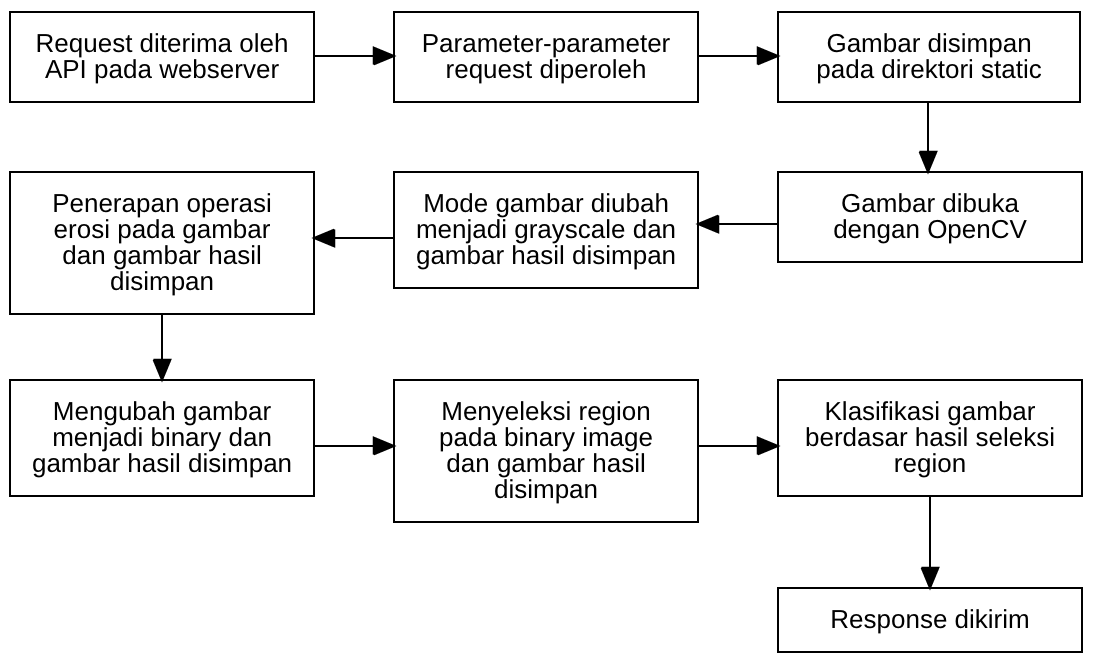
Dokumentasi Program

Klasifikasi Keretakan Jalan

1. 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 webserver

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 3: 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 4: Gambar dibuka dengan OpenCV
* Proses 5: Mode gambar diubah menjadi *grayscale* dan gambar hasil disimpan

Tujuan mode gambar diubah menjadi *grayscale* adalah agar gambar lebih mudah diproses. Ketika gambar bermode RGB, piksel-pikselnya 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.

* Proses 6: Penerapan operasi erosi pada gambar dan gambar hasil disimpan

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.

* Proses 7: Mengubah gambar menjadi *binary* dan gambar hasil disimpan

Gambar *binary* adalah gambar yang piksel-pikselnya 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.

* Proses 8: Menyeleksi *region* pada *binary image* dan gambar hasil disimpan

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*.

* 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.

1. 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.
* Nilai pada variabel-variabel ini sebenarnya bebas.

1. FILE: *image\_processory.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