



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 atau parah suatu kerusakan (Tenriajeng, tt).

Di dalam pendekatan mekanika retak, diasumsikan ada bagian yang lemah pada setiap material. Ketika pembebanan terjadi, ada konsentrasi tegangan yang lebih tinggi di sekitar bagian tersebut, sehingga material tersebut tidak lagi memiliki distribusi tegangan yang seragam dan terjadilah kerusakan atau retak pada bagian tersebut dan berkembang ke bagian yang lainnya. Mekanika retak juga menggambarkan perkembangan retak tergantung pada sifat material tersebut (Tenriajeng, tt).

Kerusakan yang terjadi pada perkerasan konstruksi jalan dapat disebabkan oleh :

1. Lalu lintas yang dapat berupa peningkatan beban dan repetisi beban.
2. Air yang dapat berasal dari air hujan, sistem drainase jalan yang tidak baik, naiknya air dengan sifat kapilaritas.
3. Material konstruksi perkerasan. Dalam hal ini dapat disebabkan oleh sifat material itu sendiri atau dapat pula disebabkan oleh sistem pengolahan yang tidak baik.

4. Iklim. Indonesia beriklim tropis, dimana suhu udara dan curah hujan umumnya tinggi, yang dapat merupakan salah satu penyebab kerusakan jalan.
5. Kondisi tanah dasar yang tidak stabil. Kemungkinan disebabkan oleh *system* pelaksanaan yang kurang baik, atau dapat juga disebabkan oleh sifat tanah dasar yang memang kurang bagus.
6. Proses pemadatan di atas lapisan tanah dasar yang kurang baik.

Umumnya kerusakan-kerusakan yang timbul tidak disebabkan oleh satu faktor saja, tetapi dapat merupakan gabungan dari penyebab yang saling kait-mengait. Dalam mengevaluasi kerusakan jalan perlu ditentukan:

1. Jenis kerusakan (*distress type*) dan penyebabnya.
2. Tingkat kerusakan (*distress severity*)
3. Jumlah kerusakan (*distress amount*)

Jenis-jenis retak (*cracking*) dan penyebabnya terbagi menjadi sembilan jenis. Oleh karena pada pembatasan masalah hanya menggunakan empat dari sembilan jenis retakan yang termasuk objek penelitian, maka keempat jenis retak tersebut sebagai berikut.

1. Retak halus (*hair cracking*)

Lebar celah lebih kecil atau sama dengan 3 mm, penyebab adalah bahan perkerasan yang kurang baik, tanah dasar atau bagian perkerasan di bawah lapis permukaan. Untuk pemeliharaan dapat dipergunakan lapis *latasir*, atau buras. Dalam tahap perbaikan sebaiknya dilengkapi dengan perbaikan *system* drainase. Retak halus dapat berkembang menjadi retak kulit buaya.

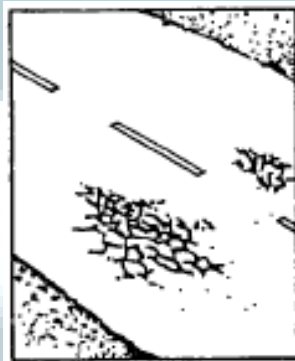


Gambar 2.1 Retak Halus

Sumber : Rekayasa Jalan Raya 2 (Tenriajeng, tt)

2. 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 lapis permukaan kurang stabil, atau bahan lapis pondasi dalam keadaan jenuh air (air tanah baik).



Gambar 2.2 Retak Kulit Buaya

Sumber : Rekayasa Jalan Raya 2 (Tenriajeng, tt)

Umumnya daerah dimana terjadi retak kulit buaya tidak luas. Jika daerah dimana terjadi retak kulit buaya luas, mungkin hal ini disebabkan oleh repetisi beban lalu lintas yang melampaui beban yang dapat dipikul oleh lapisan

permukaan tersebut. Retak kulit buaya untuk sementara dapat diperbaiki dengan mempergunakan lapis burda, burtu, ataupun lataston, jika celah kurang lebih 3 mm. Sebaiknya bagian perkerasan yang telah mengalami retak kulit buaya akibat air yang merembes masuk ke lapis pondasi dan tanah dasar diperbaiki dengan cara dibongkar dan membuang bagian-bagian yang basah, kemudian dilapis kembali dengan bahan yang sesuai. Perbaikan harus disertai dengan perbaikan drainase di sekitarnya. Kerusakan yang disebabkan oleh beban lalu lintas harus diperbaiki dengan memberi lapis tambahan. Retak kulit buaya dapat diresapi oleh air sehingga lama kelamaan akan menimbulkan lubang-lubang.

3. Retak susut (*shrinkage cracks*)

Retak yang saling bersambungan 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. Perbaikan dapat dilakukan dengan mengisi celah dengan campuran aspal dan pasir melapisi dengan burtu.



Gambar 2.3 Retak susut

Sumber : Rekayasa Jalan Raya 2 (Tenriajeng, tt)

4. 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 akibar tidak diberikan *tack coat* sebagai bahan pengikat di antara kedua lapisan. Retak selip pun terjadi akibat terlalu banyaknya pasir dalam campuran lapisan permukaan, atau kurang baiknya pemadatan lapis permukaan. Perbaikan dapat dilakukan dengan membongkar bagian yang rusak dan menggantikannya dengan lapisan yang lebih baik.



Gambar 2.4 Retak selip

Sumber : Rekayasa Jalan Raya 2 (Tenriajeng, tt)

2.2 *Image Processing*

Pengolahan citra (*Image Processing*) adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Pengolahan citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra, namun citra keluaran mempunyai kualitas lebih baik

daripada citra masukan. Termasuk dalam bidang ini juga adalah pemampatan citra (*image compression*). Perubahan kontras citra adalah contoh operasi pengolahan citra. Contoh pengolahan citra lainnya adalah penghilangan *noise*.

Umumnya, operasi-operasi pada pengolahan citra diterapkan pada citra bila:

1. Perbaikan atau memodifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung di dalam citra.
2. Elemen di dalam citra perlu dikelompokkan, dicocokkan, atau diukur. Sebagian citra perlu digabung dengan bagian citra yang lain.

2.3 Thresholding

Thresholding adalah metode paling sederhana dari segmentasi citra. Dari citra *grayscale*, *thresholding* dapat digunakan untuk membentuk citra biner. Sebuah citra biner adalah sebuah citra digital yang hanya memiliki dua kemungkinan nilai untuk tiap *pixel*. Kedua warna tersebut adalah hitam dan putih. Warna yang digunakan untuk objek dalam citra adalah warna *foreground*, sedangkan sisa dari citra adalah warna *background*.

Secara umum, proses *thresholding grayscale* untuk menghasilkan citra biner adalah sebagai berikut:

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) < T \end{cases} \quad \dots \text{Rumus 2.1}$$

Sumber : Pengolahan Citra (Kadir, 2013)

Dengan $g(x,y)$ citra biner dari citra *grayscale* $f(x,y)$, dan T menyatakan nilai ambang. Nilai T memegang peran sangat penting dalam proses *thresholding*. Kualitas hasil citra biner sangat bergantung pada nilai T yang digunakan. Dengan

operasi pengambangan tersebut, objek dibuat berwarna terang (1 atau putih), sedangkan latar berwarna gelap (0 atau hitam).

2.4 Adaptive Threshold Sauvola

Pengambangan adaptif (*Adaptive thresholding*) merupakan pengambangan yang menggunakan nilai ambang lokal, yang dihitung secara adaptif berdasarkan statistika piksel-piksel tetangga. Hal ini didasarkan kenyataan bahwa bagian-bagian kecil dalam citra mempunyai iluminasi yang sama, sehingga lebih tepat kalau nilai ambang (*threshold*) dihitung berdasarkan bagian-bagian kecil dalam citra dan bukan berdasarkan seluruh piksel dalam citra (Kadir dkk, 2013).

Suatu citra dibagi menjadi blok-blok kecil dan kemudian dilakukan *adaptive thresholding* atas setiap blok itu dengan nilai *threshold* yang berbeda. Sebagai contoh, pengambangan dilakukan terhadap daerah citra yang berukuran 3x3 atau 5x5 piksel. Nilai *threshold* ditentukan sebagai fungsi rata-rata derajat keabuan di dalam daerah citra tersebut. Intensitas piksel yang berbeda secara signifikan dari nilai rata-rata tersebut dianggap mengandung informasi kontras dan ini harus dipertahankan dalam citra biner. Dengan *adaptive thresholding*, secara subjektif citra biner yang dihasilkan terlihat lebih bagus dan sedikit informasi yang hilang.

Pada metode binerisasi *sauvola*, nilai *threshold* dikomputasi menggunakan *mean* dan *standard deviation* dari intensitas *pixel* dalam $w*w$ ukuran *window* yang berada di sekitar *pixel*. Metode binerisasi *sauvola* merupakan variasi dari metode *niblack*. Perhitungan nilai *threshold* pada metode *sauvola* seperti rumus 2.2.

$$p(x, y) = \text{mean} * (1 + k * (\frac{\text{std}}{R} - 1)) \quad \dots \text{Rumus 2.2}$$

Sumber : fiji.sc/Auto_Local_Threshold

Keterangan :

- k = *weighting factor*
- std = standar deviasi
- R = nilai konstan 128

2.5 Edge Detection

Edge Detection berfungsi untuk memperoleh tepi objek. *Edge Detection* 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, *edge detection* menggunakan dua macam *detector*, yaitu *detector* baris (Hy) dan *detector* kolom (Hx). Beberapa contoh yang tergolong jenis ini adalah operator *Roberts*, *Prewitt*, *Sobel*, dan *Frei-Chen* (Kadir, 2013).

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).

2.6 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan merupakan sebuah model seperti jaringan syaraf pada manusia, yang dapat digunakan untuk pengenalan pola dan pembelajaran mesin (Lejap, 2008). Jaringan syaraf tiruan adalah sekumpulan sel *neuron* saling terhubung secara sederhana (Lagudu & Sarma, 2013). Sel-sel *neuron* yang ada diatur sehingga setiap sel dapat mengirimkan masukan dari satu sel ke sel lainnya. Keluaran yang dihasilkan dari jaringan ini didapatkan dari masukan yang disebarkan ke seluruh jaringan dari sel-sel yang aktif (Lagudu & Sarma, 2013).

Jaringan syaraf tiruan ditentukan oleh tiga hal, yaitu (Siang, 2004):

1. Pola hubungan antar *neuron* (Arsitektur jaringan)
2. Metode untuk menentukan bobot penghubung (Metode pelatihan)
3. Fungsi aktivasi

2.6.1 Pelatihan Standar *Backpropagation*

Pelatihan *Backpropagation* meliputi tiga fase (Siang, 2004). Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, mulai dari haris yang berhubungan langsung dengan unit-unit di layar keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi (Siang, 2004). Berikut merupakan penjelasan dari fase-fase tersebut:

1. Propagasi maju

Selama propagasi maju, sinyal masukan dipropagasikan ke layar tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit layar tersembunyi tersebut selanjutnya dipropagasikan maju lagi ke layar tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan. Berikutnya, keluaran jaringan dibandingkan dengan target yang harus dicapai. Selisih antara *target value* dan *output node value* adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap baris dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi (Siang, 2004).

2. Propagasi mundur

Berdasarkan kesalahan $t_k - y_k$, dihitung faktor ∂_k ($k = 1, 2, \dots, m$) yang dipakai untuk mendistribusikan kesalahan di unit y_k ke semua unit tersembunyi yang terhubung langsung dengan y_k . ∂_k juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran.

Dengan cara yang sama, dihitung ∂_j di setiap unit di layar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga semua faktor ∂ di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung (Siang, 2004).

3. Perubahan bobot

Setelah semua faktor ∂ dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor ∂ *neuron* di layar atasnya.

Ketiga fase tersebut diulang-ulang terus hingga kondisi pemberhentian terpenuhi (Siang, 2004). Untuk setiap pengulangan dalam pelatihan jaringan syaraf tiruan disebut *epoch*. Umumnya kondisi pemberhentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diizinkan (Siang, 2004).

2.6.2 Pemilihan Bobot dan Bias Awal

Bobot awal akan mempengaruhi apakah jaringan mencapai titik minimum lokal atau global, dan seberapa cepat konvergensinya (Siang, 2004).

Bobot yang menghasilkan nilai turunan aktivasi yang kecil sedapat mungkin dihindari karena akan menyebabkan perubahan bobotnya menjadi sangat kecil. Demikian pula nilai bobot awal tidak boleh terlalu besar karena nilai turunan fungsi aktivasinya menjadi sangat kecil juga (Siang, 2004). Oleh karena itu dalam standar *Backpropagation*, bobot dan bias diisi dengan bilangan acak kecil.

Nguyen dan Widrow mengusulkan cara membuat inisialisasi bobot dan bias ke unit tersembunyi, sehingga menghasilkan iterasi lebih cepat.

Misalnya, n adalah jumlah unit masukan, p adalah jumlah unit tersembunyi, dan β adalah faktor skala yang berupa $0.7^n \sqrt{p}$. Algoritma inisialisasi Nguyen Widrow adalah sebagai berikut (Siang, 2004):

1. Inisialisasi semua bobot (v_{ji} (lama)) dengan bilangan acak dalam interval $[-0.5, 0.5]$.

2. Hitung $\|v_j\| = \sqrt{v_{j1}^2 + v_{j2}^2 + \dots + v_{jn}^2}$...Rumus 2.3

3. Bobot yang dipakai sebagai inisialisasi = $v_{ji} = \frac{\beta v_{ji}(\text{lama})}{\|v_j\|}$...Rumus 2.4
4. Bias yang dipakai sebagai inisialisasi = $v_{j0} =$ bilangan acak antara $-\beta$ dan β .

2.6.3 Fungsi Aktivasi

Dalam jaringan syaraf tiruan, fungsi aktivasi dipakai untuk menentukan keluaran suatu neuron (Siang, 2004). Beberapa fungsi aktivasi yang sering dipakai adalah sebagai berikut:

1. Fungsi *threshold* (batas ambang)

$$f(x) = \begin{cases} 1 & \text{jika } x \geq a \\ 0 & \text{jika } x < a \end{cases} \quad \dots \text{Rumus 2.5}$$

$$f(x) = \begin{cases} 1 & \text{jika } x \geq a \\ -1 & \text{jika } x < a \end{cases} \quad \dots \text{Rumus 2.6}$$

Untuk beberapa kasus, fungsi *threshold* yang dibuat tidak berharga 0 atau 1, tapi berharga -1 atau 1 (sering disebut *threshold bipolar*) (Siang, 2004).

2. Fungsi sigmoid

$$f(x) = \frac{1}{1 + e^{-x}} \quad \dots \text{Rumus 2.7}$$

$$f'(x) = f(x)(1 - f(x)) \quad \dots \text{Rumus 2.8}$$

Fungsi sigmoid sering dipakai karena nilai fungsinya yang terletak antara 0 dan 1 dan dapat diturunkan dengan mudah (Siang, 2004).

3. Fungsi identitas

$$f(x) = x \quad \dots \text{Rumus 2.9}$$

Fungsi identitas sering dipakai apabila kita menginginkan keluaran jaringan berupa sembarang bilangan riil (Siang, 2004).