

Aplikasi Graf pada *Convolutional Neural Network* untuk Klasifikasi Sampah Otomatis berbasis *Image Processing*

Muhammad Zunan Alfikri - 13518019
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13518019@std.stei.itb.ac.id

Abstract—Sampah adalah sesuatu yang tidak digunakan, tidak dipakai, tidak disenangi atau sesuatu yang dibuang yang berasal dari kegiatan manusia dan tidak terjadi dengan sendirinya. Salah satu masalah besar di Indonesia adalah kurangnya kesadaran masyarakat untuk memilah sampah. Selain itu, pengolahan sampah di hilir (Tempat Pembuangan Akhir) masih manual. Seiring pertambahan penduduk, sampah di TPA semakin lama semakin menumpuk dan membeludak. Untuk menyelesaikan masalah pengolahan sampah tersebut, dibutuhkan teknologi pemilah sampah yang dapat memilah secara efektif dan efisien. Penulis menyelesaikan masalah ini dengan penerapan teori graf pada *Convolutional Neural Network* untuk klasifikasi sampah otomatis berbasis *image processing*.

Keywords—Klasifikasi Sampah, Graf, *Convolutional Neural Network*, *Image Processing*.

I. PENDAHULUAN

Sampah merupakan zat sisa yang sudah tidak terpakai dan dibuang. Terdapat banyak sampah yang dihasilkan setiap harinya di Indonesia. Jumlah sampah yang dihasilkan Indonesia pada tahun 2019 diprediksikan akan mencapai 67 juta ton sampah (AANasional, 24 Jan 2019). Hal ini sangat mengkhawatirkan mengingat jumlah sampah yang dihasilkan Indonesia per tahunnya sekitar 64 juta ton sampah (AANasional, 24/01/2019). Sampah - sampah yang menumpuk di tempat pembuangan akhir dapat mengotori lingkungan, baik jalanan, pemukiman, maupun kualitas tanah di daerah itu sendiri.

Bertambahnya jumlah sampah di Indonesia tidak disertai dengan bertambahnya jumlah TPA di Indonesia. Bahkan beberapa TPA di sekitar daerah Jawa telah penuh akan sampah. Banyak TPA di daerah kota Jakarta dan kota Depok telah melebihi kapasitas yang dimilikinya dalam menampung sampah, namun alternatif untuk penempatan sampah selanjutnya juga kurun belum ada (PikiranRakyat, 7 Nov 2018). Bahkan, TPA Cipeucang ditutup secara paksa karena tidak mampu membendung sampah yang masuk dan membuat masyarakat sekitar kurang nyaman (INDOPOS, 09 Mar 2019).

Oleh sebab itu, diperlukan suatu solusi dalam menyelesaikan permasalahan penumpukan sampah yang berlebih di Indonesia. Salah satu solusi yang telah berhasil dijalankan adalah solusi dari TPA Benowo yang terletak di Surabaya. TPA ini

melakukan pengolahan sampah - sampah organik menjadi energi listrik melalui biogas yang diperoleh dari sampah. Saat ini, teknologi PLTSa (Pembangkit Listrik Tenaga Sampah) yang dilakukan TPA Benowo masih dilakukan melalui pemanfaatan biogas (Tribunnews, 13 Juni 2014).

Selain itu, sampah - sampah non-organik dapat dimanfaatkan melalui proses daur ulang menjadi barang - barang yang bermanfaat. Jenis pengolahan sampah tersebut berbeda disesuaikan dengan jenis sampahnya sendiri. Karena hal tersebut, dibutuhkan proses pemilahan sampah yang baik dan efektif. Namun, ternyata pemilahan sampah di TPA Indonesia masih dilakukan secara manual dengan bantuan para pemulung yang berada di sekitar TPA (RMolBanten, 19 Juni 2019). Teknik ini jelas kurang efektif mengingat jumlah sampah yang dihasilkan Indonesia per harinya.

Untuk itu, harus ada cara yang baik dan efektif agar proses pemilahan sampah dapat dilakukan secara cepat dan tepat. Masalah ini dapat diatasi dengan mengotomasi kegiatan tersebut melalui klasifikasi gambar. Klasifikasi gambar yang dilakukan dapat menggunakan banyak cara. Dalam makalah ini, penulis memilih untuk melakukan pengklasifikasian gambar dengan memakai penerapan Graf pada *Convolutional Neural Network* (CNN). Hasil dari model klasifikasi sampah ini kemudian dapat diterapkan pada mesin pemilah sampah. Harapannya, dengan diterapkannya model ini, pengolahan sampah di TPA menjadi lebih cepat, efektif, dan efisien.

II. LANDASAN TEORI

A. Graf

Graf adalah suatu struktur yang digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antar objek - objek tersebut. Pada umumnya, objek-objek direpresentasikan sebagai titik, simpul, noktah, bulatan, atau bentuk lainnya. Hubungan antara objek-objek tersebut direpresentasikan dengan suatu garis.

Graf G didefinisikan sebagai pasangan himpunan (V, E) . V merupakan himpunan tidak kosong simpul-simpul (vertices) dan E merupakan himpunan sisi-sisi (edges). Hal tersebut dinotasikan dengan:

$$G = (V, E)$$

$$V = \{v_1, v_2, \dots, v_n\}$$

$$E = \{e_1, e_2, \dots, e_n\}$$

Dari definisi di atas, V tidak boleh kosong, sedangkan E boleh kosong. Graf yang hanya mempunyai sebuah simpul tanpa satu sisi pun dinamakan graf trivial. Simpul pada graf biasanya dinamai dengan huruf (a, b, c, \dots), angka ($1, 2, 3, \dots$), ataupun keduanya (a_1, a_2, a_3, \dots). Sisi e yang menghubungkan simpul u dengan simpul v dinotasikan dengan:

$$e = (u, v)$$

Graf dapat dibedakan menjadi beberapa jenis berdasarkan ada tidaknya sisi ganda atau gelang dan orientasi arah pada sisi.

Berdasarkan ada tidaknya sisi ganda atau gelang, graf dibedakan menjadi dua jenis, yaitu sebagai berikut.

1. Graf sederhana (*simple graph*)

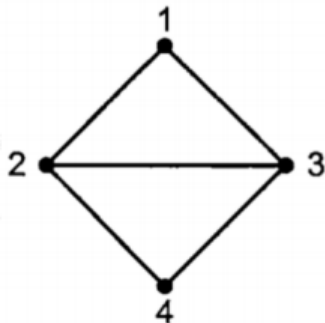
Graf sederhana tidak mengandung sisi ganda ataupun gelang. Pada graf sederhana, sisi merupakan pasangan tidak terurut (*unordered pairs*) sehingga (u, v) memiliki arti yang sama dengan (v, u) , dengan notasi:

$$(u, v) = (v, u)$$

Sisi pada graf ini menghubungkan dua simpul yang berbeda, dengan notasi:

$$e = (u, v)$$

$$u \neq v$$



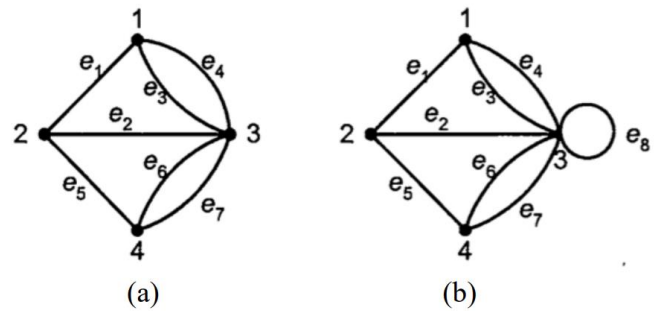
Gambar 1 Graf sederhana
Sumber : [1]

2. Graf tidak sederhana (*unsimple graph*)

Graf tidak sederhana mengandung sisi ganda atau gelang. Graf tidak sederhana dibagi menjadi dua jenis, yaitu graf ganda (*multigraph*) dan graf semu (*pseudograph*). Graf ganda adalah graf yang mengandung sisi ganda. Graf semu adalah graf yang mengandung gelang. Gelang merupakan sisi yang menghubungkan simpul dengan simpul itu sendiri, dengan notasi:

$$e = (u, v)$$

$$u = v$$



Gambar 2 (a) Graf ganda dan (b) graf semu dengan gelang pada e_8

Sumber : [1]

Berdasarkan orientasi arah pada sisi, graf dibedakan menjadi dua jenis yaitu sebagai berikut.

1. Graf tidak berarah (*undirected graph*)

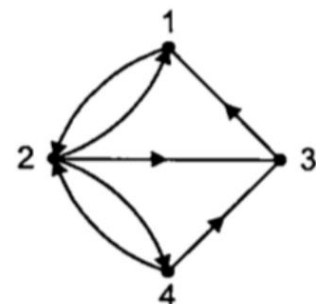
Graf tidak berarah adalah graf yang sisinya tidak mengandung arah. Urutan pasangan simpul yang dihubungkan suatu sisi tidak diperhatikan, sehingga: $(u, v) = (v, u)$ Contoh graf tidak berarah terdapat pada Gambar 1, Gambar 2 (a), dan Gambar 2 (b).

2. Graf berarah (*directed graph* atau *digraph*)

Graf berarah adalah graf yang sisinya memiliki orientasi arah. Orientasi arah umumnya direpresentasikan oleh tanda panah. Sisi berarah disebut busur (*arc*). Pada graf berarah, urutan pasangan simpul diperhatikan, sehingga:

$$(u, v) \neq (v, u)$$

Pada busur (u, v) , u adalah simpul asal (*initial vertex*), sedangkan v adalah simpul terminal (*terminal vertex*).



Gambar 3 Graf berarah
Sumber: [1]

Graf memiliki beberapa terminology (istilah) dasar. Beberapa istilah dasar yang berkaitan dengan graf adalah :

1. Ketetangaan (*Adjacency*)

Ketetangaan dalam graf didefinisikan sebagai dua buah simpul pada graf tak berarah G yang mana keduanya terhubung langsung dengan sebuah sisi. Hal ini dilambangkan dengan (u,v) yang berarti simpul u bertetangga dengan simpul v .

2. Beririsan (*Indidency*)
Beririsan dalam graf didefinisikan sebagai sembarang sisi $e = (u,v)$, menandakan e beririsan dengan simpul u dan simpul v .
3. Simpul Terpencil (*Isolated Vertex*)
Simpul terkecil dalam graf didefinisikan sebagai simpul yang tidak mempunyai sisi yang beririsan dengannya atau simpul yang tidak satupun bertetangga dengan simpul-simpul lainnya.
4. Graf Kosong (*Null Graph* atau *Empty Graph*)
Graf Kosong didefinisikan sebagai graf yang himpunan sisinya merupakan himpunan kosong.



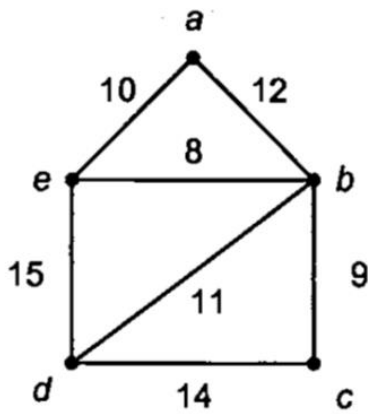
Gambar 4 Graf kosong
Sumber:[5]

5. Derajat (*Degree*)
Derajat adalah jumlah sisi yang beririsan dengan simpul pada graf tidak berarah. Pada gambar 1, simpul 1 dan 4 berderajat 2, sedangkan simpul 2 dan 3 berderajat 3, dengan notasi :

$$d(1) = d(4) = 2$$

$$d(2) = d(3) = 3$$
Derajat simpul terpencil adalah 0. Simpul dengan derajat 1 disebut simpul anting-anting (*pendant vertex*).
6. Lintasan (*Path*)
Lintasan yang panjangnya n dari simpul awal v_0 ke simpul tujuan v_n pada graf G adalah barisan berselangseling simpul, dengan notasi $v_0, e_0, v_1, e_1, v_2, e_2, \dots, v_{n-1}, e_{n-1}, v_n, e_n$ dengan sisi $e_1 = (v_0, v_1), e_2 = (v_1, v_2) \dots, e_n = (v_{n-1}, v_n)$ himpunan sisi $E = \{e_1, e_2, \dots, e_n\}$. Lintasan juga dapat disebut jalur. Panjang lintasan adalah jumlah sisi dalam lintasan tersebut. Lintasan sederhana (*simple path*) adalah lintasan dengan semua simpulnya berbeda. Lintasan tertutup (*closed path*) adalah lintasan yang berawal dan berakhir pada simpul yang sama, sedangkan lintasan terbuka (*opened path*) adalah lintasan yang berawal dan berakhir pada simpul yang berbeda. Pada Gambar 1, lintasan 1, 2, 3, 4 adalah lintasan dengan barisan sisi (1, 2), (2, 3), dan (3, 4) dengan panjang lintasan sebesar 3.

7. Siklus (*Cycle*) atau Sirkuit (*Circuit*)
Siklus atau sirkuit adalah lintasan yang berawal dan berakhir pada simpul yang sama. Siklus atau sirkuit termasuk lintasan tertutup. Panjang sirkuit adalah jumlah sisi pada sirkuit tersebut. Pada Gambar 1, lintasan 1, 2, 3, 1 merupakan sebuah siklus atau sirkuit dengan panjang 3. Lintasan tersebut berawal dan berakhir pada simpul 1.
8. Terhubung (*Connected*)
Dua buah simpul pada dikatakan terhubung jika terdapat lintasan dari simpul awal ke simpul akhir. Pada graf tidak berarah, graf disebut graf terhubung (*connected graf*) jika untuk setiap pasang simpul u dan v pada himpunan V , terdapat lintasan dari u ke v , sedangkan graf disebut graf tidak terhubung (*disconnected graf*) jika untuk setiap pasang simpul u dan v pada himpunan V , tidak terdapat lintasan dari u ke v . Pada graf berarah, graf disebut graf terhubung jika graf tidak berarahnya terhubung. Pada graf berarah, graf terhubung dibedakan menjadi dua jenis, yaitu graf terhubung kuat (*strongly connected*) dan graf lemah (*weakly connected*). Graf terhubung kuat adalah graf yang memiliki lintasan dari u ke v dan juga lintasan dari v ke u . Graf terhubung lemah adalah graf yang memiliki lintasan dari u ke v , tetapi tidak memiliki lintasan dari v ke u .
9. Upagraf (*Subgraph*) dan Komplemen Upagraf
Sebuah graf $G_1 = (V_1, E_1)$ dikatakan sebagai upagraf dari graf $G = (V, E)$ jika $V_1 \subseteq V$ dan $E_1 \subseteq E$. Sebuah graf $G_2 = (V_2, E_2)$ dikatakan komplemen dari upagraf G terhadap G_1 jika $E_2 = E - E_1$ dan V_2 merupakan himpunan simpul yang anggotanya bersisian dengannya. Komponen graf (*connected component*) adalah jumlah maksimum upagraf terhubung dalam graf tersebut. Pada graf berarah, komponen terhubung kuat (*strongly connected component*) adalah jumlah maksimum upagraf yang terhubung kuat. Graf pada Gambar 2 (a) merupakan upagraf dari graf pada Gambar 2 (b)
10. Upagraf Rentang (*Spanning Subgraph*)
Upagraf $G_1 = (V_1, E_1)$ dari graf $G = (V, E)$ dikatakan upagraf merentang jika $V_1 = V$. Hal tersebut berarti G_1 mengandung semua simpul dari G .
11. *Cut-set*
Cut-set dari graf terhubung G adalah himpunan sisi yang jika dihilangkan dari G menyebabkan G tidak terhubung sehingga cut-set dua buah komponen. Istilah lain dari cut-set adalah jembatan (*bridge*).
12. Graf Berbobot (*Weighted Graph*)
Graf berbobot adalah graf yang pada setiap sisinya diberi sebuah bobot. Graf berbobot memiliki istilah lain, yaitu graf berlabel.

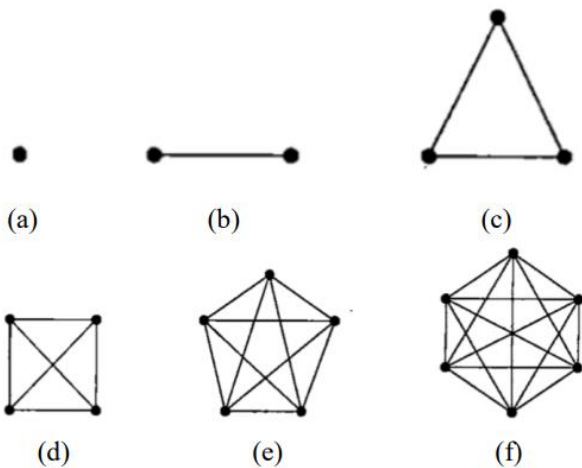


Gambar 5 Graf berbobot
Sumber : [1]

Graf graf yang memiliki bentuk atau pola khusus disebut graf khusus. Graf khusus ini banyak ditemukan penerapannya. Graf-graf khusus tersebut antara lain sebagai berikut.

1. Graf Lengkap (*Complete Graph*)

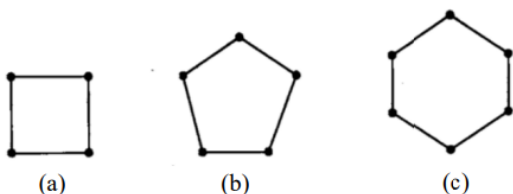
Graf lengkap adalah graf sederhana yang setiap simpulnya memiliki sisi ke semua simpul lainnya. Artinya, setiap simpul pada graf lengkap bertetangga dengan simpul lainnya. Graf lengkap dengan n buah simpul dinotasikan dengan K_n . Derajat setiap simpul pada K_n adalah $n - 1$.



Gambar 6 Graf lengkap (a) K_1 (graf trivial), (b) K_2 , (c) K_3 , (d) K_4 , (e) K_5 , dan (f) K_6
Sumber : [1]

2. Graf Lingkaran

Graf lingkaran adalah graf sederhana yang derajat setiap simpulnya berjumlah dua. Graf lingkaran dengan n buah simpul dinotasikan dengan C_n . Pada Gambar 6 (c), graf K_3 termasuk graf lingkaran.



Gambar 7 Graf lingkaran (a) C_4 , (b) C_5 , dan (c) C_6
Sumber : [1]

3. Graf Teratur (*Regular Graph*)

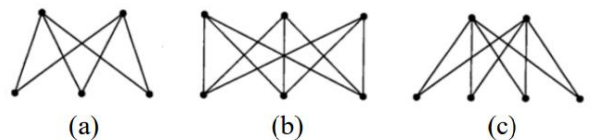
Graf teratur adalah graf yang setiap simpulnya berderajat sama. Graf teratur derajat r adalah graf yang setiap simpulnya berderajat r . Graf lengkap dan graf lingkaran termasuk graf teratur. Jumlah sisi e pada graf teratur dengan n buah simpul dirumuskan dengan:

$$e = \frac{nr}{2}$$

4. Graf Bipartit (*Biparte Graph*)

Graf bipartit adalah graf yang himpunan simpulnya dapat dikelompokkan menjadi dua himpunan bagian. Graf bipartit G dengan himpunan bagian simpul V_1 dan V_2 dinotasikan dengan $G(V_1, V_2)$. Setiap pasang simpul di himpunan bagian tidak bertetangga dengan simpul lain di himpunan bagian tersebut. Graf bipartit lengkap (*complete bipartite graph*) adalah graf bipartit yang setiap simpul di himpunan bagian satu bertetangga dengan semua simpul di himpunan bagian lainnya. Graf bipartite lengkap dinotasikan dengan $K_{m,n}$ dengan m dan n jumlah simpul di masing-masing himpunan bagian. Jumlah sisi e pada graf bipartit lengkap dirumuskan dengan:

$$e = m n$$



Gambar 8 Graf bipartit (a) $K_{2,3}$, (b) $K_{3,3}$, dan (c) $K_{2,4}$
Sumber : [1]

B. Artificial Neural Network (ANN)

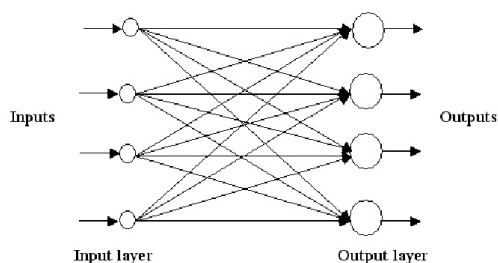
Artificial neural network atau jaringan saraf tiruan adalah jaringan dari sekelompok unit pemroses yang dimodelkan berdasarkan jaringan saraf manusia. ANN merupakan system adaptif yang dapat mengubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal.

Neuron-neuron dalam *neural network* disusun dalam grup yang disebut dengan layer (lapis). Pada dasarnya ANN terdiri dari 3 lapisan (layer), yaitu input layer, process, dan output layer. Pada input layer berisi variable data input, process layer berisi langkah pengenalan objek dan output layer berisi hasil pengenalan suatu objek.

Pada umumnya, artificial neural network terdapat 2 jenis yang sering digunakan, yaitu : *Sinle-Layer Neural Network* dan *Multilayer Neural Network*.

1. *Single-Layer Neural Network*

Pada *single-Layer Neural Network*, *input layer* terhubung langsung ke *output layer*. Kelemahan dari jenis ini adalah hanya bias digunakan pada kasus sederhana.

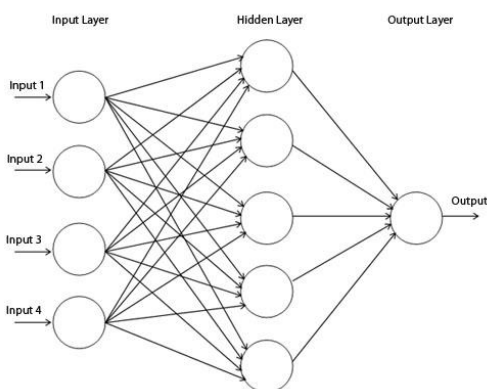


Gambar 9 Single-Layer Neural Network

Sumber : [researchgate.net/figure/A-single-layer-feed-forward-neural-network_fig1_228394623](https://www.researchgate.net/figure/A-single-layer-feed-forward-neural-network_fig1_228394623)

2. Multilayer Neural Network

Pada *Multilayer Neural Network* terdapat hidden layer yang terletak diantara input layer dan output layer.



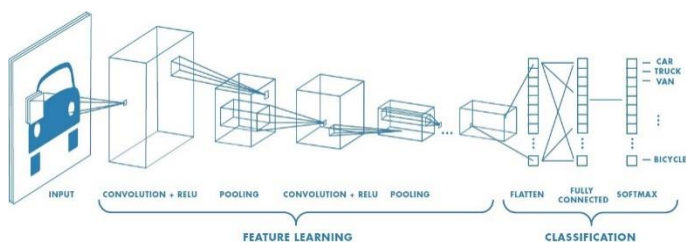
Gambar 10 Multilayer Neural Network

Sumber : [nicolamanzini.com/single-hidden-layer-neural-network/](https://www.nicolamanzini.com/single-hidden-layer-neural-network/)

C. Convolutional Neural Network (CNN)

Convolutional Neural Network atau CNN merupakan salah satu jenis dari *Artificial Neural Network*. CNN termasuk ke dalam jenis *Multilayer Neural Network*. CNN bisa digunakan untuk mendeteksi dan mengenali objek pada sebuah image. CNN sendiri telah banyak digunakan pada data-data berbasis image.

Secara garis besar CNN tidak jauh berbeda dengan neural network biasanya. CNN terdiri dari neuron - neuron yang memiliki bias, weight, serta fungsi aktivasi. Yang membedakan CNN dengan neural network lainnya terdapat dalam arsitekturnya, dimana dalam CNN terjadi pengekstraksian fitur - fitur penting lewat metode konvolusi, selanjutnya diikuti dengan proses klasifikasi.



Gambar 11 Ilustrasi Layer CNN

Sumber : [medium.com/@samuelsena/pengenalan-deep-](https://www.medium.com/@samuelsena/pengenalan-deep-)

learning-part-7-convolutional-neural-network-cnn-b003b477dc94

Layar konvolusi dari CNN sendiri digunakan untuk mengurangi jumlah kekompleksan data masukan dengan 2D convolution layers. Oleh sebab itu, jenis model *neural network* ini sangat ideal dalam melakukan klasifikasi gambar.

Dalam membuat model *Convolutional Neural Network*, penulis menggunakan beberapa *layers*. Layer berperan sebagai suatu tahapan dalam machine learning yang akan mengoptimasi kemampuan mesin tersebut. Terdapat beberapa layer yang digunakan pada model CNN ini.

1. Conv2D

Layer Conv2D berperan untuk mendapatkan fitur dari suatu gambar dengan melakukan scanning pada gambar tersebut menggunakan suatu filter yang telah ditentukan. Pada percobaan ini, penulis mengatur filter, *kernel_size*, dan *activation* pada layer Conv2D yang digunakan.

Filter merupakan ukuran dimensi output yang diinginkan. Penulis mengatur ukuran dimensi keluaran pada range 32 dan 64.

Kernel_size merupakan sebuah integer atau tuple yang akan menentukan ukuran dari 1D Convolutional Window. Pada percobaan ini, penulis mengatur *kernel_size* dengan ukuran (3,3).

2. MaxPooling2D

Layer MaxPooling2D berperan untuk mengubah ukuran fitur dari gambar yang berupa matriks. Pada matriks, akan diperoleh fitur-fitur yang penting yang akan dimasukkan ke matriks yang baru yang ukurannya lebih kecil. Ukuran matriks dibuat menjadi lebih kecil agar mesin dapat belajar lebih cepat. Pada MaxPooling2D dikenal istilah *pool_size*. *Pool_size* merupakan suatu matriks kecil yang menyatakan ukuran kelompok-kelompok kecil pada matriks besar yang akan diekstraksi fitur pentingnya. Penulis mengatur *pool_size* dengan ukuran 2x2.

3. Dropout

Layer Dropout berperan sebagai pencegah *overfitting* pada mesin. Cara kerja Dropout adalah dengan mematikan beberapa node penghubung, oleh karena itu, node sebelumnya harus mencari node yang lain untuk dapat melanjutkan perjalanannya ke layer terakhir.

4. Flatten

Flatten berperan untuk menghubungkan *convolutional layer* dengan *fully connected layer*, yang kita sebut dengan Dense. Flatten akan menghasilkan vektor-vektor sebagai representasi tiap matriks pada layer sebelumnya. Hal ini dilakukan untuk memudahkan mesin dalam mengklasifikasi nantinya.

5. Dense

Dense berperan untuk mengklasifikasi vektor-vektor pada layer sebelumnya ke beberapa jenis. Pada

percobaan kali ini, dense terakhir berjumlah 7 buah, karena terdapat 7 klasifikasi pada percobaan ini, yaitu plastic bottle, metal, cardboard, paper, glass, organic, dan other trash.

6. *Activation*

Pada layer-layer seperti Conv2D dan Dense, penulis menggunakan *activation*. *Activation* merupakan fungsi yang dapat mengoperasikan nilai dengan operasi non-linear yang diperlukan dalam pembelajaran mesin.

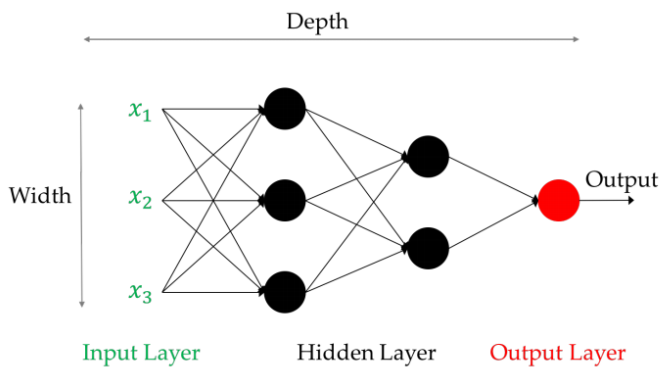
Activation yang digunakan pada percobaan ini adalah ReLu dan Softmax.

ReLu merupakan singkatan dari *Rectified Linear Unit*, yang berperan dengan cara menghasilkan 0 untuk setiap nilai negatif yang ada dan tidak mengubah nilainya jika nilai tersebut bersifat positif. Fungsi ReLu dilakukan untuk meningkatkan non-linearity dari gambar, karena gambar pada dasarnya tidak linear. Sedangkan softmax diimplementasikan di layer dense terakhir untuk menampilkan distribusi probabilitas setiap kelas oleh suatu gambar.

Arsitektur CNN adalah komposisi *layer-layer* hasil dari percobaan dengan berbagai persoalan. CNN Memiliki jenis-jenis arsitektur yang berbeda beda dan arsitektur ini telah teruji dan digunakan di banyak persoalan. Contoh dari arsitektur CNN yaitu VGG, LeNet, AlexNet, MobileNet, Inception, Xception, dan lain sebagainya.

III. APLIKASI GRAF PADA CONVOLUTIONAL NEURAL NETWORK (CNN)

Convolutional Neural Network pada dasarnya tersusun dalam bentuk graf. CNN memiliki *neuron* atau *vertex* jika di dalam graph dan memiliki sinapsis atau edge jika di dalam graph. Berikut ilustrasi dari permodelan graf di dalam CNN.



Gambar 12 Ilustrasi Graf CNN
Sumber : [3]

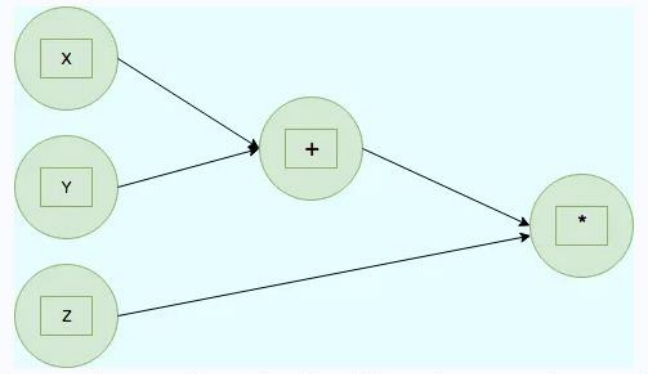
Pada gambar tersebut, antar layer CNN bisa di pandang sebagai kumpulan graf biparte yang saling terhubung. Kedalaman graf juga mengacu pada banyaknya layer suatu CNN.

Untuk dapat menghasilkan *output*, satu parameter input atau biasa disebut *feature* harus memiliki lintasan untuk sampai ke node output. Lintasan ini melewati tepat satu node dalam tiap

layer.

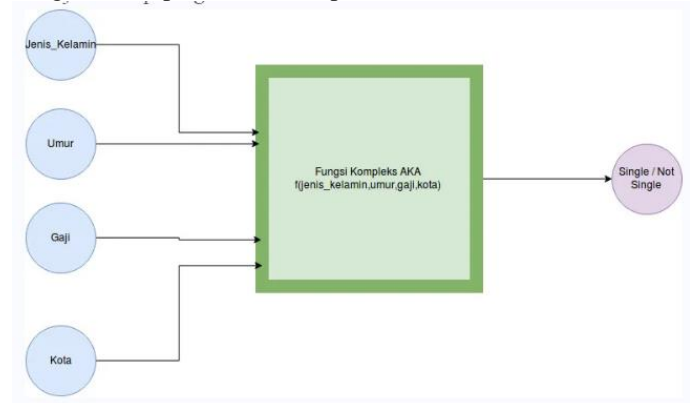
Dalam perhitungannya, CNN juga menerapkan Graf Komputasi sebagai fungsi untuk menghasilkan *output*. Graf Komputasi adalah cara merepresentasikan fungsi matematika dengan Bahasa yang mirip dengan Teori Graf. Didalam Graph Komputasi setiap Node itu bisa berupa suatu nilai placeholder atau merupakan suatu fungsi matematika dan Edge berfungsi sebagai tempat aliran data yang diperlukan untuk menyelesaikan tugas tertentu.

Misalnya kita ingin membuat suatu fungsi $f(x,y,z) = (x + y) * z$. Jika diterjemahkan ke dalam Graph Komputasi, fungsi tersebut menjadi seperti berikut .



Gambar 13 Ilustrasi Graf Komputasi
Sumber : [4]

Di dalam CNN, sebenarnya dapat di ilustrasikan sebagai fungsi juga, dengan parameter input adalah fitur-fitur dari gambar dan output adalah kelas yang ingin di prediksi. Hal tersebut tentunya menerapkan Graf Komputasi sebagai dasarnya. Berikut ilustrasi kasus dari cara mendeteksi suatu karyawan sudah menikah atau belum menggunakan CNN dengan menerapkan Graf Komputasi.



Gambar 14 Ilustrasi Permasalahan CNN
Sumber : [4]

IV. KLASIFIKASI SAMPAH MENGGUNAKAN CNN

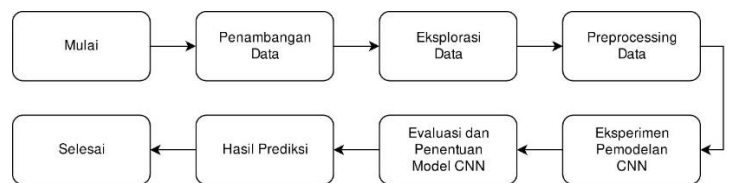


Diagram 1 Alur Klasifikasi Sampah Menggunakan CNN

Sumber : Dokumentasi pribadi

A. Penambahan Data

Data yang penulis peroleh berasal dari data open source yang ada di github. Data ini diambil dengan meletakkan objek pada sebuah papan tulis di sebuah ruangan yang terang. Gambar yang diperoleh memiliki ukuran 512 x 384 px yang kemudian akan diubah ukurannya menjadi 256 x 256 px. Gambar-gambar tersebut merupakan foto diambil menggunakan gawai.

B. Eksplorasi Data

Gambar sampah yang penulis peroleh diklasifikasikan ke dalam tujuh kelas berdasarkan cara pengolahannya. Tujuh kelas data gambar tersebut yaitu kaca, kertas, karton, aluminium, plastik, sampah organik, dan sampah lainnya.

Pada percobaan ini, penulis menggunakan 3089 data gambar. Beberapa sampel gambar yang penulis gunakan antara lain :

1. Kaca, dengan sampel gambar seperti dibawah ini.



Gambar 15 Sampel data kaca

2. Kertas, dengan sampel gambar seperti dibawah ini.



Gambar 16. Sampel data kertas

3. Karton, dengan sampel gambar seperti dibawah ini.



Gambar 17 Sampel data Karton

4. Aluminium, dengan sampel gambar seperti dibawah ini.



Gambar 18 Sampel data Aluminium

5. Plastik, dengan sampel gambar seperti dibawah ini.



Gambar 19 Sampel data Plastik

6. Sampah Organik, dengan sampel gambar seperti dibawah ini.



Gambar 20 Sampel data Organik

7. Sampah lainnya

Sampah lainnya merupakan tipe sampah yang tidak termasuk dalam 6 jenis sampah sebelumnya. Sampel dari sampah lainnya seperti gambar dibawah ini.



Gambar 21 Sampel data sampah lainnya

Jumlah setiap jenis data dapat dilihat pada tabel dibawah.

Jenis	Jumlah
Kaca	531
Kertas	624
Karton	433
Aluminium	440
Plastik	482
Organik	412
Lainnya	167

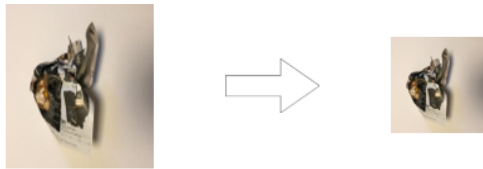
Tabel 1 Jumlah Gambar Tiap Jenis Sampah

C. Data Preprocessing

Sebelum dilakukan eksperimen dengan pemodelan CNN, data gambar terlebih dahulu diproses agar lebih mudah diolah. Preprocessing data gambar dilakukan dengan menggunakan bantuan ImageDataGenerator yang telah tersedia di library Tensorflow. Tahap preprocessing data gambar dibagi menjadi 2 (dua) tahap, yakni:

1. *Rescale*

Di tahap ini, penulis melakukan perubahan bentuk dimensi gambar dari (512 x 384) px menjadi (256 x 256) px. Rescaling ini dilakukan untuk mengurangi jumlah data yang perlu diproses sewaktu pemodelan dilangsungkan.



Gambar 22 Ilustrasi Rescale

2. Data Augmentation

Tahap ini perlu dilakukan ketika menggunakan neural network dan model deep learning. Hal ini dikarenakan dalam melakukan training terhadap model CNN diperlukan data yang sangat banyak. Ketika data yang digunakan kurang mencapai yang diperlukan, maka dibutuhkan augmentasi data. Untuk itu, penulis memakai ImageDataGenerator untuk melakukan augmentasi data dengan memakai ImageDataGenerator yang telah disediakan di library Tensorflow.

D. Eksperimen Pemodelan CNN

Dalam eksperimen kali ini, penulis membandingkan 5 (lima) macam arsitektur CNN yaitu VGG 8 blok, Inception, Resnet, Exception, dan VGG 16 blok. Eksperimen ini dilakukan dengan tujuan utama menemukan model terbaik yang dapat digunakan untuk mengklasifikasikan sampah dari kelima macam arsitektur CNN.

Pada Eksperimen kali ini, penulis membandingkan beberapa arsitektur tersebut dengan melatih model dengan 25 epoch. Berikut hasil perbandingan skor validasi dan skor loss dari masing masing arsitektur.

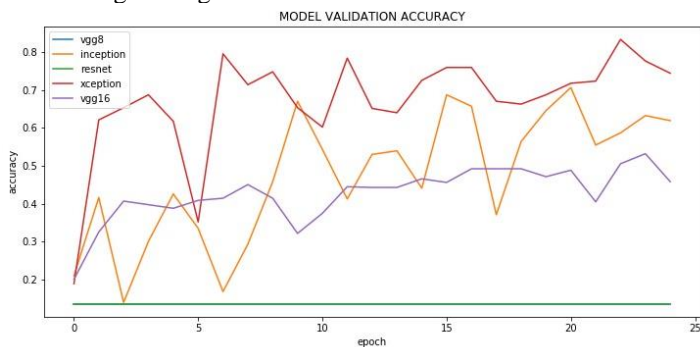
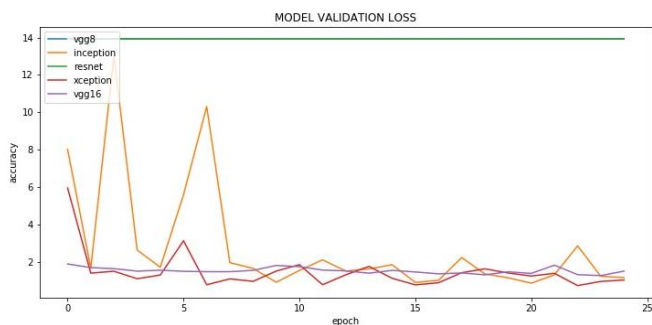


Diagram 2 Perbandingan Validation Accuracy



Diragram 3 Perbandingan Validation Loss

E. Evaluasi dan Penentuan Model CNN

Dari grafik tersebut, dapat dilihat bahwa Xception memiliki nilai tertinggi untuk validation accuracy dan memiliki nilai yang

cenderung menurun untuk loss accuracy. Maka, dari kelima arsitektur model CNN tersebut, dapat disimpulkan bahwa model terbaik untuk metode learning from scratch adalah arsitektur model Xception.

F. Hasil Prediksi

Dalam tahap prediction, model digunakan untuk memprediksi data pada 291 gambar yang telah disiapkan. Model dilatih dengan epoch yang lebih tinggi yaitu 100 epoch. Berikut hasil confusion matrix dan tabel dari hasil prediksi yang telah dilakukan.

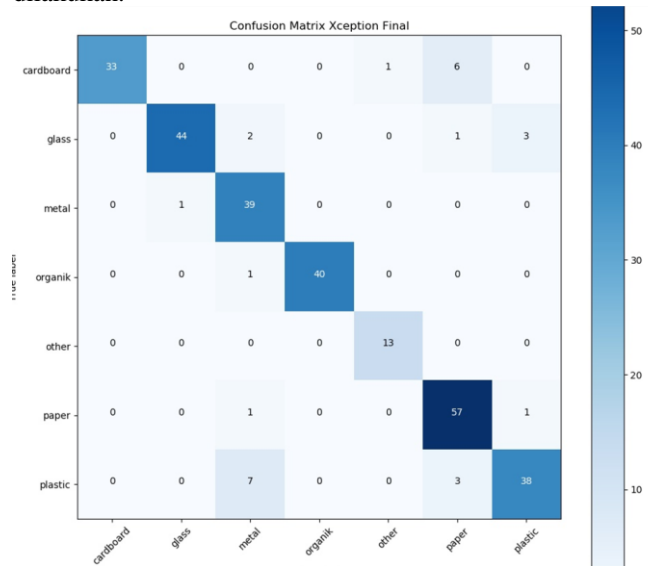


Diagram 4 Confusion Matrix Hasil Prediksi

Jenis Sampah	Jumlah Benar	Jumlah Salah
Kardus	33	7
Kaca	44	6
Aluminium	39	1
Organik	40	1
Lainnya	13	0
Kertas	57	2
Plastik	38	10
Total	264	27

Tabel 2 Hasil Prediksi

Dari hasil prediksi tersebut, model Xception dapat memprediksi 264 gambar benar dari 291 gambar dan 27 gambar salah. Skor untuk hasil prediksi ini yaitu 90,7%. Waktu yang digunakan untuk memprediksi satu gambar yaitu 0,03 detik.

V. KESIMPULAN

Simpulan yang dapat diambil dari makalah ini adalah terdapat banyak produk dari Matematika Diskrit yang aplikatif di berbagai bidang. Salah satunya yaitu aplikasi graf pada CNN untuk Klasifikasi Sampah Otomatis. Aplikasi graf ini bisa menyelesaikan salahsatu masalah lingkungan di Indonesia yaitu sampah. Dengan Aplikasi graf pada CNN, sampah menjadi terklasifikasi lebih efektif dengan tingkat akurasi 90,7% dan kecepatan memprediksi 0.03 detik untuk setiap gambar.

VI. UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kepada Allah SWT karena atas pertolongan, izin, berkat, rahmat, dan hidayah-Nya, penulis dapat menyelesaikan tugas makalah ini dengan lancar. Pada kesempatan ini, dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T. selaku dosen Kelas 01 Mata Kuliah IF2120 Matematika Diskrit atas segala ilmu dan bimbingannya. Penulis juga ingin menyampaikan terima kasih kepada kedua orang tua yang selalu memberikan dukungan kepada penulis. Ucapan terima kasih juga penulis ucapkan kepada teman-teman yang telah memberikan bantuan dan dukungan dalam penyusunan makalah ini.

REFERENCES

- [1] Munir, Rinaldi. 2016. *Matematika Diskrit*. Edisi Revisi Keenam. Bandung: Informatika Bandung.
- [2] Geron, Aurelion. 2017. *Hands-On Machine Learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. United States of America : O'Reilly Media, Inc.
- [3] Putra, Jan Wira Gotama. 2018. *Pengenalan Pembelajaran Mesin dan Deep Learning*.
- [4] Deep Learning Sebagai Graf Komputasi . 2018. <http://omahti.web.id/blog/deep-learning-sebagai-graf-komputasi/>. Diakses 5 Desember 2019 pada pukul 23:23 WIB.
- [5] SuperDataScienceTeam.2018.*Convolutional Neural Networks (CNN): Step 1(b) ReLU Layer*. <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-1b-relu-layer/> . Diakses 25 November pada pukul 19.00 WIB
- [6] Rosebrock, Adrian.2018.*Keras Conv2D and Convolutional LayersTensorFlow Conv2D Layers: A Practical Guide*. <https://www.pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/>. Diakses 27 November pada pukul 07.27 WIB

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 5 Desember 2019



Muhammad Zunan Alfikri 13518019