LOGIC: Jurnal Penelitian Informatika
Vol. 3, Issue 2, pp. 59–67 (2025)
doi: http://doi.org/10.25124/logic.v3i2.6467

RESEARCH ARTICLE

Analisis Kemampuan Beta-VAE Pada Dataset Yang Berbeda

Bramantya Purbaya, Bedy Purnama* and Edward Ferdian

Fakultas Informatika, Universitas Telkom, Bandung, 40257, Jawa Barat, Indonesia
*Corresponding author: bedypurnama@telkomuniversity.ac.id,

Abstrak

Data sintetis sudah menjadi beberapa penelitian untuk kasus *machine learning*, salah satunya adalah menambah data baru dikarenakan kurangnya data yang sudah ada. Tetapi bagaimana untuk menghasilkan dan mengatur berbagai variasi dari distribusi data masukan masih menjadi bahan penelitian. Pada penelitian ini menggunakan salah satu variasi metode *Variational Auto Encoder* (VAE) untuk menghasilkan data sintetis, yaitu Beta-*Variational Auto Encoder* (Beta-VAE). VAE sendiri merupakan metode *unsupervised learning* yang dapat menghasilkan data sintetis, tetapi variasi yang dihasilkan tidak terlalu teratur dibandingkan Beta-VAE. Pada penelitian ini digunakan metode Beta-VAE asli untuk menghasilkan data sintetis yang dilatih dengan empat dataset yang berbeda. Digunakan metrik PSNR, SSIM dan FID score untuk mengevaluasi model Beta-VAE. Dibandingkan setiap model Beta-VAE yang dilatih dengan dataset berbeda dan dilakukan analisis pada setiap model. Hasil dari penelitian didapati model yang dilatih dengan CelebA memiliki hasil terbaik terlihat dari metrik evaluasi.

Key words: data sintetis, dataset, *autoencoder*, *variational autoencoder*.

Pendahuluan

Teknologi kecerdasan buatan (AI) sudah dipakai pada kehidupan sehari-hari dan membuka pintu inovasi di berbagai bidang pekerjaan. Salah satu implementasinya adalah AI generatif. Kegunaan AI generatif dapat menghasilkan suatu konten baru dari data-data yang sudah dipelajari sebelumnya. Contoh konten yang dihasilkan berupa tulisan, citra, ataupun suara. Pada penelitian ini dikhususkan untuk AI generatif yang dapat menghasilkan citra. Membuat AI generatif untuk citra dibutuhkan model *machine learning* yang dapat mempelajari citra dataset untuk menghasilkan citra yang mirip dengan data input. Salah satu cara pembuatannya menggunakan metode *neural network* agar dapat mempelajari fitur fitur dari citra input. Beberapa metode untuk AI generatif, yaitu GAN [1], infoGAN [2], DCIGN [3] dan Variational *AutoEncoder* [4]. Metode-metode tersebut mempunyai kemampuan yang mirip, dimana hanya memerlukan dataset citra untuk melatih model. Pada penelitian ini difokuskan untuk metode VAE, dikarenakan berbeda dengan metode seperti GAN, hasil yang dihasilkban VAE dapat lebih bervariasi, karena GAN cenderung untuk menghasilkan subset dari data asli [5]. VAE juga digunakan pada salah satu metode stable diffusion untuk menghasilkan citra yang lebih baik [6].

Meskipun VAE sudah dapat menghasilkan citra baru yang bervariasi, tetapi variasi citra yang dihasilkan tidak terlalu terkontrol. Maka digunakan metode lanjutan VAE, yaitu Beta-VAE. Metode ini menambahkan *hyperparameter* beta yang dapat diatur nilainya untuk mengatur kualitas dari *disentangle* (pemisahan) karakteristik data pada ruang laten [7]. Tujuan pada penelitian ini adalah untuk membuat data sintetis menggunakan Beta-VAE dan menguji kinerja Beta-VAE jika dilatih menggunakan dataset yang berbeda. Dengan menggunakan dataset yang berbeda, maka dapat diketahui bagaimana dataset yang baik untuk Beta-VAE. Pada penelitian ini dapat diketahui kemampuan model *machine learning* BetaVAE untuk menghasilkan data sintetis dan distribusi data pada ruang laten. Diharapkan dengan penelitian ini dapat memahami potensi dari Beta-VAE dalam menghasilkan citra.

Topik dan Batasannya

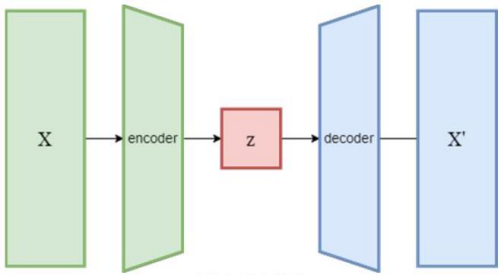
Topik penelitian ini, pertama, bagaimana memperoleh hasil data baru yang dihasilkan oleh model Beta-VAE yang dilatih dengan empat dataset yang berbeda. Kedua, bagaimana menguji model Beta-VAE untuk melihat kinerja dari setiap model Beta-VAE. Dataset yang digunakan untuk penelitian ini adalah dataset citra mobil, wajah manusia, wajah kucing dan motif batik. Dua dataset merupakan data alami (wajah manusia dan kucing) dan dua yang lainnya tidak / buatan (mobil dan batik). Motif batik yang digunakan hanya satu jenis, yaitu motif Megamendung agar mempermudah proses pelatihan. Metode penelitian menggunakan Beta-VAE asli (original). Model Beta-VAE yang diteliti hanya menggunakan nilai beta 10 dan 100 untuk setiap model.

Tujuan

Tujuan pada tugas akhir ini adalah untuk menyelidiki kemampuan model Beta-VAE untuk dataset yang berbeda dan menghasilkan data sintetis yang mirip dengan keempat dataset tersebut yang mampu dikontrol oleh parameter tertentu.

Organisasi Tulisan

Pertama dijelaskan terlebih dahulu topik tentang AI generatif dan VAE. Dijelaskan juga tentang batasan topik dan tujuan penelitian tentang kinerja Beta-VAE. Selanjutnya studi terkait tentang *auto encoder*, VAE, Beta-VAE dan metrik evaluasi yang dipakai akan dijelaskan. Kemudian penjelasan tentang alur sistem dan arsitektur VAE yang dibangun. Setelah itu bab evaluasi menjelaskan tentang proses yang dilakukan untuk menguji model Beta-VAE dan hasil yang diperoleh. Terakhir kesimpulan menjelaskan tentang hasil yang didapat dari keseluruhan pengujian.



Gambar 1. Contoh *autoencoder* sederhana

Tinjauan Pustaka

Berikut berupa beberapa penelitian yang dijadikan referensi untuk penelitian tugas akhir ini:

1. Penelitian oleh Miroslav Fil, Munib Mesinovic, Matthew Morris dan Jonas Wildberger yang berjudul "*Beta-VAE Reproducibility: Challenges and Extensions*" [8] meneliti tentang Beta-VAE dengan melatih model memakai dataset yang kompleks. Dataset yang dipakai untuk penelitian *disentanglement*, yaitu *2D Shapes*, *3D Shapes*, dan MPI3DToy. Penelitian untuk rekonstruksi, yaitu CIFAR10 dan CIFAR100. Dilakukan juga investigasi paper Beta-VAE yang asli. Beberapa konstruibusi yang dilakukan, yaitu mendemonstrasi jika beta ¿ 1 tidak selalu menghasilkan hasil *disentangle* kuantitatif yang baik untuk dataset yang kompleks. Kesimpulan yang didapat pada studi ini, pertama, yaitu nilai skor metrik *disentanglement* yang tinggi tidak berarti *disentanglement* kualitatif dan kedua, dinilai secara kuantitatif nilai beta yang lebih rendah memberikan rekonstruksi yang lebih baik dari image original.
2. Penelitian oleh Aditya Firman Ihsan yang berjudul "*Initial Study of Batik Generation using Variational Autoencoder*" [9] meneliti tentang bagaimana hasil rekonstruksi menggunakan VAE terhadap tiga jenis dataset batik. Dataset motif batik yang digunakan, yaitu Megamendung, Lereng dan Kawung. Menggunakan VAE, tiga jenis motif batik tersebut akan diimplementasikan dan dianalisis. Dataset batik yang digunakan diubah menjadi bentuk *gray scale* untuk mengurangi dimensi dari input data. Pada penelitian ini terbukti bahwa VAE secara tidak langsung bertindak sebagai *image embedder* pada ruang laten. Telah ditunjukkan juga bahwa performa dari model pada setiap jenis motif Batik berbeda-beda. Hasil image yang dihasilkan dari model dengan dimensi laten yang tinggi juga dianalisis. Diidentifikasi juga pengaruh *batch normalization* terhadap performa model. Model batik yang menggunakannya terlihat memiliki KL loss yang tidak stabil, dapat disimpulkan penggunaan *batch normalization* menyebabkan ketidakstabilan pada optimisasi KL loss saat pelatihan dengan data batik.
3. Penelitian oleh Higgins dengan judul "*beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework*" [7] merupakan penelitian dari beta-vae asli. Saat paper ini dibuat, tidak ada metode general untuk mengukur tingkat pembelajaran dari *disentanglement* (keteruraian). Dilakukan penelitian pada dataset celebA [10], *3D chairs* [11] dan *3D faces* [12]. Secara keseluruhan beta-VAE cenderung secara konsisten menemukan lebih banyak faktor laten dan mempelajari representasi yang lebih bersih daripada infoGAN dan DC-IGN. Ini berlaku pada dataset seperti celebA.

Autoencoder

Autoencoder merupakan sebuah *neural network* yang dilatih untuk merekonstruksi kembali inputnya. Arsitektur *Autoencoder* dibagi menjadi dua, yaitu *encoder* dan *decoder*. Data input akan menjadi representasi yang lebih kecil setelah masuk ke *encoder*, kemudian representasi tersebut akan direkonstruksi kembali menjadi sebuah data setelah masuk ke *decoder*. *Autoencoder* menggunakan cara *representation learning*, yaitu mengubah sebuah data menjadi representasi yang lebih kecil dengan *neural network* dan direkonstruksi kembali. Biasanya *autoencoder* digunakan untuk kompresi data. Kompresi data menjadi representasi yang lebih kecil disebut *encoding*, kemudian data yang sudah dikompres disebut *code* dan terakhir proses rekonstruksi kembali disebut *decoding*. Terdapat dua komponen yang digunakan, yaitu *encoder* untuk proses *encoding* dan *decoder* untuk *decoding* sebagaimana ditampilkan pada gambar 1 [13]. X merupakan input citra yang akan dikompres melalui *encoder*, kemudian z adalah representasi hasil kompresi. *Decoder* akan menerima fitur/kode laten z dan merekonstruksi menjadi X' yang direkonstruksi semirip mungkin dengan input. *Autoencoder* tidak dapat menghasilkan citra baru atau menghasilkan variasi dari data input. Untuk memanfaatkan kemampuan rekonstruksi *autoencoder* dan menghasilkan variasi baru, maka dibutuhkan mekanisme baru.

Variational Auto Encoder (VAE)

Variational Auto Encoder (VAE) merupakan pengembangan dari *Autoencoder* asli. Perbedaannya pada VAE, data akan didistribusikan pada dimensi laten. Dengan menggunakan mean dan standar deviasi dari distribusi laten, maka bisa diambil sampel dari distribusi laten yang selanjutnya direkonstruksi menggunakan *decoder*. Untuk perhitungan loss terdapat dua macam perhitungan, yaitu loss untuk rekonstruksi dan *loss kullback leibler divergence* (*KL Divergence*). Loss pada rekonstruksi menggunakan *Mean Square Error* (MSE) dengan menggunakan citra input dan citra rekonstruksi didapatkan hasil loss rekonstruksi. VAE hanya belajar dari ditribusi variabel laten. Sebelum variabel laten di-*decode*, digunakan output dari *encoder* berupa prediksi mean (μ) dan strandar deviasi (σ) dari distribusi laten untuk mendapatkan sampel kode laten (z). Mean dan standar deviasi pada VAE tidak didapatkan melalui perhitungan matematis seperti pada umumnya perhitungan rata-rata ataupun standar deviasi. Tetapi dua vektor tersebut diasumsikan merepresentasikan sebuah distribusi yang dalam hal ini merupakan distribusi normal, sesuai dengan distribusi prior. Dengan adanya vektor mean dan standar deviasi, maka dapat dilakukan sampling untuk mendapatkan laten vektor z dari kedua vektor tersebut. Untuk itu dilakukan *reparameterization trick* untuk mendapatkan sampel z dengan $\varepsilon \sim \mathrm{N}(0,1)$ [4].

$$z = \mu + \sigma\varepsilon \tag{1}$$

KL Divergence adalah metode untuk membandingkan distribusi yang satu dengan yang lainnya. Pada VAE, *KL Divergence* digunakan untuk memastikan distribusi dari variabel laten mendekati distribusi normal atau Gaussian. Distribusi yang didapatkan dengan meng-encode data x , yaitu $\mathrm{q(z \mid x)}$ diinginkan untuk mendekati distribusi saat ini (distribusi normal) $\mathrm{p(z)} \sim \mathrm{N}(0,1)$.

$$L_{KL} = D_{KL}\left(q\left(z|x\right) \parallel \left(p\left(z\right)\right)\right) \tag{2}$$

Gambar 2. Contoh hasil manipulasi dimensi laten (*traversal plot*) pada paper beta-VAE.

Untuk menghitung KLD Loss dibutuhkan mean (μ) dan standar deviasi (σ) yang kemudian pada equasi (3) dilakukan penjumlahan (sum) sebanyak jumlah dimensi dalam distribusi laten (D) dari setiap mean dan standar deviasi pada setiap dimensi.

KLD Loss

$$KLDLoss = -\frac{1}{2}\sum_{i=1}^{D}\left(1 + log\left(\sigma_i^2\right) - \mu_i^2 - \sigma_i^2\right) \quad (3)$$

Model VAE dapat menghasilkan citra yang mempunyai variasi dari input. Tetapi dengan variasi dan karakteristik output yang dihasilkan tidak terlalu menonjol dan tidak terpisahkan tidak lebih baik dibandingkan Beta-VAE. Maka digunakan salah satu varian VAE, yaitu Beta-VAE.

Beta-Variational Auto Encoder (Beta-VAE)

Beta-VAE merupakan modifikasi dari framework VAE dengan memperkenalkan *hyperparameter* Beta. Nilai Beta digunakan untuk mengubah berat (*weight*) terhadap KLD loss. Berdasarkan penelitian sebelumnya [7, 14], disimpulkan bahwa semakin besar nilai Beta, hasil *disentangle* atau pemisahan karakteristik yang didapatkan semakin tinggi akan tetapi kualitas hasil rekonstruksi citra menurun. Jika Beta = 1, maka Beta-VAE akan sama dengan VAE original [7]. Pada penelitian ini digunakan Beta-VAE untuk menghasilkan citra dengan variasi dan karakteristik dapat diatur. *Traversal plot* didapatkan dengan melakukan pengambilan sampel pada dimensi laten dengan mengubah standar deviasi antara[-3, 3]. Pada paper beta-VAE umumnya digunakan dataset CelebA untuk menguiji model beta-VAE [7]. Maka dari itu pada penelitian ini salah satu dataset yang digunakan adalah CelebA dan hasilnya dibandingkan dengan dataset yang lain.

Metrik Evaluasi

Untuk mengevaluasi model yang dibangun ada beberapa kriteria:

1. Kualitas rekonstruksi
2. Traversal dimensi laten
3. Kemampuan *disentanglement* (penguraian)

Metrik-metrik yang digunakan untuk mengukur kualitas rekonstruksi adalah *Peak Signal-to-Noise Ratio* (PSNR), *Structural Similarity Index* (SSIM) dan *Frechet Inception Distance score* (FID score). Evaluasi metrik PSNR dan SSIM digunakan untuk mengevaluasi hasil rekonstruksi dari model. Pada Beta-VAE, selain dapat menghasilkan data baru, tetapi dapat merekonstruksi data yang sama. Kemampuan rekonstruksi ini didapatkan jika laten vektor z sama dengan vektor mean. Penambahan sedikit standar deviasi menimbulkan mean mendekati z menghasilkan sedikit variasi. Berarti mean merupakan vektor utama yang digunakan untuk rekonstruksi data dan standar deviasi memberikan variasi pada data memungkinkan menjadikannya data baru. Jadi penggunaan metrik PSNR dan SSIM dapat digunakan karena salah satu dari optimasi yang digunakan adalah rekonstruksi (laten vektor z sama dengan mean).

Umumnya PSNR digunakan untuk mengevaluasi kualitas dari transmisi dan kompresi dari sinyal citra atau video, dengan mengacu pada MSE dari citra yang diterima atau diproses dibandingkan dengan sumber citra [15, 16]. Nilai yang diberikan PSNR berupa desibel dan semakin tinggi nilainya semakin dekat citra hasil rekonstruksi dengan citra asli. Nilai yang perlu diketahui untuk menghitung PSNR adalah nilai piksel tertinggi pada citra $\mathrm{C_{max}}$ dan nilai MSE.

$$PSNR = 10log_{10}\frac{C_{MAX^2}}{MSE} \quad (4)$$

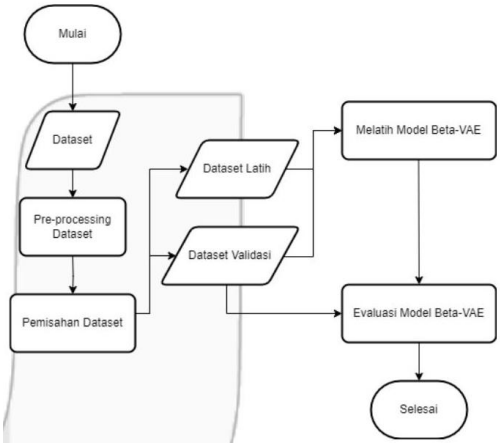
SSIM dirancang untuk lebih mencerminkan dengan persepsi visual manusia. Walaupun SSIM terdapat kasus yang dapat menyebabkan kesalahan dalam penilaian kualitas citra [14], tetapi SSIM sudah menjadi metrik umum yang dapat digunakan untuk menilai kualitas citra dibandingkan PSNR [15]. Batas nilai SSIM yang dapat dihasilkan adalah antara -1 dan 1. Mendekati nilai 1 berarti citra hasil rekonstruksi sangat mirip dengan citra asli dan mendekati nilai -1 berarti citra hasil rekonstruksi dan citra asli sangat berbeda.

$$SSIM\,(x,y) = \frac{(2\mu_x\mu_y + C_1)\,(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 C_2\right)} \quad (5)$$

Di mana μx dan μy adalah mean dari citra asli (x) dan citra hasil rekonstruksi (y). Sedangkan σx dan σy merupakan variasi dari citra asli dan citra hasil rekonstruksi, σxy merupakan kovarians antara citra asli dan citra rekonstruksi. Kemudian $C1$ dan $C2$ adalah konstanta untuk stabilitas perhitungan dan menghindari nilai 0. Selanjutnya metrik evaluasi yang digunakan adalah FID score. FID score digunakan pada paper [16] yang tujuan awalnya untuk mengevaluasi sebuah model GAN. Tujuan dari FID adalah untuk mengevaluasi kualitas dari sampel yang dihasilkan model *machine learning*. Terkadang dengan mata manusia tidak terlihat perbedaan jika sampel yang dihasilkan model yang berbeda memiliki hasil yang terlihat sama. Oleh karena itu digunakan metrik FID score untuk membandingkan model-model yang baik dalam menghasilkan sampel. Semakin kecil nilai FID score maka semakin baik kualitasnya. Menggunakan FID lebih baik jika digunakan bila ada pembandingnya. Pada penelitian ini dibandingkan model Beta-VAE dari setiap dataset. Selanjutnya dilakukan evaluasi dengan melakukan uji traversal. Evaluasi ini dilakukan dengan melakukan *traversal plot* yang merupakan penelusuran dari setiap dimensi pada ruang laten. Pada satu dimensi diubah dengan rentang [-3, 3] kemudian direkonstruksi menggunakan *decoder*. Evaluasi ini juga untuk melihat hasil *disentanglement* dari setiap model. Contoh hasilnya dari paper beta-VAE [7] pada penggunaan dataset CelebA, yaitu warna kulit seseorang dari cerah menjadi gelap (Gambar 2).

Metodologi Penelitian

Pada tugas akhir ini dibangun sistem untuk menghasilkan data sintetis yang karakteristik fiturnya dapat dipisahkan (*disentangle*) dari data masukan. Metode yang digunakan adalah Beta-VAE [7] dan kode yang digunakan untuk penelitian ini terdapat dari Github [17]. Berikut pada Gambar 3 merupakan rancangan sistem yang akan dibuat. Pembuatan sistem dimulai dengan menyiapkan dataset terlebih dahulu. Setelah itu dilakukan augmentasi data, hal yang dilakukan adalah *horizontal flip*, *random crop* dan data di-*resize* agar memiliki ukuran yang sama Data akan dibagi menjadi ratio 75% dan 25% yang masing-masing



Gambar 3. *Flowchart* perancangan sistem

Table 1. Nama-nama dataset dengan jumlahnya

Nama Dataset	Jumlah
Dataset citra mobil (Oxford Car Dataset)	9755
Dataset wajah manusia (CelebA)	8144
Dataset wajah kucing	15747
Dataset motif batik	2400

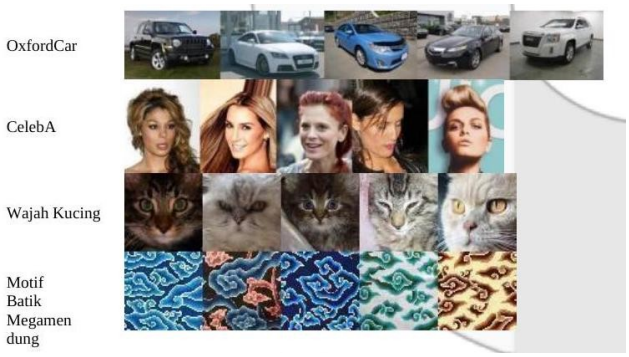
merupakan data latih dan data validasi. Setelah itu dilakukan pelatihan model dengan dataset latih dan validasi. Terakhir akan dilakukan evaluasi model menggunakan evaluasi matrik.
Evaluasi yang dilakukan terdiri dari:

1. Evaluasi rekonstruksi (SSIM dan PSNR)
2. Perbandingan parameter beta yang digunakan
3. Evaluasi kualitas citra
4. Evaluasi dimensi laten *disentangle* menggunakan *traversal plot*

Algortima *learning* yang dipakai pada Beta-VAE adalah *unsupervised learning* dimana model dilatih data tanpa label. Kemudian data tersebut diubah menjadi distribusi kode laten menggunakan encoder. Perbedaan dari *autoencoder* adalah kemampuan dari model ini mengubah kode laten menjadi sebuah distribusi normal menggunakan trik reparameterisasi. Setelah itu digunakan untuk mengambil sampel laten vektor tersebut untuk direkonstruksi kembali menjadi data melalui *decoder*.

Dataset

Dataset-dataset yang digunakan untuk tugas akhir ini masing-masing mempunyai pola dan jumlah yang berbeda. Pada Tabel 1 terlihat dataset-dataset dan jumlahnya yang digunakan untuk penelitian ini. Untuk dataset citra mobil digunakan Stanford Cars Dataset [18] yang pada awalnya digunakan sebagai representasi objek 3D, dataset wajah manusia mengunakan CelebA [10], untuk dataset wajah kucing dan motif batik didapatkan dari *open source* kaggle.com [22, 23]. Sebelum dataset siap dipakai, isi dataset dipilah terlebih dahulu. Untuk dataset mobil, gambar-gambar yang dipilih adalah gambar mobil yang menghadap ke arah miring ke kanan dan miring ke kiri yang dilihat dari depan, contoh dapat dilihat pada Gambar 4. Untuk dataset motif batik dipilih motif batik Megamendung. Untuk seluruh isi dataset wajah manusia dan kucing tidak ada yang dipilah.



Gambar 4. Contoh gambar dari setiap dataset yang dipakai. Setiap baris mewakilkan 5 contoh gambar darisetiap dataset.

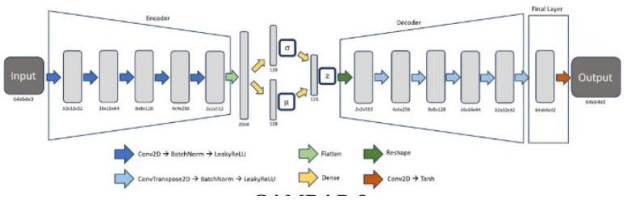
Gambar 5. Contoh gambar dataset setelah ditransformasi. Setiap baris terdiri dari contoh 5 gambar dari setiap dataset.

Transformasi Data

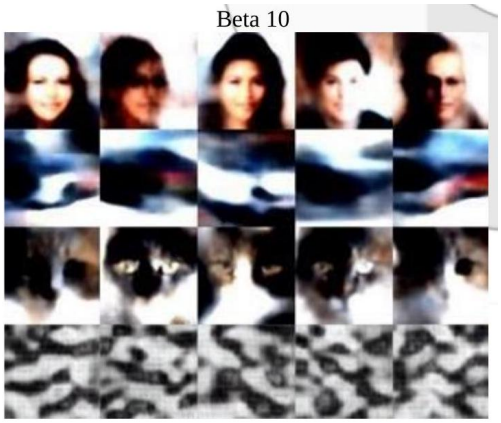
Transformasi data atau preparasi data adalah proses mempersiapkan dan memastikan data untuk siap dipakai untuk *training model*. Transformasi yang dilakukan pada penelitian ini meliputi *Random Horizontal Flip*, *Resize*, *CenterCrop* dan *Normalize*. *Random Horizontal Flip* secara acak melakukan pembalikan gambar secara horizontal. *Resize* mengubah ukuran data berdasarkan nilai pada parameter, untuk penelitian ini digunakan citra berukuran 64×64. *CenterCrop* memotong citra sesuai parameter yang dimasukkan. *Normalize* atau menormalisasi data sesuai nilai dari parameter mean dan standard deviation dari setiap channel. Untuk nilai mean dan standard deviation yang dipakai untuk penelitian ini memakai sesuai pada ImageNet [19]. Setelah itu dataset yang telah ditransformasi siap dipakai untuk *training model*. Pada dataset Batik Megamendung ditambahkan transormasi *grayscale* dikarenakan pada Batik difokuskan untuk diteliti pattern-nya. Dikarenakan jumlah dataset dari Batik sangat sedikit, maka dilakukan RandomCrop, maka jumlah dataset yang sebelumnya hanya 48 menjadi 2400. Referensi untuk transformasi batik didapatkan dari paper berikut [9].

Arsitektur VAE

Arsitektur VAE menggunakan mean dan standar deviasi untuk mengambil sampel dari distribusi laten. Untuk ukuran input menggunakan ukuran 64x64 RGB begitu pula hasil output-nya. *Encoder* terdiri dari beberapa blok konvolusi diikuti dengan *BatchNormalization* dan fungsi aktivasi LeakyReLU. Setiap blok konvolusi memiliki channel yang berbeda, terdiri dari 32, 64, 128, 256 dan 512 channel. Konvolusi



Gambar 6. Arsitektur VAE dengan ukuran input dan output 64x64 dengan 3 channel.



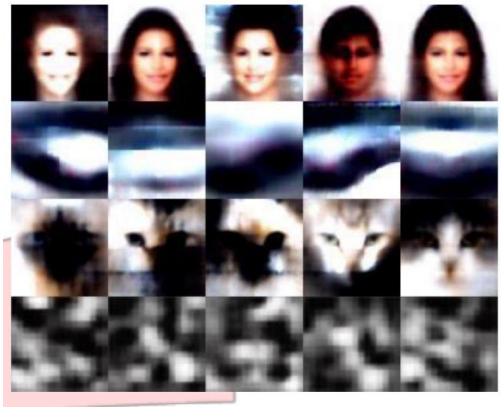
Gambar 7. Contoh sampel acak yang dihasilkan model Beta-VAE.

menggunakan kernel size 3 dan stride 2. Dekoder memiliki jumlah blok yang sama, tetapi dilakukan operasi transpose. Pelatihan model menggunakan Adam optimizer dengan *learning rate* sebesar 0.005. Untuk dataset motif batik, blok konvolusi yang digunakan hanya terdiri dari 3 blok, dengan jumlah channel sebesar 32, 64 dan 128. Selain itu, jumlah channel input dan output adalah 1 (*grayscale*). Jumlah blok dibedakan dikarenakan untuk dataset batik yang memiliki jumlah paling sedikit dari dataset lainnya maka digunakan 3 blok saja. Selain itu digunakan juga referensi dari paper tentang VAE pada batik [9]. Untuk semua model, dimensi laten (z) yang digunakan adalah sebesar 128. Sesuai dengan teori betaVAE, setiap dimensi dari dimensi 1 ke 128 merepresentasikan karakteristik atau atribut data yang diuraikan. Pada Gambar 6 terdapat arsitektur VAE, beta digunakan untuk menambahkan bobot pada fungsi lossnya KLD, yang berguna untuk mengubah kekuatan penguraian dari dimensi laten. Input model berupa citra berukuran 64x64 RGB, kemudian dimasukkan kedalam *encoder* yang terdiri dari beberapa blok Konvolusi2D. Hasil dari *encoder* kemudian dilakukan flatten agar menjadi vektor satu dimensi yang kemudian direparameterisasi untuk menghasilkan dua vektor yang dianggap sebagai mean (μ) dan standar deviasi (σ). Setelah itu dilakukan sampling kode laten (z). Dalam hal ini, kode laten z merupakan vektor satu dimensi yang lalu diubah menjadi vektor dua dimensi untuk kemudian diproses oleh *decoder*.

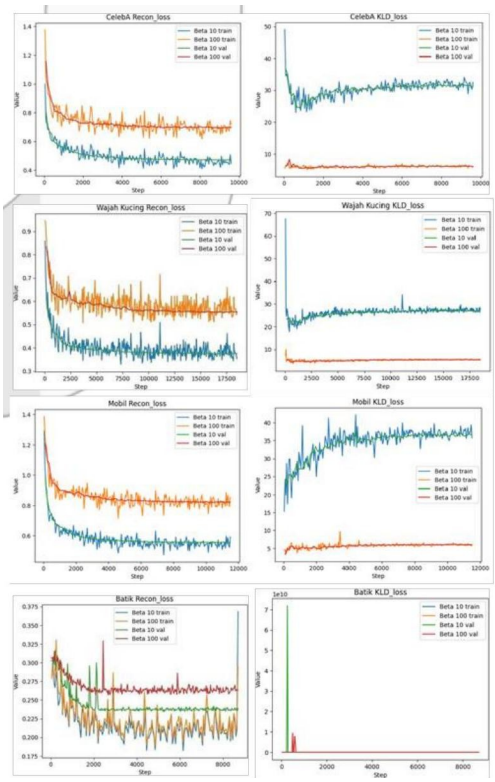
Hasil dan Pembahasan

Hasil Eksperimen

Untuk hasil pengujian pada model beta-VAE dihitung menggunakan PSNR dan SSIM untuk melihat seberapa baik hasil rekonstruksi data dan untuk melihat seberapa baik sampel yang dihasilkan model digunakan FID score. Dapat terlihat pada Gambar 7 dan gambar 8 untuk contoh hasil dari penelitian ini.



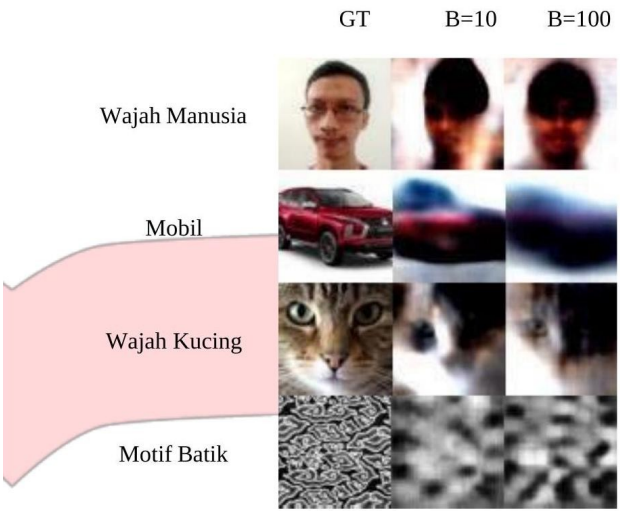
Gambar 8. Contoh sampel acak yang dihasilkan model Beta-VAE.



Gambar 9. Grafik dari loss KLD dan loss rekonstruksi untuk data latih (train) dan validasi (val) untuk setiapmodel CelebA, Wajah Kucing, Mobil dan Motif Batik.

Hasil Pelatihan Model

Dari data yang didapatkan dari pelatihan model dibuat grafik untuk melihat bagaimana loss rekonstruksi dan KLD. Masing-masing dari loss tersebut nilainya disimpan dari setiap step pada tahap pelatihan model. Nilai dari loss rekonstruksi didapatkan menggunakan MSE dari data input dan data rekonstruksi. Nilai dari loss KLD didapatkan dari ekuasi (3). Pada Gambar 9 terdapat 8 jumlah gambar yang merupakan *training curves* hasil dari pelatihan model Beta-VAE yang dilatih dari empat dataset yang berbeda. Baris pertama merupakan grafik pelatihan dari model yang dilatih menggunakan dataset wajah manusia (CelebA), kedua dari dataset wajah kucing, ketiga dari dataset mobil (OxfordCar) dan terakhir dari dataset motif batik. Kolom pertama dari sebelah kiri merupakan grafik untuk Loss Rekonstruksi dan kolom kedua untuk



Gambar 10. Citra Ground Truth (bagian kiri), citra rekonstruksi dengan beta=10 (bagian tengah), citra rekonstruksi dengan beta=100 (bagian kanan).

Loss KL Divergence. Setiap grafik memiliki empat garis yang masing-masing merupakan hasil pelatihan model Beta-VAE dari dataset latih (*training*) dan validasi dengan beta 10 dan 100.

Nilai beta merupakan tambahan *hyperparameter* pada perhitungan loss KLD untuk mengubah kekuatan penguraian dimensi laten. Karena besar nilai beta berpengaruh pada hasil penguraian, untuk itu dicoba dua nilai beta yang berbeda. Nilai beta kecil (beta=10) dan beta besar (beta=100). Secara umum, dari keempat model yang dilatih dengan dataset yang berbeda, beta kecil memiliki loss rekonstruksi yang lebih kecil yang berarti hasil rekonstruksi lebih baik. Sedangkan untuk *loss KL Divergence*, beta besar memiliki *loss KL Divergence* lebih kecil yang berarti kekuatan *disentangle*-nya lebih baik. Dapat dilihat untuk model wajah, kucing, dan mobil, hasil *training* curve memiliki hasil yang mirip, yaitu pada loss rekonstruksi, beta kecil memiliki hasil yang rendah dibandingkan beta besar dan untuk *loss KL Divergence*, beta kecil memiliki hasil yang tinggi dibandingkan beta besar. Namun untuk dataset batik,terlihat pada loss rekonstruksi untuk hasil kurva *training* beta kecil dan besar memiliki hasil yang sedikit mirip, tetapi hasil kurva validasi beta kecil dan besar terlihat beta kecil memiliki hasil lebih rendah dibandingkan beta besar.

Evaluasi Rekonstruksi

Untuk evaluasi pertama, dilakukan evaluasi untuk citra hasil rekonstruksi dari setiap model Beta-VAE yang diteliti. Citra yang digunakan untuk evaluasi rekonstruksi didapatkan dari internet ataupun diambil langsung di dunia nyata untuk menghindari mendapati citra yang sama dari dataset latih. Selanjutnya citra di*resize* 64x64 dan setelah itu tidak dilakukan transformasi apapun lagi kecuali untuk citra evaluasi motif batik, yaitu dilakukan transformasi *grayscale*. Setelah citra tersebut direkonstruksi kembali, digunakan metrik evaluasi PSNR dan SSIM untuk menilai seberapa dekat citra hasil rekonstruksi dengan citra aslinya (gambar 10).

Untuk hasil metrik PSNR, hasil tertinggi yang didapatkan dari model CelebA dan nilai terkecil didapati oleh model Motif Batik Megamendung. Walaupun begitu, nilai yang didapatkan dari metrik PSNR tidak terlalu dapat dijadikan referensi, dikarenakan terkadang tidak cocok untuk merasakan kualitas visual dari citra dan juga hasil representasinya tidak dinormalisasikan [20]. Maka dari itu digunakan juga evaluasi metrik SSIM. Metrik evaluasi SSIM dirancang untuk mengikuti presepsi visual manusia [14]. Pada Tabel 2 terlihat nilai SSIM tertinggi didapatkan oleh model Beta-VAE CelebA. Pada Gambar 10 terlihat pada hasil rekonstruksi dengan wajah manusia masih terlihat bentuk wajah, rambut, hidung dan mulut, tetapi untuk bentuk mata tidak terlalu terlihat. Nilai SSIM terbesar didapatkan oleh model yang dilatih dari dataset CelebA dengan beta=10. Nilai terkecil didapatkan oleh model yang dilatih dengan dataset motif batik. Hasil rekonstruksi dari model Motif Batik yang didapatkan tidak terlalu jelas dan tidak terlalu terlihat bentuk motif megamendung pada citra rekonstruksi tersebut.

Table 2. Tabel hasil evaluasi rekonstruksi dari berbagai dataset dengan metrik evaluasi SSIM dan PSNR. Hasilevaluasi pada tabel merupakan rata–rata dari jumlah data (n) yang dihitung masing–masing evaluasinya.

Dataset	Beta	SSIM	PSNR (db)
OxfordCar $(\mathrm{n} = 2439)$	10	0.189	29.202
	100	0.132	28.924
CelebA $(\mathrm{n} = 2036)$	10	0.293	29.668
	100	0.242	29.475
Wajah Kucing $(\mathrm{n} = 3937)$	10	0.174	29.208
	100	0.126	28.905
Motif Batik	10	-0.010	27.904
Megamendung $(\mathrm{n} = 600)$	100	-0.002	27.917

Table 3. Tabel hasil evaluasi kualitas sampel dari berbagai dataset dengan metrik evaluasi FID.

Dataset	Beta	FID score
OxfordCar	10	305.426
	100	293.873
CelebA	10	**138.452**
	100	149.211
Wajah Kucing	10	188.945
	100	171.680
Motif Batik	10	431.118
Megamendung	100	402.966

FID score

Selanjutnya digunakan FID score untuk menilai seberapa baik citra yang dihasilkan model. Pada penelitian ini diambil sampel acak dari distribusi normal sebanyak 2400 sampel untuk masing-masing model. Kemudian sampel-sampel tersebut dibandingkan dengan dataset asli menggunakan FID score. Semakin kecil nilai FID-nya berarti semakin mirip sampel yang dihasilkan model dengan dataset aslinya. Pada Tabel 3 terlihat hasil model yang dilatih CelebA memiliki kemampuan memberikan sampel yang baik pada nilai beta $= 10$. Umumnya semakin besar nilai beta, maka semakin buruk hasil rekonstruksinya [21], tetapi pada beberapa dataset terdapat beta yang lebih besar memiliki hasil sampel yang lebih baik dibandingkan beta yang kecil. Contohnya pada Tabel 3 dari model yang dilatih dengan dataset OxfordCar, Wajah Kucing dan Motif Batik Megamendung, pada beta=100 memiliki nilai FID score yang lebih kecil dibandingkan beta $= 10$.

Penelitian Tambahan Dataset Batik

Dikarenakan hasil dari dataset batik yang tidak sebanding dengan dataset yang lain dilakukan penelitian tambahan untuk batik. Penelitian yang dilakukan dengan membuat model dengan dataset batik dengan nilai

Table 4. Tabel SSIM dan PSNR dari model motif batik dengan KLD *weight* yang berbeda

KLD *Weight*	Beta	SSIM	PSNR
0.0025	1	-0.008	27.930
0.000001	1	-0.018	27.903

beta yang sangat kecil untuk mengurangi nilai KLD loss yang sangat besar pada model batik dengan beta 10 dan 100. Ditambahkan KLD *Weight* yang dikalikan dengan beta untuk mengatur nilai dari beta agar tidak terlalu besar. KLD *Weight* yang digunakan untuk penelitian yang lain adalah 0.0025, sedangkan untuk penelitian tambahan tambahan ini diubah menjadi 0.000001. Terlihat pada Tabel 4 motif batik dengan KLD *Weight* yang sangat kecilpun, hasil dari SSIM dan PSNR-nya tidak jauh berbeda dengan model batik dengan beta 10 dan 100. Menggunakan dataset motif batik yang sama dan menurunkan nilai KLD *Weight*nya diharapkan dapat membuat hasil rekonstruksinya lebih baik, tetapi dilihat dari nilai SSIM dan PSNR tidak jauh berbeda. Jadi untuk evaluasi penelitian terhadap model motif batik akan berfokus pada model dengan beta 10 dan 100.

Terlihat bahwa nilai SSIM disini sangat rendah walaupun dengan nilai KLD *Weight* dan beta yang kecil. Kualitas hasil rekonstruksi citra juga buruk. Asumsi untuk penelitian batik yang memiliki hasil yang buruk, mungkin dikarenakan memiliki pelatihan model batik menggunakan dimensi yang tinggi, yaitu 128. Pada salah satu penelitan batik [9] telah diteliti bahwa penggunaan dimensi yang tinggi hanya menghasilkan sesuatu yang lebih tidak berarti yang berarti hasil rekonstruksi lebih buruk. Penggunaan *batch normalization* juga deteliti bahwa hasil dari rekonstruksi pada model batik lebih baik jika tidak menggunakan *batch normalization*. Tetapi pada penlitian ini digunakan nilai dimensi yang tinggi juga penggunaan *batch normalization* yang memungkinkan menyebabkan hasil rekonstruksi lebih buruk.

Traversal Plot

Evaluasi selanjutnya dilakukan pengujian menggunakan *traversal plot* untuk mengetahui pada setiap dimensi laten, karakteristik apa yang bisa didapatkan. Model menggunakan besar dimensi laten 128. Untuk menguji *traversal plot*, digunakan salah satu citra sebelumnya yang digunakan untuk tes rekonstrusi citra. Sebelum dimasukkan ke *decoder*, nilai dari satu dimensi pada z diubah dengan besar nilai $-3, -2, -1, 0, 1$, 2, 3 dengan nilai pada dimensi laten lainnya tetap. Setelah itu didapatkan 7 hasil citra dari perubahan pada satu dimensi. Pada model dengan beta $= 10$, hasil *traversal plot* untuk model OxfordCar tidak terlalu jelas perubahan karakteristik apa yang didapat. Untuk model CelebA secara berurutan dimensi laten ke-11, 16, dan 31 berdampak pada perubahan gender, panjang rambut, dan umur. Untuk model Wajah Kucing secara berurutan dimensi laten ke-3, 31, 104, dan 124 berdampak pada perubahan kecerahan wajah, warna kulit, besar mata, dan sudut pencerahan. Untuk model Motif Batik hanya terlihat sedikit corak-corak batik, tetapi tidak terlihat perubahan karakteristik yang berarti.

Untuk model dengan beta $= 100$, memiliki *traversal plot* dengan pemisahan karakteristik yang lebih padat dibandingkan beta=10. Untuk model OxfordCar secara berurutan dimensi laten ke-1, 5, dan 22 berdampak pada perubahan besar mobil, warna latar, dan rotasi. Untuk model CelebA secara berurutan dimensi laten ke-43, 47, dan 100 berdampak pada perubahan rotasi, sudut pencerahan, dan kecerahan wajah. Model Wajah Kucing secara berurutan dimensi laten ke-5, 25, 42, 51, dan 69 berdampak pada perubahan sudut menghadap wajah, warna mata, besar mata, rotasi, dan kecerahan wajah. Untuk model Motif Batik tidak terlihat perubahan karakteristik yang berarti dan hasilnya lebih buram dari beta=10. Perbedaan dari beta besar dan kecil terlihat dari hasil rekonstruksi dan *disentangle*-nya. Beta yang lebih besar menghasilkan hasil *disentangle* yang lebih baik, tetapi hanya menampilkan warna yang rata-rata dominan pada citra dataset. Pada hasil *traversal plot* tidak ditampilkan seluruh perubahan pada setiap dimensi karena tidak setiap dimensi memiliki perubahan karakteristik. Untuk melihat hasil dari *traversal plot* dari setiap model, dapat dilihat pada bagian Lampiran.

Hasil Sampel

Dari model beta-VAE yang telah dilatih, dapat dijadikan sebagai generator penghasil citra baru atau sampel yang diambil dari ruang laten. Sampel acak yang didapat dengan cara mendapatkan nilai acak dari distribusi normal dengan jumlah sesuai dimensi laten yang ditentukan, pada penelitian ini digunakan 128 dimensi laten. Kemudian nilai tersebut dimasukkan ke bagian *decoder* dari model tersebut dan didapat hasil citra sampel. Menggunakan *decoder* dari model BetaVAE yang sudah dilatih, biasa didapatkan generator citra untuk menghasilkan citra yang baru. Sampel yang didapat merupakan hasil citra baru walaupun begitu hasil rekonstruksinya tidak selalu baik tergantung dari kemampuan model itu sendiri. Contoh sampel acak dapat dilihat pada Gambar 8.

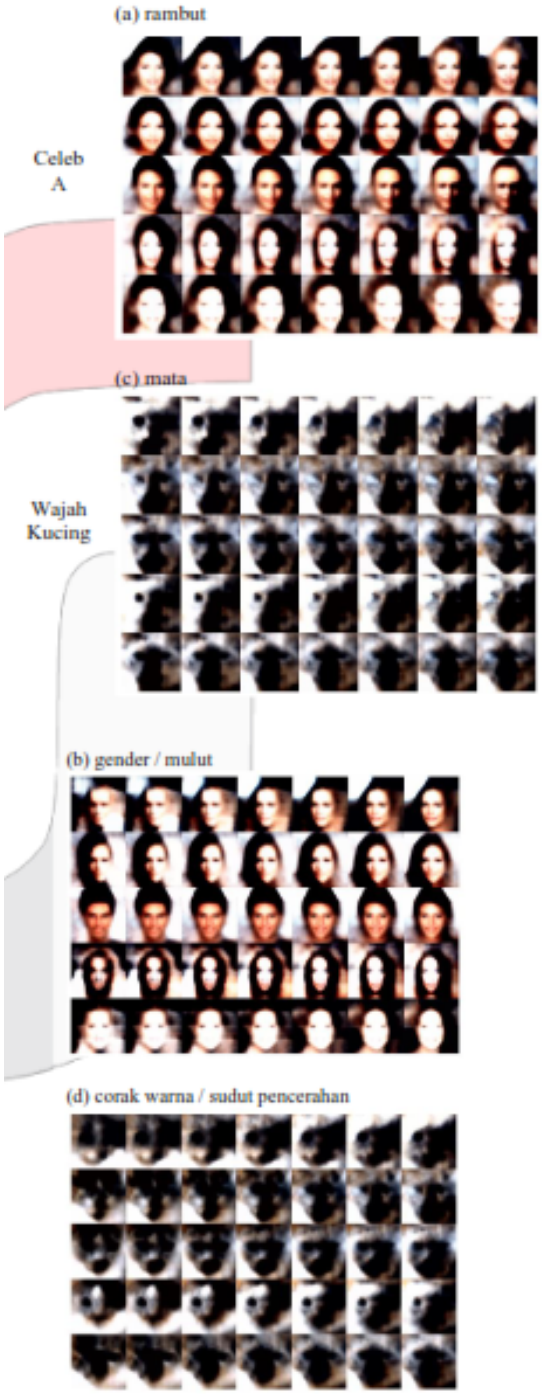
Hasil *Disentangle*

Selain mendapat citra baru, model beta-VAE dapat memanipulasi variabel laten untuk mendapatkan perubahan karakteristik. Menggunakan citra input yang dimasukkan pada *encoder* atau menggunakan nilai acak dari distribusi normal, kemudian didapatkan variabel laten yang dapat dimanipulasi nilainya untuk mengontrol karakteristik dari citra input. Untuk menemukan dimensi laten yang memiliki perubahan karakteristik, dilakukan pengecekan satu per satu dengan mengubah salah satu dimensi tersebut dengan jarak nilai [-3, 3] dengan nilai dimensi yang lainnya sama. Pada Gambar 11 terlihat hasil manipulasi model untuk CelebA dan Wajah Kucing. Contoh manipulasi variabel laten yang didapat untuk CelebA adalah perubahan panjang rambut dan perubahan gender. Untuk Wajah Kucing terdapat perubahan bentuk mata dan perubahan corak warna atau sudut pencerahan. Contoh didapat dari model dengan beta=10. Setiap baris citra dari kiri ke kanan merupakan hasil dari perubahan nilai dimensi yang diubah dengan rentang nilai [-3, 3]. Pada model CelebA, hasil yang didapat terdapat perubahan pajang rambut (a) dan perubahan gender dari laki-laki ke perempuan atau perubahan mulut yang tidak senyum ke senyum (b). Untuk model Wajah Kucing terdapat perubahan bentuk mata (a) dan perubahan corak atau sudut pencerahan (d). data sudah dilampaui oleh *Diffusion Model*. Walaupun bergitu, untuk menghasilkan citra resolusi tinggi, *Diffusion Model* masih bergantung pada VAE untuk proses kompresi dan rekonstruksi data. Hal ini menarik untuk diteliti tentang pengaruh varian VAE yang berbeda pada *Diffusion Model*, karena saat ini *Diffusion Model* menjadi *state-of-the-art*.

Kesimpulan

Pada penelitian ini, model Beta-VAE telah diterapkan kepada empat citra dataset, terdiri dari dataset citra mobil, wajah manusia, wajah kucing dan motif batik megamendung. Terlihat kemampuan BetaVAE dalam melakukan *disentangle* karakteristik dari distribusi data. Pada umumnya, beta-VAE dapat melakukan rekonstruksi dan menghasilkan variasi data baru dari pengaturan dimensi laten. Tetapi, kemampuan model berbeda-beda tergantung pada dataset yang digunakan. Terlebih lagi pengaturan atribut yang dihasilkan pun berbeda tergantung dari kompleksitas data. Berikut kesimpulan dari penelitian tentang kemampuan model Beta-VAE terhadap dataset yang berbeda:

1. Pada hasil penelitian, model CelebA beta=10 memiliki hasil terbaik dari model beta-VAE lainnya. Didapatkan dari metrik evaluasi bahwa



Gambar 11. Hasil manipulasi laten variabel pada CelebA dan Wajah Kucing

beta dengan nilai kecil memiliki hasil lebih baik dibandingkan yang beta besar.

2. Beta yang berbeda menghasilkan hasil *disentangle* di dimensi laten yang berbeda.
3. Dilihat dari *traversal plot* yang didapatkan, setiap dimensi laten memiliki karakteristik yang berbeda antara satu dimensi dengan dimensi yang lainnya. Bisa dilihat bahwa karakteristik dari setiap dimensi laten tidak dapat diatur sesuai keinginan.
4. Kemampuan *disentangle* dan rekonstruksi dari metode Beta-VAE asli ini kurang robust. Metode ini bergantung pada dataset yang memiliki kualitas yang baik. Dataset yang terlalu abstrak, pada penelitian ini yaitu dataset OxfordCar dan Batik tidak memiliki hasil *disentangle* yang baik dan rekonstruksi yang buruk dibandingkan dataset CelebA dan Wajah Kucing. Karakteristik ini mungkin dikarenakan oleh konsistensi yang ada pada dataset tersebut, contohnya seperti wajah yang menghadap arah yang sama dan warna wajah yang tidak jauh berbeda pada dataset CelebA.

Sebagai pernutup, Beta-VAE adalah arsitektur yang berguna untuk melakukan kompresi dan sampling data. Tetapi, kemampuan rekonstruksinya berbanding terbalik dengan kemampuan *disentangle* yang menghasilkan hasilnya terlihat blur. Saran untuk penelitian selanjutnya dapat dieksplorasi nilai beta atau penggunaan loss lain yang dapat mengoptimalkan kualitas citra yang dihasilkan. Untuk rekomendasi penelitian selanjutnya, menarik untuk diteliti tentang pengaruh VAE terhadap *Diffusion Model*. Kemampuan untuk menghasilkan data baru dari sebuah distribusi.

Daftar Pustaka

1. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative Adversarial Networks. 2014.
2. Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. CoRR. 2016;abs/1606.03657. Available from: http://arxiv.org/abs/1606.03657.
3. Kulkarni TD, Whitney W, Kohli P, Tenenbaum JB. Deep Convolutional Inverse Graphics Network. CoRR. 2015;abs/1503.03167. Available from: http://arxiv.org/abs/1503.03167.
4. Kingma DP, Welling M. Auto-Encoding Variational Bayes. 2022.
5. Zhou L, Deng W, Wu X. Unsupervised anomaly localization using VAE and beta-VAE. CoRR. 2020;abs/2005.10686. Available from: https://arxiv.org/abs/2005.10686.
6. Pandey K, Mukherjee A, Rai P, Kumar A. VAEs meet Diffusion Models: Efficient and High-Fidelity Generation. In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications; 2021. Available from: https://openreview.net/forum?id=-J8dM4ed_92.
7. Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, et al. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In: International Conference on Learning Representations (ICLR); 2017. Available from: https://openreview.net/forum?id=Sy2fzU9gl.
8. Fil M, Mesinovic M, Morris M, Wildberger J. Beta-VAE Reproducibility: Challenges and Extensions. CoRR. 2021;abs/2112.14278. Available from: https://arxiv.org/abs/2112.14278.
9. Ihsan AF. Initial Study of Batik Generation using Variational Autoencoder. Procedia Computer Science. 2023;227:785-94. Available from: https://doi.org/10.1016/j.procs.2023.10.584.
10. Liu Z, Luo P, Wang X, Tang X. Deep Learning Face Attributes in the Wild. In: Proceedings of the International Conference on Computer Vision (ICCV); 2015. .
11. Aubry M, Maturana D, Efros A, Russell B, Sivic J. Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2014. .
12. Paysan P, Knothe R, Amberg B, Romdhani S, Vetter T. A 3D Face Model for Pose and Illumination Invariant Face Recognition. In: 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance; 2009. p. 296-301. Available from: https://doi.org/10.1109/AVSS.2009.58.

13. Bank D, Koenigstein N, Giryes R. Autoencoders. CoRR. 2020;abs/2003.05991. Available from: https://arxiv.org/abs/2003.05991.
14. Nilsson J, Akenine-Möller T. Understanding SSIM. ArXiv. 2020;abs/2006.13846. Available from: https://api.semanticscholar.org/CorpusID:2200416.
15. Wang Z, Bovik A, Sheikh H, Simoncelli E. Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing. 2004;13:600-12. Available from: https://doi.org/10.1109/TIP.2003.819861.
16. Lucic M, Kurach K, Michalski M, Gelly S, Bousquet O. Are GANs Created Equal? A Large-Scale Study. In: Advances in Neural Information Processing Systems (NeurIPS); 2017. Available from: https://api.semanticscholar.org/CorpusID:4053393.
17. Subramanian AK. PyTorch-VAE; 2020. GitHub. GitHub repository. Available from: https://github.com/AntixK/PyTorch-VAE.
18. Krause J, Stark M, Deng J, Fei-Fei L. 3D Object Representations for Fine-Grained Categorization. In: 2013 IEEE International Conference on Computer Vision Workshops; 2013. p. 554-61. Available from: https://doi.org/10.1109/ICCVW.2013.77.
19. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. CoRR. 2015;abs/1512.03385. Available from: https://arxiv.org/abs/1512.03385.
20. Sara U, Akter M, Uddin MS. Image Quality Assessment through FSIM, SSIM, MSE and PSNR – A Comparative Study. Journal of Computer and Communications. 2019;7:8-18. Available from: https://doi.org/10.4236/jcc.2019.73002.
21. Burgess CP, Higgins I, Pal A, Matthey L, Watters N, Desjardins G, et al. Understanding disentangling in β-VAE. ArXiv. 2018;abs/1804.03599. Available from: https://api.semanticscholar.org/CorpusID:4879659.