

BAB 3

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Penelitian “Penerapan Deep Learning Style Transfer untuk Menciptakan Kombinasi Gambar dengan Batik” menggunakan beberapa tahap agar metodologi dan perancangan sistem dapat terpenuhi. Tahap-tahap yang dilaksanakan antara lain adalah sebagai berikut.

a. Identifikasi dan Perumusan Masalah

Permasalahan yang dipilih untuk diteliti adalah menggabungkan suatu gambar konten dengan gambar batik sebagai tema gaya utama dengan menggunakan beberapa kombinasi lapisan VGG19.

b. Studi Literatur

Setelah melakukan identifikasi dan perumusan masalah, selanjutnya adalah melakukan studi literatur yang bertujuan untuk memperoleh informasi dan teori-teori yang berhubungan atau relevan dengan permasalahan tersebut seperti jurnal dari penelitian mengenai topik terkait dan serupa yang sudah pernah dilakukan sebelumnya. Proses ini bertujuan untuk meningkatkan pemahaman secara teori terkait *Neural Style Transfer* dan VGG19 untuk memenuhi landasan teori penelitian.

c. Pengumpulan Data

Dataset yang akan digunakan dalam penelitian diperoleh melalui Kaggle dan Freepik dan merupakan *Free Resources*. Dataset berupa gambar dengan dua jenis yang berbeda, yaitu gambar konten dan gambar batik, dimana gambar konten

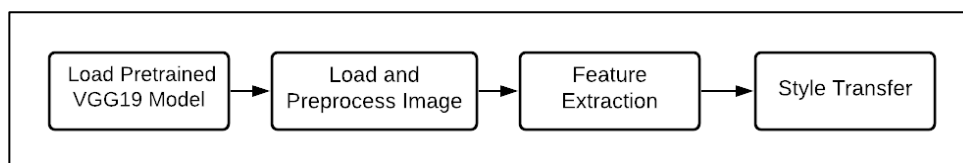
akan berjumlah dua gambar dan gambar batik akan berjumlah lima gambar.

d. Pemrograman Sistem

Berdasarkan topik permasalahan yang sudah dipilih, selanjutnya dilakukan analisis untuk mengimplementasikan algoritma yang tepat dan sesuai dengan permasalahan tersebut. *Neural Style Transfer* dan VGG19 akan digunakan dalam penelitian ini. VGG19 akan digunakan untuk ekstraksi fitur pada *layer* tertentu untuk gambar konten dan batik, oleh karena itu lapisan VGG19 yang akan digunakan hanya lapisan *features* dimana lapisan *features* tidak termasuk lapisan klasifikasi. Kemudian algoritma *Neural Style Transfer* akan digunakan untuk melakukan transfer gaya pada hasil ekstraksi fitur yang telah diperoleh.

e. Implementasi Algoritma

Pada tahap ini, VGG19 dan *Neural Style Transfer* diimplementasikan untuk melakukan ekstraksi fitur gambar dengan menggunakan *layer* VGG19 tertentu dan melakukan proses transfer gaya pada gambar konten dan batik yang diambil dari Freepik. Adapun tahapan dari implementasi algoritma dimuat pada Gambar 3.1 berikut.



Gambar 3.1 Alur kerja algoritma

f. Evaluasi Hasil Implementasi Algoritma

Setelah algoritma diimplementasikan, hal selanjutnya yang akan dilakukan adalah melakukan evaluasi terhadap gambar hasil kombinasi yang dihasilkan dari implementasi VGG19 dan algoritma *Neural Style Transfer* dalam melakukan ekstraksi fitur dan transfer gaya pada gambar konten dan batik. Adapun evaluasi

akan dilakukan secara kualitatif dengan menggunakan metode skala 5 poin *Likert* (Fischer, *et al.*, 2020). Kuesioner akan disebarluaskan secara publik dengan jumlah responden minimal sebanyak 30 orang (Sugiono, 2015) untuk mengukur tingkat kepuasan responden terhadap gambar hasil kombinasi tanpa memperhatikan aspek nilai-nilai batik secara detail. Dengan demikian, responden dapat memberikan pendapat mereka dengan melihat gambar hasil kemudian responden dapat memberikan penilaian mereka dengan melalui skala 5 poin *Likert* pada kuesioner tersebut (Bogucka, E.P. dan Meng, L., 2019).

g. Penyusunan Laporan

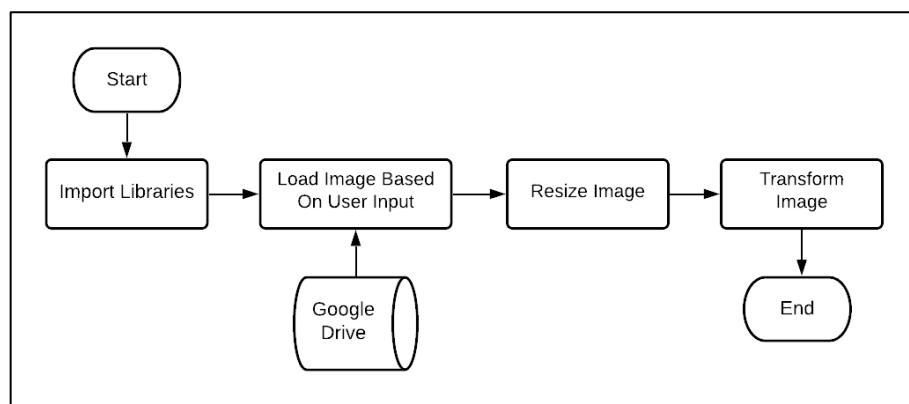
Tahap terakhir dalam penelitian ini adalah menyusun laporan yang berisikan hasil keseluruhan dari penelitian yang sudah dilakukan sebagai dokumentasi dari penelitian yang sudah dilakukan. Laporan disusun secara terstruktur sesuai dengan kaidah penyusunan dan penulisan laporan ilmiah yang sudah ditentukan, dimulai dari pendahuluan hingga kesimpulan dan saran.

3.2 Perancangan Sistem

Sistem yang dihasilkan dari penelitian ini berbasis *website* dengan menggunakan *Anvil*. Sebelum *website* dibuat, dilakukan perancangan beberapa *flowchart* yang terdiri dari *flowchart* untuk *load* dan *preprocess* gambar, *flowchart* ekstraksi fitur, *flowchart* untuk melakukan *Style Transfer*, *flowchart* alur kerja *website*, tabel daftar aset gambar yang digunakan, dan rancangan tampilan antarmuka.

3.2.1 Flowchart Load dan Preprocess Gambar

Sebelum dilakukan proses ekstraksi fitur pada VGG19, dilakukan proses pengambilan dan preprocess gambar untuk membuat gambar memiliki bentuk dan ukuran yang sesuai dengan format masukan pada lapisan VGG19. Gambar tersebut merupakan parameter yang dikirim dari *website* berdasarkan pilihan *user*. Adapun *flowchart* untuk proses pengambilan dan preprocess gambar dapat dilihat pada Gambar 3.2.



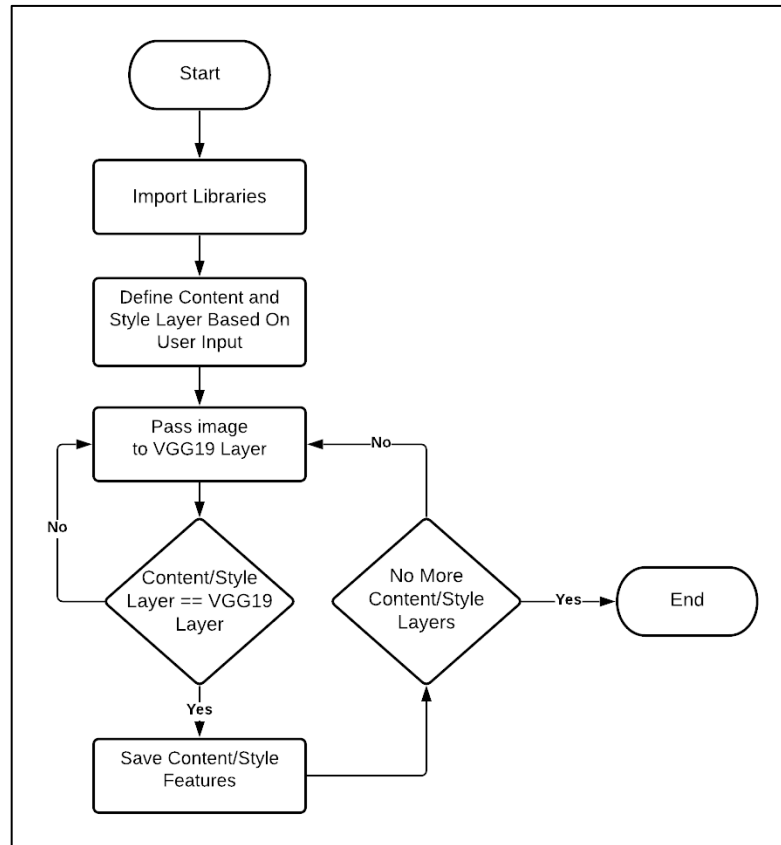
Gambar 3.2 *Flowchart load dan preprocess gambar*

Berdasarkan Gambar 3.2, *library* yang dibutuhkan akan di-*import* terlebih dahulu. Kemudian gambar konten dan batik akan dipanggil dari *Google Drive Folder* yang telah disiapkan sebelumnya. Setelah gambar dipanggil, proses *resize* akan dilakukan dan akan mengubah ukuran gambar menjadi 224x224 dalam ukuran piksel. Hal tersebut dikarenakan lapisan VGG19 dilatih dengan gambar berukuran 224x224, sehingga ukuran tersebut merupakan ukuran minimum dari gambar pada lapisan VGG19. Proses pengubahan ukuran gambar akan dilanjutkan dengan proses *transform image*. Proses tersebut bertujuan untuk mengubah gambar menjadi format tensor (B, C, H, W), dimana B merupakan jumlah gambar dari *batch*. C merupakan jumlah *channel* gambar, H merupakan *height* dari gambar, W merupakan *width* dari gambar. Gambar yang telah dipanggil memiliki jumlah

channel sebanyak tiga dimana nilai tiga merepresentasikan nilai *channel red*, *green*, dan *blue* (RGB). Setelah diubah menjadi tensor, dilakukan normalisasi pada gambar sehingga memiliki nilai per piksel yang lebih kecil tanpa mengubah warna dari gambar tersebut.

3.2.2 Flowchart Ekstraksi Fitur

Setelah dilakukan preproses pada gambar konten dan batik, selanjutnya adalah ekstraksi fitur gambar tersebut pada lapisan VGG19. Sebelum gambar dimasukkan ke lapisan VGG19, terlebih dahulu dilakukan penentuan lapisan yang akan digunakan oleh gambar konten dan batik dimana penentuan layer disesuaikan dengan pilihan *user* dari *website*. Setelah menentukan lapisan, maka gambar konten dan batik akan dimasukkan ke sepanjang lapisan *features* VGG19 dan ekstraksi fitur akan dilakukan sesuai dengan lapisan yang telah ditentukan sebelumnya. Jika lapisan konten atau batik yang telah ditentukan sebelumnya sesuai dengan lapisan VGG19 tempat gambar berada, maka lapisan tersebut akan menjadi hasil ekstraksi fitur dari gambar yang kemudian akan disimpan. Hal tersebut akan terus diulangi hingga seluruh lapisan yang telah ditentukan sebelumnya dilewati. Adapun *flowchart* untuk ekstraksi fitur gambar dapat dilihat pada Gambar 3.3.

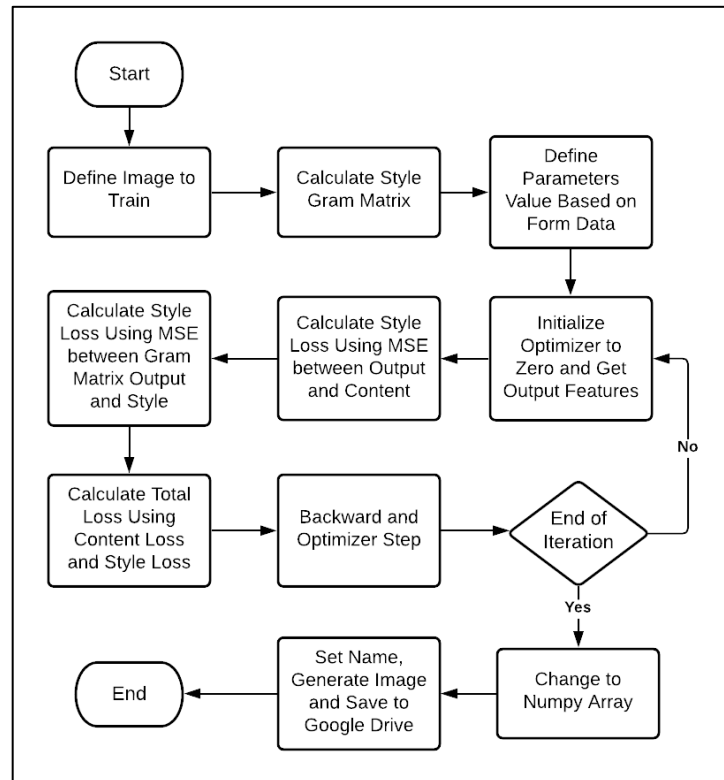


Gambar 3.3 *Flowchart* ekstraksi fitur

3.2.3 Flowchart Untuk Melakukan Style Transfer

Hasil ekstraksi fitur gambar konten dan batik akan diteruskan pada proses transfer gaya bersamaan dengan gambar yang akan menerima hasil representasi dari kedua fitur tersebut dimana gambar tersebut merupakan *clone* dari gambar konten. Namun sebelum masuk pada proses transfer gaya, fitur gambar batik yang merupakan representasi dari gaya akan diproses terlebih dahulu dengan menggunakan *Gram Matrix* (GM). Kemudian dilakukan inisialisasi terhadap beberapa nilai seperti *hyperparameter*, jumlah iterasi, *Adam optimizer*, dan *weight* dimana nilai *weight* akan diberikan untuk setiap lapisan representasi konten dan batik yang telah didefinisikan sebelumnya. Setelah semua parameter terinisialisasi, maka akan dilanjutkan pada proses *train image* secara berulang sesuai dengan

jumlah iterasi yang telah ditentukan sebelumnya. Adapun *flowchart* untuk melakukan *style transfer* dapat dilihat pada Gambar 3.4.



Gambar 3.4 *Flowchart* untuk melakukan *style transfer*

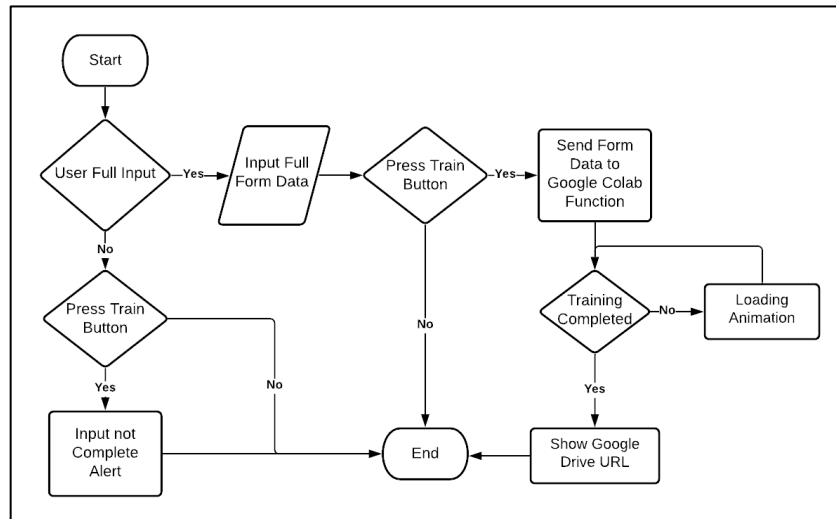
Berdasarkan Gambar 3.4, terlihat bahwa proses *training* dimulai dengan melakukan inisialisasi *optimizer* dengan menjadikannya bernilai *zero* atau *zero grad* dan melakukan ekstraksi fitur terhadap gambar hasil *cloning* gambar konten. Kemudian dilanjutkan dengan menghitung *content loss* dengan melakukan MSE dari gambar yang ingin di *train* dengan gambar konten, menghitung *style loss* dengan melakukan MSE dari *gram matrix* gambar yang ingin di *train* dengan gambar batik. Setelah kedua nilai *loss* didapatkan, maka akan dilanjutkan dengan perhitungan *total loss* yang melakukan operasi perkalian *content loss* dengan *hyperparameter alpha* dan *style loss* dengan *hyperparameter beta* dan kemudian menjumlahkan hasil perkalian kedua *loss* tersebut. Hasil *loss* akan dilanjutkan

dengan melakukan *backward* dimana *backward* akan melakukan perhitungan *derivative* terhadap fungsi yang berkaitan dengan *total loss*. Setelah proses *backward* selesai dan membentuk matriks gradien, maka dilakukan *optimizer step*. *Optimizer step* akan digunakan untuk melakukan *update* terhadap nilai matriks gambar yang ingin di *train* dengan menggunakan algoritma Adam. Setelah proses *step* selesai, proses *training* akan terus dilakukan hingga mencapai iterasi yang telah ditentukan. Jika jumlah perulangan telah mencapai jumlah yang telah ditentukan, maka gambar akan diubah bentuknya menjadi *numpy array* agar dapat di *generate* dan disimpan pada *Google Drive* tetapi sebelum di *generate*, gambar hasil akan diberi nama sesuai dengan *input* yang user masukan pada *website*.

3.2.4 Flowchart Alur Kerja Website

Website akan berbentuk *form* yang terdiri dari empat *dropdown* dan satu *text input*. *Dropdown* pertama dan kedua digunakan untuk memilih gambar konten dan gambar batik, *dropdown* ketiga digunakan untuk memilih *layer* yang akan digunakan untuk representasi gaya dan *dropdown* keempat digunakan untuk memilih *hyperparameter* yang akan diterapkan. Kemudian *text input* digunakan sebagai nama pada gambar yang telah selesai di *training*. Jika ada setidaknya satu dari lima kolom *input* yang tidak diisi atau dipilih oleh *user* dan tombol *train* ditekan, maka akan muncul *alert* yang akan memberi tahu *user* bahwa terdapat kolom *input* yang belum diisi. Jika kolom telah lengkap diisi, maka proses *train* akan dijalankan dengan mengirimkan seluruh *form data* pada fungsi *Google Colab*. Selama proses *training*, halaman akan menampilkan sebuah *loading animation* dan hanya akan berhenti apabila proses telah selesai dan pada bagian bawah tombol

train akan muncul sebuah alamat *Google Drive Folder* yang berisikan hasil gambar dengan nama sesuai dengan yang *user* masukan pada kolom text *input*. Adapun *flowchart* alur kerja *website* dapat dilihat pada Gambar 3.5.



Gambar 3.5 *Flowchart* alur kerja *website*






3.2.5 Tabel Daftar Aset Gambar Yang Digunakan

Berikut ini merupakan daftar aset gambar yang digunakan pada sistem dan disajikan dalam bentuk tabel. Terdapat gambar konten sejumlah dua gambar dan gambar batik sejumlah lima gambar. Adapun tabel daftar aset gambar yang digunakan dapat dilihat pada Tabel 3.1 dan Tabel 3.2.

Tabel 3.1 Daftar aset konten yang digunakan

No	Nama	Gambar	Deskripsi	Sumber	Nama <i>Author</i>
1	Gunung		Digunakan untuk gambar konten pada <i>website</i>	freepik	wirestock
2	Kota Prague		Digunakan untuk gambar konten pada <i>website</i>	freepik	www.slonspics

Tabel 3.2 Daftar aset batik yang digunakan

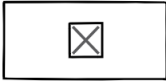
1	Batik Bali dengan Motif - Karakter Hewan		Digunakan sebagai variasi gambar batik pada <i>website</i>	kaggle	Dionisius Darryl Hermansyah
2	Batik Megamendung - Motif Bentuk Awan		Digunakan sebagai variasi gambar batik pada <i>website</i>	kaggle	Dionisius Darryl Hermansyah
3	Batik Ciamis - Motif Tumbuhan		Digunakan sebagai variasi gambar batik pada <i>website</i>	kaggle	Dionisius Darryl Hermansyah
4	Batik Sogan - Motif Bentuk Oval		Digunakan sebagai variasi gambar batik pada <i>website</i>	kaggle	Dionisius Darryl Hermansyah
5	Batik Tambal - Motif Beragam -macam		Digunakan sebagai variasi gambar batik pada <i>website</i>	kaggle	Dionisius Darryl Hermansyah

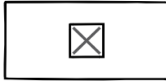
3.2.6 Rancangan Tampilan Antarmuka

Berikut ini merupakan rancangan tampilan antarmuka dari *website* yang akan digunakan dapat dilihat pada Gambar 3.6.

Neural Style Transfer

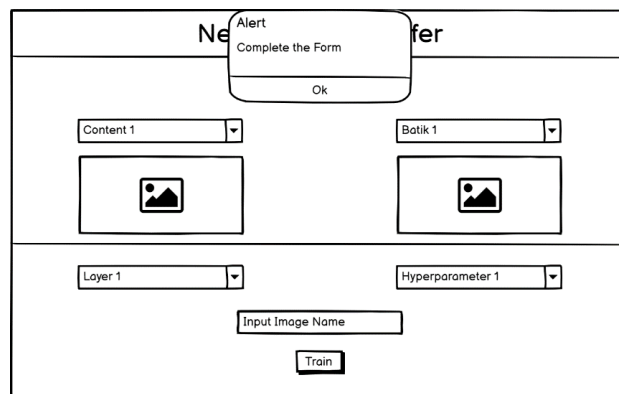
Choose Image





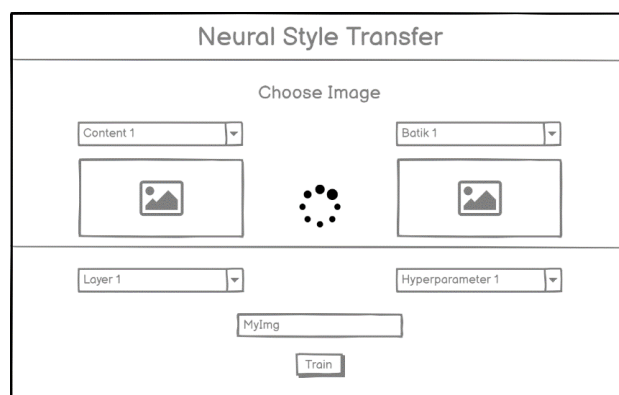
Gambar 3.6 Tampilan antarmuka *website*

Berdasarkan Gambar 3.6, pada tampilan antarmuka *website* terdapat form *input* berbentuk *dropdown input* untuk memilih gambar konten, gambar batik yang ingin digunakan dan lapisan yang akan digunakan sebagai representasi gaya. Terdapat juga *text input* untuk memasukan nama gambar dan tombol yang berfungsi untuk mengirim masukan pada form pada fungsi *Google Colab* dan menjalankan proses transfer gaya. Jika ada salah satu *input* yang tidak diisi, maka akan muncul sebuah *alert*. Adapun tampilan antarmuka saat *alert* muncul dapat dilihat pada Gambar 3.7.



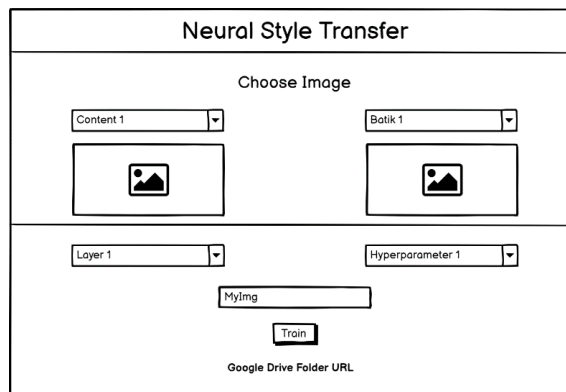
Gambar 3.7 Tampilan antarmuka saat *alert* muncul

Jika tombol *train* ditekan pada saat form data telah diisi seluruhnya, maka akan muncul sebuah *loading animation*. Adapun tampilan antarmuka saat *loading animation* muncul dapat dilihat pada Gambar 3.8.



Gambar 3.8 Tampilan antarmuka saat *loading animation* muncul

Ketika proses selesai, maka *loading animation* akan hilang dan pada bagian bawah tombol *train* akan muncul sebuah *Google Drive Folder link* yang berisikan gambar hasil transfer gaya dengan nama sesuai dengan *input* pada kolom *text input*. Adapun tampilan antarmuka saat *link* muncul dapat dilihat pada Gambar 3.9.



Gambar 3.9 Tampilan antarmuka saat *link* muncul