

# Penerapan Model Generatif Pada Kerangka Kerja *Deep Learning* Untuk Menerjemahkan Citra Sketsa Daun Menjadi Citra Alami Daun

Muhammad Eka Purbaya<sup>1</sup>, Anggit Ferdita Nugraha<sup>2</sup>

<sup>1</sup> Program Studi Magister Teknologi Informasi, DTETI, UGM

<sup>2</sup> Program Studi Magister Teknik Elektro, DTETI, UGM

e-mail: [1.muhammad.eka.p@mail.ugm.ac.id](mailto:1.muhammad.eka.p@mail.ugm.ac.id), [2.anggitferdita.mti15@mail.ugm.ac.id](mailto:2.anggitferdita.mti15@mail.ugm.ac.id),

**Abstrak**—Sebagai salah satu objek penelitian yang dikembangkan untuk klasifikasi spesies tanaman melalui proses identifikasi citra daun. Seringkali citra daun yang digunakan tidak dalam kondisi yang ideal untuk dikaji karena banyaknya gangguan terutama bila pengujian dilakukan menggunakan objek sketsa yang tentunya memiliki tingkat keserupaan yang jauh dari bentuk alaminya. Sehingga diperlukan adanya perbaikan struktur citra sketsa daun agar mempermudah klasifikasi tanaman dengan metode penerjemahan citra. Metode yang dapat digunakan untuk mengubah menerjemahkan citra adalah dengan memanfaatkan model generatif pada kerangka kerja *deep learning*. Metode yang diusulkan pada model generatif ini adalah dengan memanfaatkan jaringan permusuhan model antara model generatif dengan arsitektur U-Net dan model diskriminatif dengan arsitektur PatchGANs yang lebih dikenal dengan metode CoGANs. Kerangka kerja *deep learning* tersebut dilatih selama 50 *epoch* kemudian dianalisis secara kualitatif dan kuantitatif. Pada hasil kualitatif, semakin lama pembelajaran maka hasil penerjemahan akan semakin baik. Sedangkan pada penilaian kuantitatif model generatif melalui MAE *loss function* menghasilkan nilai 0,0926 ( $\pm$  0,0068) yang menunjukkan bahwa model generatif tersebut menghasilkan penerjemahan citra yang mampu mendekati citra aslinya

**Kata kunci**—*Deep Learning, Image Translation, CoGANs, GANs, U-Net.*

## I. PENDAHULUAN

Daun merupakan bagian dari morfologi tanaman yang mudah ditemukan dan tersedia pada berbagai macam musim. Keunggulan tersebut menjadikan daun lebih banyak digunakan sebagai objek penelitian klasifikasi tanaman daripada menggunakan objek lain seperti bunga, buah dan biji tanaman [1], [2]. Sebelum dilakukan pengujian model klasifikasi, citra daun perlu melalui tahapan pra-pemrosesan terlebih dahulu berdasarkan mekanisme pengolahan citra yang bertujuan untuk meningkatkan kualitas suatu citra. Kendala yang umum terjadi pada tahapan ini adalah citra yang digunakan tidak dalam kondisi ideal untuk diproses akibat *noise* yang bisa berupa bayangan, kualitas citra yang rusak atau kabur sehingga mempengaruhi proses analisis yang akan dilakukan. Selain kondisi citra yang ideal, penggunaan objek klasifikasi tanaman lainnya seperti citra sketsa daun turut menjadi kendala pada tahap pra-pemrosesan dikarenakan unsur pembentuk daun pada citra tersebut belum lengkap. Sehingga diperlukan adanya metode perbaikan struktur citra sketsa daun agar memiliki bentuk yang serupa dengan citra alaminya sehingga membantu mempermudah proses klasifikasi tanaman.

Dalam menerapkan upaya tersebut, diperlukan suatu metode atau teknologi untuk mengubah citra sketsa daun menjadi citra alami daun yang dikenal sebagai proses penerjemahan citra yang memanfaatkan model generatif dengan memakai kerangka kerja *deep learning*. Model generatif yang diketahui sesuai untuk menangani masalah penerjemahan citra adalah

*Conditional Generative Adversarial Networks* (CoGANs) [3]. Metode tersebut merupakan perkembangan dari metode *Generative Adversarial Networks* (GANs) yang pertama kali diperkenalkan oleh Goodfellow dkk [4] melalui konsep *game theory* yang menggambarkan adanya interaksi antara kedua pemain sebagai perlawanan individu yang rasional.

Penelitian mengenai penerjemahan citra sketsa daun menjadi citra alami daun belum pernah dilakukan sebelumnya, namun terdapat beberapa penelitian yang masih memiliki relevansi pada metode yang akan digunakan pada penelitian ini yaitu U-Net [5], GANs [4], [6] dan CoGANs [3], [7]. Metode U-Net digunakan oleh Olaf Ronneberger dkk [5] untuk segmentasi citra mikroskopik dengan cara memprediksi pelokalan citra untuk setiap piksel agar diperoleh representasi yang mudah dipahami untuk tujuan analisis. U-Net merupakan pengembangan metode *Autoencoders* yang terdiri dari proses *encoding* dan *decoding* dengan modifikasi pada arsitektur jaringan konvolusi dan tambahan *local skip-connection* sebagai solusi dari kekurangan *Autoencoder* yang mengalami kehilangan beberapa resolusi saat proses *encoding* sehingga cenderung memberikan hasil citra yang kabur.

Pengembangan metode kondisional pada GANs [4] yang dilakukan Mehdi Mirza dkk [3] dilakukan dengan menambahkan label  $y$  pada suatu data observasi  $x$  agar hasil gambar yang dihasilkan sesuai dengan umpan yang diberikan oleh label  $y$ . Metode CoGANs dapat menghasilkan citra digit tertentu sesuai dengan label yang diinginkan untuk kasus label uni-modal. Sedangkan untuk kasus dengan label multi-modal, citra

masukan akan secara otomatis diberi tag sesuai peristiwa yang dimiliki citra tersebut. Kekurangan pada penelitian ini terletak pada desain arsitektur CoGANs yang hanya menggunakan *hidden layer* sebagai pembobotan dan hasil gambar yang masih belum jelas namun arti digit yang ditampilkan masih dapat dipahami. Jenis masukan untuk pelatihan masih menggunakan *noise* dengan dimensi 100 sehingga belum bisa digunakan untuk jenis masukan berupa gambar.

Phillip Isola dkk [7] melakukan penyempurnaan metode CoGANs [3] dengan pemilihan desain arsitektur U-Net [5] untuk model generatif dan PatchGANs untuk model diskriminatif. PatchGANs merupakan modifikasi *Deep Convolutional Generative Adversarial Networks* DCGANs [8] untuk mengklasifikasi setiap bagian dari piksel gambar yang dihasilkan oleh model generatif sehingga dinamakan PatchGANs. Beberapa kasus yang ditangani oleh metode usulan Isola dkk [7] adalah menerjemahkan citra pada *dataset Cityscapes, Architecture, Aerial Photograph, Shoes & Bag* dan foto yang memiliki kerusakan piksel.

Hasil yang diharapkan pada penelitian ini adalah citra sketsa daun dapat diterjemahkan menjadi citra alami daun menggunakan metode CoGANs [7] dengan komposisi model generatif menggunakan algoritme U-Net dan model diskriminatif menggunakan algoritme PatchGANs. Untuk melihat kinerja model CoGANs [7] dalam menghasilkan citra terjemahan, maka dilakukan penilaian kualitatif dengan membandingkan hasil presisi citra terjemahan dengan *ground truth* dan penilaian kuantitatif menggunakan *L1 loss function*. Kontribusi yang kami berikan melalui penelitian ini adalah menerapkan metode CoGANs sebagai model perlawanan antara model generatif dan model diskriminatif untuk mengatasi permasalahan penerjemahan citra pada data sketsa daun menjadi citra alami daun.

## II. LANDASAN TEORI

### A. Model Generatif

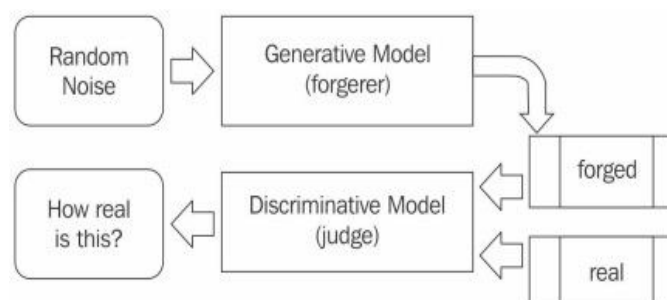
Dalam ilmu statistik, model generatif memodelkan probabilitas observasi  $X$  untuk menghasilkan instansiasi acak  $P(X|Y)$  sehingga model generatif dapat

menghasilkan data baru yang serupa dengan data yang asli. Data yang baru dapat ditransformasikan ke  $P(Y|X)$  untuk tahapan klasifikasi oleh model diskriminatif. Agar lebih banyak informasi yang dihasilkan, model generatif dapat melakukan rujukan ke model distribusi probabilitas gabungan  $X$  dan  $Y$  yang dianotasikan  $P(X, Y) = P(X|Y) P(Y)$ . Model generatif memiliki peran meniru distribusi data  $x$  yang disimbolkan  $p_{data}(x)$  dengan mengambil set contoh pelatihan dari  $p_{data}(x)$  yang belajar untuk mewakili perkiraan distribusi itu sehingga mendapat probabilitas  $p_{model}(x)$  yang dapat bekerja menghasilkan sampel baru dari distribusi yang sama melalui prinsip kemungkinan maksimum (*maximum likelihood*). Pemanfaatan model generatif ini sebelumnya dapat digunakan untuk melengkapi *missing data*, kompres set data dan menghasilkan *dataset* yang tidak terlihat.

### B. GANs

Model generatif pada penelitian ini mengacu pada jaringan permusuhan yang diperkenalkan sebagai kerangka kerja alternatif untuk melatih model menghindari kesulitan mendekati banyak perhitungan probabilitas yang sulit dipecahkan yang dikenal dengan *Generative Adversarial Networks* (GANs). GANs pertama kali diperkenalkan oleh Goodfellow dkk [4] melalui konsep *game theory* yang menggambarkan adanya interaksi antara kedua pemain yang merupakan individu yang rasional. Setiap pemain memiliki strategi yang berbeda dan akan memberikan konsekuensi bagi pemain lainnya.

Gambar 1 merupakan ilustrasi yang menunjukkan bahwa terdapat dua pemain, yaitu model generatif dan diskriminatif. Model generatif bertugas untuk menangkap distribusi data dan membantu model diskriminatif dalam meningkatkan probabilitas dalam menguji data palsu dan data asli. Model diskriminatif bertugas memperkirakan probabilitas bahwa sampel berasal dari data pelatihan dan bukan dari data generatif. Tujuan dari fungsi pembelajaran ini adalah model generatif dapat belajar bagaimana menghasilkan data palsu yang tidak dapat dibedakan dari data asli melalui bantuan algoritme klasifikasi dari model diskriminatif.



Gambar 1. Ilustrasi model GANs

GAN memiliki dua model yang dilatih secara bersamaan yakni model generatif G dan model diskriminatif D. Model generatif G mengambil masukan vektor acak  $z \sim p_z(z)$  dan mempelajari himpunan parameter  $\theta_g$  untuk menghasilkan gambar  $G(z; \theta_g)$  yang mengikuti distribusi dari gambar sebenarnya. Pada saat bersamaan, model diskriminatif D belajar berdasarkan satu set parameter  $\theta_d$  untuk mengklasifikasikan  $x \sim p(x)$  sebagai gambar nyata dan  $G(z; \theta_g)$  sebagai gambar sintetis (atau palsu). Proses pelatihan dilakukan dengan memaksimalkan probabilitas D untuk menetapkan kelas yang benar ke  $x$  dan  $G(z)$ , sementara G dilatih untuk meminimalkan  $1 - D(G(z; \theta_g))$ . Dengan menggunakan *cross-entropy* sebagai *loss function*, optimasi  $V(D, G)$  didefinisikan pada persamaan 1.

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

### C. U-Net

Sebagai salah satu teknik yang dikembangkan berdasarkan arsitektur *autoencoder*, U-Net memiliki ciri tugas khusus yang mampu memberikan kelas atau wilayah lokal di sekitar piksel pada hasil segmentasi citra. Kerangka kerja U-Net terdiri dari *contracting path* dan *expansive path* yang memberikan arsitektur berbentuk huruf U [5]. Sisi kiri dapat disebut sebagai *contracting path* yang dibentuk dari jaringan konvolusi yang berulang diikuti dengan operasi fungsi aktivasi ReLU dan *max pooling*. Selama proses *contracting path*, informasi spasial yang disimpan pada lapisan sebelumnya akan berkurang, sementara informasi fitur yang dihasilkan dari pemrosesan yang dilajukan akan semakin ditingkatkan. Pada jalur kiri atau bisa disebut juga dengan istilah *expansive path* merupakan jalur yang menggabungkan fitur dengan informasi spasial melalui urutan *up-sampling* dan penggabungan (*concatenation*) lapisan resolusi tinggi dari jalur *contracting path* dengan keluaran *up-sampling* yang sering dikenal dengan istilah *skip-connection*. *Skip-connection* merupakan solusi dari kekurangan *Autoencoder* yang mengalami kehilangan beberapa resolusi saat proses *encoding* sehingga cenderung memberikan hasil citra yang kabur.

### D. CoGANs

Algoritme GANs [4] dapat menjadi CoGANs ketika model didalam GANs diberikan informasi tambahan berupa kondisi yang dapat berupa data apapun seperti label kelas atau modalitas lain. Model generatif pada GANs [14] yang sebelumnya hanya *noise*  $p_z(z)$  yang berfungsi sebagai masukan yang menghasilkan data,

kemudian diberi masukan berlabel yang digabungkan dalam representasi tersembunyi bersama. Metode CoGANs digunakan oleh Phillip Isola dkk [7] dengan penyempurnaan pemilihan desain arsitektur U-Net [17] untuk model generatif dan PatchGANs untuk model diskriminatif.

## III. METODE

### A. Persiapan Data

Bahan utama yang digunakan dalam penelitian ini adalah *dataset* citra daun MalayaKew (MK) [16] yang dikumpulkan di Royal Botanic Gardens, Kew, Inggris. Data ini terdiri dari citra 44 jenis kelas daun. Untuk memenuhi kebutuhan dari penelitian, maka pemilahan data dilakukan dengan mengambil 3 bentuk daun: Lanceolate, Runcinate dan Lyrate. Masing-masing bentuk diambil 3 *spesies* daun seperti yang dapat dilihat pada Tabel 1 dengan jumlah sampel masing-masing 4 item sehingga total citra daun yang dipilih adalah 768 item. (1)

Tabel 1. Rincian Data Spesimen Daun.

Bentuk Daun	Lanceolate	Runcinate	Lyrate
Nama Daun	<i>Aff cerris</i>	<i>X-kewensis</i>	<i>Rubur f purpubascens</i>
	<i>X-hispanica</i>	<i>X-roseacea</i>	<i>Acutissima</i>
	<i>Agriefolia</i>	<i>Hartwissiana</i>	<i>Lobata</i>
	<i>Laurifolia</i>	<i>Serrata</i>	<i>Uercus</i>

Pada dasarnya, himpunan data MK yang diunduh [9] tidak bisa langsung diproses menggunakan model yang diusulkan dalam penelitian. Hal tersebut dikarenakan sampel pelatihan yang diharapkan untuk melatih model CoGANs adalah citra alami daun dan citra segmentasi vena daun. Persiapan yang dilakukan untuk transformasi citra alami daun yaitu menggunakan filter sederhana yang disediakan oleh pustaka OpenCV tanpa ada patokan khusus mengenai teknik yang digunakan dalam segmentasi citra vena daun. Fungsi deteksi tepi pada citra menjadi acuan dalam pembuatan *dataset* citra alami daun dan citra sketsa daun. Gambar 2 merupakan contoh hasil segmentasi vena daun yang dijadikan objek pelatihan dengan sisi kanan adalah citra alami daun dan sisi kiri adalah citra sketsa daun.



Gambar 2. Citra bentuk *Lanceolate*

### B. Pelatihan dan Pengujian Model

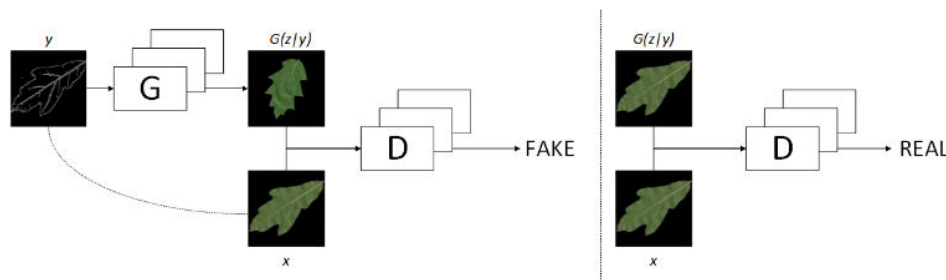
skenario model pelatihan dapat dijelaskan menggunakan Gambar 3. data  $x$  yang merupakan distribusi citra alami daun bentuk *lanceolate*, dan data  $y$  yang merupakan label menjadi identitas dari data  $x$  atau dari sketsa daun yang akan di proses. Model generatif yang digunakan akan memprediksi masukan dari data  $y$  sehingga menghasilkan distribusi data  $x$  tiruan yaitu  $G(z)$ . Model diskriminatif kemudian membandingkan citra alami daun bentuk *lanceolate* dengan data yang diperoleh berdasarkan  $G(z)$ .

Ketika  $G(z)$  diketahui menghasilkan citra alami daun bentuk *runcinate*, maka model diskriminatif akan memberikan nilai "fake" atau "0", sebaliknya jika  $G(z)$  menghasilkan citra alami daun bentuk *lanceolate*, maka model diskriminatif akan memberikan nilai "real" atau "1".

Untuk membuat arsitektur U-Net yang memiliki masukan citra sketsa daun berdimensi 128x128 piksel, perlu dibuat *contracting path* di jalur kiri sebagai proses *encoding* yang memiliki 7 *down-block* dan *expansive*

*path* di jalur kanan sebagai proses decoding yang memiliki 6 *up-block*. Alasan dari penggunaan 7 *down-block* didasarkan pada penelitian sebelumnya, dengan anggapan bahwa untuk mendapatkan citra terjemahan perlu adanya prediksi nilai untuk setiap piksel. Oleh karena itu *feature map* menjadi salah satu opsi tambahan untuk mencapai dimensi 1x1 piksel.

Dengan menggunakan filter yang berukuran 3x3, maka proses konvolusi selanjutnya dapat dijalankan pada masing-masing lapisan konvolusi. Setiap lapisan memiliki jumlah *filter* pada *encoder* dengan urutan sebagai berikut: 64, 128, 256, 512, 512, 512, 512. Sedangkan pada bagian decoder, penggunaan *filter* dalam lapisan dilakukan berdasarkan urutan 512, 512, 256, 128, 64. Pemilihan urutan dan jumlah lapisan tersebut tidak mengacu pada aturan tertentu. Namun ketetapan tersebut diambil berdasarkan saran dari berbagai penelitian sebelumnya serta dengan adanya anggapan bahwa semakin banyak *filter* yang digunakan akan memperbanyak *feature map* yang dihasilkan namun sangat berpengaruh terhadap jumlah parameter dan kecepatan proses pembelajaran.



Gambar 3. Ilustrasi skenario pelatihan.

Mengacu pada penelitian Isola dkk [7], setiap *down-block* pada *encoder* U-Net memuat komposisi proses konvolusi>>BatchNorm>>LeakyRelu dengan nilai Alpha = 0,2. Operasi *BatchNorm* tidak digunakan pada konvolusi pertama karena sebagai konvolusi merupakan gerbang masukan distribusi data gambar. Proses pada *down-block* diakhiri dengan *subsampling* menggunakan *max pooling layer* yang berukuran 2x2 guna mereduksi dimensi *feature map* sehingga citra hasil proses nantinya akan berukuran setengah dari ukuran aslinya. Karena tujuan *down-block* adalah untuk mendapatkan fitur yang digunakan, maka tujuan dari *up-block* adalah untuk mengembalikan atau merekonstruksi kembali representasi fitur yang diperoleh untuk kemudian dikembalikan lagi menjadi citra aslinya. Berdasarkan fungsi tersebut, komposisi untuk setiap *up-block* pada *decoder* U-Net akan memuat komposisi proses konvolusi>>BatchNorm>>Dropout (0,5)>>Relu. Setiap melewati proses *up-block* perluasan piksel dilakukan menggunakan *un-pooling* yang merupakan kebalikan proses yang ada ketika mengimplementasikan *max pooling layer*.

Arsitektur U-Net identik dengan adanya *skip-connection* yang bisa diaplikasikan mulai dari setiap lapisan  $i$  dalam *encoder* dan lapisan  $n-i$  dalam *decoder*, dengan  $n$  yang merupakan total keseluruhan lapisan. *Skip-connection* ini merupakan hasil keluaran proses *down-block* sebelum proses *downsampling* yang didistribusikan langsung ke *up-block* sebagai masukan gabungan yang bertujuan untuk mengurangi adanya reduksi yang terjadi pada *downsampling* sehingga harapannya distribusi kualitas data citra yang dihasilkan akan sesuai dengan kualitas yang menjadi citra masukan. Arsitektur U-Net yang digunakan sebagai model generatif untuk pelatihan penelitian ini dapat dilihat pada gambar 4.

Setelah model generatif menghasilkan terjemahan citra sketsa daun, kerangka kerja CoGANs akan melakukan penyelidikan citra melalui model diskriminatif apakah citra masukan merupakan data asli atau tiruan dari generator. Tujuan akhir dari proses ini adalah generator dipaksa untuk menghasilkan penerjemahan citra sketsa daun sebaik mungkin hingga citra alami daun yang dihasilkan tampak nyata.

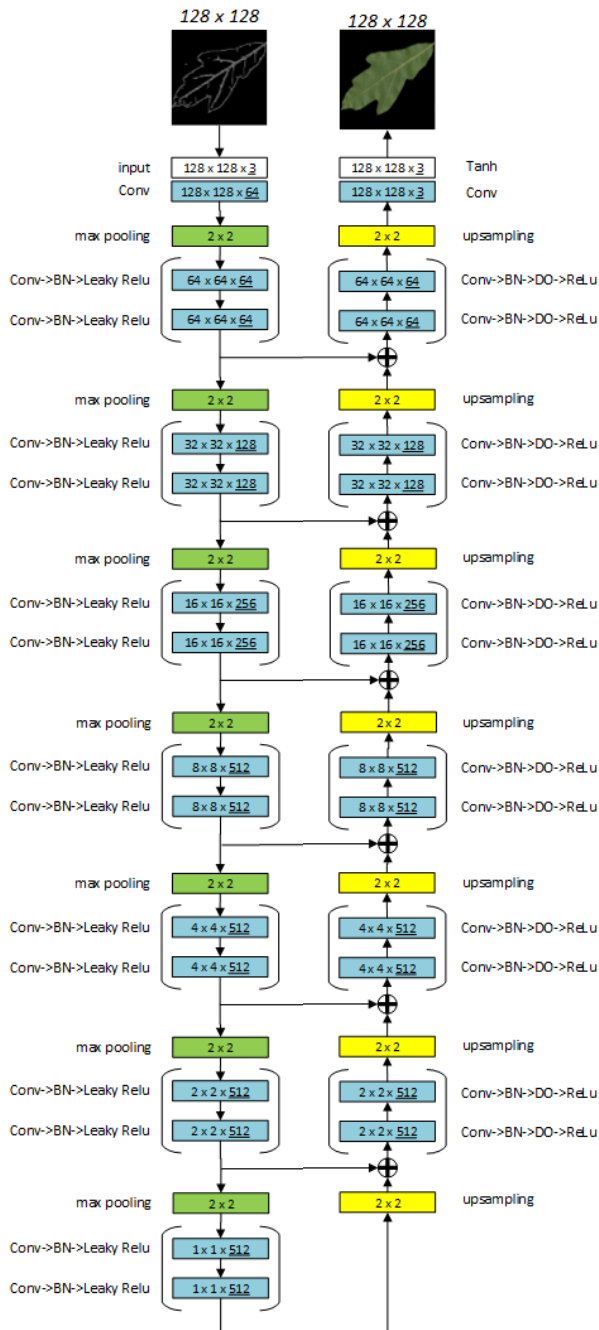


Arsitektur diskriminator CoGANs menggunakan CNNs yang mengacu pada arsitektur diskriminator DCGANs [8] dengan rangkaian proses konvolusi, lalu BatchNorm dan kemudian LeakyRelu (0,2). CNNs tersebut diaplikasikan pada potongan citra masukan yang

kemudian dikenal dengan istilah PatchGANs yang akan mengklasifikasi potongan citra per-ukuran piksel 64x64 berdasarkan rekomendasi penelitian sebelumnya [7] dan kemudian baru disatukan hasilnya. Arsitektur PatchGANs disajikan pada Gambar 5. membandingkan hasil citra terjemahan secara kualitatif. Fungsi biaya kerugian atau *loss function* sebenarnya diperlukan untuk evaluasi pembelajaran agar model dapat melakukan *back propagation* dan memberikan nilai terbaik. Model diskriminator menggunakan *binary cross entropy loss function* untuk menghitung nilai *error rate* dengan rentang nilai 0 sampai 1 untuk menguji apakah citra yang dihasilkan oleh generator asli atau palsu. *Loss function* yang digunakan untuk penilaian kinerja generator yaitu menggunakan L1 *loss function* atau biasa dikenal dengan *Mean Absolute Error* dengan prinsip mencari rerata nilai absolut dari nilai prediksi dikurangi nilai kebenaran yang kemudian ditotalkan seperti persamaan sebagai berikut:

$$MAE = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| \quad (2)$$

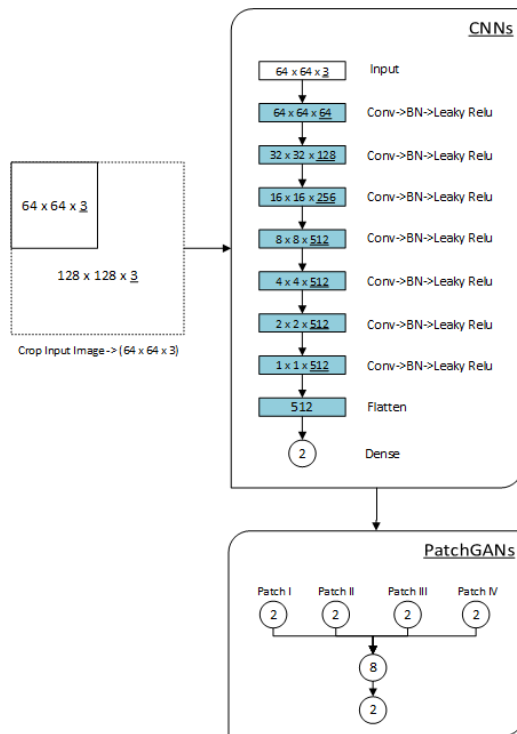
Pada penelitian ini, seluruh fitur yang dihasilkan akan dipisahkan menjadi data pelatihan (*training*) dan data pengujian (*testing*). Proses pembagian data tersebut dilakukan secara acak dengan proporsi untuk data pelatihan sebesar 60% data validasi 20% dan data pengujian 20% dari keseluruhan citra. Data *training* dan *testing* tersebut kemudian diproses menggunakan PatchGANs untuk mendapatkan nilai L1 *loss function* berdasarkan fitur yang digunakan. Proses pembagian data untuk masing-masing skenario dilakukan sebanyak 30 kali agar dapat diasumsikan bahwa distribusi data tersebut sudah berada pada distribusi normal sesuai dengan *central limit theory* [10].



Gambar 4. Arsitektur pelatihan U-Net.

### C. Evaluasi Kinerja

Evaluasi kinerja merupakan tahapan yang digunakan untuk mengetahui hasil dari model yang diusulkan sebagai bahan pertimbangan dalam proses penarikan kesimpulan. Pada penelitian ini, evaluasi kinerja akan dilakukan dalam dua mdoel evaluasi, yang pertama adalah perbandingan kuantitatif pada fungsi biaya kerugian (*loss function*) dan yang kedua adalah



Gambar 5. Arsitektur PatchGANs.

#### IV. HASIL

Proses pelatihan data dilakukan menggunakan layanan *cloud computing* yang dibuat dan disediakan oleh Google *research project* [11] dengan nama *Google Collaboratory*. Layanan *cloud* gratis ini digunakan untuk pembelajaran *machine learning* secara umum dan *deep learning* secara khusus. Layanan ini didukung oleh GPU Tesla K80 yang membantu proses pelatihan model *machine learning* yang dapat memangkas beban komputasi daripada melatih model pada komputer pribadi.

##### A. Hasil Kualitatif

Pengukuran secara kualitatif dilakukan dengan cara membandingkan hasil citra terjemahan dari data pengujian yang dihasilkan oleh metode CoGANs dengan citra *ground truth*. Setelah dilatih hingga *epoch* ke-60, Gambar 6-11 merupakan hasil terjemahan citra sketsa daun dengan baris citra bagian atas merupakan citra sketsa daun sebagai citra masukan, baris tengah merupakan citra prediksi penerjemahan dari model generatif yang diusulkan (CoGANs), baris bawah merupakan citra *ground truth* sebagai pembanding.

##### B. Hasil Kuantitatif

Pada dasarnya terdapat dua usulan metrik antara MSE (*Mean Square Error*) dan MAE yang sering dipakai untuk analisis regresi. Karena representasi data yang dihasilkan generator memiliki rentang nilai (0-255) pada masing-masing kanal, maka MAE merupakan pilihan yang tepat. Hal ini disebabkan MSE memiliki operasi kuadrat yang mengakibatkan ketika

ada data *outlier* pada kumpulan data prediksi eror, maka nilai MSE akan melambung daripada MAE. Tujuan dari penggunaan metrik ini adalah nilai prediksi mendekati nilai sebenarnya atau hasil terjemahan citra dari generator mampu semirip mungkin dengan *ground truth*. Semakin kecil nilai MAE menandakan semakin akurat nilai prediksi. Setelah dilakukan 30 kali pengujian maka dapat diasumsikan bahwa distribusi data tersebut sudah berada pada distribusi normal sesuai dengan *central limit theory* [10], seluruh nilai *loss function* yang dihasilkan akan dihitung rerata dan nilai persebaran (standar deviasi) dengan hasil 0,0926 ( $\pm 0,0068$ ).

##### C. Hasil Kuantitatif

Pada dasarnya terdapat dua usulan metrik antara MSE (*Mean Square Error*) dan MAE yang sering dipakai untuk analisis regresi. Karena representasi data yang dihasilkan generator memiliki rentang nilai (0-255) pada masing-masing kanal, maka MAE merupakan pilihan yang tepat. Hal ini disebabkan MSE memiliki operasi kuadrat yang mengakibatkan ketika ada data *outlier* pada kumpulan data prediksi eror, maka nilai MSE akan melambung daripada MAE. Tujuan dari penggunaan metrik ini adalah nilai prediksi mendekati nilai sebenarnya atau hasil terjemahan citra dari generator mampu semirip mungkin dengan *ground truth*. Semakin kecil nilai MAE menandakan semakin akurat nilai prediksi. Setelah dilakukan 30 kali pengujian maka dapat diasumsikan bahwa distribusi data tersebut sudah berada pada distribusi normal sesuai dengan *central limit theory* [10], seluruh nilai *loss function* yang dihasilkan akan dihitung rerata dan nilai persebaran (standar deviasi) dengan hasil 0,0926 ( $\pm 0,0068$ ).

#### V. KESIMPULAN

Berdasarkan hasil dan analisis dari penelitian yang telah dilakukan maka berikut kesimpulan yang dapat disampaikan:

1. Model generatif pada kerangka kerja *deep learning* dengan usulan metode CoGANs yang tersusun dari U-Net sebagai model generatif dan PatchGANs sebagai model diskriminatif mampu menerjemahkan citra sketsa daun menjadi citra alami daun.
2. Semakin lama pelatihan (*epoch*), maka model generatif pada kerangka kerja *deep learning* dapat menghasilkan prediksi penerjemahan citra yang lebih baik.
3. Melalui penilaian kuantitatif dengan patokan nilai MAE *loss function* pada hasil pelatihan model generatif yang mendekati 0 menandakan bahwa hasil penerjemahan mampu mendekati *ground truth* atau citra aslinya.



Gambar 6. Epoch ke-1



Gambar 7. Epoch ke-10



Gambar 8. Epoch ke-20



Gambar 9. Epoch ke-30



Gambar 10. Epoch ke-40



Gambar 11. Epoch ke-50

#### DAFTAR PUSTAKA

- [1] V. Metre and J. Ghorpade, "An overview of the research on texture based plant leaf classification," *arXiv Prepr. arXiv1306.4345*, vol. 2, no. 3, pp. 61–64, 2013.
- [2] H. Scharr, M. Minervini, A. P. French, C. Klukas, D. M. Kramer, X. Liu, I. Luengo, J. P. Gerrit, P. Danijela, V. Xi, and S. A. Tsafaris, "Leaf segmentation in plant phenotyping: a collation study," *Mach. Vis. Appl.*, vol. 27, no. 4, pp. 585–606, 2016.
- [3] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," pp. 1–7, Nov. 2014.
- [4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," pp. 1–9, Jun. 2014.
- [5] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," pp. 1–8, 2015.
- [6] J. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative Visual Manipulation on the Natural Image Manifold," pp. 1–16, Sep. 2016.
- [7] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017–Janua, pp. 5967–5976, 2017.
- [8] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *CoRR*, vol. abs/1511.0, 2015.
- [9] "MalayaKew Dataset + Ground Truth." [Online]. Available: [http://web.fsktm.um.edu.my/~cschan/downloads\\_MKLeaf\\_dataset.html](http://web.fsktm.um.edu.my/~cschan/downloads_MKLeaf_dataset.html). [Accessed: 10-May-2018].
- [10] M. Mether, "The history of the central limit theorem," *Sovell. Mat. erikoistyöt*, vol. 2, no. 1, p. 08, 2003.
- [11] "Google Colaboratory." [Online]. Available: <https://colab.research.google.com/>. [Accessed: 03-Mar-2018].