



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

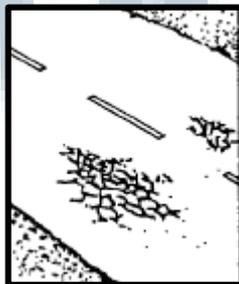
LANDASAN TEORI

2.1 Retak

Retak adalah suatu gejala kerusakan atau pecahnya permukaan perkerasan sehingga akan menyebabkan air pada permukaan perkerasan masuk ke lapisan dibawahnya dan hal ini merupakan salah satu faktor yang akan membuat luas dan parah suatu kerusakan (Andi, tt).

Jenis retak yang terjadi pada lapisan permukaan jalan dapat dibedakan atas empat jenis. Oleh karena pada pembatasan masalah hanya tiga jenis dari sembilan jenis retak yang masuk objek penelitian, maka ketiga jenis retak tersebut sebagai berikut (Andi, tt):

1. Retak kulit buaya (*alligator cracks*), lebar celah lebih besar atau sama dengan 3 mm. Saling merangkai membentuk serangkaian kotak-kotak kecil yang menyerupai kulit buaya. Retak ini disebabkan oleh bahan perkerasan yang kurang baik, pelapukan permukaan, tanah dasar atau bagian perkerasan di bawah lapisan permukaan kurang stabil atau bahan lapis pondasi dalam keadaan jenuh air (air tanah baik).



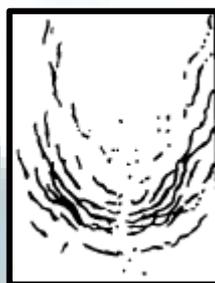
Gambar 2.1 Retak kulit buaya

2. Retak susut (*shrinkage cracks*), retak yang saling berhubungan membentuk kotak-kotak besar dengan sudut tajam. Retak disebabkan oleh perubahan *volume* pada lapisan permukaan yang memakai aspal dengan penetrasi rendah, atau perubahan *volume* pada lapisan pondasi dan tanah dasar.



Gambar 2.2 Retak susut

3. Retak selip (*slippage cracks*), retak yang bentuknya melengkung seperti bulan sabit. Hal ini terjadi disebabkan oleh kurang baiknya ikatan antara lapis permukaan dengan lapis di bawahnya. Kurang baiknya ikatan dapat disebabkan oleh adanya debu, minyak, air, atau benda non-adhesif lainnya, atau akibat tidak diberinya *tack coat* sebagai bahan pengikat di antara kedua lapisan.



Gambar 2.3 Retak selip

2.2 Image Processing

Istilah citra digital sangat populer pada masa sekarang (Abdul, 2013). Banyak peralatan elektronik misalnya, *scanner*, kamera digital, mikroskop digital, dan *fingerprint reader* (pembaca sidik jari), yang menghasilkan citra digital. Perangkat lunak untuk mengolah citra digital juga sangat populer digunakan oleh pengguna untuk mengolah foto atau untuk berbagai keperluan lain. Sebagai contoh, *Adobe Photoshop* dan *GIMP (GNU Image Manipulation Program)* menyajikan berbagai fitur untuk memanipulasi citra digital.

Secara umum menurut istilah pengolahan citra digital menyatakan “pemrosesan gambar berdimensi-dua melalui komputer digital” (Jain, 1989). Pengolahan citra adalah istilah umum untuk berbagai teknik yang keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara (Efford, 2000). Foto adalah contoh gambar berdimensi-dua yang dapat diolah dengan mudah. Setiap foto dalam bentuk citra digital (misalnya berasal dari kamera digital) dapat diolah melalui perangkat lunak tertentu. Sebagai contoh, apabila hasil bidikan kamera terlihat agak gelap, citra dapat diolah menjadi lebih terang. Dimungkinkan pula untuk memisahkan foto orang dari latar belakangnya. Gambaran tersebut menunjukkan hal sederhana yang dapat dilakukan melalui pengolahan citra digital. Tentu saja, banyak hal pelik lain yang dapat dilakukan melalui pengolahan citra digital.

Sebuah citra mengacu kepada fungsi intensitas dalam bidang dua dimensi yaitu $f(x,y)$. Fungsi ini adalah representasi matematis dari citra, dimana $f(x,y)$ adalah tingkat kecerahan (*brightness*) pada koordinat spasial (x,y) . Pemrosesan citra menggunakan komputer membutuhkan citra digital sebagai input. Yang dimaksud

dengan citra digital adalah sebuah citra dengan fungsi (x,y) yang nilai kecerahan maupun posisi koordinatnya telah didiskritkan, sehingga nilainya bukan merupakan nilai yang tak berhingga, melainkan berada di dalam jangkauan tertentu. Sebuah citra digital direpresentasikan sebagai sebuah matriks, dimana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya (piksel) menyatakan tingkat kecerahan pada titik tersebut.

2.3 Thresholding

Citra pada umumnya terdiri dari piksel-piksel dengan variasi nilai RGB (komponen warna merah, hijau, dan biru). Citra jenis ini disebut sebagai citra berwarna. Citra abu-abu (*grayscale*) adalah citra dengan satu macam warna yakni abu-abu, dengan variasi derajat keabuan (Abdul, 2013). Proses mengubah citra warna menjadi *grayscale* digunakan *image processing* untuk menyederhanakan elemen matriks dari suatu citra. Citra berwarna terdiri dari 3 layer matriks yaitu *R-layer*, *G-layer*, *B-layer*. Untuk mengubah citra berwarna menjadi citra abu-abu maka 3 *layer* tersebut diubah menjadi 1 *layer* matrik *grayscale*. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan.

Thresholding merupakan proses pemisahan piksel-piksel berdasarkan derajat keabuan yang dimilikinya (Abdul, 2013). Piksel yang memiliki derajat keabuan lebih kecil dari nilai batas yang ditentukan akan diberikan nilai 0, sementara piksel yang memiliki derajat keabuan yang lebih besar dari batas akan diubah menjadi bernilai 1. Metode ini digunakan untuk mengkonversi data *image* menjadi data biner dengan tujuan agar proses selanjutnya menjadi lebih mudah.

2.4 Edge Detection

Deteksi tepi berfungsi untuk memperoleh tepi objek. Deteksi tepi memanfaatkan perubahan nilai intensitas yang drastis pada batas dua area. Definisi tepi di sini adalah “himpunan piksel yang terhubung yang terletak pada batas dua area” (Gonzalez & Woods, 2002). Perlu diketahui, tepi sesungguhnya mengandung informasi sangat penting. Informasi yang diperoleh dapat berupa bentuk maupun ukuran objek.

Umumnya, deteksi tepi menggunakan dua macam detektor, yaitu detektor baris (H_y) dan detektor kolom (H_x). Beberapa contoh yang tergolong jenis ini adalah *operator* Roberts, Prewitt, Sobel, dan Frei-Chen.

Deteksi tepi dapat terbagi menjadi dua golongan. Golongan pertama disebut deteksi tepi orde pertama, yang bekerja dengan menggunakan turunan atau diferensial *order* pertama. Termasuk kelompok ini adalah *operator* Roberts, Prewitt, dan Sobel. Golongan kedua dinamakan deteksi tepi orde kedua, yang menggunakan turunan orde kedua. Contoh yang termasuk kelompok ini adalah *Laplacian of Gaussian* (LoG).

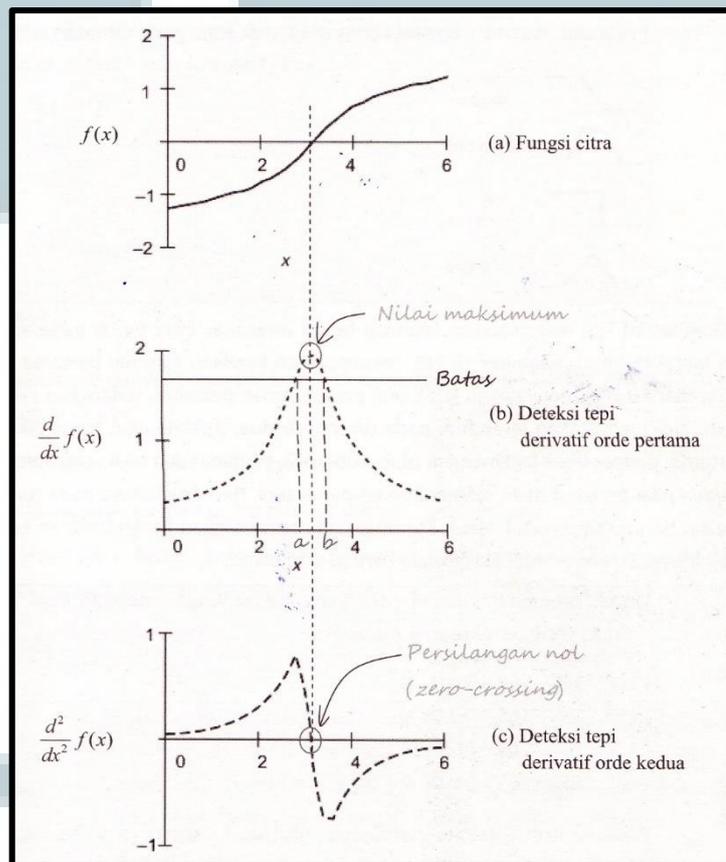
Tabel memberikan definisi turunan orde pertama dan kedua baik pada bentuk kontinu maupun diskret. Bentuk diskret sangat berguna untuk melakukan deteksi tepi orde pertama dan orde kedua. Perlu diketahui, terkait dengan turunan, tepi sesungguhnya terletak pada:

1. Nilai absolut maksimum pada turunan pertama.
2. Persilangan nol (*zero-crossing*) pada turunan kedua.

Tabel 2.1 Turunan orde pertama dan kedua pada bentuk kontinu dan diskret

(Abdul, Susanto, 2013)

Turunan	Bentuk kontinu	Bentuk diskret
$\frac{df}{dx}$	$\lim_{\Delta x \rightarrow 0} \frac{f(y, x + \Delta x) - f(y, x)}{\Delta x}$	$f(y, x+1) - f(y, x)$
$\frac{df}{dy}$	$\lim_{\Delta y \rightarrow 0} \frac{f(y + \Delta y, x) - f(y, x)}{\Delta y}$	$f(y+1, x) - f(y, x)$
$\nabla^2 f(y, x)$	$\left[\frac{df}{dy}, \frac{df}{dx} \right]$	$[f(y, x+1) - f(y, x), f(y+1, x) - f(y, x)]$
$\frac{d^2 f}{dx^2}$	$\lim_{\Delta x \rightarrow 0} \frac{\left(\frac{df}{dx} \right)(y, x + \Delta x) - \left(\frac{df}{dx} \right)(y, x)}{\Delta x}$	$f(y, x+1) - 2f(y, x) + f(y, x-1)$
$\frac{d^2 f}{dy^2}$	$\lim_{\Delta y \rightarrow 0} \frac{\left(\frac{df}{dy} \right)(y + \Delta y, x) - \left(\frac{df}{dy} \right)(y, x)}{\Delta y}$	$f(y+1, x) - 2f(y, x) + f(y-1, x)$
$\nabla^2 f(y, x)$	$\frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2}$	$f(y, x+1) + f(y, x-1) - 4f(y, x) + f(y+1, x) + f(y-1, x)$



Gambar 2.4 Deteksi tepi orde pertama dan orde kedua pada arah x (Abdul,

Susanto, 2013)

Contoh di gambar 2.5(a) menunjukkan fungsi intensitas citra $f(y, x)$ pada arah x dengan bentuk tepi tanjakan landai. Gambar 2.5(b) menunjukkan keadaan turunan pertama pada arah x . Puncak di gambar 2.5(b) menyatakan letak tepi pada turunan pertama, sedangkan persilangan nol di gambar 2.5(c) menyatakan letak tepi pada turunan kedua. Apabila nilai batas dikarenakan pada turunan pertama, puncak tidak lagi menjadi tepi. Akibatnya, terdapat dua nilai yang memenuhi (yaitu a dan b). Kedua nilai tersebut akan menjadi piksel-piksel tepi. Berbeda halnya pada turunan kedua, tepi akan selalu berupa satu piksel. Hal itu terlihat pada perpotongan fungsi turunan kedua dengan sumbu x . Akibatnya, ketebalan tepi akan selalu berupa satu piksel.

Deteksi tepi dengan turunan orde pertama dilakukan dengan menggunakan operator gradien. Operator gradien didefinisikan sebagai vektor

$$v_y = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix} \dots\dots\dots \text{Rumus 2.1}$$

Besaran vektor dihitung menggunakan rumus

$$\nabla f = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2} = \left[\left(\frac{df}{dx}\right)^2 + \left(\frac{df}{dy}\right)^2 \right]^{1/2} \dots\dots\dots \text{Rumus 2.2}$$

Namun untuk alasan penyederhanaan komputasi, operasi akar ditiadakan sehingga besaran vektor tersebut dihipotesis melalui

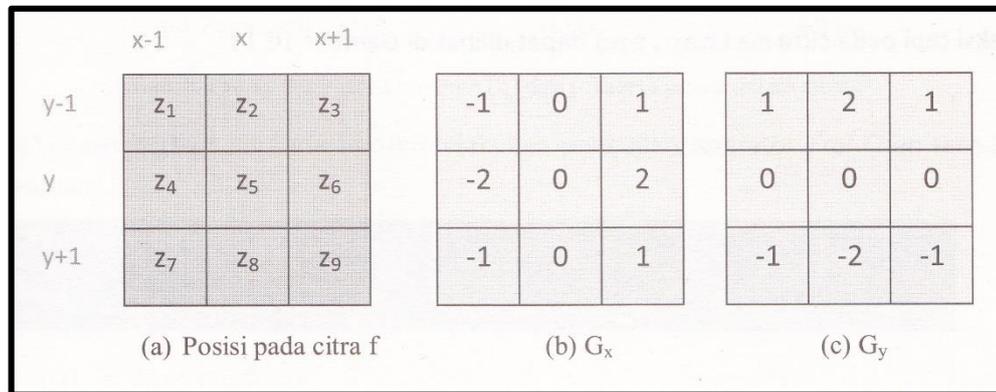
$$\nabla f \approx G_x^2 + G_y^2 \dots\dots\dots \text{Rumus 2.3}$$

Perlu diketahui, besaran gradien sering disebut sebagai “gradien” saja.

Adapun turunan orde kedua yang biasa digunakan dalam pengolahan citra dihitung dengan menggunakan Laplacian. Perhitungannya seperti berikut:

$$\nabla^2 f(y, x) = \frac{d^2 f(y, x)}{dx^2} + \frac{d^2 f(y, x)}{dy^2} \dots\dots\dots \text{Rumus 2.4}$$

Operator Sobel dapat dilihat di gambar 2.6. *Operator Sobel* lebih sensitif terhadap tepi diagonal daripada tepi vertikal dan horizontal. Hal ini berbeda dengan *operator Prewitt*, yang lebih sensitif terhadap tepi vertikal dan horizontal.



Gambar 2.5 Operator Sobel (b) dan (c) serta posisi dalam citra f (Abdul, Susanto, 2013)

2.5 Template Matching

Metode *template matching* adalah salah satu teknik dalam pengolahan citra digital yang berfungsi untuk mencocokkan tiap-tiap bagian dari suatu citra dengan citra yang menjadi *template*. Teknik ini banyak digunakan dalam bidang industri sebagai bagian dari *quality control*.

Prinsip metode ini adalah membandingkan antara citra *streaming* yang akan dikenali dengan citra *template*. Citra objek yang akan dikenali mempunyai tingkat kemiripan sendiri terhadap masing-masing citra *template*. Pengenalan dilakukan dengan melihat nilai tingkat kemiripan tertinggi dan nilai batas ambang pengenalan dari citra objek tersebut. Bila nilai tingkat kemiripan berada di bawah nilai batas ambang maka citra objek tersebut dikategorikan sebagai objek tidak dikenal.

Pada prinsipnya metode *template matching* memiliki karakteristik antara lain:

1. Relatif mudah untuk diaplikasikan dalam teknik pengolahan citra digital.
2. Hasilnya relatif sangat akurat karena mendeteksi kesalahan hingga ukuran piksel.
3. Metode ini cukup rentan terhadap perbedaan orientasi antara citra *template* dengan citra yang diidentifikasi, yang meliputi: ukuran, posisi dan kualitas citra.
4. Untuk mendapatkan hasil yang maksimal maka metode ini sangat tergantung pada teknik pengolahan citra digital yang lain seperti *enhancement*, *color filtering*, dan lain-lain.

Rumus yang digunakan dalam *template matching* adalah *Sum of Absolute Difference*.

$$D(t) = \sum_{(i,j) \in W} |I_1(i,j) - I_2(i+x, j+y)| \dots \dots \dots \text{Rumus 2.5}$$

Sum of Absolute Difference adalah algoritma yang banyak digunakan untuk mengukur kesamaan antar blok gambar. Algoritma ini bekerja dengan mengambil nilai mutlak untuk perbedaan setiap piksel di blok asli dengan piksel yang sesuai di blok yang digunakan untuk perbandingan.

Proses ini dilakukan dari piksel per piksel dari kedua gambar. Jika $D(t) = 0$ maka dapat disimpulkan bahwa bagian blok tersebutlah yang memiliki kemiripan. Dengan metode ini tingkat akurasi sempurna karena dilakukan operasi piksel per piksel. Kelemahan dari operasi ini adalah jika salah satu gambar mengalami modifikasi baik bentuk maupun komposisi warna, tidak dapat ditemukan nilai $D(t) = 0$. Sebagai pendekatan yang realistis adalah gambar perlu diubah ke mode

grayscale. Jadi dapat disimpulkan semakin kecil SAD yang diperoleh, semakin cocok *template* yang dibandingkan terhadap citra objek.

2.6 Android

Android adalah sistem operasi yang berbasis Linux untuk *smartphone* dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai macam gadget. Awalnya, Google Inc membeli Android Inc, pendatang baru yang membuat piranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan piranti keras, piranti lunak dan telekomunikasi, termasuk Google, HT, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia (Tri, 2013).

Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Fitur-fitur Android antara lain adalah:

1. *Framework* aplikasi, memungkinkan daur ulang dan penggantian komponen.
2. *Browser* terintegrasi berbasis *engine* Open Source WebKit.
3. Rancangan *handset platform* disesuaikan dengan kebutuhan VGA (*Video Graphic Adapter*) yang lebih besar, *library* grafik 2D dan 3D yang berdasarkan spesifikasi OpenGL ES 1.0 serta *layout smartphone* yang tradisional.
4. *Multi-touch*.

5. Dukungan *hardware* tambahan. Android mendukung penggunaan kamera, layar sentuh, GPS (*Global Positioning System*), pengukur kecepatan, *magnetometer*, akselerasi 2D *bit blits* (dengan orientasi *hardware*, *scaling*, konversi format piksel) dan akselerasi grafis 3D.

