Different Versions of K-Means in two Spaces". *Global Advanced Research Journal of Engineering, Technology and Innovation* 1.9, pp. 30–41.
- [6] Rizki, M. (2022). "DETEKSI KELILING LUKA KRONIS MENGGUNAKAN ACTIVE CONTOUR (SNAKE) DAN ACTIVE CONTOUR YANG DITAMBAHKAN INTERPOLASI".
- [7] Satoshi Suzuki, K. A. (1985). "Topological structural analysis of digitized binary images by border following". *Computer Vision, Graphics, and Image Processing* 30.1, pp. 32–46.
- [8] Lockyer, P. (2006). *Controlling the Interpolation of NURBS Curves and Surfaces*. University of Birmingham.
- [9] Aryani, R. et al., (2018). "Buku Panduan: Rancang Bangun Aplikasi Mobile Android Sebagai Alat Deteksi Warna Dasar Luka Dalam Membantu Proses Pengkajian Luka Kronis dengan Nekrosis".