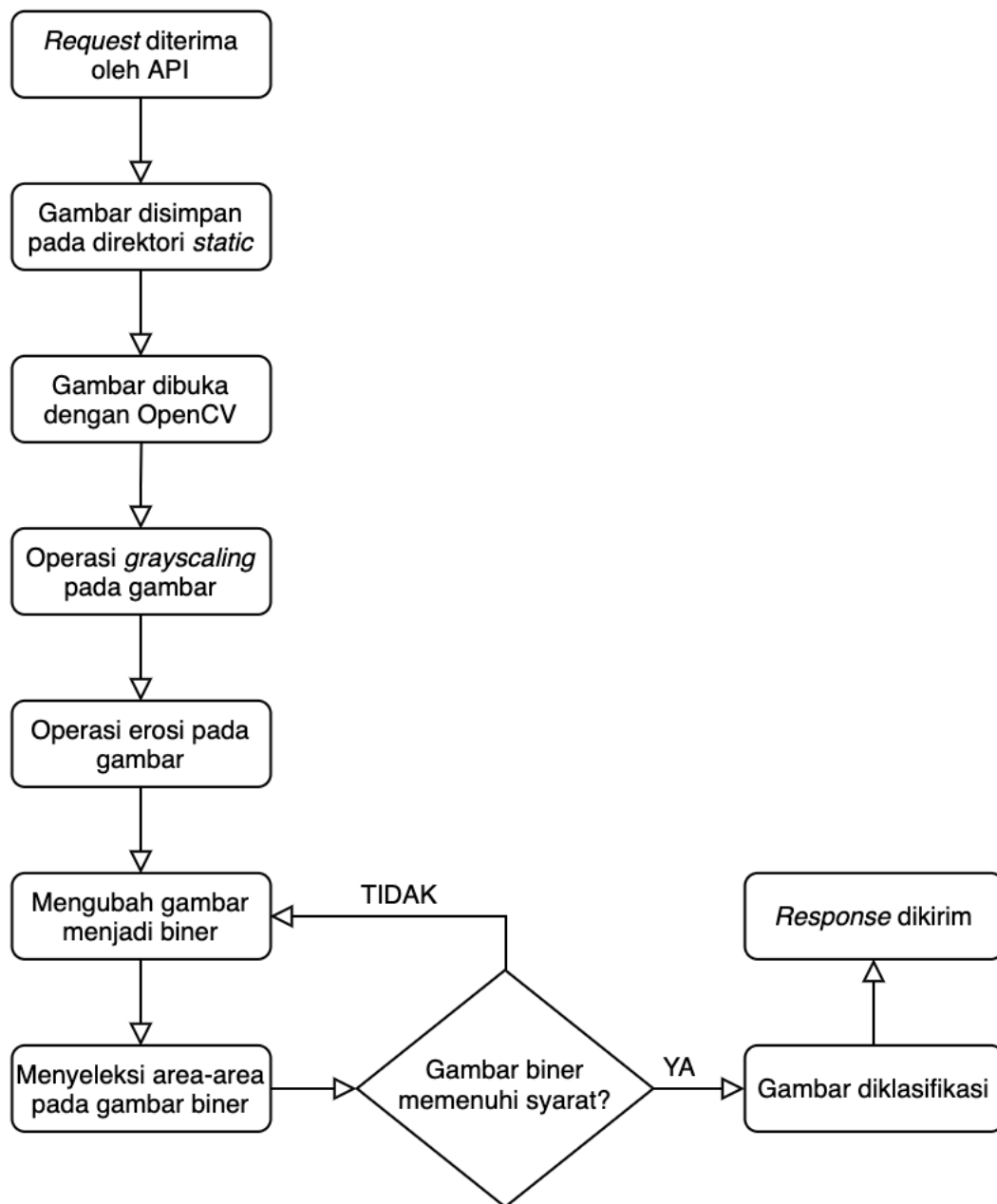


## Dokumentasi Program Klasifikasi Keretakan Jalan

### A. Diagram *Flowchart*



Terdapat 10 proses pada diagram *flowchart* diatas yang akan dijelaskan sebagai berikut. Masing-masing proses bersesuaian dengan keterangan pada *source code*, sehingga lebih urutan-urutan proses dapat lebih mudah dipahami.

- Proses 1: *Request* diterima oleh API  
Pada proses ini, *request* diterima oleh API yang memiliki alamat “/process\_image”

—Proses 2: Parameter-parameter *request* diperoleh

Terdapat tiga parameter yang akan diperoleh, yakni:

1. *kernel\_size*, diperlukan sebagai ukuran kernel untuk melakukan operasi erosi
2. *threshold\_binary\_image*, diperlukan sebagai *threshold* untuk mengubah gambar menjadi *binary* (hitam atau putih / 0 atau 255)
3. *threshold\_region\_selection*, diperlukan sebagai *threshold* untuk memilih *region* tertentu pada gambar *binary* berdasarkan ukuran dari *region* tersebut.

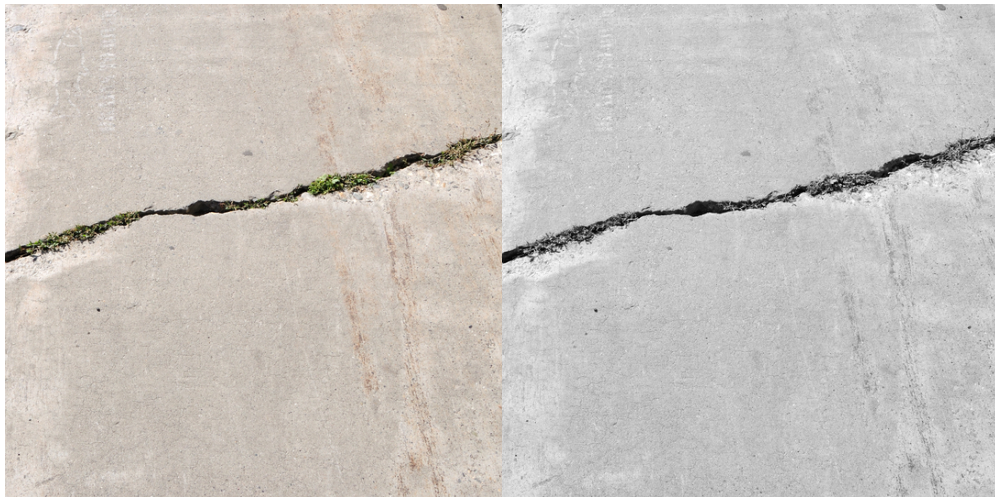
- Proses 2: Gambar disimpan pada direktori *static*

Tujuan gambar ini disimpan agar nanti dapat dibuka lagi untuk pemrosesan. Gambar disimpan dengan nama *image\_original*. Jika sudah ada gambar dengan nama *image\_original*, maka gambar tersebut akan ditimpa (*overwrite*).

- Proses 3: Gambar dibuka dengan OpenCV

- Proses 4: Operasi *grayscale* pada gambar

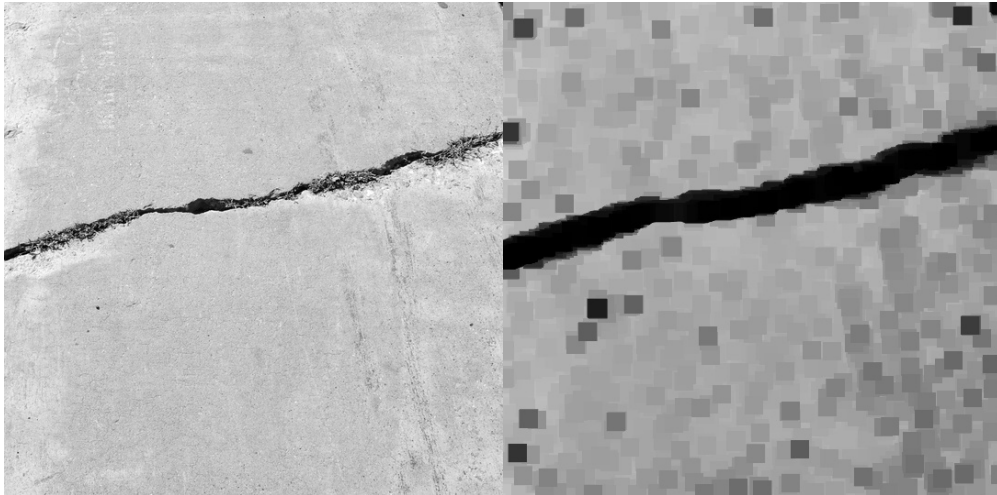
Tujuan mode gambar diubah menjadi *grayscale* adalah agar gambar lebih mudah diproses. Ketika gambar bermode RGB, piksel-piksennya memiliki tiga *channel* warna, yakni *Red*, *Green*, dan *Blue*. Untuk menyederhanakannya, gambar diubah menjadi mode *grayscale*. Dengan mode *grayscale*, piksel-piksel gambar hanya memiliki satu *channel* warna, yakni intensitas cahaya saja.



Gambar 1. Gambar kiri adalah gambar asli, sedangkan gambar kanan adalah gambar hasil operasi *grayscale*

- Proses 5: Operasi erosi pada gambar

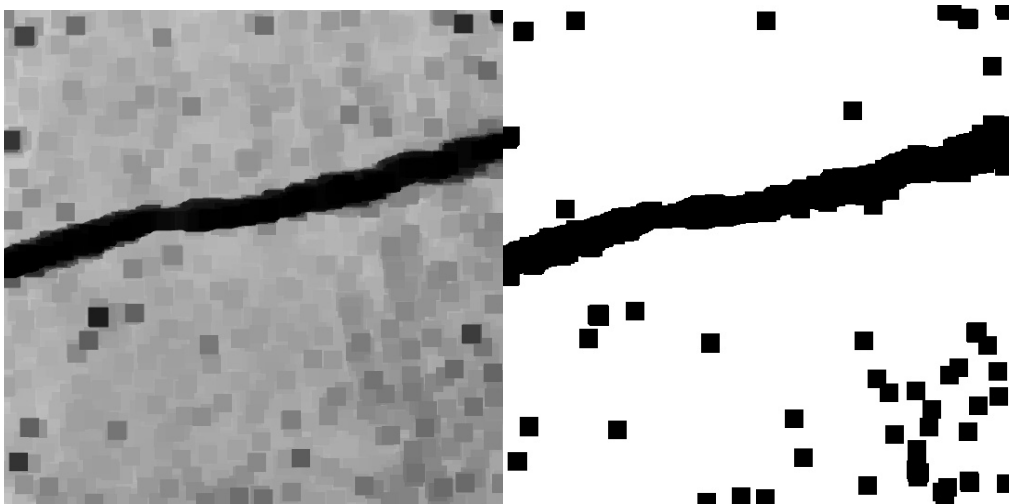
Operasi erosi adalah salah satu operasi morfologi pada gambar. Selain operasi erosi, terdapat operasi dilasi. Operasi erosi bertujuan untuk menghilangkan *noises* pada gambar dan membuat fitur gambar yang diinginkan, yakni keretakan jalan, menjadi lebih besar.



Gambar 2. Gambar kiri adalah gambar hasil operasi *grayscale*, sedangkan gambar kanan adalah gambar hasil operasi erosi

- Proses 6: Mengubah gambar menjadi biner

Gambar *binary* adalah gambar yang piksel-piksennya hanya memiliki dua kemungkinan warna, yakni hitam atau putih / 0 atau 255. Hal ini dilakukan dengan melakukan *thresholding*. Nilai *threshold* ditentukan oleh pengguna. Pada program ini, nilai *default threshold* telah ditentukan. Jika piksel memiliki nilai dibawah *threshold*, maka nilai piksel tersebut diubah menjadi 0. Jika piksel memiliki nilai diatas *threshold*, maka nilai piksel tersebut diubah menjadi 255.



Gambar 3. Gambar kiri adalah gambar hasil operasi erosi, sedangkan gambar kanan adalah gambar yang sudah diubah menjadi biner

- Proses 7: Menyeleksi area-area pada gambar biner

Seleksi *region* dilakukan pada *binary image* untuk mendapatkan fitur yang benar-benar diinginkan untuk klasifikasi, yakni keretakan jalan itu sendiri. Pada Proses 7, ketika diperoleh gambar *binary*, keretakan jalan masih belum teridentifikasi karena masih terdapat

banyak *region* dalam gambar. Dengan demikian, perlu dipilih *region* yang memiliki besar diatas *threshold* agar *region* yang tersisa pada gambar adalah keretakan jalan. Nilai *threshold* ini ditentukan oleh masukan dari pengguna. Jika pengguna tidak memberi masukan, maka digunakan nilai *default*.



Gambar 4. Gambar kiri adalah gambar biner, sedangkan gambar kanan adalah gambar biner yang area-areanya hitamnya sudah diseleksi.

- Proses 8: Pengecekan apakah gambar biner memenuhi syarat  
Gambar biner yang area-areanya sudah diseleksi, akan diklasifikasi. Namun sebelum itu, perlu dilakukan pengecekan apakah gambar tersebut sudah layak diklasifikasi. Bagaimana menentukan sebuah gambar sudah layak untuk diklasifikasikan atau belum?

Caranya adalah dengan membandingkan piksel hitam (*wanted pixel*) pada gambar dengan keseluruhan piksel yang ada. Gambar yang layak diklasifikasi memiliki piksel hitam sebanyak 7% - 15% dari keseluruhan piksel yang ada di gambar.

Bagaimana jika piksel hitam lebih dari 15%? Ulangi Proses 6 untuk mengubah gambar menjadi biner dengan *threshold* yang lebih rendah, sehingga jumlah piksel hitam dapat berkurang.

Bagaimana jika piksel hitam kurang dari 7%? Ulangi Proses 6 untuk mengubah gambar menjadi biner dengan *threshold* yang lebih tinggi, sehingga jumlah piksel hitam dapat bertambah.

Pada program ini, penambahan atau pengurangan *threshold* dilakukan secara otomatis dengan menerapkan prinsip *binary search* untuk mendapatkan *threshold* yang baik. Bagaimana caranya?

- a. Pada awalnya, *threshold* untuk gambar biner bernilai 128. Lakukan pengecekan.
  - b. Jika hasil pengecekan menandakan *threshold* harus dinaikkan, maka naikkan *threshold* sebanyak 64. Jika hasil pengecekan menandakan *threshold* harus diturunkan, maka turunkan *threshold* sebanyak 64. Lakukan pengecekan.
  - c. Jika hasil pengecekan menandakan *threshold* harus dinaikkan, maka naikkan *threshold* sebanyak 32. Jika hasil pengecekan menandakan *threshold* harus diturunkan, maka turunkan *threshold* sebanyak 32. Lakukan pengecekan.
  - d. Ulangi proses tersebut hingga dua kondisi ini terpenuhi:
    - Pengecekan berhasil, artinya gambar biner sudah layak diklasifikasi.
    - Pengecekan selalu gagal sampai unit penaikan atau penurunan adalah 1. Jika gambar biner belum juga layak diklasifikasi, maka gambar tidak bisa diklasifikasi.
- Proses 9: Klasifikasi gambar berdasar hasil seleksi *region*  
 Klasifikasi bertujuan untuk menentukan apakah keretakan jalan berjenis TRAVERSAL, LONGITUDINAL, atau TURTLE. Klasifikasi dilakukan menggunakan hasil pada Proses 8.
- Jika gambar hasil Proses 8 memiliki *region* di bagian kiri dan kanan dan tidak memiliki *region* di bagian atas dan bawah, maka jenis keretakan adalah TRAVERSAL.
- Jika gambar hasil Proses 8 memiliki *region* di bagian atas dan bawah dan tidak memiliki *region* di bagian kiri dan kanan, maka jenis keretakan adalah LONGITUDINAL.
- Jika gambar hasil Proses 8 memiliki *region* di bagian atas, kanan, dan kiri, maka jenis keretakan adalah TURTLE.
- Proses 10: *Response* dikirim  
 Setiap gambar hasil Proses sebelumnya, berikut hasil klasifikasi dikirimkan kembali ke pengguna menggunakan JSON.

#### B. FILE: *constants.py*

- Kelas Pixel berisi atribut warna piksel untuk gambar binary.
- Piksel dengan nilai 0 berwarna hitam, sedangkan piksel dengan nilai 255 berwarna putih.
- Kelas ImageClass berisi jenis-jenis keretakan.
- Kelas ThresholdRecommendation berisi apakah *threshold* untuk gambar biner harus dinaikkan atau diturunkan.

#### C. FILE: *image\_processor.py*

- Kelas ImageProcessor berisi fungsi-fungsi yang diperlukan untuk memproses gambar
- Fungsi *get\_binary\_image* digunakan untuk mengubah gambar menjadi gambar binary
- Fungsi *get\_wanted\_regions* digunakan untuk menyeleksi region-region tertentu pada gambar binary
- Fungsi *classify* digunakan untuk mengklasifikasikan gambar