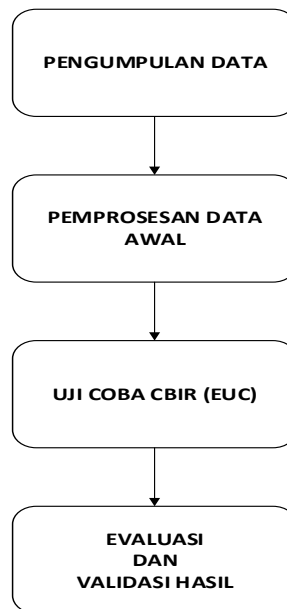


BAB III METODOLOGI PENELITIAN

3.1 Alur Penelitian

Bagan alur penelitian yang dilakukan dapat di ilustrasikan pada gambar 3.1



Gambar 3.1 Alur penelitian

3.2 Alat dan Bahan

Pada tahap ini peneliti menggunakan beberapa aplikasi dan perangkat keras pendukung untuk memudahkan peneliti dalam menguji algoritma, antara lain:

Perangkat Keras :

1. PC Acer Exstensa
2. Core i5, 4 core, 2,8 Ghz
3. Ram 8GB
4. Graphic Nvidia Gforce 730 DDR3 2GB
5. Resolusi layar 1376x768

Perangkat lunak :

1. Python versi 3.8
2. Matplotlib modul versi 3.1.3
3. Numpy modul versi 1.18.1
4. Image modul versi 1.5.28
5. Scikit-image modul versi 0.16.2
6. Opencv-python versi 4.2.0.32
7. Imageio modul versi 2.8.0
8. Sublime text versi 3

3.3 Pengumpulan Data

Penelitian ini mengumpulkan data melalui *study literatur* dari jurnal – jurnal yang relevan dan *ebook* yang didapatkan dari penyedia jurnal yang disediakan oleh pihak google. Sedangkan data *ground truth* didapatkan melalui *dataset* yang disediakan oleh *Oxford* dan *Kaggle* yang dapat diunduh melalui alamat web www.robots.ox.ac.uk/~vgg/data/ dan www.kaggle.com/paramaggarwal/fashion-product-images-dataset pada alamat ini disediakan dataset yang sangat banyak dan berbagai jenis citra ada di alamat tersebut. Karena penelitian ini hanya menggunakan data berupa citra yang memiliki JPEG dan berjenis *artificial* maka peneliti hanya mengunduh file yang dibutuhkan sesuai kebutuhan peneliti. Adapun jumlah *dataset* yang akan peneliti kumpulkan yaitu sebesar 15000 *dataset*.

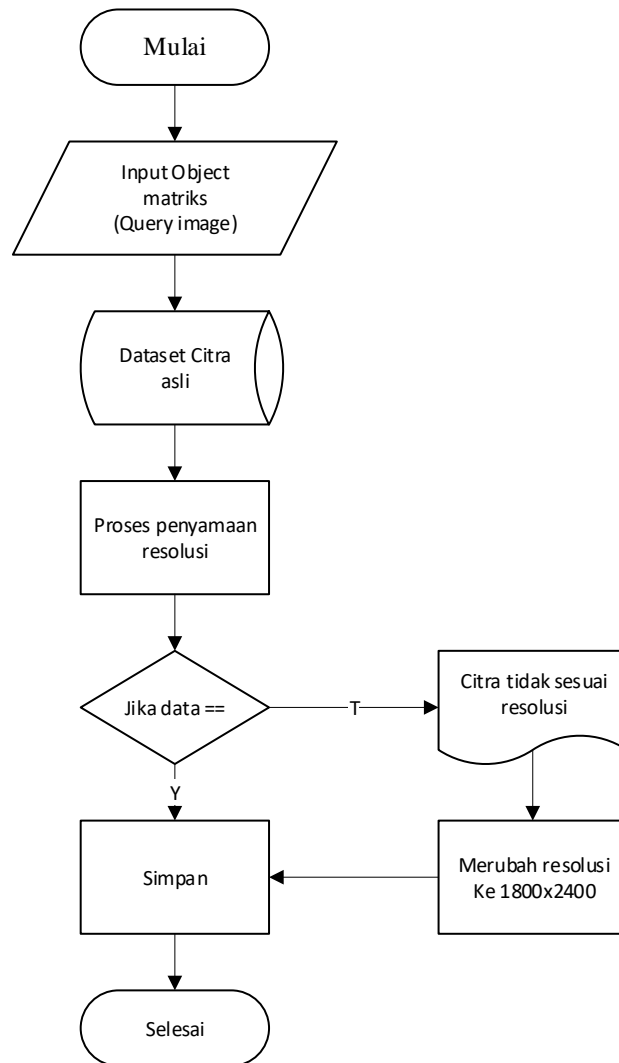
3.4 Pemrosesan data awal

Pemrosesan data awal ini dilakukan setelah *dataset* yang dibutuhkan telah terkumpul, pemrosesan data awal ini bertujuan untuk menghasilkan citra *ground truth* yang akan digunakan pada proses CBIR dengan metode *Euclidean distance*. Adapun beberapa proses pengolahan data awal adalah sebagai berikut.

3.4.1 Penyetaraan Resolusi Gambar

Proses ini dilakukan untuk melakukan konversi dari seluruh citra yang sudah dikumpulkan menjadi satu resolusi yang sama yaitu 1800x2400 piksel. Konversi dilakukan secara otomatis menggunakan program yang dibangun menggunakan python dengan merubah resolusi gambar yang resolusinya tidak sesuai dengan

resolusi citra yang ditentukan peneliti menjadi sesuai dengan ketentuan. Alur proses perubahan resolusi atau sering disebut resize dapat dilihat pada Gambar 3.2.



Gambar 3.2 Proses *Resize*

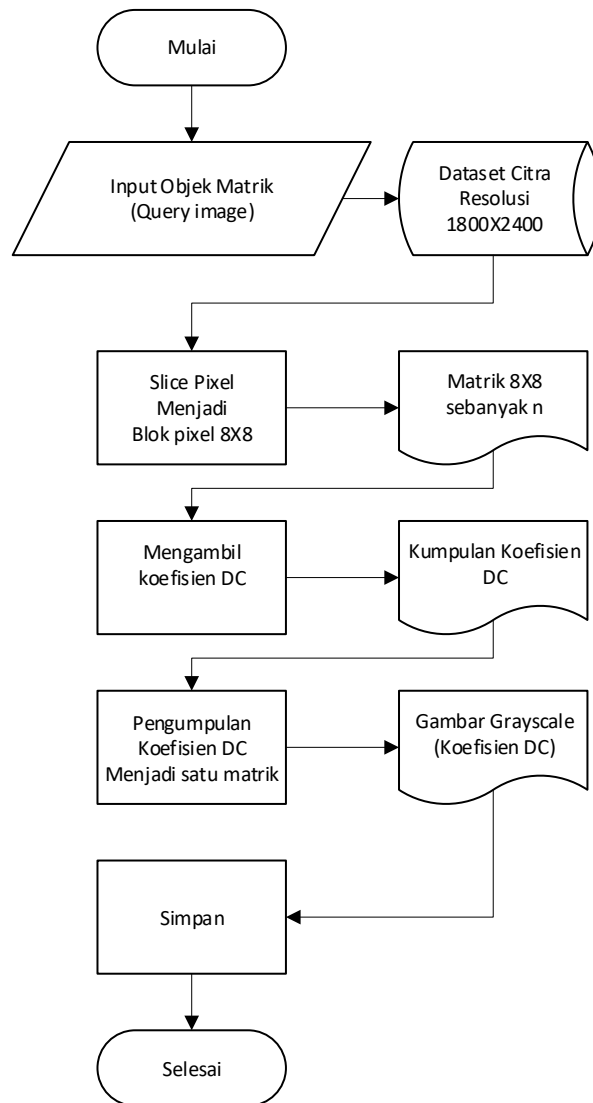
3.4.2 Ekstraksi DC

Dalam penelitian ini ekstraksi fitur DC akan digunakan untuk membuat bahan yang digunakan untuk pencarian citra (*image indexing* dan *image retrieval*). Algoritma DC ini hampir sama dengan algoritma DCT tetapi hanya koefisien DC yang digunakan sebagai indexing DC baik di citra query ataupun di dalam database. Oleh karena itu hanya selisih dari koefisien DC saja yang akan dihitung.

Hal ini dikarenakan koefisien DC mengandung jumlah energi dari suatu citra yang sangat mempengaruhi citra tersebut secara signifikan. Persamaan algoritma DC dapat ditulis sebagai berikut :

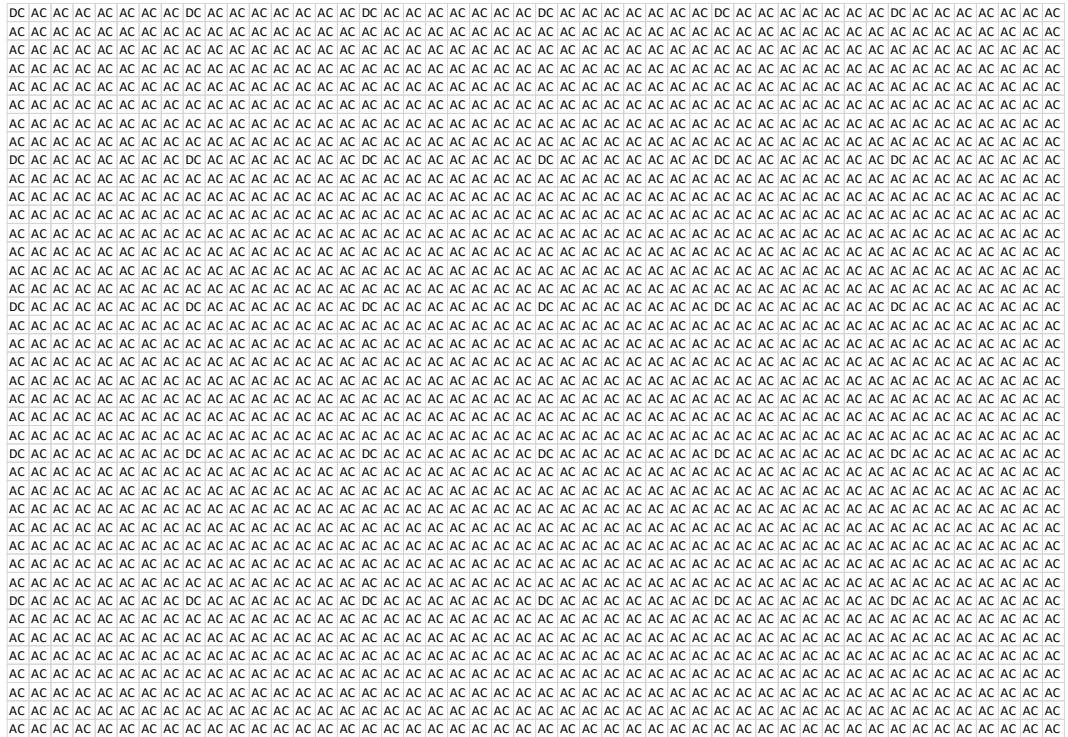
$$H = \sum_{i=1}^{64} C_i$$

Dimana H adalah kunci peng-indeks-an dan $C_i = \frac{\sum_{k=1}^{64} (category)_k}{N}$, C adalah nomor kategori ke i dari ke 64 koefisien DCT. Dalam penelitian ini resolusi gambar yang akan di ambil koefisien DC nya yaitu 1800x2400 yang kemudian akan dilakukan *grayscale* dan *pixel* dari citra *grayscale* tersebut akan dipecah menjadi blok 8x8 yang kemudian akan diambil koefisien pertama yaitu koefisien DC. Berikut adalah proses dari ekstraksi koefisien DC dapat dilihat pada gambar gambar 3.3.

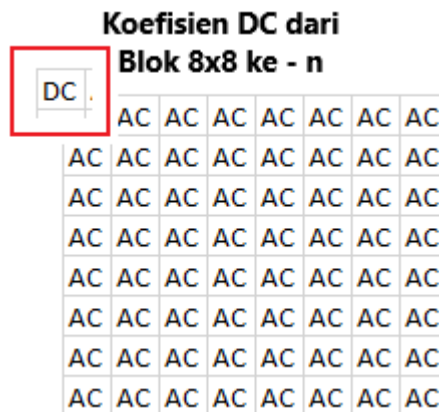


Gambar 3.3 Proses Ekstraksi DC

Ilustrasi ekstraksi koefisien DC dimulai dari pemecahan piksel menjadi blok 8x8 sampai menjadi gambar grayscale dapat dilihat pada gambar 3.4 sampai gambar 3.6.



Gambar 3.4 Ilustrasi koefisien didalam satu citra resolusi 48x40



Gambar 3.5 ilustrasi ekstraksi koefisien DC dari blok 8x8 ke – n

DC	DC	DC	DC	DC	DC
DC	DC	DC	DC	DC	DC
DC	DC	DC	DC	DC	DC
DC	DC	DC	DC	DC	DC
DC	DC	DC	DC	DC	DC

Gambar 3.6 ilustrasi citra dengan koefisien DC

3.4.3 Segmentasi *Multi Otsu Thresholding*

Segmentasi citra merupakan proses yang ditujukan untuk mendapat objek – objek yang terkandung didalam citra atau membagi citra ke beberapa daerah dengan setiap objek atau daerah memiliki kemiripan atribut. Metode otsu ini untuk menentukan nilai ambang. Pada metode otsu nilai ambang akan didapatkan secara otomatis, yakni dengan mengubah citra digital abu – abu menjadi hitam putih berdasarkan perbandingan nilai ambang dengan nilai warna piksel citra digital. Selanjutnya akan dilakukan tahap binerisasi sehingga piksel hanya akan bernilai 1 dan 0, kemudian tahap selanjutnya adalah proses negasi untuk penggantian warna, agar proses penghapusan noise dapat lebih akurat. Berikut ini adalah algoritma *otsu* menentukan nilai *threshold* k [10] dengan nilai k berkisar antara 0 sampai 255.

1. Probabilitas setiap pixel pada setiap gray level i

$$p_i = \frac{n_i}{N}$$

Dimana n_i = jumlah pixel pada level ke i

2. Jumlah kumulatif (ZerothCM)

$$\omega(k) = \sum_{i=0}^k p_i$$

3. Rata – rata kumulatif (firstCM)

$$\mu(k) = \sum_{i=0}^k i \cdot p_i$$

4. Rata – rata intensitas global (tMeans)

$$\mu_T = \sum_{i=0}^L i \cdot p_i$$

5. Nilai ambang ditentukan dengan memaksimumkan persamaan dari

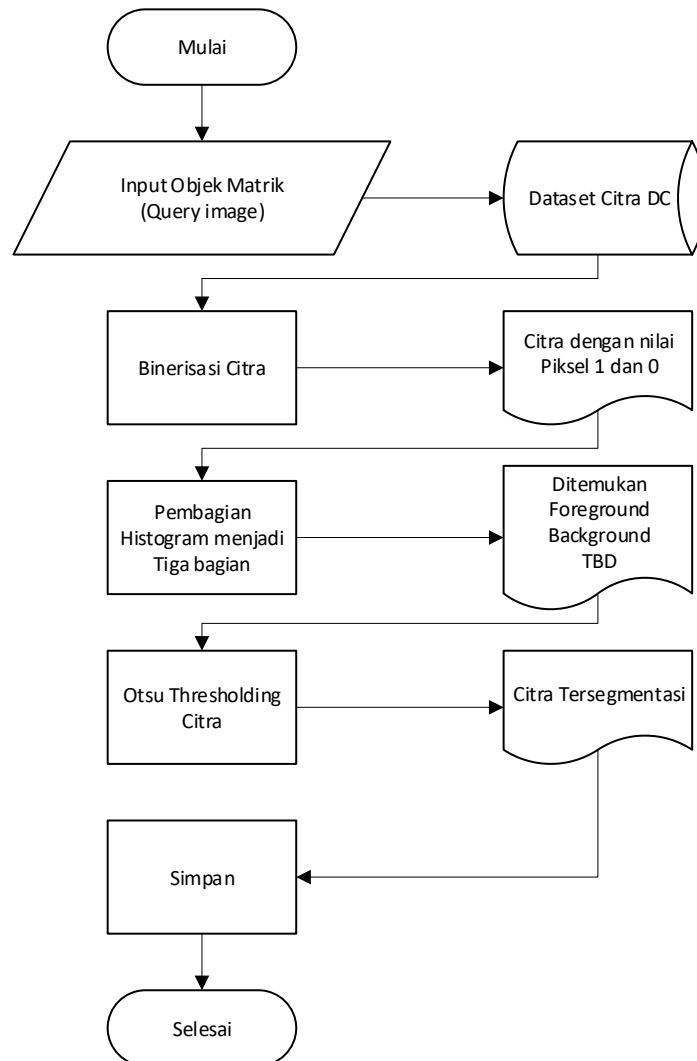
$$\sigma_B^2(k^*) = \max (\sigma_B^2(k))$$

Dengan persamaan

$$\sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]}$$

Metode *Otsu* merupakan metode populer diantara semua metode *thresholding* dan metode terbaik dalam mendapatkan nilai *threshold* secara otomatis. Sedangkan metode *multi otsu thresholding* merupakan perbaikan pada *otsu treshloding* dengan

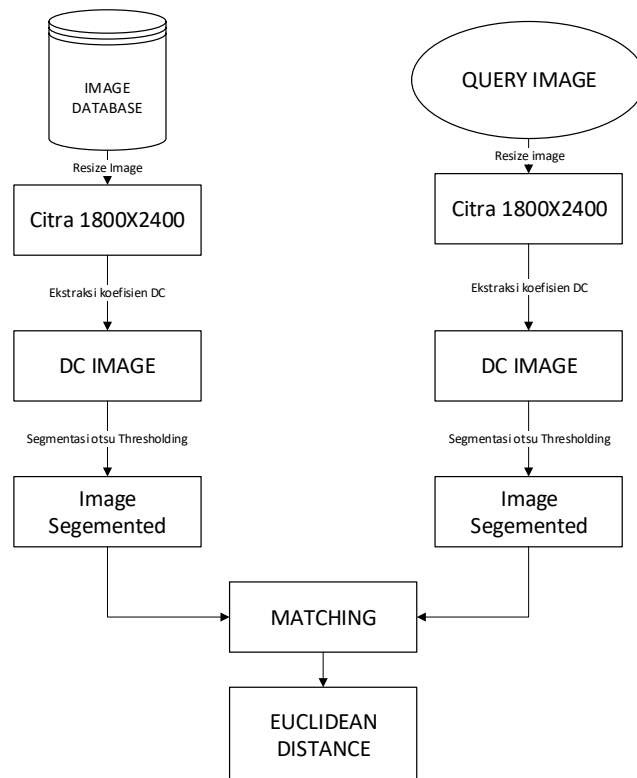
membagi histogram menjadi tiga bagian dimana bagian piksel yang memiliki nilai tinggi adalah *foreground* dan dan piksel yang memiliki nilai lebih rendah adalah *background*. Sedangkan piksel diantara kedua tersebut merupakan piksel yang sering disebut dengan sebutan TBD(*to-be Determined*). Proses segmentasi dapat dilihat pada gambar 3.4 berikut.



Gambar 3.7 Proses Segmentasi Multi Otsu Thresholding

3.5 Model Content Base Image Retrieval (CBIR)

Sistem CBIR yang akan dibangun yaitu mengikuti pada gambar 3.5 berikut ini.



Gambar 3.5 Model CBIR

Penjelasan proses-proses dari gambar diatas adalah sebagai berikut:

1. Pembuatan Dataset

Tahap ini adalah membangun dataset citra artificial yang berjumlah 15.000 citra yang didapatkan dari dataset yang disediakan oleh *Oxford* dan *Kaggle*.

2. Citra Query

Pada tahap ini pengguna memasukkan citra query yang ingin dicari

3. Proses Resize

Proses merupakan proses dimana citra yang diperoleh dari internet sebelumnya disetarakan resolusinya menjadi 1800X2400.

4. Proses Ekstraksi DCT

Kemudian pada tahap ini akan diekstrak fitur DC baik citra query maupun database citra kemudian akan disegmentasi.

5. Proses segmentasi

Proses segmentasi citra menggunakan metode otsu thresholding.

6. Proses kemiripan citra

Pada tahap ini hasil yang didapat dari hasil proses segmentasi tadi yang kemudian akan diproses dengan metode *Euclidean Distance* untuk mendapatkan hasil pengenalan

3.6 Efektifitas pencarian citra

Dalam penelitian ini akan dikerjakan *query* sebanyak 200 *query* yang akan di retrieve dari setiap database. Untuk setiap *query* sebanyak 20 akan ditampilkan yang kemudian akan dihitung presisi dan *recall*. Berikut adalah persamaan yang digunakan untuk menghitung presisi dan *recall* :

$$presisi = \frac{jumlah\ citra\ relevan\ yang\ terpanggil}{jumlah\ citra\ yang\ terpanggil}$$

$$recall = \frac{jumlah\ citra\ relevan\ yang\ terpanggil}{jumlah\ citra\ dalam\ dataset}$$

3.7 Metode Pengembangan Perangkat Lunak

Dalam penelitian ini metode pengembangan perangkat lunak yang dapat digunakan yaitu prototipe, dimana pada proses ini dilakukan pengumpulan data dan mengidentifikasi algoritma serta proses apa saja yang akan dimasukkan didalam sistem yang akan digunakan untuk menguji algoritma yang digunakan.