**LAPORAN UJIAN AKHIR VISI KOMPUTER**

**IMAGE GENERATION MENGGUNAKAN GENERATIVE ADVERSARIAL NETWORK**



Disusun Oleh :

Fachriza Dian Adhiatma (18051204061)

**PRODI S1 TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS NEGERI SURABAYA**

**2021**

**BAB I**

**PENDAHULUAN**

1. Latar Belakang

Untuk mencapai hasil yang optimal, sebuah model deep learning membutuhkan data pembelajaran yang sangat besar. Namun menghimpun data yang besar dan melakukan proses labeling secara manual akan memakan biaya yang besar karena membutuhkan infrastruktur dan tenaga manusia yang mumpuni. Generative Adversarial Network atau GAN adalah salah satu metode machine learning yang dapat digunakan untuk menghasilkan sebuah data baru yang sangat menyerupai data training. Dalam penelitian ini akan dibuat model GAN yang akan dilatih dengan menggunakan dataset MNIST fashion sehingga nantinya model ini dapat menghasilkan gambar-gambar fashion yang berbeda namun serupa dengan data training.

1. Rumusan Masalah
   1. Bagaimana cara membuat model GAN yang optimal untuk menghasilkan gambar baru?
2. Tujuan
   1. Mengetahui cara membuat model GAN yang optimal untuk menghasilkan gambar fashion MNIST yang baru dan identik.
   2. Membantu mengatasi masalah keterbatasan data training kepada pihak yang ingin mengadopsi deep learning pada bisnis mereka.

**BAB II**

**TINJAUAN PUSTAKA**

1. Generative Model

Generative model adalah salah satu jenis machine learning yang digunakan untuk membuat data baru (data generation) seperti gambar, teks, dan audio. Generative model kontras dengan discriminative model yang digunakan untuk membedakan data berdasarkan kelasnya. Secara formal, jika diberikan data X dan label Y, maka

* Generative model menghitung probabilitas gabungan p(X,Y) atau p(X) jika tidak ada label yang diberikan
* Descriminative model menghitung probabilitas dari suatu label jika diberikan suatu observasi, p(Y|X)

Generative model yang digunakan untuk memproses gambar memungkinkan model untuk menangkap korelasi seperti “objek yang menyerupai kapal akan muncul didekat objek yang menyerupai laut” sementara discriminative model hanya mempelajari perbedaan antara kapal dan bukan kapal hanya dengan melihat beberapa pola penting dan mengabaikan korelasi lain yang harus ditangkap oleh generative model.

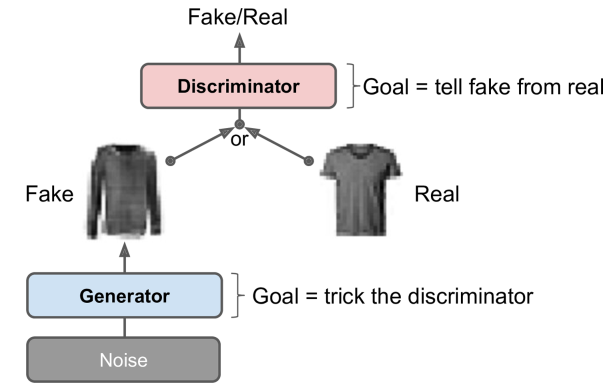
Discriminative model akan menempatkan garis-garis batas dalam ruang vektor, sementara generative model akan mencoba menempatkan data-data ke tempat yang tepat dalam ruang vektor.

Two graphs, one labelled 'Discriminative Model'
          and the other labelled 'Generative Model'. Both graphs show
          the same four datapoints. Each point is labeled with the image
          of the handwritten digit that it represents. In the discriminative
          graph there's a dotted line separating two data points from the
          remaining two. The region above the dotted line is labelled 'y=0' and
          the region below the line is labelled 'y=1'. In the generative graph
          two dotted-line circles are drawn around the two pairs of points. The
          top circle is labelled 'y=0' and the bottom circle is labelled 'y=1

Gambar 1.0. Perbedaan discriminative model dan generative model

1. Generative Adversarial Network (GAN)

Menurut Ian Goodfellow et all (2014), Generative Adversarial Network adalah sebuah generative model yang terdiri dari dua buah model dan saling diadu satu sama lain dengan tujuan meningkatkan performa kedua model teserbut. Dua model penyusun GAN yaitu discriminative model (Discriminator) yang bertugas untuk menentukan apakah sebuah sample berasal dari dataset atau dari model. Generative model (Generator) yang bertugas untuk membuat sample palsu. Secara sederhana generative model dapat dianalogikan sebagai pemalsu yang berusaha mencetak uang palsu, sedangkan discriminative model adalah polisi yang berusaha untuk mendeteksi keaslian uang tersebut. Proses training yang kompetitif ini akan mendorong kedua model tesebut untuk bekerja lebih baik hingga generative model akan mampu membuat sebuah sample palsu yang tidak dapat dibedakan dari sample asli.



Gambar 1.1. Alur training GAN

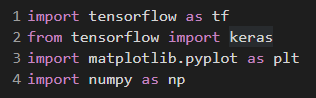
Proses training pada GAN terdiri dari dua tahap:

* 1. Fase pertama berfokus pada melatih discriminator. Data latihan yang digunakan dalam fase ini ada dua yaitu gambar asli yang diambil langsung dari dataset dalam bentuk batch dan gambar palsu yang dihasilkan oleh generator dalam bentuk batch. gambar asli akan diberi label 1 dan gambar palsu akan diberi label 0. Fase ini bertujuan untuk meningkatkan performa discriminator dalam membedakan gambar palsu dan asli.
  2. Fase kedua berfokus untuk melatih generator. mula-mula generator digunakan untuk memproduksi gambar palsu dalam bentuk batch dan kemudian discriminator digunakan untuk memprediksi labelnya, namun dalam kasus ini label data yang dihasilkan generator akan diberi nilai 1. Hal ini bertujuan untuk mengecoh discriminator. Dalam tahap ini, model discriminator harus di freeze karena pada tahap backpropagation bisa terfokus untuk memperbarui parameter pada generator.

**BAB III**

**HASIL DAN PEMBAHASAN**

1. Implementasi program
   1. Import library yang diperlukan



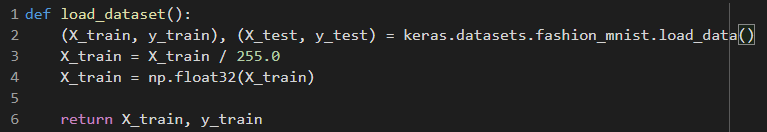
* 1. Membuat variabel tetap untuk menyimpan besarnya dimensi latent dan ukuran batch

Generator bertugas untuk membuat gambar baru dari angka random yang didapatkan dari distribusi gaussian, besarnya dimensi untuk angka random ini diatur oleh variabel latent\_dim. GAN perlu dilatih secara batch, maka dari itu perlu didefinisikan terlebih dahulu besarnya batch yang akan digunakan untuk proses training.



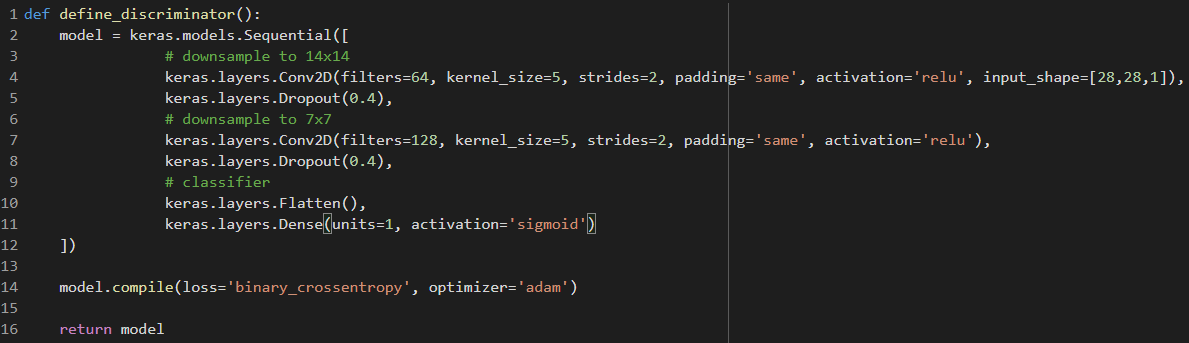
* 1. Fungsi untuk load dataset dan preprocessing

Dataset yang digunakan adalah fashion MNIST yang dapat digenerate langsung dari library keras. Fashion MNIST adalah sebuah gambar greyscale berikuran 28 x 28 pixel. Setelah berhasil di load, data akan dinormalisasikan sehingga range pixelnya menjadi 0 dan 1.

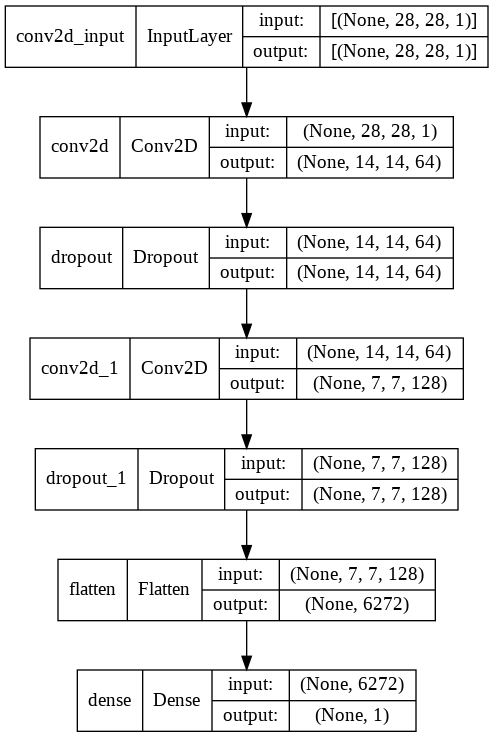


* 1. Fungsi untuk membuat model discriminator

Arsitektur model discriminator ini terdiri dari dua buah layer konvolusi dengan regularisasi dropout, kemudian satu layer dense sebagai classifier.



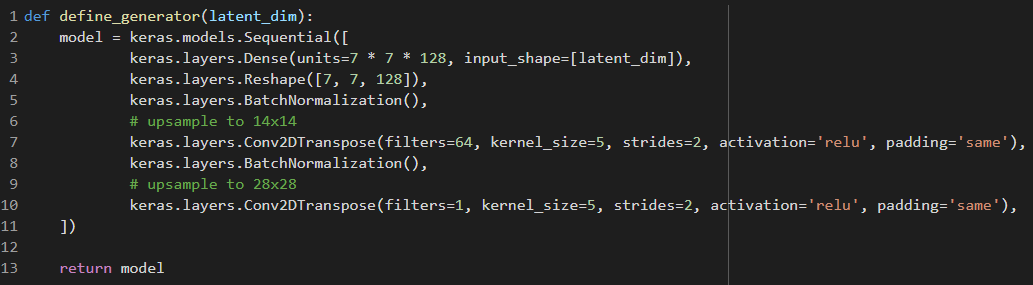
Arsitektur modelnya dapat digambarkan seperti berikut



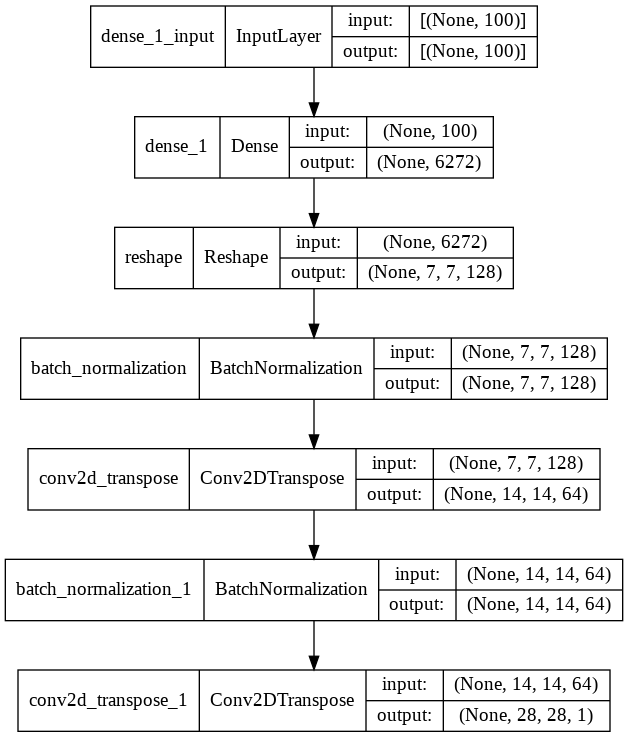
Gambar 1.2. Arsitektur discriminator

* 1. Fungsi untuk membuat model generator

Jika pada discriminator gambar masukan akan di downsampling untuk diklasifikasikan, maka dalam generator gambar masukan akan dilakukan upsampling hingga berukuran sama dengan gambar asli. Oleh karena itu perlu digunakan layer konvolusi transpose



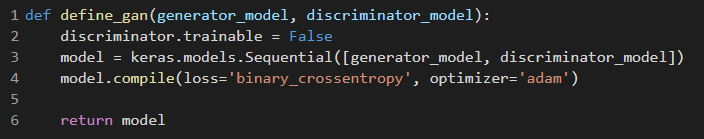
Arsitektur dari generator dapat digambarkan seperti berikut



Gambar 1.3. Arsitektur generator

* 1. Fungsi untuk membuat model GAN

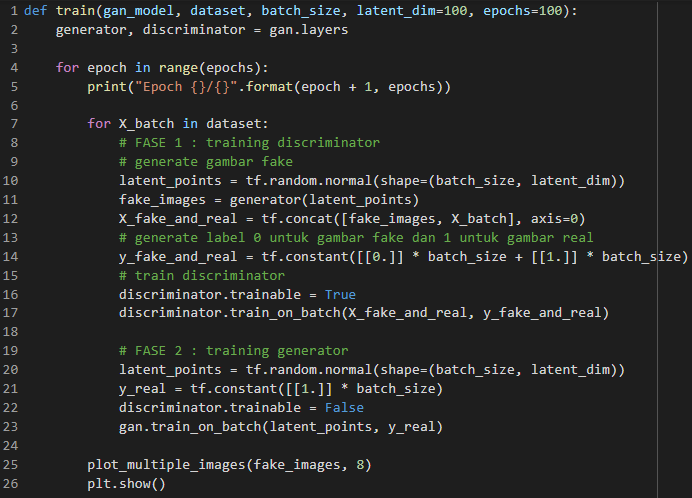
Model GAN dapat dibuat dengan menggabungkan model discriminator dan generator secara berurutan dalam suatu model Sequential. Discriminator bertugas melakukan klasifikasi biner yaitu membedakan antara gambar asli dan palsu, oleh karena itu digunakan binary\_crossentropy sebagai perhitungan loss.



* 1. Fungsi untuk melatih model GAN

Proses pelatihan GAN terdiri dari dua tahap, yaitu melatih discriminator kemudian generator.

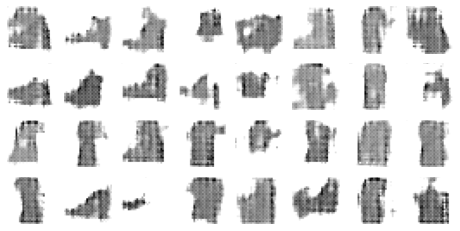
* + 1. Tahap 1 : generator akan membuat gambar palsu dari angka-angka acak yang dibuat dan diberi label 0. Gambar asli akan diambil dari dataset dan diberi label 1. Dengan dataset inilah discriminator akan dilatih untuk membedakan antara gambar yang asli dan palsu.
    2. Tahap 2 : pada tahap ini discriminator akan di freeze dan proses training dilakukan pada model GAN, tidak langsung dari generator. angka-angka random akan digenarate dan diberi label 1.



1. Hasil Proses Training

Model GAN dilatih dengan ukuran batch sebesar 32, dimensi latent sebesar 100 dan epochs sebanyak 50.

Hasil epoch ke 1



Gambar 1.4. hasil training GAN pada epoch ke 1

Hasil epoch ke 10



Gambar 1.5. hasil training GAN pada epoch ke 10

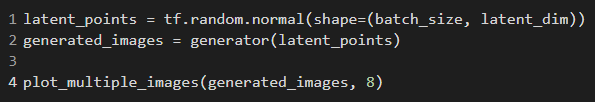
Hasil epoch ke 50



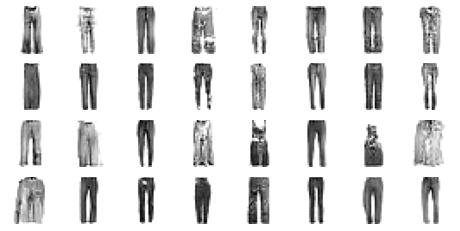
Gambar 1.6. hasil training GAN pada epoch ke 50

1. Membuat gambar baru

Pembuatan gambar baru dapat dilakukan dengan memasukkan angka-angka yang digenerate secara random pada generator. berikut kode program untuk melakukan prediksi dan menampilkan 8 diantaranya.



Hasilnya



Gambar 1.7. hasil gambar yang berhasil dibuat oleh GAN

**BAB IV**

**PENUTUP**

1. Kesimpulan

Dari penelitian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut

* 1. GAN dapat digunakan untuk menghasilkan gambar baru yang identik dengan gambar dari data latih.
  2. Kombinasi antara CNN dengan GAN atau bisa disebut DCGAN (Deep Convolutional Generative Adversarial Network) dapat menjadi model yang efektif untuk menyelesaikan permasalahan image generation.

1. Saran

Penulis menyadari bahwa penelitian ini masih memiliki banyak kekurangan dan keterbatasan. Berikut saran yang dapat digunkan untuk penelitian lebih lanjut

* 1. Menyertakan metrics yang digunakan untuk mengukur error ataupun akurasi dari model generator dan discriminator.
  2. Menyertakan label dalam proses generate gambar sehingga gambar yang dihasilkan akan lebih terarah.
  3. Memperbarui model supaya bisa memproses gambar RGB.

**DAFTAR PUSTAKA**

1. Ian Goodfellow, et all. 2014. Generative Adversarial Nets
2. Aurelien Geron. 2019. Hands-on Machine Learning with Scikit-Learn, Keras & Tensorflow Concepts, Tools and Techniques to Build Intelligent Systems. Sebastopol. O’Reilly Media, Inc.
3. <https://machinelearningmastery.com/how-to-develop-a-conditional-generative-adversarial-network-from-scratch/>
4. <https://developers.google.com/machine-learning/gan>