



TESIS - ES235401

**INTEGRASI EFFICIENT TRANSFORMER KE DALAM
ENHANCED GROUP CONVOLUTIONAL NEURAL
NETWORK UNTUK MENINGKATKAN KINERJA
SINGLE IMAGE SUPER-RESOLUTION**

**FIKRI YOMA ROSYIDAN
6026232019**

**DOSEN PEMBIMBING
Dr. Rarasmaya Indraswari, S.Kom.
1995202012057**

**PROGRAM MAGISTER
DEPARTEMEN SISTEM INFORMASI
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2025**



THESIS - ES235401

**INTEGRATION OF EFFICIENT TRANSFORMER
INTO ENHANCED GROUP CONVOLUTIONAL
NEURAL NETWORK TO IMPROVE SINGLE IMAGE
SUPER-RESOLUTION PERFORMANCE**

**FIKRI YOMA ROSYIDAN
6026232019**

**SUPERVISOR
Dr. Rarasmaya Indraswari, S.Kom.
1995202012057**

**MASTER PROGRAM
DEPARTMENT OF INFORMATION SYSTEMS
FACULTY OF INTELLIGENT ELECTRICAL AND INFORMATICS
TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2025**

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Sistem Informasi (M.Kom.)
di
Institut Teknologi Sepuluh Nopember

Oleh:

Fikri Yoma Rosyidan
NRP: 6026232019

Tanggal Ujian: 14 Mei 2025
Periode Wisuda ITS: 132

Disetujui oleh:
Pembimbing:

Dr. Rarasmaya Indraswari, S.Kom
NIP: 1995202012057

**Penguji:**

Prof. Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D
NIP: 198201202005012001



Dr. Ir. Aris Tjahyanto, M.Kom
NIP: 196503101991021001



Surabaya, 20 Juni 2025
Kepala Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas



LP/P/25/288

Prof. Dr. Wriwik Anggraeni, S.Si, M.Kom
NIP: 197601232001122002

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP : Fikri Yoma Rosyidan / 6026232019
Program studi : S2 Sistem Informasi
Dosen Pembimbing : Dr. Rarasmaya Indraswari, S.Kom / 1995202012057 / NIP

dengan ini menyatakan bahwa Tesis dengan judul "Integrasi Efficient Transformer Ke Dalam Enhanced Group Convolutional Neural Network Untuk Meningkatkan Kinerja Single Image Super-Resolution" adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 20 Juni 2025

Mengetahui

Dosen Pembimbing



Dr. Rarasmaya Indraswari, S.Kom
NIP. 1995202012057

Mahasiswa



Fikri Yoma Rosyidan
NRP. 6026232019



INTEGRASI EFFICIENT TRANSFORMER KE DALAM ENHANCED GROUP CONVOLUTIONAL NEURAL NETWORK UNTUK MENINGKATKAN KINERJA SINGLE IMAGE SUPER-RESOLUTION

Nama Mahasiswa : Fikri Yoma Rosyidan
NRP : 6026232019
Dosen Pembimbing 1 : Dr. Rarasmaya Indraswari, S.Kom.

ABSTRAK

Single Image Super-Resolution (SISR) merupakan salah satu tantangan krusial dalam pengolahan citra digital, di mana tujuan utamanya adalah mengubah citra beresolusi rendah menjadi citra beresolusi tinggi dengan detail yang tajam dan akurasi visual yang tinggi. Di era digital saat ini, kebutuhan akan citra berkualitas tinggi sangat mendesak di berbagai bidang, mulai dari diagnosa medis yang membutuhkan interpretasi detail citra MRI dan CT scan, hingga aplikasi pengawasan keamanan dan hiburan digital yang mengandalkan ketajaman visual. Meskipun banyak pendekatan telah dikembangkan dengan menggunakan *Convolutional Neural Networks* (CNN) seperti *Enhanced Group Convolutional Neural Network* (EGCNN), keterbatasan dalam menangkap hubungan jarak jauh antar piksel menjadi penghambat dalam mencapai hasil rekonstruksi yang optimal. Di sisi lain, perkembangan *Transformer* dengan mekanisme *self-attention* telah membuka peluang baru untuk memodelkan konteks global, namun kompleksitas komputasinya sering kali menjadi kendala. Untuk itu, penelitian ini mengusulkan integrasi *Efficient Transformer* (ET) ke dalam arsitektur EGCNN guna mengatasi kekurangan tersebut dan menghasilkan model SISR yang tidak hanya mampu merekonstruksi detail tekstur halus dan pola global, tetapi juga efisien secara komputasi. Pendekatan yang diusulkan memanfaatkan kekuatan EGCNN dalam mengekstraksi fitur lokal melalui *group convolution*, yang kemudian diperkaya dengan konteks global melalui mekanisme *efficient self-attention* pada ET. Output fitur yang telah diproses diubah menjadi bentuk spasial kembali melalui tahap rekonstruksi dan upsampling, sehingga menghasilkan citra beresolusi tinggi yang mendekati *ground truth*. Evaluasi dilakukan secara menyeluruh dengan menggunakan metrik kuantitatif seperti *Peak Signal-to-Noise Ratio* (PSNR) dan *Structural Similarity Index* (SSIM) pada dataset standar pengujian seperti Set5, Set14, BSD100, dan Urban100, serta didukung oleh analisis kualitatif. Hasil penelitian menunjukkan bahwa integrasi ET secara signifikan meningkatkan performa model dibandingkan dengan EGCNN *baseline*, khususnya pada skala pembesaran $\times 2$, dan menawarkan alternatif kompetitif yang efisien dibandingkan model-model *state-of-the-art* seperti SwinIR.

Kata kunci: *Single Image Super-Resolution, Enhanced Group Convolutional Neural Network, Efficient Transformer*

INTEGRATION OF EFFICIENT TRANSFORMER INTO ENHANCED GROUP CONVOLUTIONAL NEURAL NETWORK TO IMPROVE SINGLE IMAGE SUPER- RESOLUTION PERFORMANCE

Student Name : Fikri Yoma Rosyidan
NRP : 6026232019
Advisor : Dr. Rarasmaya Indraswari, S.Kom.

ABSTRACT

Single Image Super-Resolution (SISR) is one of the critical challenges in digital image processing, aiming to transform low-resolution images into high-resolution images with sharp details and high visual accuracy. In today's digital era, the demand for high-quality images is increasingly urgent across various fields, from medical diagnosis which requires detailed interpretation of MRI and CT scan images to security surveillance and digital entertainment that rely on visual sharpness. Although many approaches have been developed using Convolutional Neural Networks (CNNs), such as the Enhanced Group Convolutional Neural Network (EGCNN), their limitation in capturing long-range pixel relationships has hindered achieving optimal reconstruction results. On the other hand, the advent of Transformers with self-attention mechanisms has opened new opportunities for modeling global context, although their computational complexity often poses challenges. Therefore, this research proposes the integration of Efficient Transformer (ET) into the EGCNN architecture to overcome these shortcomings and produce a SISR model that not only reconstructs fine texture details and global patterns but is also computationally efficient. The proposed approach leverages the strength of EGCNN in extracting local features through group convolution, which is then enriched with global context via the efficient self-attention mechanism in ET. The output features processed by ET are transformed back into a spatial representation through reconstruction and upsampling stages, resulting in high-resolution images that closely approximate the ground truth. The model is evaluated comprehensively using quantitative metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) on standard testing datasets including Set5, Set14, BSD100, and Urban100, complemented by qualitative analysis. The research findings indicate that the integration of ET significantly enhances model performance compared to the baseline EGCNN, particularly at a scaling factor of $\times 2$, and offers a competitive, efficient alternative to state-of-the-art models like SwinIR.

Keywords: *Single Image Super-Resolution, Enhanced Group Convolutional Neural Network, Efficient Transformer*

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, karunia, dan petunjuk-Nya sehingga tesis yang berjudul “Integrasi *Efficient Transformer* ke dalam *Enhanced Group Convolutional Neural Network* untuk Meningkatkan Kinerja *Single Image Super-Resolution*” dapat diselesaikan dengan baik. Tesis ini disusun sebagai bagian dari pemenuhan persyaratan akademik dalam menyelesaikan Program Magister pada Departemen Sistem Informasi, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember. Selama proses penelitian dan penyusunan tesis ini, Penulis memperoleh berbagai dukungan dalam bentuk bimbingan ilmiah, masukan substantif, serta motivasi yang konstruktif dari berbagai pihak. Oleh karena itu, Penulis menyampaikan apresiasi dan ucapan terima kasih yang sebesar-besarnya kepada:

1. Orang tua tercinta, Dr. Yoso Wiyarno, M.Kes dan Dra. Mamik Pujowati, M.Pd., yang selalu memberikan doa dalam berbagai hal kepada Penulis.
2. Kakak dan adik, Ir. Bonanza Yoma Pratama, S.ST., M.T. dan Yomi Fauziyah Salsabila, yang telah memberikan dukungan kepada Penulis.
3. Ibu Dr. Rarasmaya Indraswari, S.Kom., selaku dosen pembimbing dan dosen wali yang telah memberikan arahan, dukungan, motivasi, dan pemahaman yang berharga selama masa perkuliahan, terutama terkait pengembangan tesis ini.
4. Ibu Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D. dan Bapak Dr. Ir. Aris Tjahyanto, M.Kom. selaku dosen penguji I dan II yang telah memberikan evaluasi dan masukan berharga terhadap tesis ini.
5. Seluruh dosen di Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember, yang telah berbagi ilmu dan pengalaman yang berharga dengan Penulis selama studi.
6. Seluruh staf dan karyawan di Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember, yang telah memberikan bantuan dalam administrasi dan keperluan lainnya selama perkuliahan.
7. Teman seperjuangan Penulis, Ganendra Aby Bhamakerti dan Alfado Rafly Hermawan, yang senantiasa menemani perjalanan kuliah Penulis.
8. Teman-teman dari Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember, yang telah mengisi hari-hari Penulis selama kuliah.
9. Semua pihak lainnya yang tidak dapat disebutkan satu per satu, yang telah membantu Penulis selama masa perkuliahan ini.

Penulisan tesis ini masih jauh dari kata sempurna. Oleh karena itu, Penulis terbuka menerima segala kritik dan saran yang dapat membangun dan meningkatkan kualitas tesis ini. Semoga hasil tesis ini dapat memberikan manfaat bagi berbagai pihak.

Surabaya, 14 Mei 2025
Penulis,

Fikri Yoma Rosyidan

DAFTAR ISI

LEMBAR PENGESAHAN TESIS	v
PERNYATAAN ORISINALITAS.....	vii
ABSTRAK.....	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xiv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xix
DAFTAR KODE	xx
DAFTAR LAMPIRAN	xxi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	4
1.5 Kontribusi Penelitian	4
1.6 Batasan Penelitian.....	5
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI	6
2.1 Kajian Pustaka	6
2.2 Dasar Teori	10
2.2.1 <i>Single Image Super-Resolution (SISR)</i>	11
2.2.2 <i>Convolutional Neural Networks (CNN)</i>	13
2.2.3 Pendekatan Berbasis CNN dalam SISR	14
2.2.4 <i>Group Convolution</i>	16

2.2.5	<i>Enhanced Group Convolutional Neural Network (EGCNN)</i>	19
2.2.6	<i>Transformer</i> dan Mekanisme <i>Self-Attention</i>	20
2.2.7	<i>Efficient Transformer (ET)</i>	23
2.2.8	Metrik Evaluasi Kinerja Model SISR	26
2.2.8.1	<i>Peak Signal-to-Noise Ratio (PSNR)</i>	26
2.2.8.2	<i>Structural Similarity Index (SSIM)</i>	27
BAB 3 METODOLOGI PENELITIAN.....		29
3.1	Diagram Metodologi Penelitian	29
3.2	Uraian Metodologi Penelitian	29
3.2.1	Studi Literatur	30
3.2.2	Identifikasi Masalah dan Rumusan Penelitian	31
3.2.3	Pengumpulan Data Sekunder	31
3.2.4	<i>Pre-processing</i> Data.....	34
3.2.5	Desain Arsitektur Model.....	35
3.2.6	Proses Pelatihan Model.....	38
3.2.7	Uji Coba dan Evaluasi Model	41
3.2.8	Penulisan Penelitian	43
3.3	Rencana Jadwal Pelaksanaan Penelitian	43
BAB 4 HASIL DAN PEMBAHASAN.....		44
4.1	Deskripsi Data.....	44
4.2	Lingkungan Implementasi.....	45
4.3	Hasil Penelitian	46
4.3.1	Hasil <i>Pre-processing</i> Data	46
4.3.2	Modifikasi Integrasi Arsitektur Model.....	47
4.3.2.1	Eksperimen terhadap Jumlah Iterasi.....	51
4.3.2.2	Eksperimen terhadap Jumlah Blok.....	54

4.3.2.3	Eksperimen terhadap <i>Hyperparameter</i>	58
4.3.3	Modifikasi Modul <i>Upsampling</i> untuk Faktor Skala Pecahan.....	60
4.3.4	Evaluasi Kuantitatif.....	62
4.3.4.1	Hasil Kuantitatif Skala Integer.....	62
4.3.4.2	Hasil Kuantitatif Skala Pecahan.....	71
4.3.5	Evaluasi Kualitatif.....	77
4.4	Pembahasan	80
BAB 5 KESIMPULAN DAN SARAN		84
5.1	Kesimpulan.....	84
5.2	Saran	85
DAFTAR PUSTAKA.....		86
LAMPIRAN		92
BIODATA PENULIS.....		103

DAFTAR GAMBAR

Gambar 2.1 Proses SISR (Viriyavisuthisakul et al., 2022).....	11
Gambar 2.2 Komponen lapisan dalam CNN (Szegedy et al., 2016).....	14
Gambar 2.3 Arsitektur SRCNN (Dong et al., 2014).....	15
Gambar 2.4 Operasi <i>group convolution</i> (G. Huang et al., 2018).....	17
Gambar 2.5 Arsitektur EGCNN (Tian et al., 2022).....	20
Gambar 2.6 Mekanisme <i>self-attention</i> (L. Zhang, 2018).....	21
Gambar 2.7 Arsitektur <i>Efficient Transformer</i> (Cai et al., 2023).....	24
Gambar 3.1 Diagram metodologi penelitian.....	29
Gambar 3.2 Jumlah citra pada setiap kategori dalam dataset DF2K.....	32
Gambar 3.3 Prosentase jumlah citra pada setiap kategori dalam dataset DF2K... 33	
Gambar 3.4 Diagram <i>pre-processing</i> data.....	34
Gambar 3.5 Desain arsitektur usulan.....	35
Gambar 4.1 sampel data citra dari dataset DF2K.....	44
Gambar 4.2 Hasil pre-processing citra HR yang telah di-downscale.....	46
Gambar 4.3 Diagram MFC Module.....	49
Gambar 4.4 Proses integrasi dengan Performer.....	50
Gambar 4.5 Perkembangan metrik PSNR dan SSIM berdasarkan jumlah iterasi 52	
Gambar 4.6 Perbandingan hasil rekonstruksi $\times 2$ berdasarkan jumlah iterasi.....	53
Gambar 4.7 Perbandingan hasil rekonstruksi $\times 2$ pada tiga variasi arsitektur.....	56
Gambar 4.8 Perbandingan hasil rekonstruksi $\times 3$ pada tiga variasi arsitektur.....	57
Gambar 4.9 Perbandingan hasil rekonstruksi $\times 4$ pada tiga variasi arsitektur.....	58
Gambar 4.10 Grafik nilai PSNR faktor skala integer pada keempat data uji.....	64
Gambar 4.11 Grafik nilai SSIM faktor skala integer pada keempat data uji.....	65
Gambar 4.12 Perbandingan hasil rekonstruksi $\times 2$ pada tiga arsitektur berbeda... 68	
Gambar 4.13 Perbandingan hasil rekonstruksi $\times 3$ pada tiga arsitektur berbeda... 69	
Gambar 4.14 Perbandingan hasil rekonstruksi $\times 4$ pada tiga arsitektur berbeda... 70	
Gambar 4.15 Grafik nilai PSNR faktor skala pecahan pada keempat data uji.....	73
Gambar 4.16 Grafik nilai SSIM faktor skala pecahan pada keempat data uji.....	74
Gambar 4.17 Perbandingan hasil rekonstruksi $\times 5/4$ pada tiga arsitektur berbeda75	

Gambar 4.18 Perbandingan hasil rekonstruksi $\times 3/2$ pada tiga arsitektur berbeda	76
Gambar 4.19 Perbandingan hasil rekonstruksi $\times 7/4$ pada tiga arsitektur berbeda	77
Gambar 4.20 Skor per citra berdasarkan model	79
Gambar 4.21 Desain arsitektur hybrid EGCNN+ET	81

DAFTAR TABEL

Tabel 2.1 Perbandingan metode sebelumnya dengan metode usulan.....	9
Tabel 3.1 Rencana jadwal pelaksanaan penelitian.....	43
Tabel 4.1 Spesifikasi perangkat keras.....	45
Tabel 4.2 Spesifikasi perangkat lunak	45
Tabel 4.3 Eksperimen model EGCNN+ET (6-2) terhadap jumlah iterasi.....	52
Tabel 4.4 Hasil kuantitatif tiga variasi arsitektur EGCNN+ET pada skala integer	55
Tabel 4.5 Konfigurasi eksperimen hyperparameter pada EGCNN+ET (6-2).....	59
Tabel 4.6 Hasil Eksperimen hyperparameter pada EGCNN+ET (6-2)	60
Tabel 4.7 Keterangan model untuk skala integer.....	62
Tabel 4.8 Hasil PSNR seluruh arsitektur pada skala integer	63
Tabel 4.9 Hasil SSIM seluruh arsitektur pada skala integer	63
Tabel 4.10 Perbandingan jumlah parameter dan waktu inferensi pada semua model	66
Tabel 4.11 Keterangan model untuk skala pecahan.....	71
Tabel 4.12 Hasil PSNR seluruh arsitektur pada skala pecahan	71
Tabel 4.13 Hasil SSIM seluruh arsitektur pada skala pecahan	72
Tabel 4.14 Analisis kualitatif penilaian upscaling citra.....	78
Tabel 4.15 Data mentah penilaian responden berdasarkan citra dan model.....	79
Tabel 4.16 Mean Opinion Score per model	79

DAFTAR KODE

Kode 4.1 Fungsi utama kode integrasi EGCNN+ET	47
Kode 4.2 Modul <i>MeanShift</i>	48
Kode 4.3 Ekstraksi fitur lokal melalui modul MFC	48
Kode 4.4 Ekstraksi fitur global melalui blok <i>Performer</i>	49
Kode 4.5 Modul <i>Upsampling</i>	50
Kode 4.6 Metode <i>forward</i> untuk fungsi utama.....	51
Kode 4.7 Modifikasi modul <i>upsampling</i> untuk faktor skala pecahan	61

DAFTAR LAMPIRAN

Lampiran 1. Kuesioner survei penilaian kualitas citra hasil super-resolution 92

BAB 1

PENDAHULUAN

Pada bab pendahuluan ini akan dipaparkan mengenai latar belakang dilakukannya penelitian, pembangkitan pertanyaan penelitian (*research question*) berdasarkan latar belakang permasalahan, tujuan dilakukannya penelitian, manfaat yang diharapkan bisa diperoleh dari penelitian yang akan dilakukan, batasan dan asumsi yang membatasi jangkauan penelitian ini agar tetap fokus pada permasalahan, serta manfaat yang diberikan oleh penelitian ini.

1.1 Latar Belakang

Single Image Super-Resolution (SISR) adalah teknik penting dalam pengolahan citra digital yang bertujuan untuk meningkatkan resolusi spasial dari citra beresolusi rendah menjadi citra beresolusi tinggi. (Bashir et al., 2021). Teknologi ini menjadi krusial dalam berbagai aplikasi terutama karena kebutuhan akan citra berkualitas tinggi dalam berbagai bidang seperti medis, pengawasan, dan fotografi digital (Ye et al., 2023). Dalam bidang medis, SISR membantu meningkatkan kualitas citra MRI atau CT scan yang memungkinkan identifikasi detail kecil yang penting untuk diagnosis. Di sektor pengawasan, teknologi ini dapat memperjelas gambar dari kamera keamanan yang sering kali beresolusi rendah sehingga memudahkan pengenalan objek yang ingin dilakukan observasi. Selain itu, dalam fotografi digital penggunaan SISR memungkinkan pemakai memperbesar gambar tanpa kehilangan detail yang penting. Kebutuhan terhadap teknologi SISR terus meningkat seiring dengan permintaan akan citra berkualitas tinggi yang lebih detail di berbagai industri. Urgensi ini juga didukung oleh kemajuan kecerdasan buatan (AI) dan *machine learning* yang semakin mempermudah peningkatan resolusi gambar dengan akurasi yang lebih baik dibandingkan dengan metode konvensional sebelumnya (Yang et al., 2019). Namun, tantangan utama yang dihadapi adalah bagaimana meningkatkan resolusi citra tanpa mengorbankan detail penting dan kualitas visual. Hal ini kemudian membangkitkan pertanyaan penelitian yaitu bagaimana peneliti dapat mengembangkan model SISR yang efisien secara komputasi dan sekaligus mampu

merekonstruksi detail tekstur halus serta pola global dalam citra untuk memenuhi kebutuhan industri akan citra berkualitas tinggi.

Pendekatan berbasis *Deep Learning*, khususnya *Convolutional Neural Networks* (CNN) dalam beberapa tahun terakhir telah mendominasi penelitian di bidang SISR karena kemampuannya dalam mengekstraksi fitur-fitur penting dari citra secara otomatis (Zhu et al., 2021). *Enhanced Group Convolutional Neural Network* (EGCNN) adalah salah satu arsitektur CNN yang dirancang untuk meningkatkan efisiensi komputasi dan kinerja dalam tugas SISR (Tian et al., 2022). EGCNN memanfaatkan teknik grup konvolusi yang membagi saluran input menjadi beberapa grup, sehingga mengurangi jumlah parameter dan beban komputasi tanpa mengorbankan akurasi model secara signifikan. Dengan demikian, EGCNN menjadi pilihan yang efisien untuk aplikasi *real-time* dan perangkat dengan keterbatasan sumber daya (Tian et al., 2022). Namun, meskipun EGCNN efektif dalam mengekstraksi fitur lokal, model ini memiliki keterbatasan dalam menangkap hubungan jarak jauh (*long-range dependencies*) antar piksel dalam citra. Hal ini disebabkan oleh sifat lokal dari operasi konvolusi yang hanya mempertimbangkan informasi dalam jendela filter yang sempit (Hong et al., 2023). Akibatnya, EGCNN mungkin kurang optimal dalam merekonstruksi detail tekstur yang halus dan pola-pola global dalam citra beresolusi tinggi.

Di sisi lain, *Transformer*, yang awalnya dikembangkan untuk pemrosesan bahasa alami, telah menunjukkan potensi besar dalam tugas-tugas visi komputer, termasuk SISR (Lu et al., 2022). *Transformer* mampu menangkap hubungan jangka panjang dalam data melalui mekanisme perhatian diri (*self-attention mechanism*), yang memungkinkan model untuk mempertimbangkan interaksi antar piksel di seluruh citra (Han et al., 2023). Namun, penerapan *Transformer* pada citra beresolusi tinggi menghadapi tantangan komputasi yang besar karena kompleksitasnya yang kuadrat terhadap ukuran input (Pau & Aymone, 2024). Untuk mengatasi masalah ini, *Efficient Transformer* (ET) dikembangkan sebagai varian *Transformer* yang lebih efisien secara komputasi (Xu et al., 2021). ET mengurangi kompleksitas komputasi dengan berbagai teknik seperti pengurangan dimensi dan perhatian yang lebih efisien, sehingga lebih cocok untuk diaplikasikan pada data citra (Zheng et al., 2024). Penelitian sebelumnya telah berhasil

mengintegrasikan ET ke dalam model SISR seperti *Efficient Super Resolution Transformer* (ESRT) dengan arsitektur jaringan *Residual Channel Attention Network* (RCAN) dengan menghasilkan peningkatan kinerja yang signifikan dalam kualitas citra dan efisiensi komputasi (Lu et al., 2022).

Berdasarkan uraian di atas, tampak bahwa ET memiliki potensi untuk meningkatkan performa EGCNN apabila dilakukan integrasi. Dengan memanfaatkan efisiensi komputasi EGCNN dan kemampuan ET dalam menangkap hubungan jarak jauh, diharapkan integrasi ini dapat mengatasi keterbatasan yang ada pada EGCNN dan meningkatkan kualitas hasil SISR secara keseluruhan. Integrasi ini juga berpotensi menciptakan model yang lebih efisien dan efektif untuk digunakan pada aplikasi *real-time* yang memerlukan pemrosesan cepat dan akurasi tinggi. Namun, hingga saat ini integrasi ET ke dalam EGCNN belum dieksplorasi secara mendalam. Kurangnya studi ini menunjukkan adanya kesenjangan penelitian yang perlu diisi untuk meningkatkan performa model SISR. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan model baru yang menggabungkan keunggulan EGCNN dan ET serta menguji efektivitasnya dalam meningkatkan kualitas rekonstruksi citra beresolusi tinggi.

1.2 Rumusan Masalah

Dari permasalahan di atas, pertanyaan penelitian yang muncul adalah:

1. Bagaimana mengintegrasikan *Efficient Transformer* (ET) untuk meningkatkan performa akurasi (diukur dengan metrik PSNR dan SSIM) ke dalam arsitektur *Enhanced Group Convolutional Neural Network* (EGCNN) dalam tugas *Single Image Super-Resolution*?
2. Seberapa besar peningkatan performa yang dicapai oleh EGCNN setelah integrasi ET dalam hal kualitas citra hasil super-resolusi dibandingkan dengan EGCNN tanpa integrasi ET dan model SISR terkini lainnya?
3. Seberapa besar peningkatan performa model pada berbagai skenario *upsampling* setelah integrasi ET ke dalam arsitektur EGCNN?

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Mengintegrasikan *Efficient Transformer* (ET) ke dalam arsitektur *Enhanced Group Convolutional Neural Network* (EGCNN) untuk meningkatkan performa akurasi dalam tugas *Single Image Super-Resolution* yang diukur menggunakan metrik PSNR dan SSIM.
2. Mengevaluasi sejauh mana peningkatan performa yang dicapai oleh EGCNN setelah integrasi ET dalam hal kualitas citra hasil super-resolusi dengan membandingkannya terhadap EGCNN tanpa integrasi ET dan model SISR terkini lainnya.
3. Menilai peningkatan performa model pada berbagai skenario *upsampling* setelah integrasi ET ke dalam arsitektur EGCNN untuk memahami efektivitas metode pada tingkat pembesaran yang berbeda.

1.4 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah:

1. Bagi peneliti, penelitian ini memberikan pemahaman yang lebih dalam tentang integrasi *Efficient Transformer* dalam EGCNN untuk SISR dan memperkuat kemampuan merancang model *deep learning* yang lebih efisien dan akurat. Selain itu, pendekatan baru ini menjadi kontribusi penting dalam penelitian SISR yang membuka peluang eksplorasi teknik *hybrid* CNN dan *Transformer* untuk tugas visi komputer lainnya.
2. Bagi masyarakat, penelitian ini mendukung pengembangan teknologi citra beresolusi tinggi yang lebih baik untuk aplikasi medis, pengawasan, fotografi, dan bidang lainnya sehingga masyarakat dapat mengakses konten visual yang lebih tajam.

1.5 Kontribusi Penelitian

Kontribusi yang diharapkan dari penelitian ini adalah:

1. Inovasi metodologi, mengusulkan metode *hybrid* dengan mengintegrasikan *Efficient Transformer* (ET) ke dalam *Enhanced Group Convolutional Neural Network* (EGCNN) yang belum pernah dilakukan sebelumnya. Hal ini dilakukan dengan merancang arsitektur

model yang mengintegrasikan ET setelah blok-blok EGCNN untuk menangkap hubungan jangka panjang antar piksel. Dengan demikian, model dapat memanfaatkan efisiensi komputasi EGCNN dalam mengekstraksi fitur lokal dan kemampuan ET dalam memahami konteks global sehingga berpotensi untuk meningkatkan kualitas rekonstruksi citra.

2. Peningkatan kinerja model SISR, menunjukkan bahwa integrasi ET dapat meningkatkan kualitas citra hasil super-resolusi dari EGCNN sehingga memberikan model alternatif yang lebih baik untuk tugas SISR. Hal ini dikarenakan ET mampu mengatasi keterbatasan EGCNN dalam menangkap hubungan jarak jauh yang memungkinkan model untuk merekonstruksi detail tekstur halus dan pola global dengan lebih baik. Evaluasi kinerja akan dilakukan melalui metrik kuantitatif menggunakan PSNR dan SSIM.

1.6 Batasan Penelitian

Penelitian ini memiliki beberapa batasan, yaitu:

1. Penelitian ini menggunakan dataset DF2K untuk data pelatihan yang merupakan gabungan dari dua dataset, yaitu DIV2K dan Flickr2K. Dataset ini berisi total 3.450 gambar beresolusi 2K, yang menghadirkan variasi yang luas dalam hal tekstur, detail, dan konten visual.
2. Performa model diuji menggunakan empat dataset standar, yaitu Set5, Set14, BSD100, dan Urban100. Dataset ini umum digunakan dalam penelitian SISR untuk mengevaluasi kualitas gambar hasil super-resolusi.
3. Penelitian mencakup skala pembesaran secara bertingkat supaya perubahan resolusi bisa diamati secara signifikan, adapun gradasi pembesaran dimulai dari $\times\frac{5}{4}$, $\times\frac{3}{2}$, $\times\frac{7}{4}$, $\times 2$, $\times 3$, $\times 4$.

BAB 2

KAJIAN PUSTAKA DAN DASAR TEORI

Bab kajian pustaka dan dasar teori menjelaskan pustaka yang dikaji dan dasar teori yang digunakan sebagai acuan dalam melakukan penelitian tesis. Kajian pustaka berisi hasil-hasil penelitian terdahulu yang berkaitan dengan penelitian tesis ini, sedangkan dasar teori berisi konsep-konsep yang berkaitan dengan penelitian tesis ini.

2.1 Kajian Pustaka

Penelitian dalam bidang *Single Image Super-Resolution* (SISR) telah mengalami perkembangan signifikan dengan adopsi metode berbasis *Deep Learning*, khususnya penggunaan *Convolutional Neural Networks* (CNN). CNN telah membuktikan kemampuannya dalam mengekstraksi fitur penting dari citra dan merekonstruksi detail yang hilang pada citra beresolusi rendah (Yang et al., 2019). Dong et al. memperkenalkan *Super-Resolution Convolutional Neural Network* (SRCNN), model CNN pertama untuk SISR, yang membuka jalan bagi penelitian lanjutan dalam bidang ini. SRCNN berhasil menunjukkan bahwa pendekatan *end-to-end* menggunakan CNN dapat secara efektif mempelajari pemetaan antara citra beresolusi rendah dan tinggi sehingga menghasilkan peningkatan kualitas yang signifikan dibandingkan metode konvensional (Dong et al., 2014). Setelah SRCNN, Kim et al. mengusulkan *Very Deep Super-Resolution* (VDSR), yang menggunakan jaringan yang lebih dalam dengan hingga 20 lapisan konvolusi untuk meningkatkan akurasi rekonstruksi citra. VDSR memperkenalkan teknik *residual learning* untuk mengatasi masalah *vanishing gradient* yang sering terjadi pada jaringan yang sangat dalam. Dengan menggunakan *residual learning*, VDSR dapat mencapai konvergensi yang lebih cepat dan kinerja yang lebih baik dibandingkan model sebelumnya (Kim et al., 2016).

Penelitian lain yang dilakukan oleh Zhang et al. dengan mengembangkan *Residual Dense Network* (RDN), yang memanfaatkan koneksi *residual density* untuk memfasilitasi aliran informasi dan meningkatkan kemampuan representasi fitur. RDN menggabungkan keunggulan *residual learning* dan *dense connections* untuk membentuk jalur informasi yang lebih kaya dan mendalam. Hal ini

memungkinkan model untuk mengekstraksi fitur dengan kompleksitas tinggi dan meningkatkan kualitas rekonstruksi citra secara signifikan. Selain itu, Zhang et al. memperkenalkan *Residual Channel Attention Network* (RCAN), yang menggunakan mekanisme *channel attention* untuk menyesuaikan bobot setiap saluran fitur. RCAN memanfaatkan informasi antar saluran untuk memperkuat fitur yang relevan dan menekan fitur yang kurang penting sehingga meningkatkan kemampuan model dalam menangkap informasi penting (Y. Zhang et al., 2018). Dengan arsitektur yang sangat dalam dan mekanisme perhatian, RCAN mencapai kinerja *state-of-the-art* pada saat itu yang menunjukkan kemampuan unggul dalam merekonstruksi detail-detail halus pada citra.

Untuk meningkatkan efisiensi komputasi, terutama pada perangkat dengan keterbatasan sumber daya, Tian et al. memperkenalkan *Enhanced Group Convolutional Neural Network* (EGCNN). EGCNN memanfaatkan *group convolution* yang pertama kali diperkenalkan dalam *AlexNet* untuk mengurangi jumlah parameter dan operasi komputasi tanpa mengorbankan kinerja model secara signifikan. Dalam *group convolution*, saluran input dan filter dibagi menjadi beberapa grup dan konvolusi dilakukan secara terpisah dalam setiap grup. Hal ini memungkinkan EGCNN untuk menjadi lebih ringan dan cepat sehingga cocok untuk aplikasi *real-time* atau pada perangkat dengan keterbatasan komputasi (Tian et al., 2022). Meskipun efektif, EGCNN memiliki keterbatasan dalam menangkap hubungan jarak jauh antar piksel karena sifat lokal dari operasi konvolusi. Operasi konvolusi standar hanya mempertimbangkan konteks lokal dalam *receptive field* yang terbatas, sehingga informasi global atau hubungan jangka panjang tidak ditangkap dengan baik (R. Zhang et al., 2022). Hal ini dapat membatasi kemampuan model dalam merekonstruksi detail tekstur yang kompleks atau pola global dalam citra.

Di sisi lain, *Transformer* yang diperkenalkan oleh Vaswani et al. dalam bidang pemrosesan bahasa alami telah mengubah paradigma dalam menangani data sekuensial. *Transformer* menggunakan mekanisme *self-attention* untuk menangkap hubungan jangka panjang antar elemen dalam urutan data, tanpa bergantung pada struktur urutan yang ketat seperti dalam RNN (Vaswani et al., 2017). Hal ini memungkinkan *Transformer* untuk melakukan komputasi secara paralel dan

menangkap konteks global secara efektif. Penggunaan *Transformer* dalam visi komputer mulai dieksplorasi dengan diperkenalkannya *Vision Transformer* (ViT) oleh Dosovitskiy et al. ViT mengadaptasi arsitektur *Transformer* untuk tugas klasifikasi citra dengan membagi citra menjadi patch kecil dan memperlakukan mereka sebagai token input untuk *Transformer*. ViT menunjukkan bahwa dengan dataset pelatihan yang besar, *Transformer* dapat bersaing dengan CNN dalam tugas visi komputer (Dosovitskiy et al., 2021). Namun, ViT memiliki kompleksitas komputasi yang tinggi, terutama untuk citra beresolusi tinggi, karena mekanisme *self-attention* memiliki kompleksitas kuadrat terhadap jumlah token input. Untuk mengatasi masalah kompleksitas ini, Lu et al. mengusulkan *Efficient Super-Resolution Transformer* (ESRT), yang mengintegrasikan *Efficient Transformer* (ET) ke dalam model SISR. ESRT berhasil meningkatkan kualitas rekonstruksi citra dengan efisiensi komputasi yang lebih baik dibandingkan *Transformer* konvensional. ET mengurangi kompleksitas komputasi melalui pengurangan dimensi dan mekanisme perhatian yang efisien, seperti *linear attention* atau *locality-sensitive hashing*. Dengan demikian, ET memungkinkan penerapan *Transformer* pada citra beresolusi tinggi dengan beban komputasi yang lebih rendah (Lu et al., 2022).

Adapun penelitian lain oleh Liang et al. memperkenalkan SwinIR, sebuah model restorasi citra inovatif yang memanfaatkan arsitektur *Swin Transformer* untuk berbagai tugas pemrosesan citra, seperti SISR, denoising, dan restorasi citra secara umum. *Swin Transformer* yang digunakan dalam SwinIR memiliki mekanisme perhatian unik bernama *shifted window*, yang membagi citra menjadi jendela-jendela kecil yang kemudian bergeser untuk memungkinkan pemodelan perhatian pada wilayah lokal dan global secara bergantian. Pergeseran jendela ini memungkinkan model untuk menangkap konteks spasial yang lebih luas tanpa perlu peningkatan ukuran pemrosesan yang signifikan, sehingga meningkatkan efisiensi komputasi. Pendekatan *shifted window* dalam *Swin Transformer* memperbaiki keterbatasan model-model sebelumnya yang cenderung hanya dapat menangkap informasi lokal atau global secara parsial. SwinIR berhasil mengatasi tantangan ini dengan memanfaatkan kombinasi pengolahan spasial lokal yang detail dan jangkauan perhatian global, yang pada akhirnya memberikan SwinIR

kemampuan untuk menghasilkan citra hasil restorasi dengan akurasi yang sangat tinggi (Liang et al., 2021). Berdasarkan uji kinerja, SwinIR menunjukkan keunggulan dalam mencapai *state-of-the-art* untuk tugas-tugas seperti SISR, denoising, dan lainnya, yang memperkuat perannya sebagai pendekatan yang sangat efektif dalam visi komputer, terutama dalam skenario di mana akurasi dan efisiensi sangat dibutuhkan.

Berikut adalah tabel yang berisi rangkuman poin-poin penting dari metode sebelumnya dan perbandingan dengan metode yang diusulkan:

Tabel 2.1 Perbandingan metode sebelumnya dengan metode usulan

Metode	Fitur utama	Keunggulan	Keterbatasan
SRCNN (Dong et al., 2014)	<ul style="list-style-type: none"> - Model CNN pertama untuk SISR - Pendekatan <i>end-to-end</i> dengan CNN dangkal 	<ul style="list-style-type: none"> - Memperkenalkan CNN dalam SISR - Peningkatan kualitas dibanding metode konvensional 	<ul style="list-style-type: none"> - Arsitektur dangkal - Keterbatasan dalam menangkap fitur kompleks - Kinerja terbatas pada detail halus
VDSR (Kim et al., 2016)	<ul style="list-style-type: none"> - Jaringan sangat dalam (20 lapisan) - Teknik <i>residual learning</i> 	<ul style="list-style-type: none"> - Kinerja rekonstruksi lebih baik - Konvergensi lebih cepat dengan <i>residual learning</i> 	<ul style="list-style-type: none"> - Beban komputasi lebih tinggi - Risiko vanishing gradient tanpa <i>residual learning</i>
RDN (Y. Zhang et al., 2018)	<ul style="list-style-type: none"> - <i>Residual dense connections</i> - Jalur informasi yang kaya dan mendalam 	<ul style="list-style-type: none"> - Ekstraksi fitur kompleks - Peningkatan kualitas rekonstruksi signifikan 	<ul style="list-style-type: none"> - Kompleksitas komputasi tinggi - Jumlah parameter besar
EGCNN (Tian et al., 2022)	<ul style="list-style-type: none"> - <i>Group convolution</i> untuk efisiensi - Mengurangi jumlah parameter dan operasi komputasi 	<ul style="list-style-type: none"> - Efisien dalam komputasi - Cocok untuk aplikasi <i>real-time</i> dan perangkat terbatas 	<ul style="list-style-type: none"> - Keterbatasan dalam menangkap hubungan jarak jauh

			- Kurang optimal dalam merekonstruksi detail tekstur halus
ESRT (Lu et al., 2022)	- Integrasi <i>Efficient Transformer</i> ke dalam model SISR - Berbasis arsitektur RCAN	- Meningkatkan kualitas rekonstruksi citra - Efisiensi komputasi lebih baik dibanding <i>Transformer</i> konvensional	- Kompleksitas masih tinggi untuk perangkat terbatas - Peningkatan efisiensi belum maksimal
SwinIR (Liang et al., 2021)	- Menggunakan <i>Swin Transformer</i> dengan <i>shifted window attention</i> - Memodelkan perhatian pada wilayah lokal dan global	- Menangkap informasi spasial luas secara efisien - Kinerja unggul dalam SISR dan restorasi citra lainnya	- Kompleksitas arsitektur tinggi - Beban komputasi signifikan
Metode Diusulkan (EGCNN + ET)	- Integrasi <i>Efficient Transformer</i> ke dalam EGCNN - Menggabungkan <i>group convolution</i> dan <i>efficient self-attention</i>	- Memanfaatkan efisiensi komputasi EGCNN - Menangkap hubungan jarak jauh dengan ET - Potensi peningkatan kualitas rekonstruksi dan efisiensi	- Belum dieksplorasi sebelumnya - Efektivitas dan kinerja perlu dibuktikan melalui penelitian

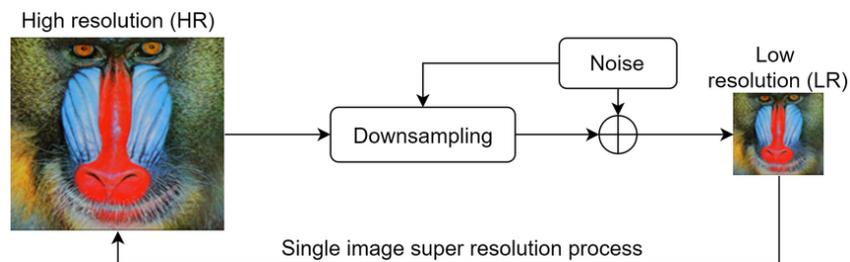
2.2 Dasar Teori

Pada sub bab ini, akan dibahas beberapa teori yang digunakan untuk mendukung penelitian ini. Teori tersebut antara lain: *Single Image Super-Resolution* (SISR), *Convolutional Neural Networks* (CNN), *group convolution*, *Enhanced Group Convolutional Neural Network* (EGCNN), *Transformer* dan Mekanisme *Self-Attention*, *Efficient Transformer* (ET), serta Metrik Evaluasi Kinerja Model SISR.

2.2.1 *Single Image Super-Resolution (SISR)*

Single Image Super-Resolution (SISR) adalah proses rekonstruksi gambar beresolusi tinggi (*High-Resolution, HR*) dari satu gambar beresolusi rendah (*Low-Resolution, LR*) (Bashir et al., 2021). Teknik ini berfungsi untuk memperbaiki atau meningkatkan kualitas gambar dengan mengisi detail yang hilang dan memiliki aplikasi luas di berbagai bidang seperti pengawasan, fotografi digital, pemrosesan citra medis, serta analisis citra satelit (Ye et al., 2023) (Le-Tien et al., 2020). Dalam pengawasan misalnya, SISR sangat penting untuk memperjelas citra rekaman dari kamera keamanan yang memiliki resolusi terbatas sehingga objek atau individu dapat diidentifikasi dengan lebih akurat. Demikian pula dalam fotografi digital, SISR berguna untuk meningkatkan kualitas gambar yang diambil dengan perangkat resolusi rendah sehingga detail yang halus seperti tekstur wajah atau latar belakang yang kompleks dapat dipertahankan dengan lebih baik (Zamfir et al., 2023).

Salah satu tantangan utama dalam SISR adalah hilangnya informasi penting pada gambar LR selama proses *downsampling* seperti tekstur halus, kontur, serta detail visual lainnya yang berkontribusi terhadap persepsi kualitas tinggi. Ketika sebuah gambar dikonversi ke resolusi yang lebih rendah maka informasi ini sering kali hilang secara permanen yang menyebabkan kesulitan dalam proses rekonstruksi (Xin et al., 2022), seperti yang diilustrasikan pada Gambar 2.1. Untuk menangani masalah ini maka model SISR harus mampu melakukan estimasi atau prediksi informasi yang hilang dengan akurasi tinggi. Strategi ini mengandalkan berbagai pendekatan berbasis *deep learning* seperti *Convolutional Neural Networks (CNNs)* dan metode berbasis Transformasi (Y. Zhang et al., 2018). CNNs khususnya telah menunjukkan kinerja yang mengesankan dalam mempelajari pola tekstur dan tepi yang kompleks meskipun memerlukan data pelatihan yang ekstensif untuk mencapai hasil yang diinginkan (Dong et al., 2014).



Gambar 2.1 Proses SISR (Viriyavisuthisakul et al., 2022)

Sebagai contoh, pendekatan berbasis CNN memungkinkan jaringan untuk mengekstrak fitur multiskala dari gambar LR yang kemudian dikombinasikan untuk merekonstruksi gambar HR yang kaya akan detail. Namun, model yang lebih kompleks ini sering kali membutuhkan daya komputasi besar yang menjadi kendala dalam aplikasi *real-time* seperti dalam pengawasan dan sistem otomasi (Yu et al., 2021). Peningkatan kualitas citra tidak dapat dipisahkan dari tantangan untuk menghindari artefak visual seperti *oversharpening*, *aliasing*, atau distorsi yang dapat merusak kualitas gambar hasil (Lim et al., 2017). Model SISR juga harus memastikan bahwa gambar yang direkonstruksi terlihat alami dan realistis dengan mempertahankan keseimbangan antara ketajaman detail dan kelancaran struktur (Cruz et al., 2018).

Selain masalah rekonstruksi detail yang halus, keberagaman jenis citra yang harus ditangani oleh model SISR juga menambah kerumitan proses. Citra wajah misalnya, memiliki pola tekstur yang sangat berbeda dibandingkan dengan citra alam atau objek buatan manusia seperti kendaraan atau bangunan. Oleh karena itu, pengembangan model SISR yang serbaguna dan mampu beradaptasi dengan berbagai jenis gambar adalah aspek penelitian yang sangat aktif dan menantang (Nan et al., 2022). Keberhasilan model dalam mengatasi masalah ini sangat bergantung pada kemampuan untuk menangkap fitur spesifik dari setiap kategori gambar serta generalisasi yang baik di berbagai domain citra (Muhammad et al., 2021).

Untuk mengatasi tantangan efisiensi komputasi, beberapa pendekatan seperti model *lightweight* dan optimasi berbasis GPU telah diusulkan. Tujuannya adalah untuk memastikan bahwa model dapat berjalan secara efisien pada perangkat dengan daya terbatas, seperti *smartphone* atau kamera pengawasan. Misalnya, metode seperti *MobileNet* dan teknik kuantisasi jaringan telah diterapkan untuk mempercepat proses inferensi tanpa mengorbankan kualitas hasil secara signifikan (H. Huang et al., 2023). Hal ini menjadi sangat relevan untuk aplikasi yang memerlukan pemrosesan *real-time*, di mana *trade-off* antara akurasi dan kecepatan menjadi faktor utama yang dipertimbangkan (Gankhuyag et al., 2023).

Secara keseluruhan, berbagai pendekatan penelitian SISR telah diusulkan mulai dari metode berbasis interpolasi klasik hingga algoritma *deep learning* yang

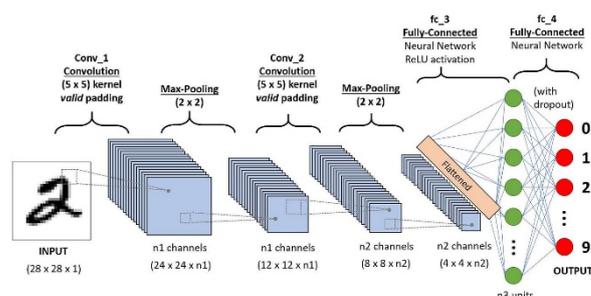
kompleks. Model-model berbasis *deep learning* yang paling canggih saat ini mampu mempelajari hubungan spasial yang rumit dalam gambar tetapi masih diperlukan inovasi untuk mengatasi batasan dalam hal memori dan komputasi (Wang et al., 2021). Oleh karena itu, penelitian bidang SISR terus berkembang dengan terus mencari solusi yang lebih baik untuk meningkatkan kualitas visual dan efisiensi yang tinggi.

2.2.2 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) adalah jenis jaringan saraf tiruan yang dirancang khusus untuk memproses data dengan struktur grid dua dimensi seperti pada citra atau video (Gu et al., 2018). CNN terkenal karena keunggulannya dalam mengenali pola visual dan mengklasifikasikan gambar secara efisien. Keberhasilannya dalam tugas-tugas terkait pengolahan citra terletak pada penggunaan lapisan konvolusi yang dirancang untuk mengekstraksi fitur-fitur penting secara hierarkis. Melalui penerapan filter yang terlatih, CNN dapat mendeteksi pola seperti tepi, sudut, dan tekstur sederhana pada lapisan awal yang kemudian menggabungkan pola-pola tersebut untuk membentuk fitur yang lebih kompleks pada lapisan-lapisan yang lebih dalam (Gu et al., 2018).

Lapisan Konvolusi adalah elemen utama dari CNN di mana filter (juga disebut *kernel*) diaplikasikan pada input untuk menghasilkan peta fitur. Setiap filter dalam lapisan konvolusi belajar mendeteksi pola tertentu dari citra input. Misalnya, filter pada lapisan awal mungkin fokus pada mendeteksi tepi horizontal atau vertikal, sementara filter pada lapisan lebih dalam dapat mengidentifikasi bentuk dan tekstur yang lebih kompleks (Albawi et al., 2017). Operasi konvolusi ditandai oleh dua sifat penting: parameter *sharing* dan *sparsity of connections*. Parameter *sharing* berarti bahwa bobot dari satu filter diterapkan di seluruh citra untuk mengurangi jumlah parameter yang perlu dipelajari dan memungkinkan efisiensi komputasi yang lebih baik. *Sparsity of connections* mengacu pada kenyataan bahwa setiap *neuron* dalam lapisan konvolusi hanya terhubung ke sebagian kecil dari *neuron* di lapisan sebelumnya yang dikenal sebagai *receptive field*. Ini memungkinkan CNN untuk mengenali pola lokal dalam citra yang penting untuk tugas-tugas seperti deteksi objek atau segmentasi gambar (Goodfellow, 2015).

Lapisan *Pooling* seperti *max pooling* dan *average pooling* digunakan dalam CNN untuk mengurangi dimensi spasial dari peta fitur yang berguna untuk mempercepat komputasi, mengurangi risiko *overfitting*, dan meningkatkan generalisasi model. *Max pooling* misalnya, memilih nilai maksimum dalam setiap *patch* kecil dari peta fitur sehingga mempertahankan informasi penting sambil mengurangi kompleksitas data (Brownlee, 2019). Pooling ini membantu jaringan menjadi lebih kuat terhadap perubahan posisi objek dalam citra yang dikenal sebagai invariansi translasi.



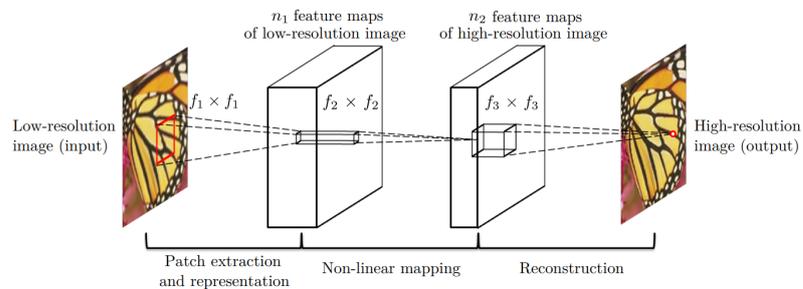
Gambar 2.2 Komponen lapisan dalam CNN (Szegedy et al., 2016)

Selain itu, CNN biasanya dilengkapi dengan lapisan *fully connected* (FC) yang berfungsi untuk mengintegrasikan semua fitur yang diekstraksi oleh lapisan konvolusi dan *pooling*. Pada bagian akhir CNN, fitur-fitur ini digunakan untuk membuat keputusan seperti klasifikasi atau regresi. Lapisan *fully connected* bertindak sebagai “penyusun keputusan” dengan cara mengubah representasi fitur menjadi hasil akhir yang sesuai dengan tugas yang sedang dihadapi seperti mengenali objek dalam citra atau memperkirakan nilai tertentu (Hirata & Takahashi, 2023). Keseluruhan lapisan CNN diilustrasikan pada Gambar 2.2.

2.2.3 Pendekatan Berbasis CNN dalam SISR

Pendekatan berbasis *Convolutional Neural Networks* (CNN) dalam *Single Image Super-Resolution* (SISR) telah merevolusi cara rekonstruksi gambar beresolusi tinggi dari gambar beresolusi rendah. Penelitian ini dimulai dengan diperkenalkannya *Super-Resolution Convolutional Neural Network* (SRCNN) oleh Dong et al., yang menunjukkan bahwa CNN mampu secara efektif mempelajari pemetaan *end-to-end* antara gambar beresolusi rendah (*Low-Resolution*, LR) dan gambar beresolusi tinggi (*High-Resolution*, HR) (Dong et al., 2014). Model

SRCNN menggunakan arsitektur sederhana dengan beberapa lapisan konvolusi untuk mengekstraksi fitur, memetakan ke ruang gambar HR, dan kemudian merekonstruksi gambar yang diperhalus seperti yang ditunjukkan pada Gambar 2.3. Keberhasilan SRCNN menginspirasi penelitian selanjutnya untuk memperdalam arsitektur jaringan dan meningkatkan kapasitas model dalam menangkap fitur yang lebih kompleks sekaligus menjaga efisiensi komputasi.



Gambar 2.3 Arsitektur SRCNN (Dong et al., 2014)

Dalam upaya untuk meningkatkan kualitas rekonstruksi, Kim et al. memperkenalkan *Very Deep Super-Resolution* (VDSR) yang menggunakan jaringan CNN lebih dalam hingga 20 lapisan konvolusi untuk menangkap representasi fitur yang lebih kaya. Salah satu inovasi utama VDSR adalah penerapan *residual learning* yang membantu model mengatasi masalah *vanishing gradient* yang kerap terjadi pada jaringan yang sangat dalam (Kim et al., 2016). *Residual learning* memungkinkan model untuk mempelajari perubahan kecil yang diperlukan untuk meningkatkan gambar LR menjadi HR, sehingga mempercepat proses konvergensi dan meningkatkan akurasi. Teknik ini membuka jalan bagi arsitektur yang lebih dalam dan lebih canggih dalam domain SISR.

Penelitian berikutnya, Lim et al. mengembangkan *Enhanced Deep Super-Resolution Network* (EDSR), yang menghilangkan lapisan normalisasi batch (*batch normalization*) untuk memaksimalkan kapasitas jaringan dalam merepresentasikan fitur. EDSR terbukti mampu menangkap informasi detail dengan lebih efisien dan menghasilkan gambar yang jauh lebih tajam dan realistis dibandingkan model sebelumnya (Lim et al., 2017). Penghapusan *batch normalization* meningkatkan fleksibilitas jaringan dalam mempelajari distribusi data yang bervariasi meskipun ini membuat model lebih sulit untuk dilatih secara stabil. Untuk mengatasi

tantangan ini, EDSR menggunakan blok residu yang disesuaikan untuk memperkuat aliran informasi antar-lapisan.

Selanjutnya Zhang et al. memperkenalkan *Residual Dense Network* (RDN), yang memanfaatkan koneksi langsung antar-lapisan untuk memperkuat aliran informasi dan gradien selama pelatihan. RDN memanfaatkan *dense connections* yang sebelumnya sukses digunakan dalam arsitektur DenseNet tetapi dengan modifikasi yang memungkinkan model untuk menangkap fitur multi-level dengan lebih efisien (Y. Zhang et al., 2021). Pendekatan ini membantu dalam mempertahankan informasi penting dari gambar LR sambil memperbaiki detail gambar HR yang sulit seperti tekstur halus dan kontur. RDN menunjukkan bahwa koneksi densitas dapat secara signifikan meningkatkan performa SISR terutama pada citra dengan tekstur yang rumit dan beragam.

Selain itu, perkembangan baru dalam *attention mechanisms* telah diadopsi dalam arsitektur CNN untuk meningkatkan kemampuan model dalam fokus pada fitur penting dalam gambar. Salah satu contohnya adalah *Channel Attention Mechanism* yang digunakan dalam model seperti RCAN (*Residual Channel Attention Network*) untuk menyoroti fitur yang relevan dan menekan fitur yang tidak penting. RCAN terbukti efektif dalam meningkatkan kualitas citra HR dengan mengalokasikan lebih banyak perhatian pada area gambar yang memerlukan detail yang lebih halus (Y. Zhang et al., 2018).

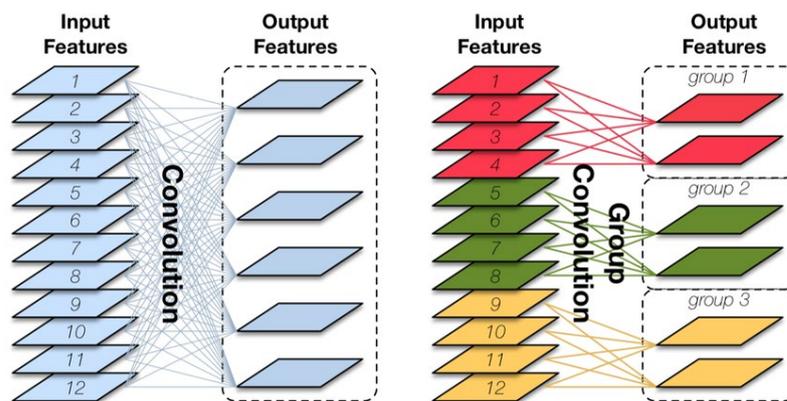
Pendekatan CNN dalam SISR terus berkembang seiring dengan pengembangan arsitektur yang lebih efisien dan efektif. Misalnya, LapSRN (*Laplacian Pyramid Super-Resolution Network*) mengadopsi strategi hierarki untuk merekonstruksi gambar secara bertahap, yang mengurangi beban komputasi sambil mempertahankan kualitas gambar (Lai et al., 2019). MemNet, model lain yang berbasis *Memory Network* dengan kemampuan mempertahankan representasi fitur yang telah dipelajari sebelumnya sehingga memberikan kemampuan model untuk mengingat dan memanfaatkan kembali informasi penting selama proses rekonstruksi (Tai et al., 2017).

2.2.4 Group Convolution

Group convolution adalah varian dari operasi konvolusi standar yang bertujuan untuk meningkatkan efisiensi komputasi dan mengurangi risiko

overfitting terutama dalam jaringan saraf tiruan yang sangat dalam dan kompleks. Konsep ini diperkenalkan untuk mengatasi kendala yang muncul ketika model *deep learning* diperluas dengan menambah jumlah filter dan lapisan konvolusi. Dalam operasi konvolusi biasa, setiap filter terhubung ke semua saluran input yang menyebabkan peningkatan eksponensial jumlah parameter dan operasi komputasi seiring bertambahnya kedalaman jaringan. *Group convolution* mengatasi masalah ini dengan membagi saluran input dan filter ke dalam beberapa grup independen yang memungkinkan konvolusi dilakukan secara terpisah dalam setiap grup (Lee et al., 2022).

Sebagai contoh (lihat Gambar 2.4), jika ada 12 saluran input dan 12 filter yang dibagi menjadi 3 grup maka setiap grup hanya akan memiliki 2 saluran input dan 2 filter. Ini secara signifikan mengurangi jumlah parameter yang perlu dipelajari karena setiap filter hanya terkait dengan subset tertentu dari saluran input dan bukan dari keseluruhan input. Dengan demikian, *group convolution* membantu mengurangi kompleksitas komputasi tanpa mengorbankan kemampuan jaringan untuk mempelajari fitur yang bermakna. Sebagai perbandingan pendekatan ini jauh lebih hemat sumber daya dibandingkan konvolusi penuh terutama untuk arsitektur yang sangat dalam seperti ResNet atau DenseNet yang memiliki ratusan lapisan (C. Zhang et al., 2021).



Gambar 2.4 Operasi *group convolution* (G. Huang et al., 2018)

Keuntungan utama lain dari *group convolution* adalah peningkatan efisiensi dalam hal memori dan waktu komputasi yang membuatnya sangat berguna dalam aplikasi yang membutuhkan kecepatan tinggi dan sumber daya terbatas seperti pada perangkat *mobile* dan sistem tertanam. Teknik ini pertama kali digunakan secara

populer dalam AlexNet, sebuah jaringan yang memperkenalkan pendekatan ini untuk memecah masalah komputasi yang sangat besar dengan memungkinkan pelatihan pada dua unit *GPU* secara paralel (Lee et al., 2022). AlexNet menunjukkan bahwa menggunakan *group convolution* tidak hanya meningkatkan efisiensi pelatihan tetapi juga membantu dalam menghindari *overfitting* dengan memaksa jaringan untuk belajar fitur yang lebih terdistribusi.

Seiring perkembangan arsitektur *deep learning*, *group convolution* diadopsi dalam model-model canggih lainnya seperti ResNeXt dan MobileNet. ResNeXt, yang diperkenalkan oleh Xie et al., menggunakan strategi ini untuk membangun jaringan yang lebih modular dan fleksibel. Arsitektur ini memperkenalkan ide “*cardinality*”, yaitu jumlah grup dalam *group convolution* yang terbukti lebih efektif dalam meningkatkan performa model dibandingkan sekadar memperdalam atau memperlebar jaringan (Xie et al., 2016). Dengan meningkatkan *cardinality*, ResNeXt mampu mempelajari fitur yang lebih beragam dan meningkatkan generalisasi model tanpa menambah jumlah parameter secara signifikan.

Selain itu, MobileNet memanfaatkan varian dari *group convolution* yang disebut *depthwise separable convolution* yang membagi operasi konvolusi menjadi dua tahap: *depthwise convolution* dan *pointwise convolution*. *Depthwise convolution* menerapkan satu filter per saluran input sementara *pointwise convolution* menggabungkan output dari *depthwise convolution* menggunakan konvolusi 1x1. Kombinasi ini dapat mengurangi jumlah operasi secara drastis dan membuat MobileNet sangat efisien untuk digunakan pada perangkat dengan daya komputasi terbatas seperti *smartphone* dan perangkat *Internet of Things* (Hwang et al., 2023).

Group convolution tidak hanya membantu dalam efisiensi komputasi tetapi juga memungkinkan jaringan untuk belajar fitur yang lebih beragam dan spesifik. Karena setiap grup konvolusi fokus pada subset yang berbeda dari data serta model yang dihasilkan dapat menangkap berbagai aspek fitur visual. Pendekatan ini juga membantu dalam regularisasi dan dapat mengurangi kemungkinan model menjadi terlalu cocok (*overfit*) dengan data pelatihan yang telah menjadi masalah umum dalam *deep learning* dengan dataset terbatas (C. Zhang et al., 2021).

2.2.5 *Enhanced Group Convolutional Neural Network (EGCNN)*

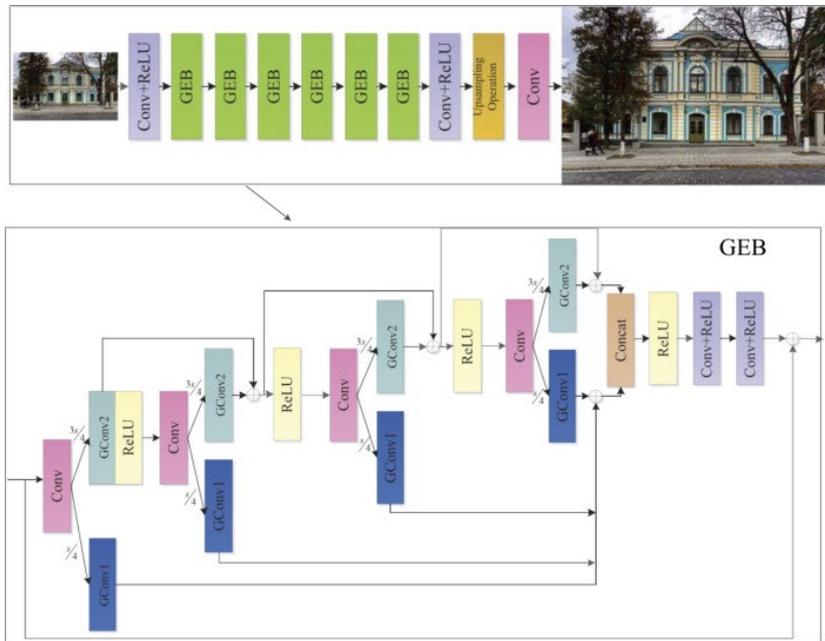
Enhanced Group Convolutional Neural Network (EGCNN) adalah arsitektur CNN yang dirancang untuk meningkatkan efisiensi komputasi dalam tugas *Single Image Super-Resolution (SISR)* dengan memanfaatkan keunggulan *group convolution*. EGCNN bertujuan untuk mengurangi jumlah parameter dan beban komputasi secara signifikan tanpa mengorbankan performa model dalam mengekstraksi fitur-fitur penting dari citra. Hal ini memungkinkan EGCNN untuk menghasilkan citra beresolusi tinggi (HR) yang berkualitas tinggi dengan efisiensi yang lebih baik dibandingkan dengan CNN konvensional.

Group convolution menjadi fondasi dalam arsitektur EGCNN karena kemampuannya dalam membagi saluran input dan filter ke dalam beberapa grup independen. Proses ini memungkinkan konvolusi dilakukan secara terpisah dalam setiap grup sehingga mengurangi jumlah operasi dan parameter yang harus dipelajari (Lee et al., 2022). Dengan mengurangi kompleksitas model, EGCNN dapat diimplementasikan pada perangkat dengan daya komputasi terbatas seperti *smartphone* atau kamera pengawas sambil tetap memberikan hasil yang memuaskan dalam meningkatkan resolusi gambar.

Arsitektur EGCNN menggabungkan mekanisme *residual learning* yang dirancang untuk mengatasi masalah *vanishing gradient* yang sering terjadi pada jaringan yang sangat dalam. Mekanisme residual menyediakan *shortcut connections* yang memungkinkan gradien mengalir langsung ke lapisan-lapisan awal selama pelatihan. Hal ini mempercepat proses konvergensi dan meningkatkan stabilitas jaringan (Y. Zhang et al., 2021). *Residual learning* membantu EGCNN dalam mempertahankan informasi penting dari input awal, sehingga memungkinkan model untuk merekonstruksi citra HR yang lebih akurat dengan detail yang lebih terjaga.

Proses kerja EGCNN seperti yang diilustrasikan pada Gambar 2.5 dimulai dengan menerima citra beresolusi rendah (LR) sebagai input. Citra ini melewati beberapa lapisan *group convolution* untuk mengekstraksi fitur-fitur penting seperti tepi dan tekstur. Setiap lapisan *group convolution* memproses subset data input yang memungkinkan model untuk belajar berbagai fitur lokal secara efisien. Setelah fitur-fitur ini diekstraksi kemudian model EGCNN menggunakan lapisan

upsampling untuk meningkatkan resolusi citra ke tingkat yang diinginkan. Teknik *upsampling* ini membantu menghasilkan citra HR yang memiliki kualitas lebih tinggi dibandingkan dengan metode interpolasi tradisional seperti *bilinear* atau *bicubic interpolation* (Dong et al., 2014).



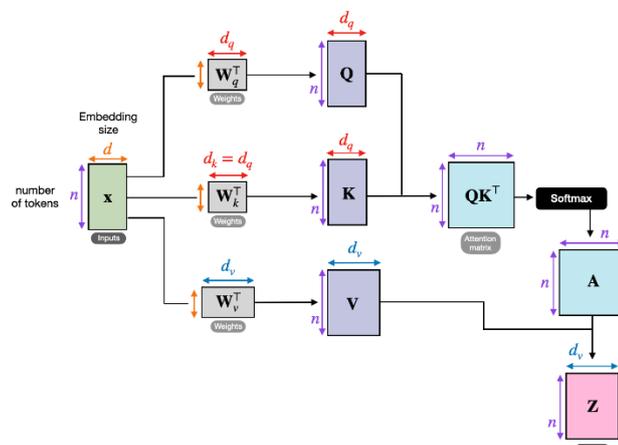
Gambar 2.5 Arsitektur EGCNN (Tian et al., 2022)

Namun, meskipun EGCNN sangat efisien dalam mengekstraksi fitur lokal model ini memiliki keterbatasan dalam menangkap hubungan jarak jauh antar piksel. Hal ini disebabkan oleh sifat lokal dari operasi konvolusi yang hanya mempertimbangkan informasi dalam *receptive field* yang terbatas. Akibatnya, EGCNN mungkin kurang optimal dalam merekonstruksi detail tekstur halus dan pola global yang ada dalam citra HR. Untuk mengatasi kelemahan ini penelitian lebih lanjut sedang dilakukan termasuk integrasi mekanisme seperti *self-attention* yang dapat memperhitungkan hubungan jarak jauh dan meningkatkan kemampuan model dalam memahami struktur global (Wang et al., 2021).

2.2.6 Transformer dan Mekanisme Self-Attention

Transformer adalah arsitektur model yang diperkenalkan oleh Vaswani et al. pada tahun 2017, yang telah merevolusi berbagai tugas dalam pemrosesan bahasa alami (*Natural Language Processing*, NLP) dan diadopsi secara luas dalam berbagai aplikasi lain termasuk pengolahan citra dan analisis data sekuensial. Salah satu inovasi utama dalam *Transformer* adalah penggunaan mekanisme *self-*

attention, yang memungkinkan model untuk memperhitungkan hubungan antara setiap elemen dalam input secara langsung tanpa bergantung pada urutan data secara eksplisit (Vaswani et al., 2017). Kemampuan ini membuat *Transformer* sangat efektif dalam menangkap hubungan jangka panjang dalam data, yang sangat penting untuk memahami konteks global dalam tugas-tugas kompleks.



Gambar 2.6 Mekanisme *self-attention* (L. Zhang, 2018)

Mekanisme *self-attention* berfungsi dengan mengubah setiap elemen input menjadi tiga representasi berbeda: *Query* (Q), *Key* (K), dan *Value* (V), lihat Gambar 2.6. *Query* mewakili elemen input yang sedang dipertimbangkan, *Key* merepresentasikan elemen lain dalam input yang mungkin relevan, dan *Value* adalah informasi yang akan diambil jika elemen tersebut dinyatakan relevan. Relevansi antar elemen dihitung dengan mengambil *dot product* antara *Query* dan *Key* yang mengukur kesamaan atau perhatian antara elemen-elemen tersebut. Hasilnya kemudian dinormalisasi menggunakan fungsi *softmax* untuk menghasilkan bobot perhatian dengan menentukan seberapa besar perhatian yang diberikan pada elemen lain. Bobot perhatian ini digunakan untuk mengombinasikan *Value* dengan menghasilkan output yang memperhitungkan hubungan konteks secara global (Vaswani et al., 2017).

Keunggulan utama *Transformer* dibandingkan dengan model-model sekuensial tradisional seperti *Recurrent Neural Networks* (RNN) atau *Long Short-Term Memory* (LSTM) adalah kemampuannya untuk melakukan *parallelization*. Model-model berbasis RNN memproses data secara berurutan yang menghambat efisiensi komputasi terutama untuk data sekuensial yang sangat panjang.

Sebaliknya, *Transformer* tidak bergantung pada urutan sehingga memungkinkan percepatan komputasi yang signifikan. Fitur ini sangat menguntungkan untuk pelatihan model pada data besar karena mengurangi waktu pelatihan secara drastis dan meningkatkan efisiensi penggunaan sumber daya komputasi (Devlin et al., 2019).

Dengan mekanisme *self-attention*, *Transformer* mampu menangkap hubungan jangka panjang dalam data dengan cara yang lebih efektif daripada RNN atau LSTM. Misalnya, dalam tugas penerjemahan bahasa, *Transformer* dapat dengan mudah menghubungkan kata-kata yang berhubungan dalam kalimat panjang meskipun kata-kata tersebut dipisahkan oleh elemen lain. Hal ini memberikan keunggulan dalam memahami konteks global yang sangat penting untuk menghasilkan output yang lebih akurat dan kohesif. Keunggulan ini juga diterapkan dalam bidang-bidang seperti pengolahan citra di mana *Transformer* telah digunakan untuk meningkatkan pemahaman citra global dengan mempertimbangkan hubungan antar piksel yang jauh (Dosovitskiy et al., 2021).

Selain itu, *Transformers* telah diadopsi dalam tugas *Single Image Super-Resolution* (SISR), di mana pemahaman konteks global sangat penting untuk merekonstruksi citra beresolusi tinggi dengan akurasi yang tinggi. Mekanisme *self-attention* dalam *Transformer* membantu model menangkap hubungan antara piksel yang berjauhan dan sering kali terlewat oleh arsitektur CNN yang hanya mempertimbangkan *receptive field* dengan terbatas. Dalam arsitektur SISR berbasis *Transformer*, pendekatan ini dapat digunakan untuk memperbaiki detail tekstur halus dan mengurangi artefak yang muncul selama proses peningkatan resolusi (Sun et al., 2022).

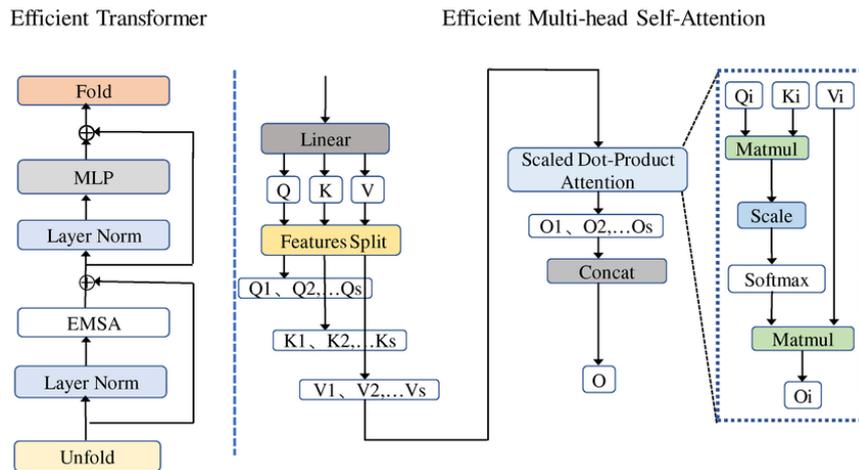
Transformers juga telah mengalami berbagai modifikasi untuk meningkatkan efisiensi dan performanya dalam tugas-tugas visual. Misalnya, *Vision Transformers* (ViT), yang diperkenalkan oleh Dosovitskiy et al., memanfaatkan mekanisme *self-attention* untuk mengolah citra dalam bentuk *patch* dua dimensi yang memberikan hasil kompetitif dibandingkan dengan model CNN tradisional (Dosovitskiy et al., 2021). ViT membagi citra menjadi blok-blok kecil (*patches*) dan memprosesnya seperti token dalam NLP yang memungkinkan model untuk menangkap hubungan global antar blok-blok tersebut. Meskipun ViT

memerlukan data pelatihan yang besar, model ini telah menunjukkan kinerja yang luar biasa dalam berbagai tugas visi computer seperti klasifikasi citra, deteksi objek, dan segmentasi. Mekanisme *self-attention* juga telah dimodifikasi menjadi *multi-head attention* yang memungkinkan *Transformer* untuk mempelajari berbagai aspek hubungan antar elemen dalam data secara bersamaan. *Multi-head attention* memecah representasi *Query*, *Key*, dan *Value* menjadi beberapa subruang yang di mana setiap subruang dapat fokus pada fitur yang berbeda. Pendekatan ini memperkaya kapasitas model dalam memahami data kompleks dan meningkatkan akurasi prediksi dalam berbagai aplikasi (Vaswani et al., 2017).

Dengan keunggulan-keunggulan ini, *Transformer* telah menjadi arsitektur yang sangat fleksibel dan kuat serta memberikan kontribusi signifikan dalam banyak domain termasuk NLP, pengolahan citra, dan pengolahan data sekuensial. Meskipun demikian, *Transformer* memiliki tantangan dalam hal kebutuhan komputasi yang tinggi sehingga mendorong pengembangan model-model lain yang lebih ringan dan efisien seperti *Efficient Transformers* dan *Performer* yang menggunakan mekanisme *attention* yang lebih hemat sumber daya (Choromanski et al., 2020).

2.2.7 Efficient Transformer (ET)

Efficient Transformer (ET) adalah varian dari arsitektur *Transformer* yang dirancang untuk mengatasi masalah kompleksitas komputasi tinggi yang terkait dengan mekanisme *self-attention*. Meskipun *Transformer* memiliki keunggulan dalam menangkap hubungan jangka panjang dalam data sekuensial dan citra, kompleksitas komputasinya meningkat secara kuadrat terhadap panjang input yaitu $O(n^2)$, di mana n adalah jumlah elemen dalam input. Hal ini menjadi kendala signifikan ketika *Transformer* diterapkan pada data dengan resolusi tinggi seperti gambar atau sekuens panjang dalam tugas pemrosesan bahasa alami (Vaswani et al., 2017). Untuk menangani tantangan ini *Efficient Transformer* (ET) diperkenalkan sebagai pendekatan yang lebih hemat sumber daya serta memungkinkan pemrosesan data berskala besar dengan efisiensi yang jauh lebih baik.



Gambar 2.7 Arsitektur *Efficient Transformer* (Cai et al., 2023)

Masalah utama dalam *Transformer* tradisional adalah ketidakmampuannya untuk menangani input yang sangat besar secara efisien. Dalam konteks data citra, setiap piksel dalam gambar dianggap sebagai elemen input yang menyebabkan peningkatan eksponensial dalam beban komputasi dan kebutuhan memori. ET menawarkan solusi untuk mengurangi beban ini dengan berbagai teknik optimasi seperti pengurangan dimensi dan mekanisme *self-attention* yang lebih efisien (Tay et al., 2022), seperti yang diilustrasikan pada Gambar 2.7. Salah satu teknik yang digunakan dalam ET adalah mereduksi dimensi representasi input sebelum diterapkan ke mekanisme *self-attention* yang secara langsung mengurangi jumlah elemen yang diproses. Dengan demikian, ET dapat memproses data citra dengan lebih efisien tanpa mengorbankan akurasi dan kemampuan model dalam memahami konteks global.

Chunked attention adalah salah satu pendekatan inovatif yang digunakan dalam ET untuk mengatasi masalah skalabilitas. Dalam *chunked attention*, input dibagi menjadi beberapa potongan atau *chunk* yang lebih kecil dan perhatian hanya dihitung dalam *chunk* tersebut atau di antara *chunk* tertentu. Teknik ini tidak hanya mengurangi jumlah operasi yang harus dilakukan tetapi juga memungkinkan pemrosesan paralel sehingga mempercepat pelatihan dan inferensi. Dengan memfokuskan perhatian dalam lingkup *chunk* yang lebih kecil, ET dapat mengelola hubungan lokal dengan lebih efisien sambil tetap mempertahankan informasi penting dari hubungan global (Tay et al., 2022).

Teknik lain yang digunakan dalam *Efficient Transformer* termasuk *sparse attention* dan *low-rank factorization* yang dirancang untuk mengurangi jumlah interaksi yang diproses dalam mekanisme *self-attention*. *Sparse attention* adalah pendekatan di mana perhatian dihitung hanya untuk elemen-elemen yang paling relevan sehingga meminimalkan interaksi yang tidak penting. Metode ini sangat berguna untuk data dengan pola sparsitas tinggi seperti citra atau teks yang memiliki ketergantungan jauh antar elemen. Di sisi lain, *Low-rank factorization* mengurangi dimensi *tensor* yang digunakan dalam perhitungan *attention* sehingga dapat menghasilkan pengurangan signifikan dalam penggunaan memori dan waktu komputasi (Khalitov et al., 2022).

Dengan teknik-teknik ini, ET mampu menurunkan kompleksitas komputasi dari $O(n^2)$ menjadi $O(n \log n)$ atau bahkan $O(n)$ tergantung pada implementasi dan optimasi yang digunakan. Hal ini membuat *Transformer* jauh lebih praktis untuk digunakan dalam tugas-tugas yang melibatkan data citra beresolusi tinggi seperti pengenalan objek, segmentasi citra, dan super-resolusi citra (Tay et al., 2022). Selain itu, ET memungkinkan aplikasi *Transformer* dalam perangkat dengan daya komputasi terbatas seperti smartphone, perangkat IoT, atau sistem *edge computing* di mana efisiensi adalah hal yang sangat penting.

Penggunaan *Efficient Transformer* dalam tugas visi komputer telah menunjukkan hasil yang luar biasa khususnya dalam menangani data gambar yang sangat besar. Arsitektur seperti *Performer* dan *Linformer* adalah contoh dari ET yang mengadopsi pendekatan inovatif untuk meningkatkan efisiensi komputasi. *Performer* misalnya, menggunakan *kernelized self-attention* untuk memperkirakan perhatian dengan efisiensi linier yang sangat bermanfaat dalam tugas-tugas yang membutuhkan kecepatan tinggi (Choromanski et al., 2020). Sementara itu, *Linformer* mengurangi dimensi matriks perhatian melalui proyeksi linier sehingga mengurangi jumlah parameter yang diperlukan dan membuat arsitektur *Transformer* menjadi lebih ringan (Tay et al., 2022).

Penerapan ET dalam *Single Image Super-Resolution* (SISR) juga memberikan manfaat yang besar. Dengan kemampuan untuk menangkap hubungan jangka panjang antar piksel dalam citra, ET dapat meningkatkan resolusi gambar dengan lebih efisien tanpa mengorbankan kualitas hasil. Model-model SISR

berbasis ET memanfaatkan perhatian global untuk memperbaiki detail tekstur dan pola sehingga dapat menghasilkan gambar yang lebih realistis dan tajam dibandingkan metode interpolasi tradisional (Sun et al., 2022). Secara keseluruhan, *Efficient Transformer* menawarkan pendekatan yang sangat diperlukan untuk mengatasi tantangan dalam skala besar dan pemrosesan data beresolusi tinggi dengan efisiensi komputasi yang lebih baik dan hasil yang tetap berkualitas tinggi. Dengan adopsi yang semakin luas dalam berbagai aplikasi, ET terus berkembang dengan inovasi baru untuk meningkatkan kinerja dan efisiensi lebih jauh.

2.2.8 Metrik Evaluasi Kinerja Model SISR

Dalam evaluasi kinerja model Single Image Super-Resolution (SISR), metrik kuantitatif sangat penting untuk mengukur seberapa baik model dalam merekonstruksi citra beresolusi tinggi dari citra beresolusi rendah. Dua metrik kuantitatif yang paling umum digunakan adalah Peak Signal-to-Noise Ratio (PSNR) dan Structural Similarity Index (SSIM). Kedua metrik ini memberikan indikasi numerik tentang kualitas rekonstruksi citra dan sering digunakan sebagai standar dalam penelitian SISR.

2.2.8.1 Peak Signal-to-Noise Ratio (PSNR)

Peak Signal-to-Noise Ratio (PSNR) adalah metrik yang digunakan untuk mengukur kualitas rekonstruksi citra dengan membandingkan perbedaan antara citra asli (*ground truth*) dan citra hasil rekonstruksi (National Instruments Australia, 2013). PSNR dinyatakan dalam satuan *desibel* (dB) dan secara matematis didefinisikan dalam Persamaan (2.1):

$$PSNR = 10 * \log_{10} \left(\frac{(MAX_I^2)}{MSE} \right) \quad (2.1)$$

Di mana:

- MAX_I adalah nilai maksimum intensitas piksel pada citra. Untuk citra 8-bit, $MAX_I = 255$
- MSE (*Mean Squared Error*) adalah rata-rata kuadrat kesalahan antara citra asli dan citra hasil rekonstruksi yang didefinisikan pada Persamaan (2.2) dengan rumus sebagai berikut:

$$MSE = \frac{1}{MN} \sum_{i=1}^m \sum_{j=1}^n [I(i,j) - K(i,j)]^2 \quad (2.2)$$

- $I(i,j)$ adalah nilai intensitas piksel pada posisi (i,j) di citra asli
- $K(i,j)$ adalah nilai intensitas piksel pada posisi (i,j) di citra hasil rekonstruksi.
- M dan N adalah dimensi citra

PSNR yang lebih tinggi menandakan kesalahan yang lebih kecil antara kedua citra dan kualitas rekonstruksi yang lebih baik. Kelebihan PSNR adalah kemudahan perhitungan dan interpretasi yang menjadikannya sebagai pilihan umum dalam evaluasi kinerja model *Single Image Super-Resolution* (SISR) dan standar industri untuk perbandingan yang konsisten antar model. Namun, PSNR memiliki keterbatasan karena tidak sesuai dengan persepsi visual manusia dengan hanya mempertimbangkan perbedaan per piksel dan sensitif terhadap perubahan kecil yang mungkin tidak signifikan secara visual sehingga tidak selalu mencerminkan kualitas visual yang sebenarnya (Horé & Ziou, 2010).

2.2.8.2 Structural Similarity Index (SSIM)

Structural Similarity Index (SSIM) adalah metrik yang dirancang untuk menilai kualitas citra dengan mempertimbangkan persepsi visual manusia. SSIM dapat menjadi metode dalam menganalisis perbedaan gambar sehingga diketahui anomali dari perbandingan dua gambar berdasarkan data struktural dari sebuah gambar. Metrik ini mengukur kesamaan struktural antara citra asli dan citra hasil rekonstruksi dengan mempertimbangkan *luminance*, *contrast*, dan *structure* (Wang et al., 2004). SSIM secara matematis didefinisikan pada Persamaan (2.3):

$$SSIM(x,y) = [l(x,y)]^\alpha \cdot [c(x,y)]^\beta \cdot [s(x,y)]^\gamma \quad (2.3)$$

Di mana:

- x dan y adalah dua *patch* citra dari citra asli dan citra hasil rekonstruksi.

- $l(x, y)$ adalah fungsi *luminance* pada Persamaan (2.4):

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (2.4)$$

- $c(x, y)$ adalah fungsi *contrast* Persamaan (2.5):

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (2.5)$$

- $s(x, y)$ adalah fungsi *structure* Persamaan (2.6):

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (2.6)$$

- μ_x dan μ_y adalah rata-rata intensitas pada *patch* x dan y .
- σ_x^2 dan σ_y^2 adalah variansi pada *patch* x dan y .
- σ_{xy} adalah kovarians antara x dan y .
- C_1, C_2 , dan C_3 adalah konstanta kecil untuk menghindari pembagian dengan nol.
- α, β , dan γ biasanya diatur ke 1.

SSIM menghasilkan nilai antara -1 hingga 1, di mana nilai 1 menunjukkan kesamaan sempurna. Kelebihan SSIM adalah kemampuannya menilai kualitas dengan cara yang lebih selaras dengan persepsi visual manusia sehingga lebih sensitif terhadap perubahan struktural yang signifikan secara visual (Wang et al., 2004). Namun, SSIM juga memiliki keterbatasan dalam kompleksitas perhitungannya dibandingkan PSNR dan kepekaannya terhadap perubahan skala atau rotasi sehingga membutuhkan citra yang terdaftar dengan baik untuk menghasilkan hasil yang akurat (Horé & Ziou, 2010).

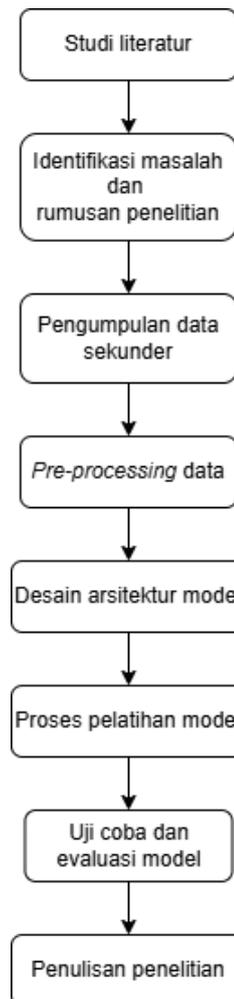
BAB 3

METODOLOGI PENELITIAN

Bab metodologi penelitian menjelaskan alur yang digunakan dalam melakukan penelitian tesis. Metodologi penelitian tersebut berisi mengenai diagram metodologi penelitian, uraian metodologi penelitian, dan rencana jadwal pelaksanaan penelitian dari penelitian tesis ini.

3.1 Diagram Metodologi Penelitian

Pada sub bab ini, akan dijelaskan metodologi atau tahapan pelaksanaan penelitian. Gambar 3.1 menunjukkan diagram metodologi penelitian.



Gambar 3.1 Diagram metodologi penelitian

3.2 Uraian Metodologi Penelitian

Pada sub bab ini, akan dijelaskan lebih rinci mengenai metodologi atau tahapan pelaksanaan penelitian.

3.2.1 Studi Literatur

Pada tahap ini, sumber literatur yang relevan dan mendukung pengerjaan penelitian dikumpulkan secara komprehensif sebagai acuan dan dasar dalam melakukan penelitian. Sumber literatur yang digunakan mencakup jurnal ilmiah, konferensi internasional, buku, dan publikasi terbaru yang berkaitan dengan *Single Image Super-Resolution (SISR)*, *Convolutional Neural Networks (CNN)*, *Transformer*, dan khususnya *Efficient Transformer (ET)*. Studi literatur ini bertujuan untuk memahami secara mendalam konsep, teori, landasan, dan teknologi yang menjadi dasar dalam penelitian ini. Penelitian dimulai dengan mengkaji literatur mengenai SISR untuk memahami tantangan utama dalam meningkatkan resolusi citra dan metode-metode yang telah dikembangkan sebelumnya. Selanjutnya, studi difokuskan pada arsitektur CNN yang memberikan performa tinggi dan efisien dalam tugas SISR. Arsitektur seperti SRCNN, VDSR, RDN, dan khususnya *Enhanced Group Convolutional Neural Network (EGCNN)* dianalisis untuk memahami mekanisme kerjanya, keunggulan, dan keterbatasannya. EGCNN dipilih karena efisiensinya dalam komputasi melalui penggunaan *group convolution*, namun diketahui memiliki keterbatasan dalam menangkap hubungan jarak jauh antar piksel.

Untuk mengatasi keterbatasan tersebut, literatur terkait *Transformer* terutama *Efficient Transformer (ET)* dikaji secara mendalam. Studi difokuskan pada bagaimana *Transformer* melalui mekanisme *self-attention* mampu menangkap hubungan jangka panjang dalam data sekuensial. Namun, kompleksitas komputasi *Transformer* standar menjadi tantangan ketika diterapkan pada citra beresolusi tinggi. Oleh karena itu, literatur mengenai ET yang menawarkan solusi dengan mengurangi kompleksitas komputasi sambil mempertahankan kemampuan menangkap hubungan global menjadi fokus utama. Penelitian sebelumnya seperti ESRT yang mengintegrasikan ET ke dalam arsitektur RCAN dianalisis untuk memahami pendekatan integrasi dan hasil yang diperoleh. Studi ini juga bertujuan untuk mengidentifikasi kesenjangan penelitian yaitu belum adanya integrasi ET ke dalam EGCNN sehingga membuka peluang untuk kontribusi baru dalam bidang ini.

3.2.2 Identifikasi Masalah dan Rumusan Penelitian

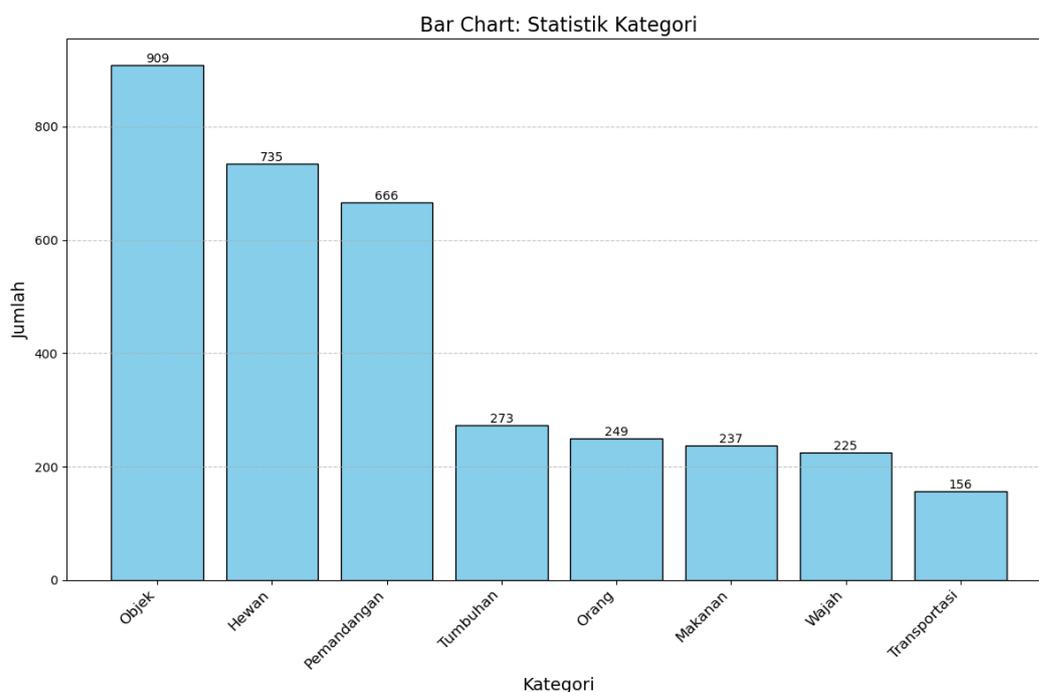
Berdasarkan hasil studi literatur yang telah dilakukan, identifikasi masalah difokuskan pada keterbatasan EGCNN dalam menangkap hubungan jarak jauh antar piksel. Meskipun EGCNN efisien dalam komputasi dan efektif dalam mengekstraksi fitur lokal melalui *group convolution*, sifat lokal dari operasi konvolusi membatasi kemampuannya dalam memahami konteks global dan struktur keseluruhan citra. Hal ini dapat mengakibatkan kurang optimalnya rekonstruksi detail tekstur halus dan pola-pola global pada citra beresolusi tinggi. Di sisi lain, *Transformer* telah terbukti mampu menangkap hubungan jangka panjang melalui mekanisme *self-attention*, namun kompleksitas komputasi yang tinggi menjadi hambatan dalam penerapannya pada tugas SISR terutama untuk citra beresolusi tinggi. *Efficient Transformer* (ET) muncul sebagai solusi dengan menawarkan mekanisme perhatian yang lebih efisien dan reduksi kompleksitas komputasi sehingga lebih cocok untuk diterapkan dalam model SISR.

Rumusan penelitian yang diangkat adalah bagaimana mengintegrasikan *Efficient Transformer* ke dalam arsitektur *Enhanced Group Convolutional Neural Network* untuk meningkatkan kinerja *Single Image Super-Resolution*. Selanjutnya penelitian ini diharapkan dapat mengatasi keterbatasan EGCNN dalam menangkap hubungan jarak jauh tanpa menambah beban komputasi secara signifikan, dengan memanfaatkan keunggulan *Efficient Transformer*.

3.2.3 Pengumpulan Data Sekunder

Pada tahap pengumpulan data sekunder, penelitian ini memanfaatkan dataset DF2K yang telah dipublikasikan di website OpenModelDB (<https://openmodeldb.info/models/4x-realsr-df2k>) sebagai sumber data utama untuk melatih model yang diusulkan. DF2K merupakan kombinasi dari dua dataset populer dalam bidang *Single Image Super-Resolution* (SISR) yaitu DIV2K dan Flickr2K yang bertujuan untuk memperkaya variasi data dan meningkatkan jumlah sampel pelatihan sehingga model dapat belajar representasi yang lebih baik dan mencapai generalisasi yang lebih kuat. DIV2K (DIVERse 2K resolution images) berisi 800 citra beresolusi tinggi untuk pelatihan, 100 citra untuk validasi, dan 100 citra untuk pengujian (Timofte et al., 2017) dengan seleksi cermat mencakup pemandangan alam, arsitektur, objek buatan manusia, dan tekstur kompleks.

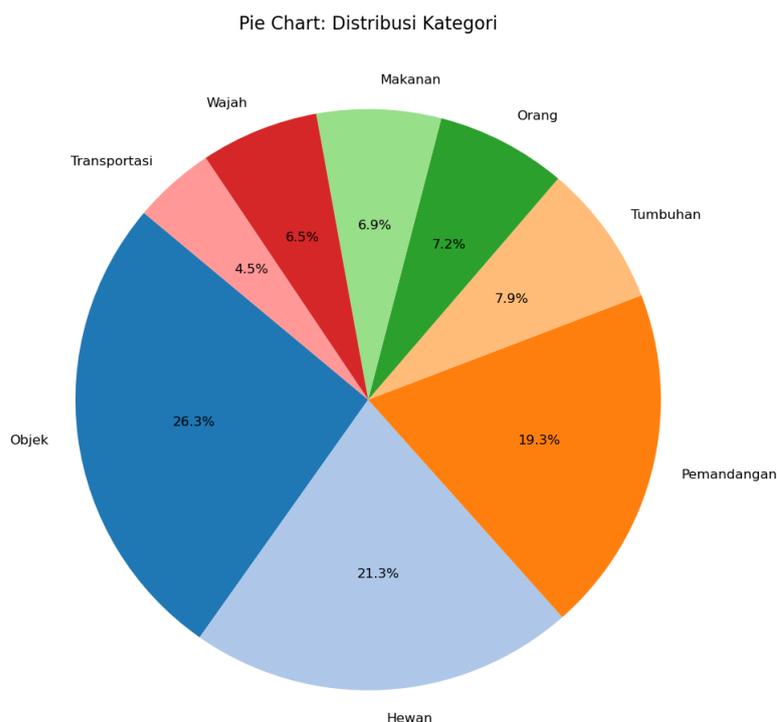
Resolusi sekitar 2K memungkinkan model mempelajari detail halus yang penting dalam SISR. Flickr2K, di sisi lain memuat 2650 citra beresolusi tinggi yang dikumpulkan dari platform Flickr, menawarkan rentang kondisi pencahayaan, perspektif, dan jenis konten yang beragam. Variasi konten ini mencakup pemandangan alam, objek manusia, dan tekstur-detail lainnya, yang memberikan tantangan bagi model SISR untuk mempelajari pola-pola pencahayaan serta fitur yang lebih kompleks.



Gambar 3.2 Jumlah citra pada setiap kategori dalam dataset DF2K

Hasil kurasi dan *captioning* data pada DF2K menghasilkan total 3450 citra yang dikategorikan ke dalam delapan kategori utama, yaitu Objek, Hewan, Pemandangan, Tumbuhan, Orang, Makanan, Wajah, dan Transportasi yang ditampilkan dalam Gambar 3.2. Kategori Objek (berjumlah 909) mencakup berbagai benda statis maupun dinamis seperti peralatan rumah tangga atau alat elektronik, sedangkan Hewan (735) berfokus pada citra beragam spesies mulai dari hewan peliharaan hingga satwa liar. Pemandangan (666) memuat panorama alam beserta latar belakang lingkungan yang bervariasi, sedangkan Tumbuhan (273) menampilkan vegetasi, bunga, serta detail struktur daun dan batang. Selanjutnya, kategori Orang (249) meliputi gambar individu atau kelompok dengan berbagai

pose dan situasi, sementara Makanan (237) menyoroti aneka kuliner dan komposisi hidangan. Kategori Wajah (225) dikhususkan untuk potret wajah dengan variasi ekspresi dan pencahayaan, dan Transportasi (156) menampilkan sarana mobilitas seperti mobil, sepeda motor, kapal, hingga pesawat. Distribusi masing-masing kategori ditunjukkan pada diagram lingkaran dalam Gambar 3.3 yang memperlihatkan keragaman data untuk membantu model mengenali pola-pola visual yang berbeda.

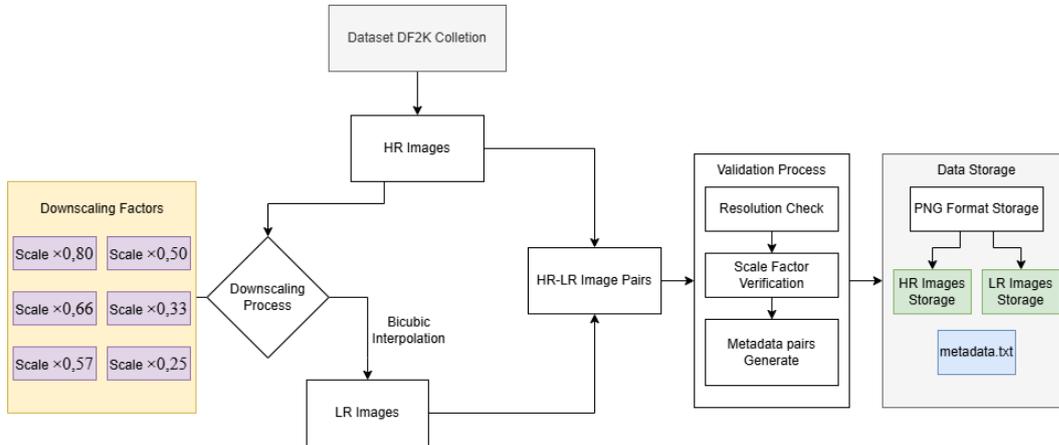


Gambar 3.3 Prosentase jumlah citra pada setiap kategori dalam dataset DF2K

Penggunaan DF2K dalam penelitian ini didasarkan pada pertimbangan bahwa jumlah data yang lebih besar dan variasi yang lebih luas akan membantu model dalam belajar representasi yang lebih kaya, mengurangi risiko *overfitting*, dan meningkatkan kemampuan generalisasi. Penggunaan dataset standar seperti DF2K juga memudahkan perbandingan kinerja model dengan penelitian lain dalam bidang yang sama. Selain itu, dataset ini telah banyak digunakan dalam kompetisi dan penelitian SISR terkini sehingga validitas dan relevansinya sudah teruji.

3.2.4 Pre-processing Data

Setelah pengumpulan data sekunder, tahap berikutnya adalah *pre-processing* data yang melibatkan persiapan citra beresolusi tinggi (*High-Resolution*, HR) dan pembuatan pasangan citra beresolusi rendah (*Low-Resolution*, LR) sebagai input untuk model. Berikut adalah gambaran diagram untuk *pre-processing* yang ditunjukkan pada Gambar 3.4.



Gambar 3.4 Diagram *pre-processing* data

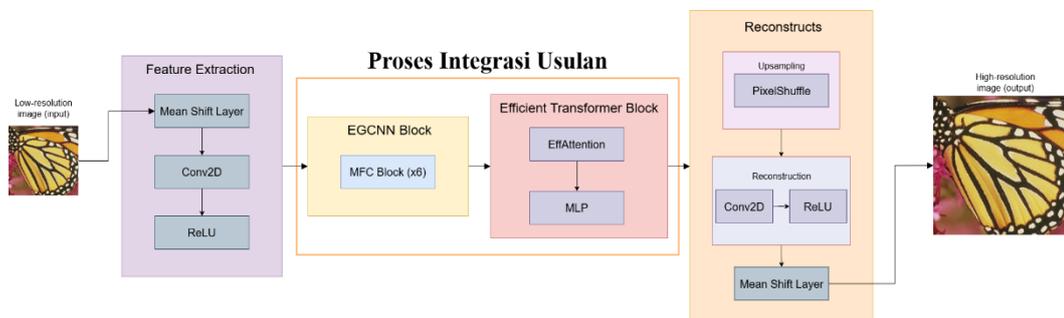
Citra HR dari dataset DF2K di-*downscaling* untuk menghasilkan citra LR menggunakan faktor skala pembesaran yang berbeda, yaitu ($\times 0.80$, $\times 0.66$, $\times 0.57$, $\times 0.50$, $\times 0.33$, $\times 0.25$). Proses *downscaling* ini dilakukan menggunakan metode interpolasi *bicubic*, yang merupakan metode standar dalam penelitian SISR. Metode ini mempertimbangkan 16 piksel tetangga terdekat untuk menghitung nilai piksel baru sehingga menghasilkan citra LR yang halus dan realistis. Dengan menerapkan berbagai skala pembesaran, model dapat dilatih untuk menangani skenario super-resolusi yang berbeda. Proses *downscaling* bertujuan untuk mensimulasikan degradasi yang terjadi pada citra beresolusi rendah sehingga model dapat belajar memetakan kembali citra LR ke citra HR yang sesuai. Setiap citra HR dipasangkan dengan citra LR yang dihasilkan dari *downscaling* untuk membentuk pasangan data yang akan digunakan dalam pelatihan. Pasangan ini penting untuk melatih model secara *supervised*, di mana citra LR menjadi input dan citra HR menjadi target output.

Selanjutnya, informasi pasangan citra HR-LR divalidasi melalui pengecekan resolusi dan kesesuaian ukuran untuk menghasilkan pasangan metadata

terkait. Hal ini penting untuk memastikan bahwa data yang masuk ke dalam model adalah konsisten dan bebas dari kesalahan yang dapat mempengaruhi proses pelatihan. Hasil dari *pre-processing* ini adalah pasangan data citra HR dan citra LR serta metadata.txt. Dengan melakukan *pre-processing* data, model dapat dilatih dengan data berkualitas tinggi dan konsisten yang penting untuk mencapai kinerja optimal. Proses ini juga memastikan bahwa *pipeline* pelatihan berjalan lancar tanpa hambatan yang disebabkan oleh kesalahan data.

3.2.5 Desain Arsitektur Model

Desain arsitektur model yang diusulkan dalam penelitian ini merupakan kombinasi dari *Enhanced Group Convolutional Neural Network* (EGCNN) dan *Efficient Transformer* (ET) dengan tujuan utama meningkatkan kinerja *Single Image Super-Resolution* (SISR).



Gambar 3.5 Desain arsitektur usulan

EGCNN berperan dalam mengekstraksi fitur-fitur lokal dari citra beresolusi rendah (*Low-Resolution*, LR) secara efisien melalui penggunaan *group convolution*, sementara ET bertugas menangkap hubungan jangka panjang antar piksel untuk memperkaya representasi fitur dengan informasi global. Integrasi kedua komponen ini diharapkan dapat mengatasi keterbatasan EGCNN dalam memahami konteks global tanpa mengorbankan efisiensi komputasi.

Proses dimulai dengan citra input LR yang memiliki ukuran tertentu (H, W, C), di mana H dan W adalah tinggi dan lebar citra, dan C adalah jumlah saluran warna (ukurannya 3 untuk citra RGB). Sebagai ilustrasi, jika kita menggunakan citra LR dengan resolusi (64, 64, 3), maka ukuran *tensor* input dalam format *PyTorch* (N, C, H, W) adalah (*Batch Size*, 3, 64, 64), di mana *Batch Size* adalah jumlah sampel dalam satu *batch* pelatihan. Langkah pertama adalah normalisasi

citra input menggunakan *mean shift*, di mana nilai rata-rata per saluran (*mean RGB* dari dataset DF2K: [0.4488, 0.4371, 0.4040]) dikurangi dari nilai piksel citra. Normalisasi ini dilakukan untuk mempercepat konvergensi selama pelatihan dan meningkatkan stabilitas numerik model. Setelah normalisasi, citra LR diproses oleh lapisan konvolusi awal untuk mengubah citra input menjadi representasi fitur awal dengan dimensi saluran yang lebih tinggi. Lapisan konvolusi yang diterapkan adalah *kernel size 3x3*, *stride 1*, *padding 1*, dan jumlah *features 64*. Ukuran *tensor* setelah lapisan ini menjadi (*Batch Size, 64, 64, 64*), di mana dimensi spasial tetap (karena *padding*), tetapi dimensi saluran meningkat menjadi 64 sehingga memungkinkan model untuk mengekstraksi fitur yang lebih kompleks. Fitur awal ini kemudian diteruskan ke blok EGCNN yang terdiri dari beberapa blok MFC (*Multi-Feature Channel*) untuk ekstraksi fitur lokal yang lebih dalam. Setiap blok MFC terdiri dari:

- *Group Convolution*: Lapisan konvolusi di mana saluran input dibagi menjadi beberapa grup. Misalnya, jika jumlah saluran adalah 64 dan jumlah grup adalah 4, maka setiap grup memiliki 16 saluran. Konvolusi dilakukan secara independen pada setiap grup sehingga mengurangi jumlah parameter dan operasi komputasi. Setelah konvolusi, saluran digabungkan kembali, dan ukuran *tensor* tetap (*Batch Size, 64, 64, 64*).
- Aktivasi *Non-Linear* (ReLU): Diterapkan setelah konvolusi untuk memperkenalkan *non-linearitas* supaya model dapat mempelajari fungsi yang lebih kompleks.
- Mekanisme Residual: *Shortcut connection* yang menambahkan input blok ke output setelah konvolusi dan aktivasi, membantu mengatasi masalah *vanishing gradient* dan memfasilitasi pelatihan jaringan yang lebih dalam.

Setelah melewati sejumlah blok EGCNN (6 blok), *tensor* fitur yang dihasilkan masih memiliki ukuran (*Batch Size, 64, 64, 64*). Pada titik ini, fitur-fitur lokal telah diekstraksi tetapi hubungan jangka panjang antar piksel belum sepenuhnya dimodelkan. Untuk mengintegrasikan *Efficient Transformer* (ET),

tensor fitur perlu diubah menjadi format yang sesuai dengan input ET. Proses ini melibatkan meratakan (*flattening*) dimensi spasial *tensor* menjadi urutan vektor. *Tensor* dengan ukuran (*Batch Size*, 64, 64, 64) diubah menjadi (*Batch Size*, 64, 4096), di mana 4096 berasal dari hasil kali 64 x 64 (dimensi spasial). Dengan demikian, setiap sampel dalam *batch* memiliki 64 urutan fitur dengan panjang 4096. Namun, untuk mengurangi kompleksitas komputasi yang tinggi dari *Transformer* (yang meningkat secara kuadrat terhadap panjang urutan), dilakukan reduksi dimensi dan proyeksi linier dengan jumlah fitur direduksi dari 64 saluran menjadi 32 saluran menggunakan lapisan konvolusi 1x1 sebelum *flattening*. Hal ini menghasilkan *tensor* dengan ukuran (*Batch Size*, 32, 64, 64), yang kemudian di-*flatten* menjadi (*Batch Size*, 32, 4096).

Efficient Transformer kemudian diterapkan pada urutan fitur ini. ET menggunakan mekanisme *self-attention* yang efisien dengan menerapkan *Linformer* atau *Performer* untuk mengurangi kompleksitas dari $O(N^2)$ menjadi $O(N)$, di mana N adalah panjang urutan (dalam hal ini 4096). Selain itu, teknik seperti *chunked attention* atau *window-based attention* digunakan dengan membagi urutan menjadi segmen-segmen yang lebih kecil (4 segmen dengan panjang 16) sehingga setiap operasi *self-attention* hanya perlu menangani urutan yang lebih pendek untuk mengurangi beban komputasi. Selama pemrosesan oleh ET, mekanisme *self-attention* memungkinkan model untuk menangkap hubungan jangka panjang antar piksel dengan mempertimbangkan interaksi antara semua posisi dalam urutan. Setelah melalui lapisan ET dengan *feed-forward network* dan *layer normalization*, fitur-fitur yang telah diperkaya informasi global diperoleh dengan ukuran yang sama (*Batch Size*, 32, 4096). Fitur-fitur ini kemudian dikembalikan ke bentuk *tensor* dengan dimensi spasial melalui *reshaping* (*unflattening*), menghasilkan *tensor* dengan ukuran (*Batch Size*, 32, 64, 64). Pada titik ini, fitur-fitur tersebut mengandung informasi lokal dari EGCNN dan informasi global dari ET untuk siap masuk ke tahap rekonstruksi.

Tahap Rekonstruksi dan *Upsampling* melibatkan beberapa lapisan konvolusi tambahan yang bertujuan untuk menggabungkan fitur-fitur dan merekonstruksi citra HR. Lapisan konvolusi dengan *kernel size* 3x3, *padding* 1, dan jumlah *features* saluran yang meningkat menjadi 64 digunakan untuk memperkaya

representasi fitur. Aktivasi *non-linear* seperti ReLU diterapkan setelah setiap lapisan untuk meningkatkan kapasitas representasi model. Untuk meningkatkan resolusi citra ke tingkat yang diinginkan maka digunakan lapisan *upsampling* yang memanfaatkan *PixelShuffle*. Skenarionya jika faktor skala pembesaran adalah $\times 2$, dan *tensor* fitur sebelum *upsampling* memiliki ukuran (*Batch Size*, 256, 64, 64) (setelah konvolusi), *PixelShuffle* dengan faktor $r=2$ akan mengubah *tensor* menjadi ukuran (*Batch Size*, 64, 128, 128). *PixelShuffle* bekerja dengan mengatur ulang elemen-elemen dalam *tensor* fitur untuk memperluas dimensi spasial sambil mengurangi dimensi saluran. Setelah *upsampling*, lapisan konvolusi akhir diikuti aktivasi *non-linear* diterapkan untuk menyempurnakan citra dan menghilangkan artefak yang mungkin timbul selama proses *upsampling*. Lapisan ini menggunakan *kernel size* 3x3 dan jumlah *features* 64.

Akhirnya, lapisan *mean shift* inversi diterapkan untuk mengembalikan nilai piksel citra ke rentang asli sebelum normalisasi. Hal ini dilakukan dengan menambahkan kembali nilai *mean* per saluran yang sebelumnya dikurangi selama normalisasi. Citra output HR yang dihasilkan memiliki ukuran (*Batch Size*, 3, $H * r$, $W * r$), misalnya dengan skenario (*Batch Size*, 3, 128, 128) untuk faktor skala $\times 2$ dan input LR berukuran (64, 64, 3). Dengan desain arsitektur ini, model diharapkan mampu memanfaatkan keunggulan EGCNN dalam mengekstraksi fitur lokal secara efisien dan kemampuan ET dalam menangkap hubungan jangka panjang antar piksel.

3.2.6 Proses Pelatihan Model

Proses pelatihan model dirancang menggunakan dataset yang telah dipersiapkan untuk menghasilkan kemampuan rekonstruksi citra beresolusi tinggi yang optimal. Konfigurasi pelatihan model diatur melalui sebuah *file* konfigurasi yang mendefinisikan berbagai parameter penting termasuk pengaturan umum, dataset, struktur jaringan, jalur penyimpanan, pengaturan pelatihan, validasi, dan *logging*. Berikut ini adalah uraian penjelasan mengenai pengaturan tersebut sesuai dengan kode parameter yang digunakan dalam program.

Pada bagian pengaturan umum, parameter seperti *name*, *model_type*, *scale*, *num_gpu*, *manual_seed*, *num_in_ch*, dan *num_out_ch* ditetapkan. *name* diberikan nilai "Experiment_EGCNN+ET", yang merupakan identitas unik untuk eksperimen

pelatihan model yang mengintegrasikan ET ke dalam EGCNN. *model_type* diatur sebagai SRModel yang menunjukkan bahwa model yang dilatih adalah model untuk *Single Image Super-Resolution*. *Scale* ditetapkan dengan nilai 2 yang berarti faktor pembesaran citra adalah 2 kali dari resolusi asli. *num_gpu* dengan nilai 1 menunjukkan bahwa pelatihan akan dilakukan menggunakan satu GPU, sesuai dengan ketersediaan perangkat keras. *manual_seed* diset ke 0 untuk memastikan reproduksibilitas hasil dengan menetapkan *seed generator* angka acak. *num_in_ch* dan *num_out_ch* masing-masing ditetapkan ke 3 yang sesuai dengan jumlah saluran pada citra RGB untuk input dan output model.

Pada pengaturan data *loader*, bagian *datasets* mendefinisikan dataset yang digunakan untuk pelatihan dan validasi. Untuk dataset pelatihan (*train*), nama dataset ditetapkan sebagai “DF2K”, dan tipe dataset adalah “PairedImageDataset” yang menunjukkan bahwa dataset terdiri dari pasangan citra beresolusi tinggi (*ground truth*) dan beresolusi rendah (*low quality*). Lokasi data ditentukan oleh *dataroot_gt* dan *dataroot_lq* yang mengarah ke masing-masing direktori tempat *file ground truth* (*gt*) dan *low quality* (*lq*). *meta_info* mengacu pada *file* berekstensi *.txt* yang berisi informasi metadata pasangan nama *file* dataset. *Backend* I/O diatur dengan *io_backend* tipe *disk* yang menunjukkan bahwa data akan diakses dari *disk*. Parameter *gt_size* dengan nilai 256 menetapkan ukuran *patch* citra GT yang akan digunakan selama pelatihan. Penggunaan augmentasi data diaktifkan dengan *use_hflip* dan *use_rot* diset ke *true* untuk memungkinkan model dapat belajar dari variasi citra yang lebih beragam melalui *flipping* horizontal dan rotasi. Pengaturan data *loader* yang lain mencakup *use_shuffle* yang diset *true* untuk mengacak data setiap *epoch*, *num_worker_per_gpu* dengan nilai 6 yang menunjukkan jumlah proses pekerja untuk memuat data per GPU, dan *batch_size_per_gpu* yang diset ke 4 sebagai satu kali iterasi yang menyesuaikan dengan keterbatasan memori GPU karena kompleksitas model. *dataset_enlarge_ratio* dengan nilai 1 berarti dataset tidak diperbesar secara buatan, sementara *prefetch_mode* diset ke *cuda* dan *pin_memory* ke *true* untuk meningkatkan efisiensi pemuatan data dengan memanfaatkan *prefetching* ke memori GPU.

Untuk dataset validasi (*val_1* dan *val_2*) digunakan dataset “Set5” dan “Set14” yang umum dalam evaluasi SISR. Kedua dataset ini juga menggunakan

tipe `PairedImageDataset` dengan direktori data yang sesuai. *Backend* I/O tetap menggunakan tipe *disk*. Bagian struktur jaringan (`network_g`) mendefinisikan tipe jaringan yang digunakan dalam hal ini type: `Net`, yang mengacu pada arsitektur model yang telah diimplementasikan yaitu EGCNN dengan integrasi ET. Pada pengaturan jalur (*path*), `pretrain_network_g` diset ke `~`, yang menunjukkan bahwa tidak ada model pra-latih yang akan dimuat. `strict_load_g` dengan nilai *true* memastikan bahwa parameter model harus cocok secara tepat saat melakukan *load* untuk membantu menghindari masalah kompatibilitas. `resume_state` juga diset ke `~`, yang menandakan bahwa pelatihan akan dimulai dari awal tanpa melanjutkan dari *checkpoint* sebelumnya. Bagian pengaturan pelatihan (*train*) mencakup beberapa parameter penting. `ema_decay` dengan nilai 0.999 menunjukkan penggunaan *Exponential Moving Average* untuk parameter model yang dapat membantu dalam stabilitas pelatihan dan peningkatan kinerja. Pengaturan *optimizer* (`optim_g`) menggunakan algoritma Adam dengan *lr* (*learning rate*) sebesar $1e-4$, `weight_decay` 0, dan `betas` [0.9, 0.999], yang merupakan parameter *default* untuk Adam dan cocok untuk berbagai tugas pembelajaran.

Scheduler untuk penyesuaian *learning rate* ditetapkan dengan *scheduler* tipe *MultiStepLR*, dengan *milestones* pada iterasi [50000, 100000, 200000], dan `gamma` 0.5. Ini berarti *learning rate* akan dikalikan dengan 0.5 pada iterasi tersebut untuk membantu dalam penurunan *learning rate* secara bertahap yang berdampak pada konvergensi yang lebih baik. `total_iter` diset ke 300000, menunjukkan bahwa pelatihan akan dilakukan selama 300.000 iterasi. `warmup_iter` dengan nilai -1 menandakan bahwa tidak ada *warm-up* yang digunakan dalam penyesuaian *learning rate* di awal pelatihan. Untuk fungsi *loss*, `pixel_opt` menggunakan type: *L1Loss* dengan `loss_weight` 1.0 dan `reduction`: *mean*. *L1Loss* dipilih karena sering menghasilkan hasil yang lebih tajam dibandingkan MSE (*Mean Squared Error*), dengan menghitung perbedaan absolut antara prediksi dan *ground truth*.

Pada pengaturan validasi (`val`), `val_freq` diset ke $5e4$ (50.000) yang berarti validasi akan dilakukan setiap 50.000 iterasi. `save_img` dengan nilai *true* menginstruksikan sistem untuk menyimpan citra hasil rekonstruksi selama validasi. `pbar` diset ke *false* untuk menonaktifkan progress bar selama validasi. Bagian *metrics* mendefinisikan metrik evaluasi yang akan digunakan yaitu PSNR dan

SSIM. Untuk PSNR, `crop_border` diset ke 2 dan `test_y_channel` diset ke `true` yang berarti evaluasi dilakukan pada komponen *luminance* (Y) saja setelah menghilangkan *border* sebesar 2 piksel. Konfigurasi ini sesuai dengan praktik umum dalam evaluasi SISR. Hal yang sama berlaku untuk SSIM.

Pada pengaturan logging (*logger*), `print_freq` dengan nilai 100 berarti informasi log akan dicetak setiap 100 iterasi. `save_checkpoint_freq` diset ke `1e4` (10.000) sehingga model akan disimpan setiap 10.000 iterasi yang memungkinkan *recovery* jika terjadi kegagalan dan memonitoring performa model pada berbagai tahap pelatihan. `use_tb_logger` dengan nilai `true` menunjukkan bahwa *TensorBoard Logger* akan digunakan untuk visualisasi metrik pelatihan secara *real-time*. Bagian *wandb* dengan `project` dan `resume_id` diset ke `~`, yang menunjukkan bahwa *Weights & Biases* tidak digunakan dalam eksperimen ini. Terakhir, pada pengaturan distribusi pelatihan (`dist_params`) *backend* diset ke `nccl` dan *port* ke 29500.

3.2.7 Uji Coba dan Evaluasi Model

Pada tahap uji coba dan evaluasi, penelitian ini tidak hanya membandingkan kinerja model usulan dengan beberapa model SISR terkini, tetapi juga menguji beberapa variasi modifikasi pada arsitektur usulan untuk memperoleh pemahaman yang lebih komprehensif tentang pengaruh setiap komponen. Adapun skenario uji coba dan prosedur pengujian diuraikan sebagai berikut:

1. Variasi jumlah blok EGCNN dan ET
 - EGCNN-ET (6 blok EGCNN, 2 blok ET). Konfigurasi *default* model usulan.
 - EGCNN-ET (4 blok EGCNN, 2 blok ET). Mengurangi jumlah blok EGCNN untuk melihat efek pengurangan kedalaman pada kemampuan ekstraksi fitur lokal.
 - EGCNN-ET (6 blok EGCNN, 4 blok ET). Menambah jumlah blok ET untuk mengukur seberapa jauh hubungan jarak jauh dapat ditangkap tanpa mengorbankan efisiensi komputasi.
2. Variasi jumlah *epoch*
 - 100K iterasi (atau ± 115 *epoch*). Tahap awal untuk melihat tren konvergensi.

- 200K iterasi (atau ± 230 *epoch*). Pengujian lebih lama untuk mengobservasi performa dan stabilitas model.
 - 300K iterasi (*default*). Menjadi patokan utama sesuai konfigurasi pelatihan awal untuk menilai apakah model masih memperoleh peningkatan kinerja signifikan.
3. Model-model pembandingan
- EGCNN tanpa ET (*Baseline*). Menilai dampak langsung integrasi *Efficient Transformer*.
 - SwinIR (Liang et al., 2021). Model restorasi citra *state-of-the-art* menggunakan *Swin Transformer*.

Keseluruhan konfigurasi di atas bertujuan untuk menilai baik kontribusi modul ET di dalam EGCNN, jumlah blok yang optimal, maupun durasi pelatihan yang dibutuhkan. Setelah setiap modifikasi selesai dilatih, metrik evaluasi akan dihitung dan direkap.

Pada tahap pengujian, setiap varian model yang telah dilatih termasuk EGCNN+ET dalam berbagai konfigurasi serta model pembandingan yaitu EGCNN tanpa ET dan SwinIR akan dimuat beserta parameter terlatihnya (*checkpoint*). Masing-masing model kemudian menjalankan proses inferensi pada dataset pengujian yaitu Set5, Set14, BSD100, dan Urban100. Selama inferensi, citra beresolusi rendah (LR) diolah oleh model untuk diubah menjadi citra beresolusi tinggi (HR), dan seluruh hasil rekonstruksi dicatat di folder khusus untuk dianalisis lebih lanjut. Setelahnya, metrik kuantitatif seperti PSNR (*Peak Signal-to-Noise Ratio*) dan SSIM (*Structural Similarity Index*) dihitung dengan membandingkan citra hasil rekonstruksi terhadap citra *ground truth* yang sesuai. Terakhir, seluruh hasil pengukuran baik PSNR dan SSIM maupun waktu inferensi dan jumlah parameter dikompilasi dalam bentuk tabel atau grafik. Hasil tersebut nantinya menjadi dasar untuk menarik kesimpulan mengenai kinerja dan efisiensi model usulan serta membandingkannya dengan model-model pembandingan.

3.2.8 Penulisan Penelitian

Pada tahap ini, dilakukan penulisan buku penelitian. Buku penelitian ditulis berdasarkan hasil dari penelitian yang telah dilaksanakan. Buku ini mencakup seluruh proses pengerjaan penelitian dari awal hingga akhir.

3.3 Rencana Jadwal Pelaksanaan Penelitian

Pada sub bab ini, akan dijelaskan mengenai rencana waktu pelaksanaan penelitian. Rencana jadwal pelaksanaan penelitian ditunjukkan pada Tabel 3.1.

Tabel 3.1 Rencana jadwal pelaksanaan penelitian

No.	Kegiatan	Bulan 1				Bulan 2				Bulan 3			
		1	2	3	4	1	2	3	4	1	2	3	4
1.	Studi literatur	■											
2.	Identifikasi masalah dan rumusan penelitian		■										
3.	Pengumpulan data sekunder			■									
4.	<i>Pre-processing</i> data				■								
5.	Desain arsitektur model					■	■	■					
6.	Proses pelatihan model								■	■	■	■	
7.	Uji coba dan evaluasi model												■
8.	Penulisan penelitian	■	■	■	■	■	■	■	■	■	■	■	■

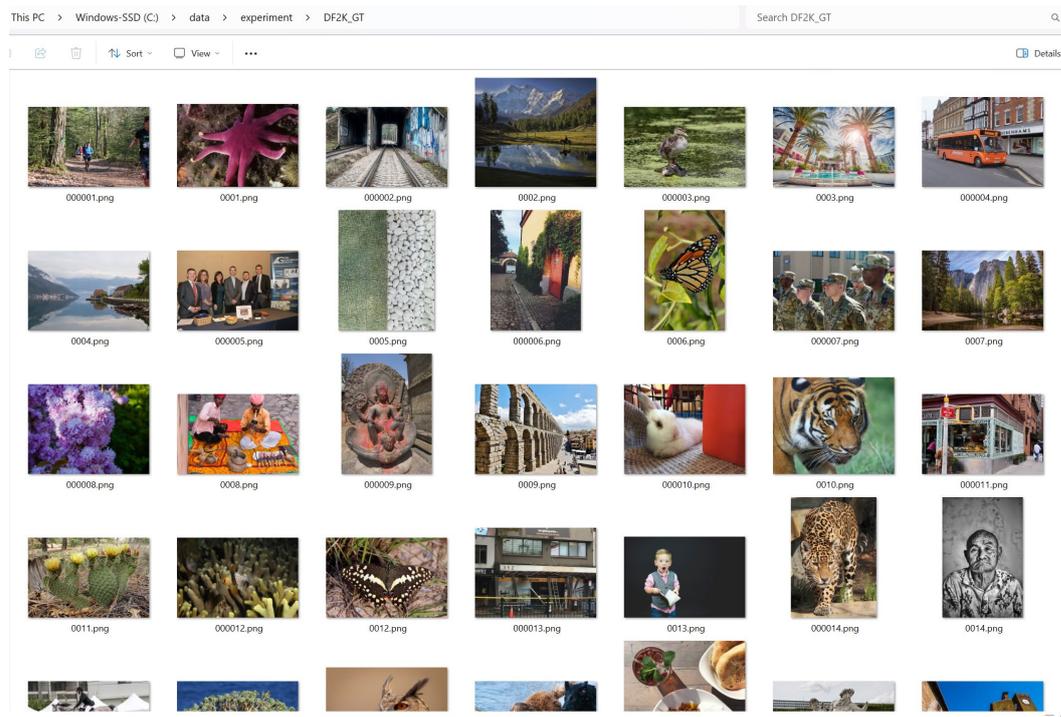
BAB 4

HASIL DAN PEMBAHASAN

Bab ini membahas secara menyeluruh hasil-hasil yang diperoleh dari penelitian, meliputi deskripsi data, konfigurasi lingkungan implementasi, pemaparan hasil uji coba, dan pembahasan temuan penelitian. Struktur penyajian bertujuan memberikan alur yang jelas, mulai dari penjelasan data, metode eksperimen, hingga analisis komparatif dengan model lain.

4.1 Deskripsi Data

Penelitian ini memanfaatkan dataset DF2K sebagai data utama untuk pelatihan model. DF2K merupakan gabungan dari DIV2K dan Flickr2K yang bertujuan untuk menghadirkan keragaman konten yang lebih luas. Sebagai gambaran, data sampel citra ditunjukkan pada Gambar 4.1. Dataset ini terdiri atas 3450 citra beresolusi tinggi yang telah dikategorikan dalam delapan jenis objek: Objek, Hewan, Pemandangan, Tumbuhan, Orang, Makanan, Wajah, dan Transportasi. Distribusi tiap kategori disajikan dalam bentuk diagram batang (Gambar 3.2) serta diagram lingkaran (Gambar 3.3) guna memudahkan pemahaman mengenai proporsi data.



Gambar 4.1 sampel data citra dari dataset DF2K

DIV2K (*DIVerse 2K resolution images*) sendiri memuat 800 citra HR untuk pelatihan, 100 untuk validasi, dan 100 untuk pengujian, dengan beragam objek seperti pemandangan alam, bangunan, objek buatan manusia, dan tekstur kompleks. Flickr2K menyediakan 2650 citra HR yang dikumpulkan dari Flickr, menawarkan variasi kondisi pencahayaan, perspektif, dan jenis konten yang lebih beragam. Dengan menggabungkan kedua dataset, model diharapkan mempelajari representasi fitur yang kaya untuk *Single Image Super-Resolution* (SISR) dengan pengurangan risiko *overfitting*, serta meningkatkan kemampuan generalisasi.

4.2 Lingkungan Implementasi

Penelitian ini diimplementasikan menggunakan sumber daya perangkat keras dan perangkat lunak dengan spesifikasi yang ditunjukkan pada Tabel 4.1 dan Tabel 4.2.

Tabel 4.1 Spesifikasi perangkat keras

Prosesor	13th Gen Intel(R) Core(TM) i9-13900HX
<i>Graphics Processing Unit</i> (GPU)	NVIDIA GeForce RTX 4090 Laptop GPU
<i>Random Access Memory</i> (RAM)	64 GB, dual channel, DDR5
<i>Operating System</i> (OS)	Windows 11 Home
Arsitektur	64-Bit

Tabel 4.2 Spesifikasi perangkat lunak

Bahasa Pemrograman	Python 3.11.5
<i>Integrated Development Environment</i> (IDE)	Visual Studio Code
<i>Package</i>	<ul style="list-style-type: none"> • PyTorch 2.3.1+cu121 • CUDA 12.1 • cuDNN 8.9 • numpy>=1.17 • opencv-python • Pillow • pyyaml • requests • scikit-image • scipy • tb-nightly • torch>=1.7 • torchvision

4.3 Hasil Penelitian

Pada sub bab ini, akan dijelaskan mengenai hasil penelitian tesis yang telah dilakukan.

4.3.1 Hasil *Pre-processing* Data

Tahap *pre-processing* dilakukan dengan mempersiapkan citra beresolusi tinggi (HR) menjadi beberapa versi beresolusi rendah (LR) melalui proses *downscaling*. Eksperimen menggunakan faktor *downscaling* ($\times 0.80$, $\times 0.66$, $\times 0.57$, $\times 0.50$, $\times 0.33$, $\times 0.25$). Gambar 3.4 memperlihatkan diagram keseluruhan tahapan *pre-processing*, termasuk bagaimana citra HR dari DF2K diproses melalui metode interpolasi *bicubic* sehingga menghasilkan beragam versi citra beresolusi rendah (LR). Setiap versi LR kemudian dipasangkan kembali dengan citra HR aslinya untuk membentuk satu set data pelatihan (*HR-LR pairs*), disertai pemeriksaan resolusi dan pembuatan berkas metadata.

Sebagai contoh, sebuah citra HR berukuran (2040×1356) jika di-*downscale* dengan faktor $\times 0.66$, akan menghasilkan citra LR berukuran (1360×904). Hasil *downscaling* ini diperiksa kecocokannya apakah rasio dan resolusi telah sesuai, kemudian disimpan dalam folder LRbicx1.5. Folder yang terbentuk selanjutnya berisi pasangan citra HR dan LR, serta berkas metadata.txt yang mencatat nama pasangan file untuk keperluan konfigurasi pelatihan. Pada akhir proses, setiap citra HR akan memiliki enam versi LR yang masing-masing mewakili faktor degradasi berbeda seperti yang diilustrasikan pada Gambar 4.2.



Gambar 4.2 Hasil pre-processing citra HR yang telah di-*downscale*

4.3.2 Modifikasi Integrasi Arsitektur Model

Modifikasi integrasi arsitektur model ini bertujuan menggabungkan keunggulan dari dua pendekatan utama, yaitu *Enhanced Group Convolutional Neural Network* (EGCNN) untuk ekstraksi fitur lokal yang efisien dan mekanisme *self-attention* berbasis *Performer* untuk menangkap hubungan jangka panjang antar piksel. Di bawah ini adalah rincian terperinci dari kode arsitektur EGCNN+ET, dari fungsi utama hingga fungsi pendukungnya, dengan kutipan kode dan penjelasan.

Inti dari model ini dibangun oleh fungsi `build_egcnn_performer()`. Fungsi ini berfungsi sebagai *factory* yang membangun dan mengembalikan contoh model *hybrid*, bernama `EGCNN_Performer`. Model ini mengintegrasikan beberapa komponen utama: modul *mean shift* untuk normalisasi input dan restorasi output, lapisan *head convolutional* untuk mengekstrak fitur awal, tumpukan MFC (*Multi-Feature Channel*) untuk ekstraksi fitur lokal, *Performer* blok untuk menangkap konteks global melalui *efficient self-attention*, dan akhirnya blok *upsample* yang memperbesar peta fitur ke resolusi tinggi yang diinginkan sebelum menghasilkan output akhir. Kode fungsi utama ini bisa dilihat pada Kode 4.1 sebagai berikut:

```
def build_egcnn_performer(...):
    class EGCNN_Performer(nn.Module):
        def __init__(self):
            super().__init__()
            # Mean Shift
            self.sub_mean = MeanShift((0.4488, 0.4371, 0.4040), sub=True)
            self.add_mean = MeanShift((0.4488, 0.4371, 0.4040), sub=False)
            # Head
            self.head = nn.Sequential(...)
            # MFC Stack
            self.mfc_stack = nn.Sequential(*[MFCModule(features, features)])
            # Post-MFC Convolution
            self.post_mfc = nn.Sequential(...)
            # Performer Stack
            self.performer_stack = nn.Sequential(*[PerformerBlock(...)])
            # Upsample
            self.upsample = UpsampleBlock(features, scale, multi_scale, group)
            # Final Layers
            self.relu = nn.ReLU(inplace=False)
            self.final_conv = nn.Conv2d(...)
    return EGCNN_Performer
```

Kode 4.1 Fungsi utama kode integrasi EGCNN+ET

Kelas MeanShift digunakan dua kali, pertama-tama untuk mengurangi rata-rata dari gambar input (normalisasi) dan kemudian untuk menambahkannya kembali ke gambar yang direkonstruksi. Implementasi ini langsung menggunakan konvolusi 1×1 yang diinisialisasi dengan matriks identitas dan bias yang diatur ke nilai rata-rata negatif atau positif. Implementasi kode ini bisa dilihat pada Kode 4.2 sebagai berikut:

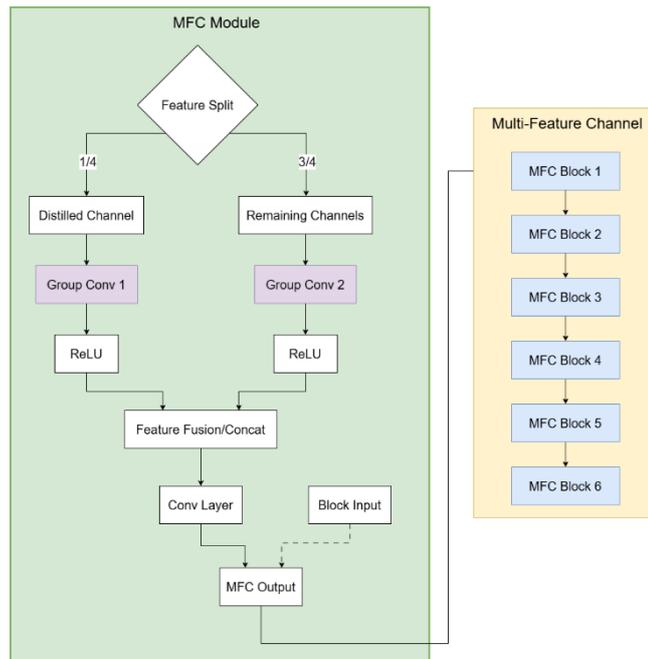
```
class MeanShift(nn.Module):
    def __init__(self, mean_rgb, sub):
        super(MeanShift, self).__init__()
        sign = -1 if sub else 1
        r = mean_rgb[0] * sign
        g = mean_rgb[1] * sign
        b = mean_rgb[2] * sign
        self.shifter = nn.Conv2d(3, 3, 1, 1, 0)
        self.shifter.weight.data = torch.eye(3).view(3, 3, 1, 1)
        self.shifter.bias.data = torch.Tensor([r, g, b])
        for params in self.shifter.parameters():
            params.requires_grad = False
```

Kode 4.3 Modul MeanShift

Model ini kemudian menggunakan setumpuk MFCModule (dikonfigurasi oleh num_mfc_blocks) untuk mengekstrak fitur lokal. Setiap modul MFC menerapkan serangkaian konvolusi, membagi fitur menjadi saluran “distilled” dan “remaining”, dan kemudian menggabungkannya melalui koneksi residual. Desain ini membantu jaringan untuk fokus pada pola lokal yang penting. Implementasi kode ini bisa dilihat pada Kode 4.3 dan ilustrasi gambaran proses ini bisa dilihat pada Gambar 4.3.

```
class MFCModule(nn.Module):
    def __init__(self, in_channels, out_channels, groups=1):
        super(MFCModule, self).__init__()
        # ... (initialization of convolution layers and parameters)
    def forward(self, x):
        out1_c = self.conv1_1(x)
        dit1, remain1 = torch.split(out1_c, (self.distilled_channels,
self.remaining_channels), dim=1)
        out1_r = self.ReLU(remain1)
        # ... (further splitting and residual summing)
        out6_r = x + self.conv1_6(self.conv1_5(out_r))
        return out6_r
```

Kode 4.6 Ekstraksi fitur lokal melalui modul MFC



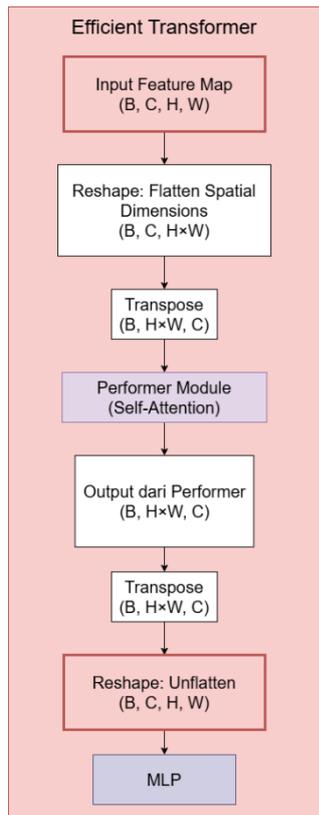
Gambar 4.3 Diagram MFC Module

Setelah fitur lokal diekstraksi, model melewati output melalui tumpukan blok Performer (dikonfigurasi oleh `num_performer_blocks`). Performerblock membungkus modul performer eksternal yang dirancang untuk bekerja dengan urutan. Di sini, peta fitur dengan bentuk (B, C, H, W) dibentuk kembali menjadi $(B, H*W, C)$, diproses oleh Performer (yang menggunakan *efficient self-attention*), dan kemudian dibentuk kembali ke dimensi aslinya. Implementasi kode ini bisa dilihat pada Kode 4.4 dan rangkaian ilustrasi proses bisa dilihat pada Gambar 4.4.

```
class PerformerBlock(nn.Module):
    def __init__(self, ...):
        super().__init__()
        # Performer expects shape (B, N, D), di mana N = panjang sekuens, D = dim
        self.performer = Performer(
            dim=dim, depth=depth, heads=heads, dim_head=dim_head, ff_mult=ff_mult,
            nb_features=nb_features
        )

    def forward(self, x):
        B, C, H, W = x.shape
        x_reshape = x.view(B, C, H*W).permute(0, 2, 1) # (B, N, C)
        out = self.performer(x_reshape) # (B, N, C)
        out = out.permute(0, 2, 1).view(B, C, H, W)
        return out
```

4.8 Ekstraksi fitur global melalui blok Performer



Gambar 4.4 Proses integrasi dengan *Performer*

Akhirnya, model menggunakan blok *upsample* untuk skala peta fitur ke resolusi target. *UpsampleBlock* adalah pembungkus yang menggunakan *pixelshuffle* (untuk skala integer) atau interpolasi *bicubic* (untuk skala pecahan). Modul ini memungkinkan arsitektur menjadi fleksibel dalam menangani berbagai faktor penskalaan. Implementasi kode ini bisa dilihat pada Kode 4.5.

```

class UpsampleBlock(nn.Module):
    def forward(self, x, scale):
        if self.multi_scale:
            if scale == 2:
                return self.up2(x)
            elif scale == 3:
                return self.up3(x)
            elif scale == 4:
                return self.up4(x)
            else:
                return F.interpolate(x, scale_factor=scale, mode='bicubic',
align_corners=False)
        else:
            return self.up(x)
  
```

Kode 4.9 Modul *Upsampling*

Di dalam kelas `EGCNN_Performer` (didefinisikan dalam `build_egcnn_performer`), metode `forward` mengatur seluruh *pipeline*: input gambar pertama kali dinormalisasi menggunakan `sub_mean`, melewati konvolusi kepala dan tumpukan MFC, kemudian disempurnakan oleh tumpukan `Performer` untuk mengintegrasikan informasi global. Setelah itu, blok *upsampling* meningkatkan resolusi, dan lapisan konvolusi akhir bersama dengan `Add_mean` mengembalikan gambar ke rentang warna aslinya. Implementasi kode ini bisa dilihat pada Kode 4.6.

```
def forward(self, x):
    x = self.sub_mean(x)          # Normalize input
    x = self.head(x)             # Extract initial features
    x = self.mfc_stack(x)        # Extract local features via MFC modules
    x = self.post_mfc(x)         # Additional processing after MFC
    x = self.performer_stack(x)  # Integrate global context with Performer blocks
    x = self.upsample(x, scale=scale) # Upsample to target resolution
    x = self.relu(x)
    out = self.final_conv(x)     # Final reconstruction to 3 channels
    out = self.add_mean(out)     # Restore original mean
    return out
```

Kode 4.11 Metode *forward* untuk fungsi utama

Dengan desain modular ini, fungsi utama `build_egcnn_performer()` bisa membangun model dengan menumpuk modul secara berurutan yang menangani normalisasi, ekstraksi fitur lokal, pemodelan konteks global, dan *upsampling*. Setiap fungsi `Meanshift`, `MFCModule`, `Performerblock`, dan `Upsampleblock` berkontribusi secara khusus pada bagian-bagian berbeda dari proses rekonstruksi citra. Desain modular ini tidak hanya membuat kode dapat dikelola dan diperpanjang tetapi juga memungkinkan penyesuaian masing-masing komponen untuk mengoptimalkan kinerja SISR secara keseluruhan.

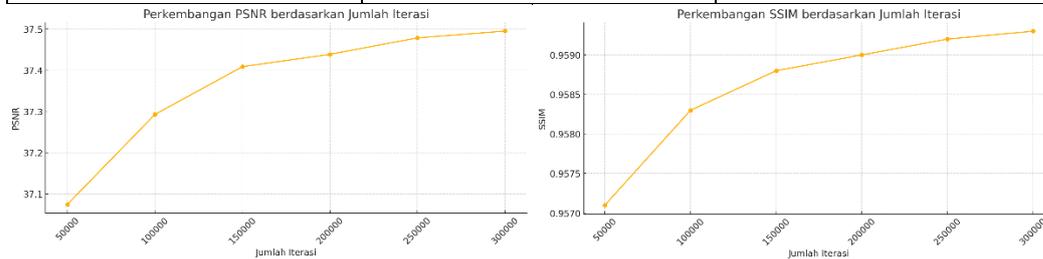
4.3.2.1 Eksperimen terhadap Jumlah Iterasi

Pada eksperimen ini, dilakukan pengujian pengaruh lamanya pelatihan pada performa model `EGCNN+ET` (6-2) dengan skala pembesaran $\times 2$ menggunakan dataset `Set5` sebagai validasi. Eksperimen ini menggunakan *mini-batch* dengan jumlah 4. Hasil pengujian diukur pada enam milestone iterasi, yaitu 50.000, 100.000, 150.000, 200.000, 250.000, dan 300.000 iterasi yang menunjukkan perkembangan kualitas rekonstruksi citra berdasarkan metrik kuantitatif, yaitu *Peak*

Signal-to-Noise Ratio (PSNR) dan *Structural Similarity Index* (SSIM), serta waktu pelatihan kumulatif. Tabel 4.3 berikut merangkum hasil yang diperoleh dan Gambar 4.5 menunjukkan grafik perkembangan metrik:

Tabel 4.3 Eksperimen model EGCNN+ET (6-2) terhadap jumlah iterasi

Jumlah Iterasi	PSNR/SSIM	Running Time (detik)
50.000	37,0744/0,9571	3.660
100.000	37,3215/ 0,9587	7.320
150.000	37,4085/0,9588	10.980
200.000	37,4358/ 0,9591	14.640
250.000	37,4784/0,9592	18.338
300.000	37,4748/ 0,9592	22.036

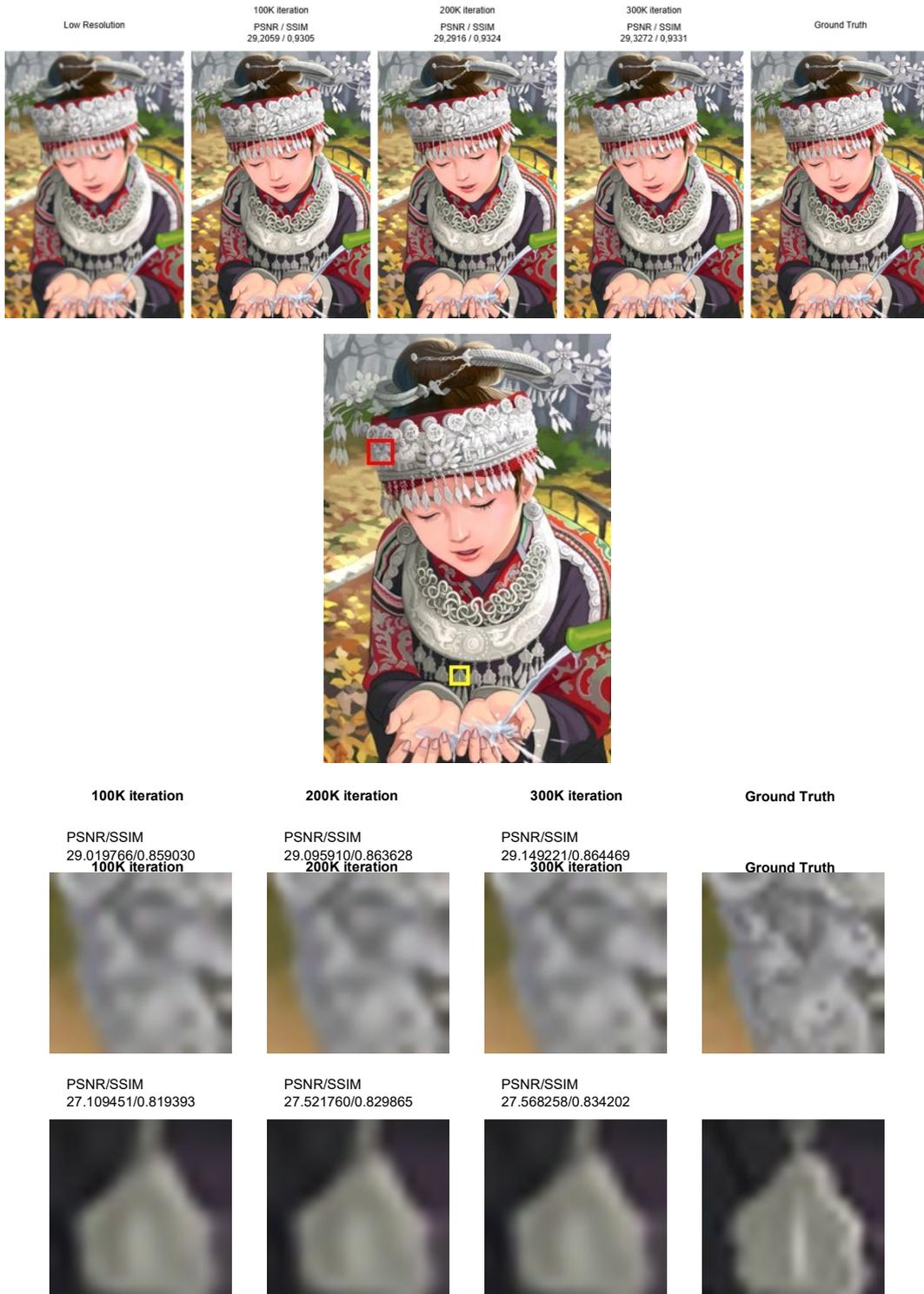


Gambar 4.5 Perkembangan metrik PSNR dan SSIM berdasarkan jumlah iterasi

Dari tabel dan grafik tersebut, terlihat bahwa pada 50.000 iterasi, model menghasilkan PSNR sebesar 37,0744 dB dan SSIM sebesar 0,9571 dengan waktu pelatihan kumulatif sekitar 3.660 detik. Pada milestone berikutnya di 100.000 iterasi, performa meningkat cukup signifikan, mencapai PSNR sebesar 37,3215 dB dan SSIM sebesar 0,9587 dengan total waktu pelatihan sekitar 7.320 detik. Selanjutnya, pada iterasi ke-150.000, terjadi peningkatan moderat dengan PSNR mencapai 37,4085 dB dan SSIM sebesar 0,9588 dalam waktu kumulatif sekitar 10.980 detik.

Peningkatan iterasi ke-200.000 memberikan perbaikan yang cukup signifikan, dengan nilai PSNR meningkat menjadi 37,4358 dB dan SSIM mencapai 0,9591, sementara waktu kumulatif meningkat menjadi 14.640 detik. Pada tahap 250.000 iterasi, performa model meningkat lebih lanjut menjadi PSNR 37,4784 dB dan SSIM sebesar 0,9592 dengan waktu pelatihan kumulatif sekitar 18.338 detik. Pada milestone terakhir (300.000 iterasi), model mencapai PSNR sebesar 37,4748

dB dan SSIM 0,9592 dengan waktu pelatihan kumulatif mencapai 22.036 detik. Sebagai gambaran hasil, disajikan gambar perkembangan hasil rekonstruksi $\times 2$ pada salah satu gambar uji.



Gambar 4.6 Perbandingan hasil rekonstruksi $\times 2$ berdasarkan jumlah iterasi

Hasil ini menunjukkan bahwa peningkatan jumlah iterasi pelatihan memberikan perbaikan performa yang konsisten, namun peningkatan tersebut mulai melambat setelah 200.000 iterasi, mengindikasikan bahwa model mendekati titik konvergensi optimal. Dengan demikian, rentang antara 200.000 hingga 300.000 iterasi dapat dianggap sebagai rentang optimal di mana kualitas rekonstruksi citra sudah sangat baik, namun peningkatan performa mulai relatif kecil dibandingkan peningkatan waktu pelatihan. Eksperimen ini memberikan wawasan penting dalam menentukan jumlah iterasi ideal untuk mendapatkan keseimbangan terbaik antara kualitas hasil rekonstruksi citra dan efisiensi waktu pelatihan.

4.3.2.2 Eksperimen terhadap Jumlah Blok

Pada eksperimen terhadap jumlah blok, dilakukan beberapa variasi konfigurasi pada arsitektur *hybrid* EGCNN+ET dengan tujuan untuk menemukan keseimbangan optimal antara ekstraksi fitur lokal dan penangkapan hubungan global. Pada konfigurasi pertama, yaitu EGCNN+ET (4-2) yang menggunakan empat blok EGCNN dan dua blok *Efficient Transformer* (ET). Dengan mengurangi jumlah blok EGCNN, model menjadi lebih ringan sehingga jumlah parameter dan waktu inferensi berkurang, meskipun hal ini dapat berpotensi mengurangi kemampuan model dalam mengekstraksi detail lokal pada citra. Pada konfigurasi kedua, EGCNN+ET (6-2) yang menjadi *baseline*, model menggunakan enam blok EGCNN dan dua blok ET dengan tujuan memberikan keseimbangan antara ekstraksi fitur lokal yang mendalam dan pemodelan konteks global melalui ET. Konfigurasi ini diharapkan mampu menghasilkan kualitas rekonstruksi yang optimal dengan efisiensi komputasi yang memadai. Selanjutnya, konfigurasi EGCNN+ET (6-4) mempertahankan enam blok EGCNN namun menambah jumlah blok ET menjadi empat, sehingga meningkatkan kemampuan model dalam menangkap hubungan jangka panjang antar piksel secara lebih menyeluruh.

Tabel 4.4 menunjukkan hasil kuantitatif (PSNR/SSIM) pada keempat dataset pengujian dengan faktor skala *integer* ($\times 2$, $\times 3$, $\times 4$). Berdasarkan hasil yang diperoleh, terlihat bahwa setiap varian model EGCNN+ET (yaitu (4-2), (6-2), dan (6-4)) memperlihatkan kinerja yang kompetitif pada ketiga faktor skala *integer* di empat dataset pengujian (Set5, Set14, BSD100, dan Urban100). Secara umum,

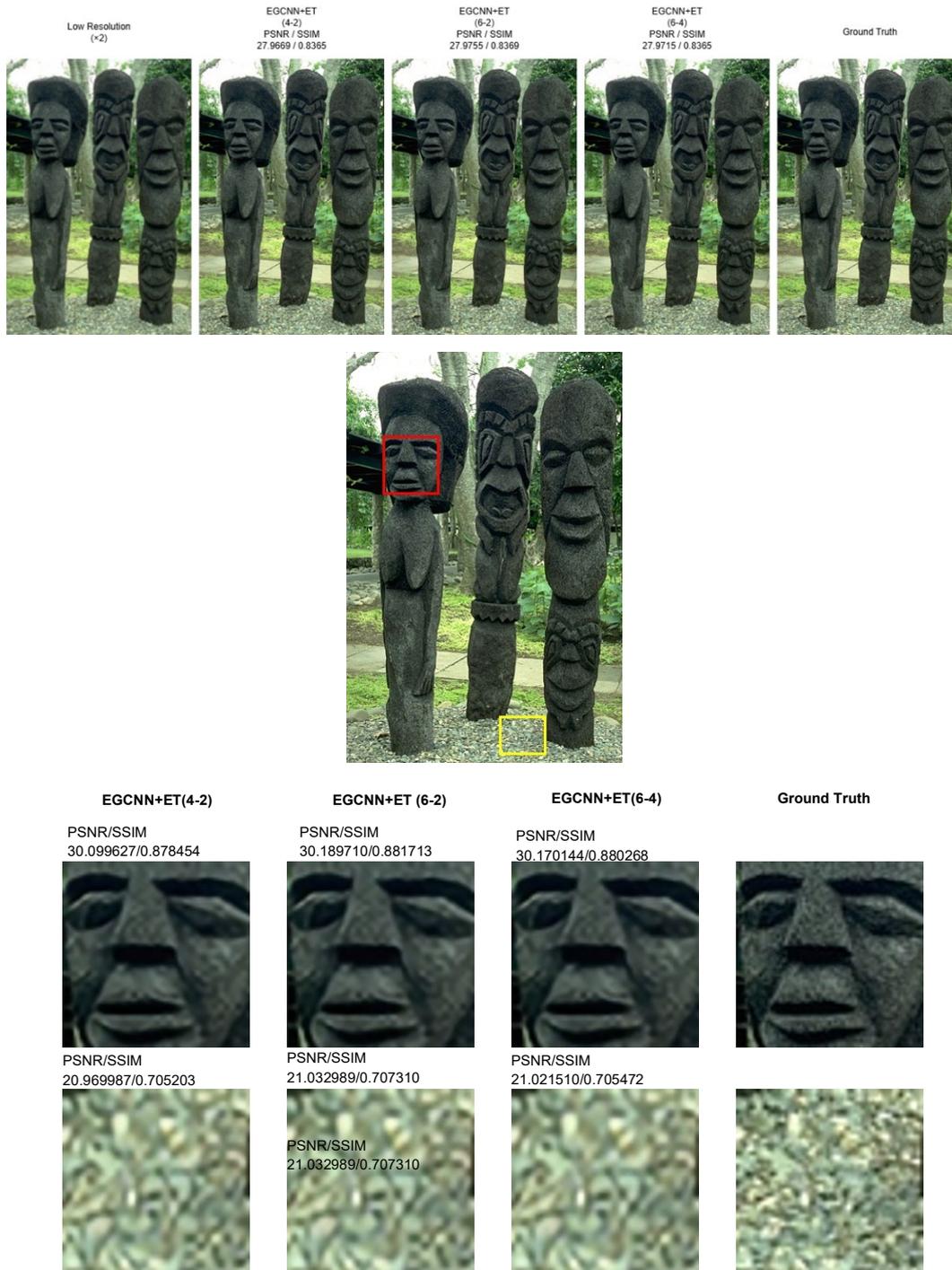
EGCNN+ET (6-2) cenderung meraih nilai PSNR dan SSIM yang sedikit lebih tinggi dibanding (4-2) pada hampir semua skenario, sementara (6-4) biasanya berada di rentang nilai yang sangat mirip dengan (6-2). Meski perbedaan angka kadang kecil, pola ini menegaskan bahwa penambahan jumlah blok EGCNN membantu model menangkap lebih banyak fitur lokal, sedangkan peningkatan blok ET meskipun menambah kapasitas global *attention* tidak selalu menghasilkan lompatan signifikan. (6-2) umumnya mengungguli (4-2) yang menegaskan pentingnya ekstraksi fitur lokal yang lebih dalam. Penambahan blok EGCNN menambah kapasitas model untuk mempelajari pola spasial, walau menambah jumlah parameter dan waktu pelatihan. Perbandingan (6-2) dan (6-4) menunjukkan bahwa menambah blok ET (menjadi 4) tidak selalu berujung peningkatan signifikan pada PSNR/SSIM. Pada dataset tertentu, (6-4) hanya terpaut sangat tipis dari (6-2) yang dapat menandakan bahwa pada jumlah blok EGCNN tertentu, *bottleneck* bukan lagi di kapasitas global *attention*.

Tabel 4.4 Hasil kuantitatif tiga variasi arsitektur EGCNN+ET pada skala *integer*

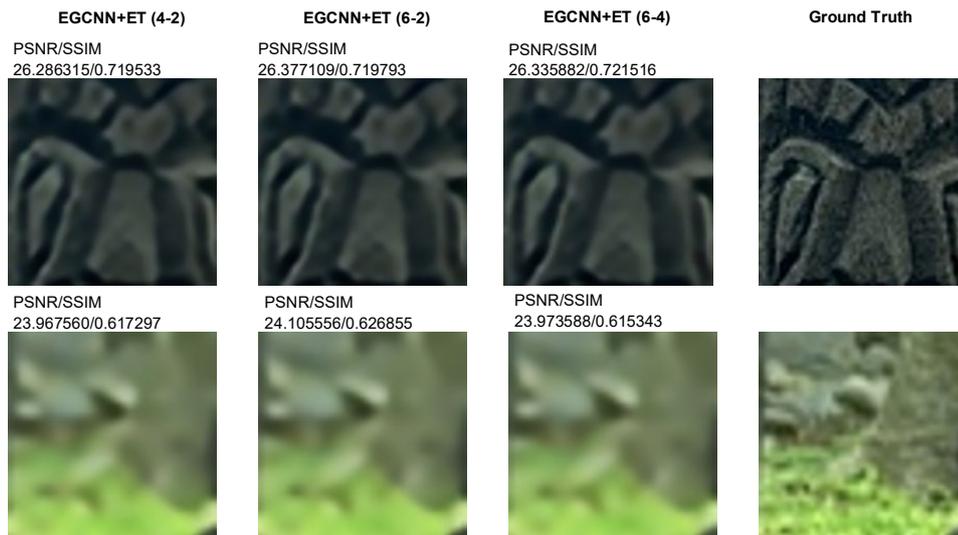
Model	SR	Set5	Set14	BSD100	Urban100
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
EGCNN+ET (4-2)	×2	37.53/0.9592	33.10/0.9135	33.02/0.9164	30.90/0.9159
EGCNN+ET (6-2)		37.58/0.9593	33.12/0.9137	33.05/0.9165	30.97/0.9165
EGCNN+ET (6-4)		37.56/0.9593	33.11/0.9137	33.03/0.9165	30.96/0.9166
EGCNN+ET (4-2)	×3	33.89/0.9234	29.95/0.8347	29.12/0.8069	27.98/0.8485
EGCNN+ET (6-2)		33.96/0.9241	30.01/0.8358	29.15/0.8078	28.11/0.8517
EGCNN+ET (6-4)		33.94/0.9236	29.99/0.8354	29.14/0.8071	28.06/0.8502
EGCNN+ET (4-2)	×4	31.69/0.8884	28.26/0.7736	27.77/0.7431	25.44/0.7635
EGCNN+ET (6-2)		31.77/0.8901	28.31/0.7748	27.79/0.7441	25.51/0.7662
EGCNN+ET (6-4)		31.75/0.8892	28.28/0.7743	27.78/0.7436	25.47/0.7651

Dari hasil di atas, EGCNN+ET (6-2) menampilkan keseimbangan optimal antara kedalaman EGCNN dan jumlah blok ET. (6-4) tidak menunjukkan perbaikan mencolok dibanding (6-2), sementara (4-2) masih tertinggal tipis pada sebagian

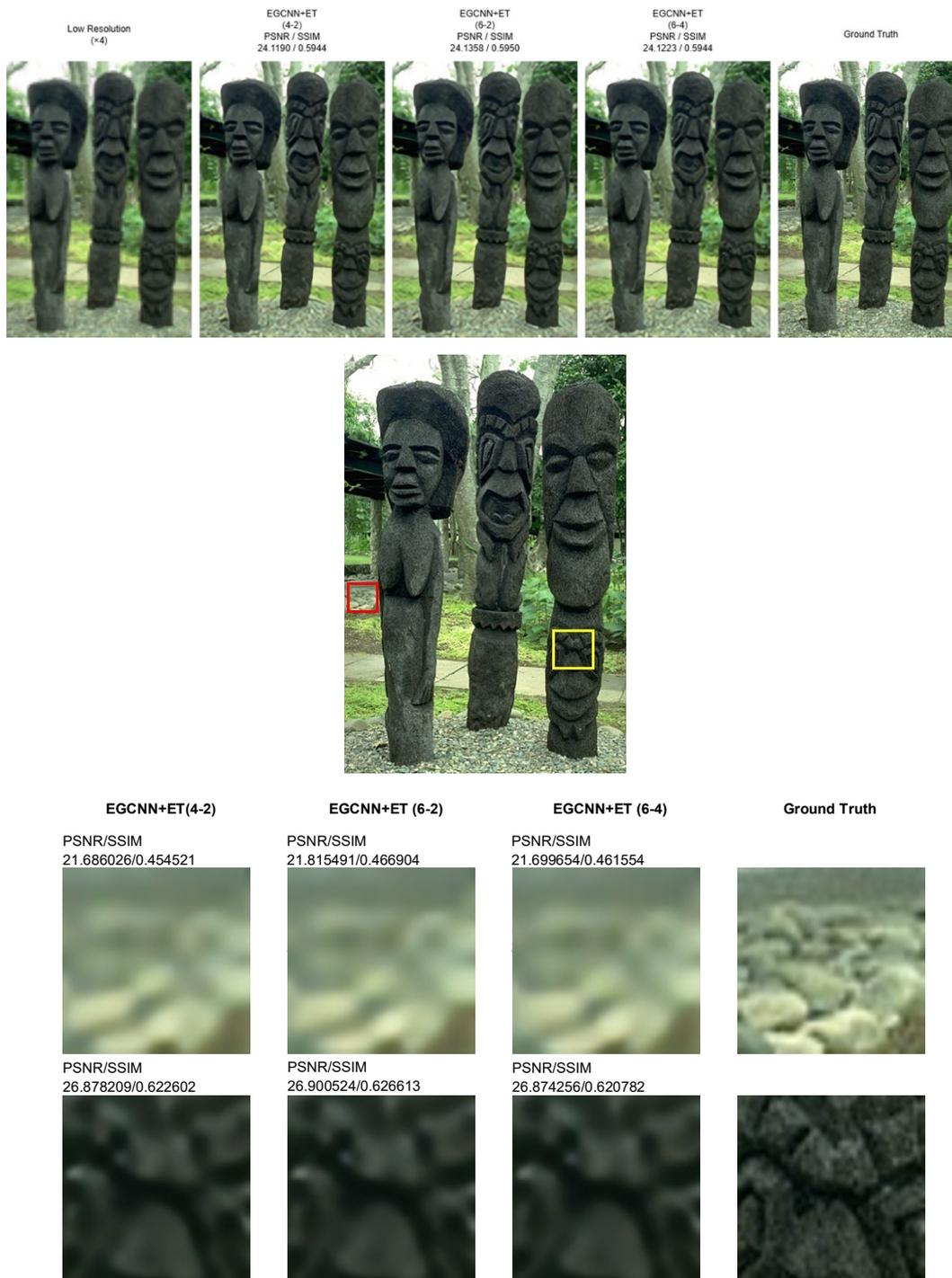
besar skenario. Artinya, menambah banyak blok ET mungkin tidak selalu sebanding dengan *overhead* komputasi tambahan, sedangkan mengurangi blok EGCNN menghemat parameter tetapi berpotensi menurunkan kualitas hasil rekonstruksi. Berikut ini adalah hasil perbandingan visual antara tiga arsitektur pada salah satu data uji yang menunjukkan perbedaan kualitas rekonstruksi.



Gambar 4.7 Perbandingan hasil rekonstruksi $\times 2$ pada tiga variasi arsitektur



Gambar 4.8 Perbandingan hasil rekonstruksi ×3 pada tiga variasi arsitektur



Gambar 4.9 Perbandingan hasil rekonstruksi $\times 4$ pada tiga variasi arsitektur

4.3.2.3 Eksperimen terhadap *Hyperparameter*

Pada sub-bab ini, dilakukan serangkaian eksperimen untuk mengevaluasi pengaruh *hyperparameter* pada modul *Performer* dalam arsitektur *hybrid* EGCNN+ET. Fokus utama dari eksperimen ini adalah untuk mengetahui sejauh mana perubahan pada parameter internal seperti *depth* dan jumlah *attention heads*

dalam *Performer* dapat meningkatkan kualitas rekonstruksi citra. Konfigurasi dasar (*baseline*) model menggunakan 6 *MFC blocks* dan 2 *Performer blocks* dengan pengaturan $\text{performer_depth} = 1$, $\text{performer_heads} = 4$, $\text{performer_dim_head} = 16$, dan $\text{performer_ff_mult} = 4$, sementara komponen lain seperti *mean shift* dan *upsample block* dipertahankan konstan untuk memastikan perbandingan yang adil.

Konfigurasi dasar (Eksperimen 1) memuat enam blok MFC dan dua blok *Performer* dengan kedalaman *self-attention* satu *depth* serta empat *attention heads* ($\text{performer_depth} = 1$, $\text{performer_heads} = 4$). Pada Eksperimen 2, kedalaman ditingkatkan menjadi dua *depth* ($\text{performer_depth} = 2$) tanpa mengubah jumlah *head*, sementara Eksperimen 3 menggandakan jumlah *head* menjadi delapan ($\text{performer_heads} = 8$) sambil mengembalikan kedalaman ke satu *depth*, sedangkan Eksperimen 4 mengombinasikan kedua peningkatan tersebut, yaitu kedalaman dua *depth* dan delapan *head*. Tabel 4.5 menyajikan konfigurasi eksperimen yang dirancang.

Tabel 4.5 Konfigurasi eksperimen *hyperparameter* pada EGCNN+ET (6-2)

Konfigurasi	performer_depth	performer_heads	Keterangan
Eksperimen 1 (<i>Baseline</i>)	1	4	Konfigurasi dasar
Eksperimen 2	2	4	Meningkatkan <i>depth</i> pada <i>Performer</i>
Eksperimen 3	1	8	Meningkatkan jumlah <i>attention heads</i>
Eksperimen 4	2	8	Mengombinasikan kedua peningkatan

Konfigurasi ini memungkinkan untuk melihat pengaruh peningkatan *depth* dan jumlah *head* pada modul *Performer* terhadap kinerja model, sehingga perbandingan antara peningkatan kapasitas global *self-attention* dan kompleksitas komputasi dapat dianalisis dengan lebih jelas. Hasil dari eksperimen ini bisa dilihat pada Tabel 4.6 sebagai berikut.

Tabel 4.6 Hasil Eksperimen *hyperparameter* pada EGCNN+ET (6-2)

Konfigurasi	Param (M)	Set5 ×4 PSNR/SSIM
Eksperimen 1 (<i>Baseline</i>)	1.596	31.77/0.8901
Eksperimen 2	1.696	31.76/0.8899
Eksperimen 3	1.631	31.75/0.8894
Eksperimen 4	1.763	31.74/0.8900

Hasil pengujian pada *benchmark* Set5 dengan faktor pembesaran ×4 menunjukkan bahwa keempat model berperforma hampir identik. *Baseline* dengan 1,596 juta parameter meraih PSNR 31,77 dB dan SSIM 0,8901, sedangkan penambahan kedalaman pada Eksperimen 2 (1,696 juta parameter) justru menurunkan PSNR tipis menjadi 31,76 dB dan SSIM ke 0,8899. Peningkatan jumlah *head* pada Eksperimen 3 (1,631 juta parameter) juga tidak memberikan lonjakan, PSNR sedikit turun ke 31,75 dB dengan SSIM 0,8894. Bahkan konfigurasi paling besar pada Eksperimen 4 (1,763 juta parameter) menghasilkan PSNR terendah, 31,74 dB, meskipun SSIM kembali setara *baseline* di 0,8900. Selisih antara seluruh konfigurasi berkisar 0,03 dB untuk PSNR dan 0,0007 untuk SSIM. Angka yang secara praktis nyaris tak terdeteksi dalam kualitas visual.

Temuan ini menegaskan adanya titik jenuh (*diminishing return*) ketika kapasitas *self-attention* diperbesar pada skala pembesaran yang tinggi. Penambahan *depth* dan *head* menaikkan kompleksitas komputasi 10–14% tanpa keuntungan berarti. Tambahan blok atau perubahan *hyperparameter* tertentu tidak selalu memberikan kontribusi yang signifikan. Sebaliknya, peningkatan parameter kadang justru meningkatkan risiko *overhead* komputasi tanpa diikuti lonjakan performa yang setara. Dengan demikian, *baseline* tetap menjadi pilihan paling efisien.

4.3.3 Modifikasi Modul *Upsampling* untuk Faktor Skala Pecahan

Dalam rangka menguji kemampuan model dalam menangani variasi degradasi citra yang lebih halus, dilakukan eksperimen modifikasi modul *upsampling* untuk faktor skala pecahan. Faktor skala pecahan yang digunakan dalam eksperimen ini adalah $\times\frac{5}{4}$, $\times\frac{3}{2}$, dan $\times\frac{7}{4}$. Modul *upsampling* yang ada pada *pipeline* awalnya dirancang untuk faktor skala integer, sehingga perlu dilakukan modifikasi agar dapat mengakomodasi transformasi citra dengan skala pecahan.

Modifikasi dilakukan dengan mendesain sebuah kelas yang fleksibel, yaitu *FlexibleUpsampleBlock* yang dapat menangani baik faktor skala integer maupun non-integer. Dalam kelas ini, ketika nilai skala yang diberikan termasuk dalam himpunan {2, 3, 4, 8} yang merupakan skala integer standar, maka modul akan mengonfigurasi serangkaian lapisan konvolusi yang diikuti oleh *PixelShuffle*. *PixelShuffle* digunakan untuk meningkatkan resolusi dengan mengatur ulang elemen *tensor* secara terstruktur sehingga menghasilkan citra dengan dimensi spasial yang diperbesar secara tepat. Untuk kasus ketika skala bukan merupakan bilangan bulat, modul tidak membangun sebuah jaringan konvolusional khusus, sebaliknya pada fungsi *forward* modul, dilakukan proses *upsampling* dengan menggunakan fungsi interpolasi (`F.interpolate`) dengan mode *'bicubic'*, yang memungkinkan peningkatan resolusi dengan faktor skala pecahan secara kontinu. Implementasi dari modifikasi modul *upsampling* ini bisa dilihat pada Kode 4.7.

```
class FlexibleUpsampleBlock(nn.Module):
    def __init__(self, n_channels, scale, group=1, mode='bicubic'):
        super(FlexibleUpsampleBlock, self).__init__()
        self.scale=scale, self.mode=mode, self.group=group, self.body=None
        if scale in [2, 3, 4, 8]:
            modules = []
            if scale == 2 or scale == 4 or scale == 8:
                for _ in range(int(math.log(scale, 2))):
                    modules += [
                        nn.Conv2d(...),
                        nn.PixelShuffle(2)]
            elif scale == 3:
                modules += [
                    nn.Conv2d(...),
                    nn.PixelShuffle(3)]
            self.body = nn.Sequential(*modules)
        else:
            self.body = None

    def forward(self, x):
        if self.scale in [2, 3, 4, 8]:
            out = self.body(x) # PixelShuffle path
        else:
            out = F.interpolate(
                x, scale_factor=self.scale, mode=self.mode, align_corners=False)
        return out
```

Kode 4.13 Modifikasi modul *upsampling* untuk faktor skala pecahan

4.3.4 Evaluasi Kuantitatif

Pada sub-bab ini, disajikan analisis numerik untuk menilai kinerja model SISR yang diusulkan dengan menggunakan metrik kuantitatif utama yaitu *Peak Signal-to-Noise Ratio* (PSNR) dan *Structural Similarity Index* (SSIM). Evaluasi kuantitatif ini bertujuan memberikan gambaran objektif tentang seberapa efektif model dalam merekonstruksi citra berkualitas tinggi dibandingkan dengan citra *ground truth*, serta untuk membandingkan kinerja model usulan dengan model-model pembanding yaitu EGCNN *baseline* dan SwinIR. Pengukuran dilakukan pada dataset standar pengujian seperti Set5, Set14, BSD100, dan Urban100.

4.3.4.1 Hasil Kuantitatif Skala Integer

Pada skala integer ($\times 2$, $\times 3$, $\times 4$), hasil kuantitatif yang ditampilkan dalam Tabel 4.8 dan Tabel 4.9 menunjukkan bahwa integrasi *Efficient Transformer* (ET) ke dalam EGCNN (*Enhanced Group Convolutional Neural Network*) secara konsisten meningkatkan nilai *Peak Signal-to-Noise Ratio* (PSNR) dan *Structural Similarity Index* (SSIM) dibandingkan *baseline* EGCNN.

EGCNN+ET secara konsisten mengungguli EGCNN (*baseline*) di semua dataset dengan menunjukkan peningkatan sekitar +0.9 hingga +1.2 dB PSNR. Menariknya, EGCNN+ET juga cenderung melebihi SwinIR pada Set5, Set14, dan BSD100 dengan margin tipis yang menandakan bahwa modul ET mampu memanfaatkan fitur lokal EGCNN dengan efisien.

Tabel 4.7 Keterangan model untuk skala integer

Model	Penjelasan
1	Sudah dilatih untuk <i>upscale</i> $\times 2$ dari citra yang di- <i>downscale</i> 0.50
2	Sudah dilatih untuk <i>upscale</i> $\times 3$ dari citra yang di- <i>downscale</i> 0.33
3	Sudah dilatih untuk <i>upscale</i> $\times 4$ dari citra yang di- <i>downscale</i> 0.25

Tabel 4.8 Hasil PSNR seluruh arsitektur pada skala integer

Model	Data Testing	EGCNN (<i>baseline</i>)	EGCNN+ ET (4-2)	EGCNN+ ET (6-2)	EGCNN+ ET (6-4)	SwinIR
		PSNR				
1	Set5	36.66	37.53	37.58	37.56	36.79
	Set14	32.48	33.10	33.12	33.11	32.79
	BSD100	32.47	33.02	33.05	33.03	32.84
	Urban100	29.75	30.90	30.97	30.96	30.18
2	Set5	33.95	33.89	33.96	33.94	34.65
	Set14	30.00	29.95	30.01	29.99	30.62
	BSD100	29.15	29.12	29.15	29.14	29.58
	Urban100	28.08	27.98	28.11	28.06	29.58
3	Set5	31.73	31.69	31.77	31.75	32.55
	Set14	28.33	28.26	28.31	28.28	28.87
	BSD100	27.80	27.77	27.79	27.78	28.24
	Urban100	25.52	25.44	25.51	25.47	26.72

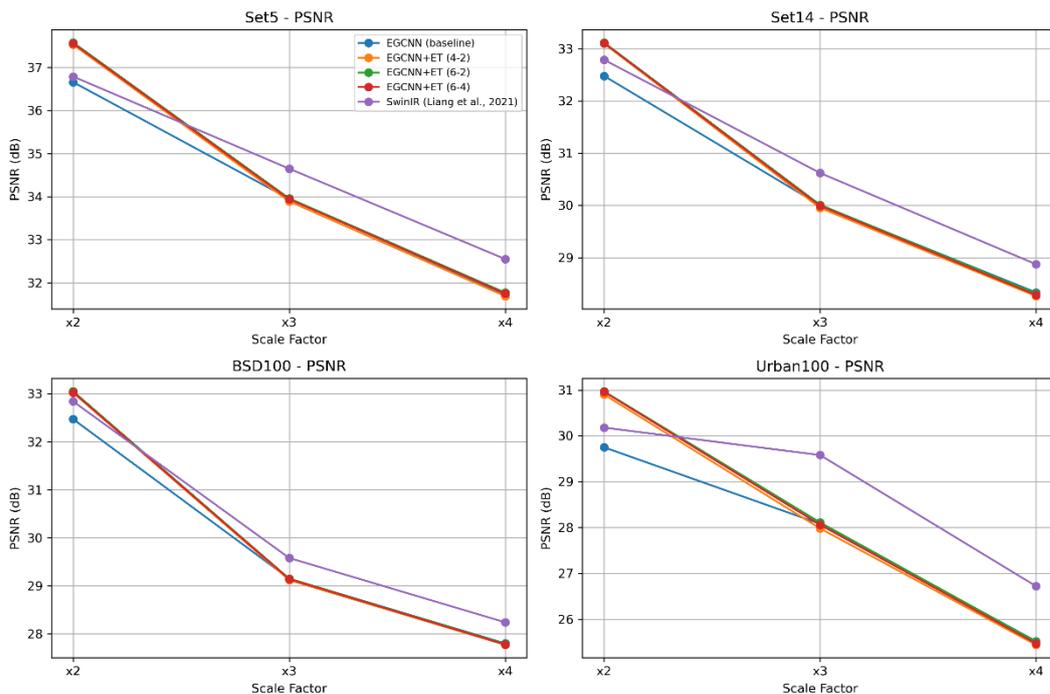
Tabel 4.9 Hasil SSIM seluruh arsitektur pada skala integer

Model	Data Testing	EGCNN (<i>baseline</i>)	EGCNN+ ET (4-2)	EGCNN+ ET (6-2)	EGCNN+ ET (6-4)	SwinIR
		PSNR				
1	Set5	0.9546	0.9592	0.9593	0.9593	0.9559
	Set14	0.9047	0.9135	0.9137	0.9137	0.9102
	BSD100	0.9059	0.9164	0.9165	0.9165	0.9108
	Urban100	0.9011	0.9159	0.9165	0.9164	0.9089
2	Set5	0.9239	0.9234	0.9241	0.9236	0.9298
	Set14	0.8359	0.8347	0.8358	0.8354	0.8484
	BSD100	0.8078	0.8069	0.8078	0.8071	0.8184
	Urban100	0.8509	0.8485	0.8517	0.8502	0.8803
3	Set5	0.8894	0.8884	0.8901	0.8892	0.9002
	Set14	0.7755	0.7736	0.7748	0.7743	0.7891
	BSD100	0.7442	0.7431	0.7441	0.7436	0.7589
	Urban100	0.7671	0.7635	0.7662	0.7651	0.8059

Pada skala $\times 2$ misalnya, EGCNN+ET (6-2) mencatat PSNR sekitar 37,58 dB di Set5, lebih tinggi daripada *baseline* EGCNN yang hanya meraih 36,66 dB. Peningkatan ini menandakan bahwa model usulan mampu memetakan detail piksel lebih akurat dan menjaga struktur citra lebih baik. Sementara itu, SwinIR (Liang et al., 2021) memiliki performa yang kompetitif, namun perbandingan di Set5 dan

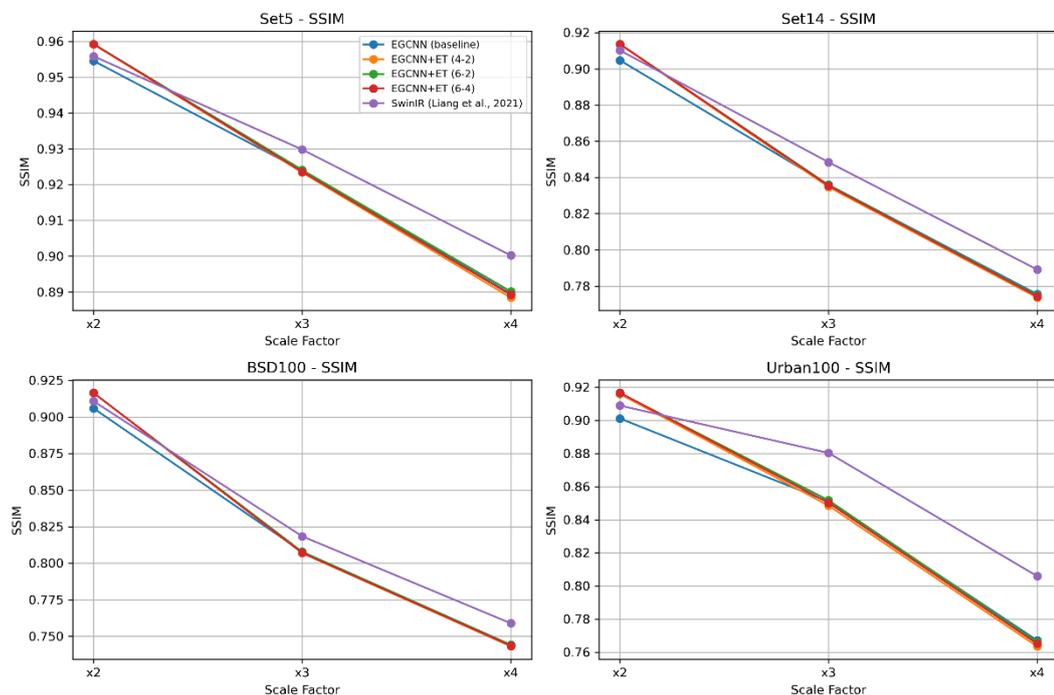
BSD100 memperlihatkan bahwa EGCNN+ET kerap menyaingi atau bahkan sedikit melebihi SwinIR dalam beberapa kasus yang mengindikasikan bahwa mekanisme *self-attention* berbasis ET efektif dalam menangkap konteks global tanpa menambah kompleksitas yang berlebihan. Skala $\times 2$ merupakan titik kuat bagi EGCNN+ET, memanfaatkan efisiensi EGCNN dan modul *self-attention* ET untuk meningkatkan PSNR/SSIM.

Ketika faktor skala ditingkatkan menjadi $\times 3$ dan $\times 4$, kinerja keseluruhan model, termasuk EGCNN+ET cenderung menurun karena tantangan rekonstruksi detail yang semakin besar. Meski demikian, EGCNN+ET tetap unggul atas EGCNN (*baseline*) yang menunjukkan peningkatan PSNR sekitar 0,1–0,5 dB dan kenaikan SSIM yang sejalan. SwinIR sering kali menempati posisi teratas di dataset yang memiliki tekstur lebih kompleks seperti Urban100 dengan memanfaatkan keunggulan mekanisme *shifted window attention* saat menghadapi pola spasial yang beragam. Namun, selisih performa antara SwinIR dan EGCNN+ET tidak terlalu besar sehingga EGCNN+ET masih dapat dianggap kompetitif. Berikut ini disajikan grafik PSNR dan SSIM pada Gambar 4.10 dan Gambar 4.11 untuk melihat tren nilai PSNR dan SSIM pada setiap faktor skala integer.



Gambar 4.10 Grafik nilai PSNR faktor skala integer pada keempat data uji

Dari grafik PSNR pada Gambar 4.10 maupun SSIM pada Gambar 4.11, terlihat bahwa kinerja semua model mengalami penurunan seiring meningkatnya faktor skala pembesaran (dari $\times 2$ ke $\times 4$) di keempat dataset (Set5, Set14, BSD100, dan Urban100). Hal ini menegaskan tantangan yang lebih besar ketika model harus merekonstruksi detail tekstur pada degradasi resolusi yang lebih tinggi. Namun, perbedaan performa antar model tetap dapat diamati, khususnya antara EGCNN (*baseline*), varian EGCNN+ET, dan SwinIR (Liang et al., 2021). Peningkatan yang diraih EGCNN+ET atas *baseline* EGCNN tetap konsisten sehingga menandakan bahwa modul *Efficient Transformer* (ET) membantu mempertahankan detail kritis meski beban rekonstruksi semakin berat. Sementara itu, SwinIR menonjol di area bertekstur tinggi terutama Urban100 karena mekanisme *shifted window attention*-nya dapat menyoroti berbagai bagian citra secara global tanpa kehilangan konteks lokal.



Gambar 4.11 Grafik nilai SSIM faktor skala integer pada keempat data uji

Untuk menganalisis efisiensi komputasi pada setiap model dilakukan pengukuran jumlah parameter dan waktu inferensi. Tabel 4.10 menyajikan perbandingan jumlah parameter dan waktu inferensi di antara model EGCNN (*baseline*), tiga varian EGCNN+ET, serta SwinIR. Data ini penting untuk menilai apakah peningkatan kualitas rekonstruksi yang dicapai melalui integrasi *Efficient*

Transformer sebanding dengan *overhead* komputasi yang ditimbulkan dan bagaimana model usulan bersaing dengan pendekatan *state-of-the-art* lain seperti SwinIR.

Tabel 4.10 Perbandingan jumlah parameter dan waktu inferensi pada semua model

Model	Param (M)	Set5 Inferensi (ms)
EGCNN (<i>baseline</i>)	1.35	2390.47
EGCNN+ET (4-2)	1.06	2979.2
EGCNN+ET (6-2)	1.45	2938.28
EGCNN+ET (6-4)	1.55	2937.25
SwinIR	11.75	4163.95

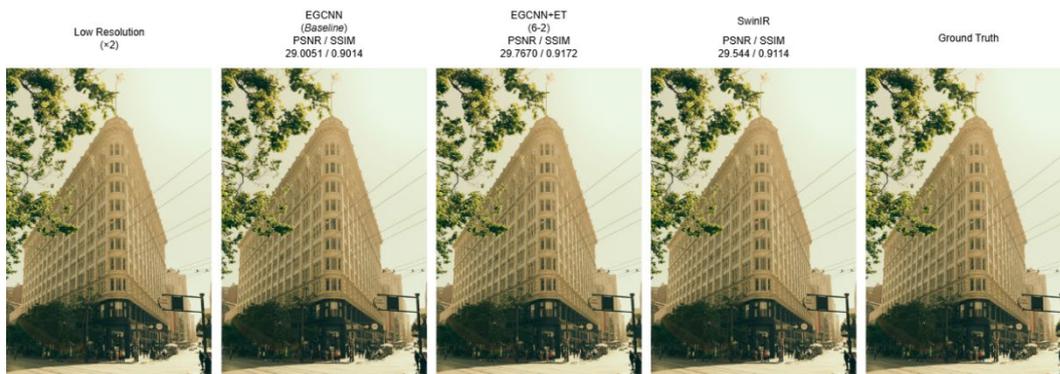
Berdasarkan tabel di atas, terlihat bahwa EGCNN (*baseline*) memiliki jumlah parameter sedikit (1,35 juta) dan waktu inferensi terendah (2390,47 ms) di antara varian EGCNN+ET. Namun, pendekatan *baseline* ini diketahui kurang optimal dalam menangkap konteks global, sehingga kualitas rekonstruksi citranya pun lebih rendah dibandingkan varian dengan ET. Ketika modul *Efficient Transformer* (ET) ditambahkan, terjadi kenaikan jumlah parameter dan juga *overhead* komputasi, meskipun tidak selalu berbanding lurus dengan waktu inferensi.

Menariknya, EGCNN+ET (4-2) memiliki jumlah parameter paling sedikit (1,06 juta) di antara varian EGCNN+ET, namun waktu inferensinya (2979,20 ms) lebih tinggi dibandingkan (6-2) dan (6-4). Hal ini mengindikasikan bahwa faktor jumlah parameter bukan satu-satunya penentu waktu inferensi, *overhead* transformasi data atau mekanisme *chunked attention* dalam *Performer* dapat memengaruhi durasi eksekusi secara signifikan. Sementara itu, EGCNN+ET (6-2) (1,45 juta parameter) dan EGCNN+ET (6-4) (1,55 juta parameter) memiliki waktu inferensi yang hampir sama, yaitu sekitar 2937–2938 ms yang menunjukkan bahwa penambahan blok *Performer* lebih banyak (6-4) tidak serta-merta menaikkan waktu inferensi secara signifikan, kemungkinan berkat optimasi paralelisasi internal.

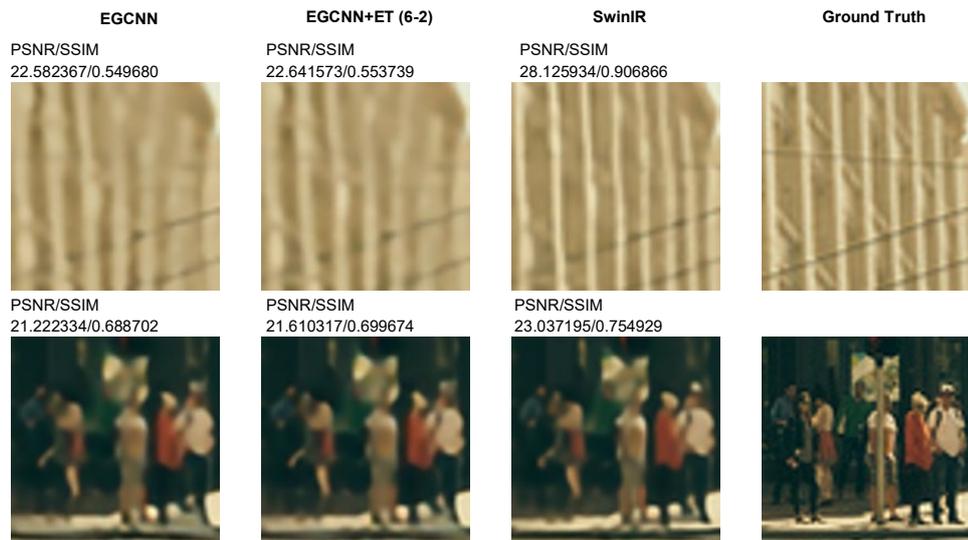
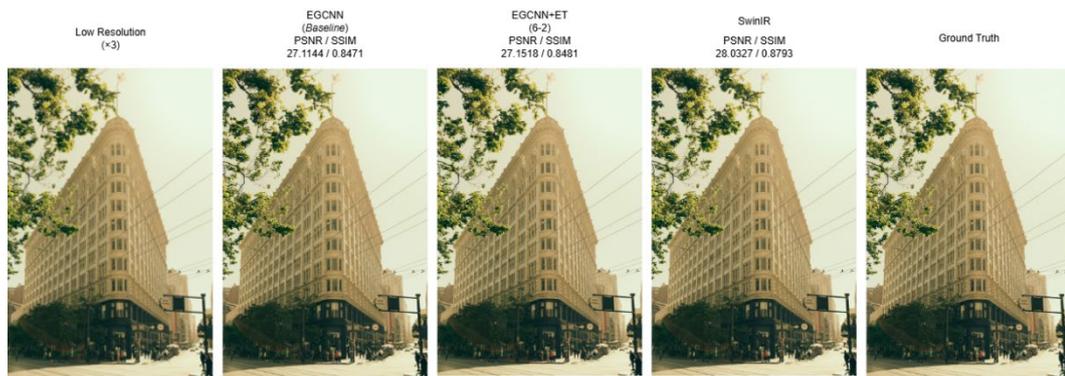
Jika dibandingkan dengan SwinIR yang memiliki 11,75 juta parameter dan memerlukan 4163,95 ms untuk inferensi, varian EGCNN+ET jauh lebih efisien, baik dalam jumlah parameter maupun waktu eksekusi. Meski SwinIR sering kali unggul dalam hal kualitas rekonstruksi, selisih waktu inferensi yang besar menegaskan adanya *trade-off* antara performa puncak dan efisiensi komputasi. Dari

perspektif aplikasi *real-time* atau perangkat dengan sumber daya terbatas, EGCNN+ET menawarkan alternatif yang kompetitif. Analisis efisiensi komputasi ini menegaskan bahwa integrasi ET pada EGCNN memang menambah *overhead*, tetapi tetap berada di bawah kompleksitas arsitektur *Transformer* yang lebih masif seperti SwinIR, sekaligus membawa peningkatan performa yang signifikan atas *baseline*.

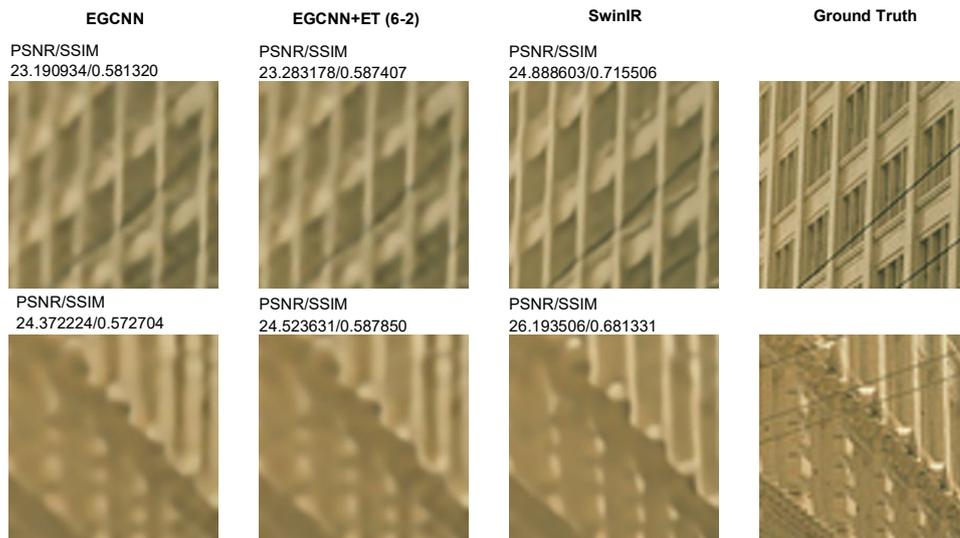
Untuk memberikan gambaran yang lebih intuitif mengenai kemampuan masing-masing model dalam merekonstruksi detail dan tekstur halus, berikut disajikan hasil perbandingan visual pada salah satu contoh citra dari dataset pengujian. Masing-masing citra diperbesar dengan faktor tertentu, kemudian diperlihatkan hasil rekonstruksi yang dihasilkan oleh EGCNN (*baseline*), EGCNN+ET, dan SwinIR, beserta *ground truth* untuk memudahkan penilaian ketajaman dan keakuratan warna.



Gambar 4.12 Perbandingan hasil rekonstruksi ×2 pada tiga arsitektur berbeda



Gambar 4.13 Perbandingan hasil rekonstruksi ×3 pada tiga arsitektur berbeda



Gambar 4.14 Perbandingan hasil rekonstruksi $\times 4$ pada tiga arsitektur berbeda

4.3.4.2 Hasil Kuantitatif Skala Pecahan

Pada skala pecahan ($\times\frac{5}{4}$, $\times\frac{3}{2}$, $\times\frac{7}{4}$), hasil kuantitatif pada Tabel 4.12 dan Tabel 4.13 menunjukkan bahwa integrasi *Efficient Transformer* (ET) ke dalam EGCNN (*Enhanced Group Convolutional Neural Network*) juga secara konsisten memperbaiki nilai PSNR dan SSIM dibandingkan *baseline* EGCNN.

Tabel 4.11 Keterangan model untuk skala pecahan

Model	Penjelasan
1	Sudah dilatih untuk <i>upscale</i> $\times 1.25$ dari citra yang di- <i>downscale</i> 0.80
2	Sudah dilatih untuk <i>upscale</i> $\times 1.5$ dari citra yang di- <i>downscale</i> 0.66
3	Sudah dilatih untuk <i>upscale</i> $\times 1.75$ dari citra yang di- <i>downscale</i> 0.57

Tabel 4.12 Hasil PSNR seluruh arsitektur pada skala pecahan

Model	Data Testing	EGCNN	EGCNN+	EGCNN+	EGCNN+	SwinIR
		(<i>baseline</i>)	ET (4-2)	ET (6-2)	ET (6-4)	
PSNR						
1	Set5	40.92	40.95	40.97	40.93	41.51
	Set14	37.09	37.12	37.17	37.14	37.49
	BSD100	35.99	36.03	36.08	36.06	36.39
	Urban100	34.44	34.47	34.55	34.48	34.79
2	Set5	40.08	40.11	40.16	40.07	40.59
	Set14	35.78	35.85	35.93	35.87	36.20
	BSD100	34.70	34.77	34.83	34.78	35.03
	Urban100	33.43	33.56	33.72	33.57	34.04
3	Set5	37.68	37.92	37.95	37.89	38.10
	Set14	33.54	33.73	33.75	33.74	33.79
	BSD100	32.42	32.57	32.60	32.58	32.64
	Urban100	30.68	31.03	31.11	31.04	31.05

Performa model EGCNN+ET (6-2) maupun (4-2) dan (6-4) cenderung unggul beberapa persepuluh dB PSNR di seluruh dataset pengujian (Set5, Set14, BSD100, dan Urban100) bila dibandingkan EGCNN (*baseline*) sehingga menandakan bahwa integrasi ET tetap efektif meskipun faktor skala yang

digunakan bukanlah integer. Pada skala $\times\frac{5}{4}$ atau $\times 1,25$ semua model relatif lebih mudah mempertahankan detail karena degradasi resolusi tidak terlalu drastic, EGCNN+ET (6-2) menampilkan sedikit peningkatan ketimbang (4-2) dan (6-4), terutama di Set5, yang relatif sederhana. Namun, Ketika skala dinaikkan menjadi $\times\frac{3}{2}$ ($\times 1,5$) dan $\times\frac{7}{4}$ ($\times 1,75$) yang dimana tantangan merekonstruksi detail halus meningkat, terlihat terjadi penurunan nilai PSNR/SSIM pada setiap model.

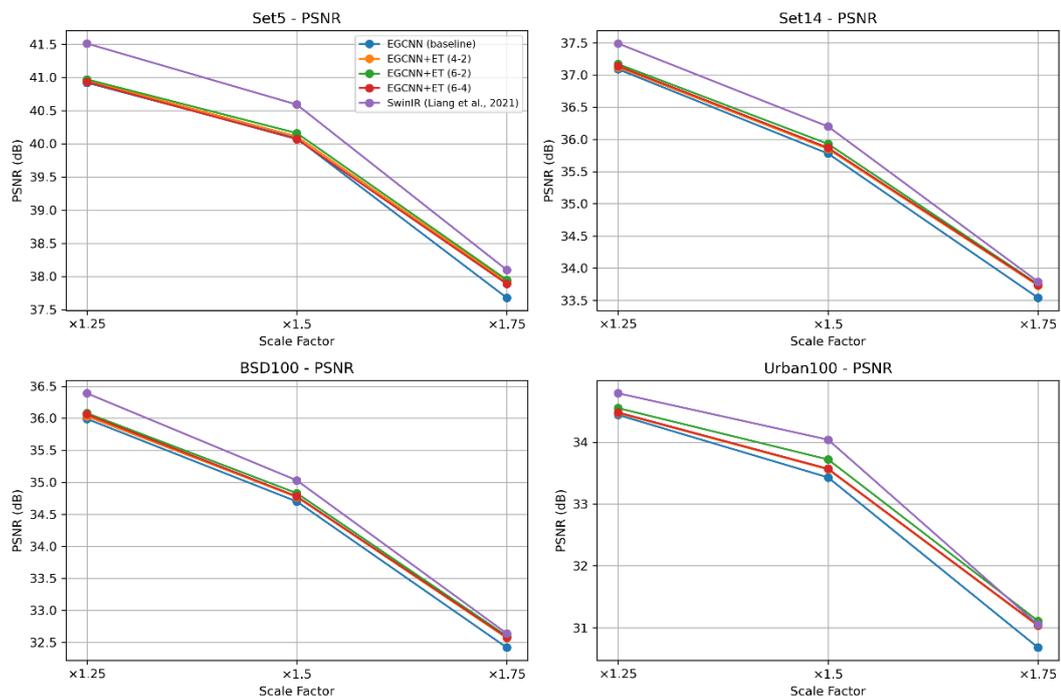
Tabel 4.13 Hasil SSIM seluruh arsitektur pada skala pecahan

Model	Data Testing	EGCNN (<i>baseline</i>)	EGCNN+ ET (4-2)	EGCNN+ ET (6-2)	EGCNN+ ET (6-4)	SwinIR
		PSNR				
1	Set5	0.9806	0.9805	0.9803	0.9804	0.9816
	Set14	0.9637	0.9632	0.9632	0.9631	0.9654
	BSD100	0.9601	0.9597	0.9599	0.9600	0.9621
	Urban100	0.9637	0.9633	0.9635	0.9633	0.9656
2	Set5	0.9752	0.9752	0.9753	0.9751	0.9760
	Set14	0.9516	0.9514	0.9519	0.9514	0.9535
	BSD100	0.9451	0.9449	0.9454	0.9451	0.9468
	Urban100	0.9519	0.9518	0.9529	0.9520	0.9555
3	Set5	0.9632	0.9638	0.9639	0.9638	0.9643
	Set14	0.9256	0.9261	0.9263	0.9262	0.9276
	BSD100	0.9106	0.9111	0.9114	0.9114	0.9129
	Urban100	0.9195	0.9217	0.9228	0.9219	0.9238

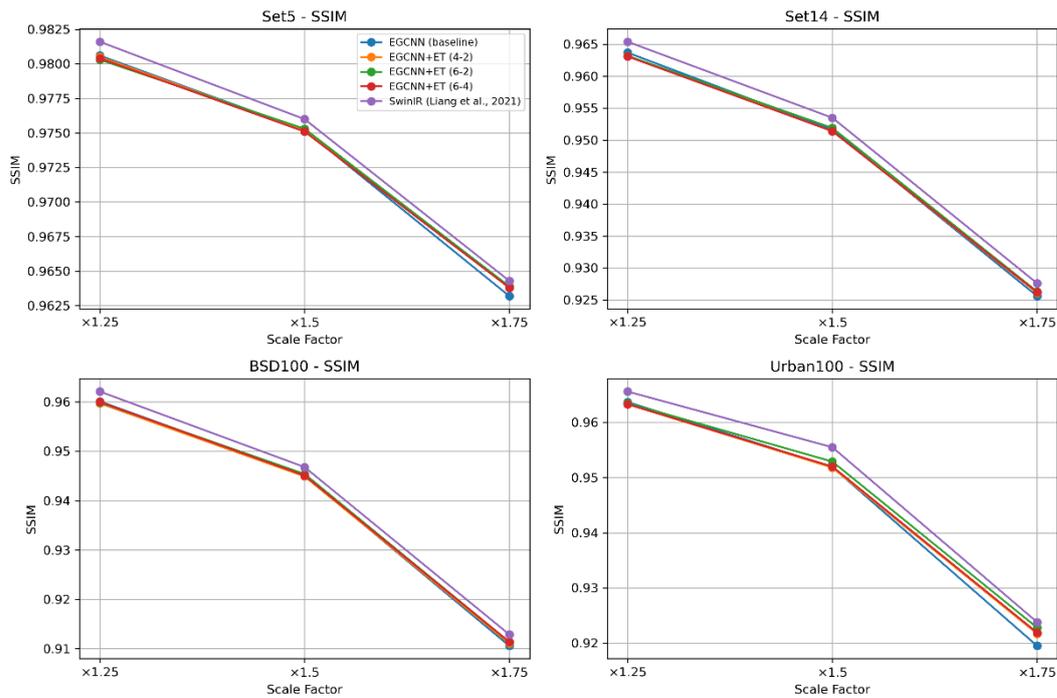
Meskipun demikian, EGCNN+ET masih menunjukkan margin keunggulan atas EGCNN (*baseline*) sekitar 0,1–0,4 dB PSNR, tergantung kompleksitas dataset. Hal ini menegaskan bahwa modul *self-attention* pada ET membantu model menangkap konteks global, bahkan untuk skala pecahan yang memerlukan penyesuaian resolusi lebih spesifik. Performa SwinIR (Liang et al., 2021) umumnya berada di posisi teratas, terutama di dataset dengan struktur spasial kompleks (BSD100, Urban100) yang mengindikasikan keunggulan mekanisme *shifted window attention* dalam memproses variasi tekstur yang luas. Namun, selisih antara SwinIR dan EGCNN+ET tidak selalu besar yang menandakan bahwa integrasi ET pada EGCNN mampu bersaing di hampir semua kondisi degradasi, baik integer maupun pecahan.

Khusus pada skala $\times\frac{7}{4}$ ($\times 1,75$), terlihat bahwa EGCNN+ET (6-2) atau (6-4) tetap menahan penurunan kinerja lebih baik dibanding *baseline* EGCNN, tetapi margin peningkatannya mulai menipis. Pada Urban100 misalnya, perbedaan PSNR antara EGCNN+ET dan *baseline* berkisar 0,4–0,5 dB, sedangkan SwinIR memimpin sedikit di atas EGCNN+ET. Kondisi ini menunjukkan bahwa penambahan ET berhasil memperkaya ekstraksi fitur lokal EGCNN dengan konteks global, walau pada tingkat degradasi yang lebih tinggi, keunggulan arsitektur *Transformer* yang lebih masif (seperti pada SwinIR) masih terlihat.

Berikut ini disajikan grafik PSNR dan SSIM pada Gambar 4.15 dan Gambar 4.16 untuk melihat tren nilai PSNR dan SSIM pada setiap faktor skala pecahan.

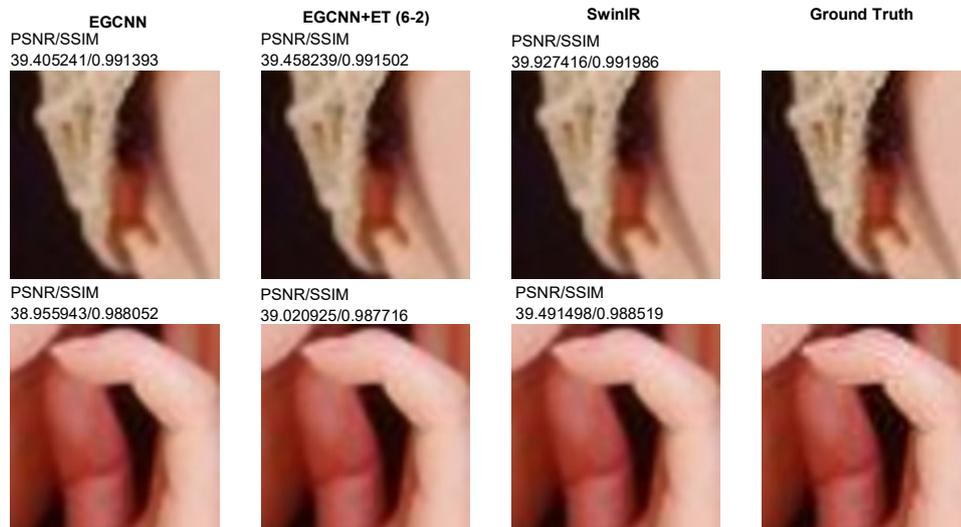
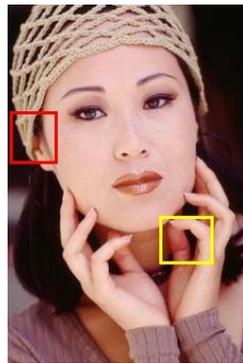


Gambar 4.15 Grafik nilai PSNR faktor skala pecahan pada keempat data uji

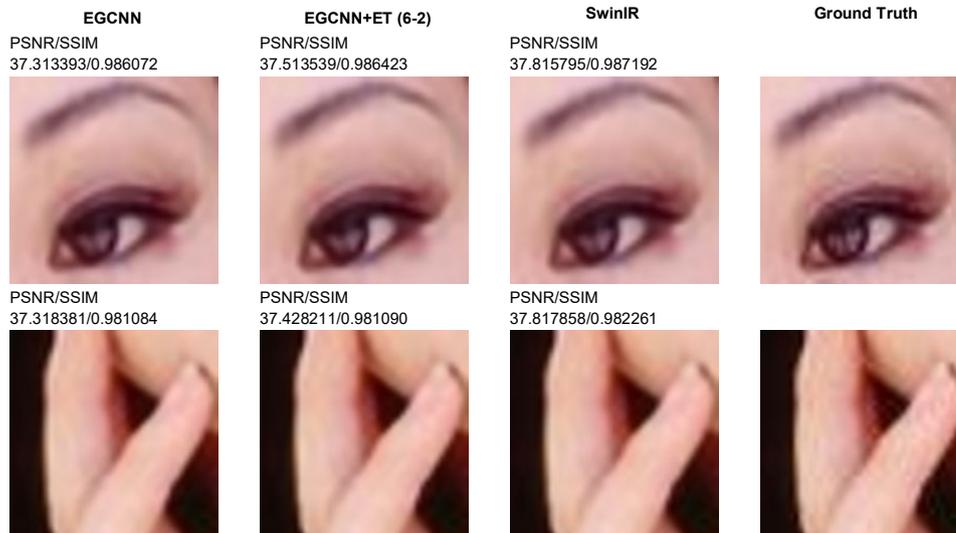
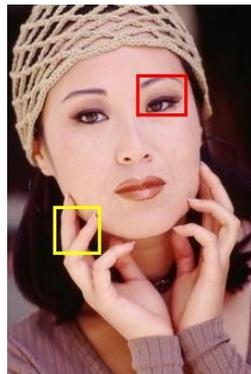
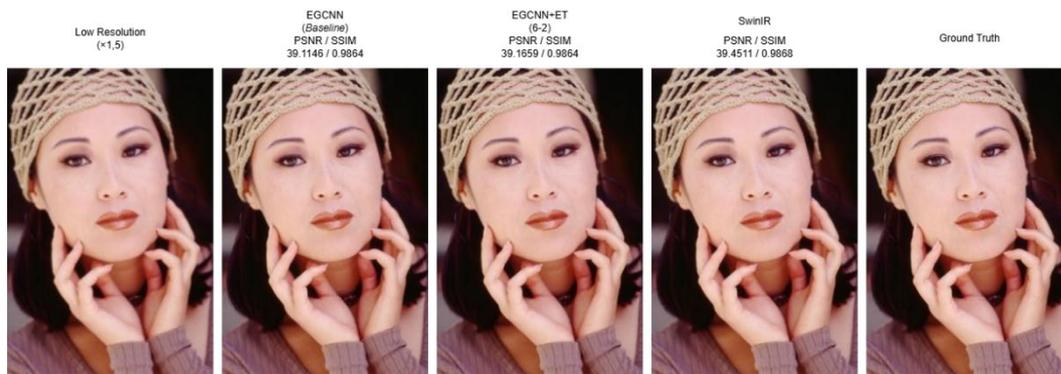


Gambar 4.16 Grafik nilai SSIM faktor skala pecahan pada keempat data uji

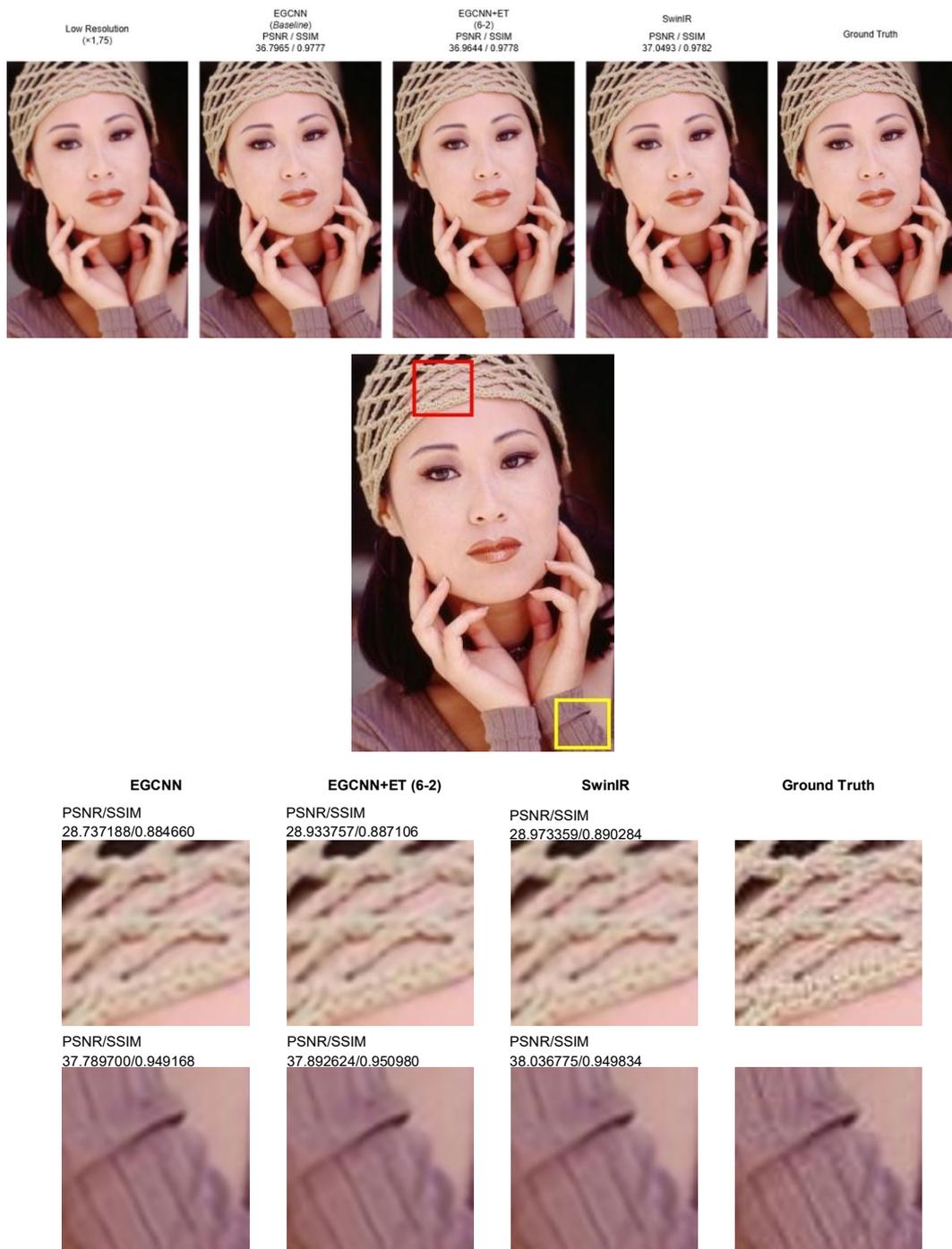
Berdasarkan Gambar 4.15 dan Gambar 4.16, terlihat bahwa semua model mengalami penurunan PSNR dan SSIM seiring meningkatnya faktor skala pecahan dari $\times\frac{5}{4}$ ke $\times\frac{7}{4}$ di setiap dataset. Kondisi ini mencerminkan tantangan yang lebih besar dalam merekonstruksi detail ketika skala pembesaran bertambah, meskipun degradasi yang dihadapi adalah pecahan. Namun, penurunan performa ini tidaklah sama besar untuk setiap model, sehingga pola persaingan antar model tetap bisa dianalisis. Untuk memberikan gambaran yang lebih intuitif mengenai kemampuan masing-masing model dalam merekonstruksi detail pada skala pecahan, berikut disajikan hasil perbandingan visual pada salah satu contoh citra dari dataset pengujian.



Gambar 4.17 Perbandingan hasil rekonstruksi $\times \frac{5}{4}$ pada tiga arsitektur berbeda



Gambar 4.18 Perbandingan hasil rekonstruksi $\times \frac{3}{2}$ pada tiga arsitektur berbeda



Gambar 4.19 Perbandingan hasil rekonstruksi $\times \frac{7}{4}$ pada tiga arsitektur berbeda

4.3.5 Evaluasi Kualitatif

Evaluasi kualitatif dilakukan untuk memastikan bahwa kenaikan metrik objektif terutama PSNR dan SSIM yang dicapai model EGCNN + ET benar-benar sejalan dengan persepsi manusia. Desain eksperimen ini mengadopsi rekomendasi *double-stimulus continuous-quality scale* (DSCQS) dari ITU-R BT.500 (Bt et al.,

2002), di mana citra referensi beresolusi tinggi (*ground truth*) selalu ditampilkan terlebih dahulu, disusul tiga citra hasil super-resolusi. Pendekatan tersebut meminimalkan *memory bias* dan memberi kesempatan pada responden untuk menilai kualitas relatif secara langsung di lingkungan tampilan terkontrol.

Lima citra uji dipilih secara acak dari Set5, Set14, BSD100, dan Urban100 kemudian diperbesar $\times 2$ yang menjadi skala unggulan penelitian untuk tiga model: EGCNN (*baseline*), EGCNN + ET, dan SwinIR. Setiap responden menyelesaikan penilaian lima citra melalui kuesioner digital yang disusun melalui *Google Form*. Sebanyak lima responden berusia 21–24 tahun dengan penglihatan normal atau terkoreksi berpartisipasi secara sukarela. Responden diminta memberi satu nilai untuk setiap citra pada skala Likert 1–5 sesuai dengan deskripsi pada Tabel 4.14 (Li et al., 2022).

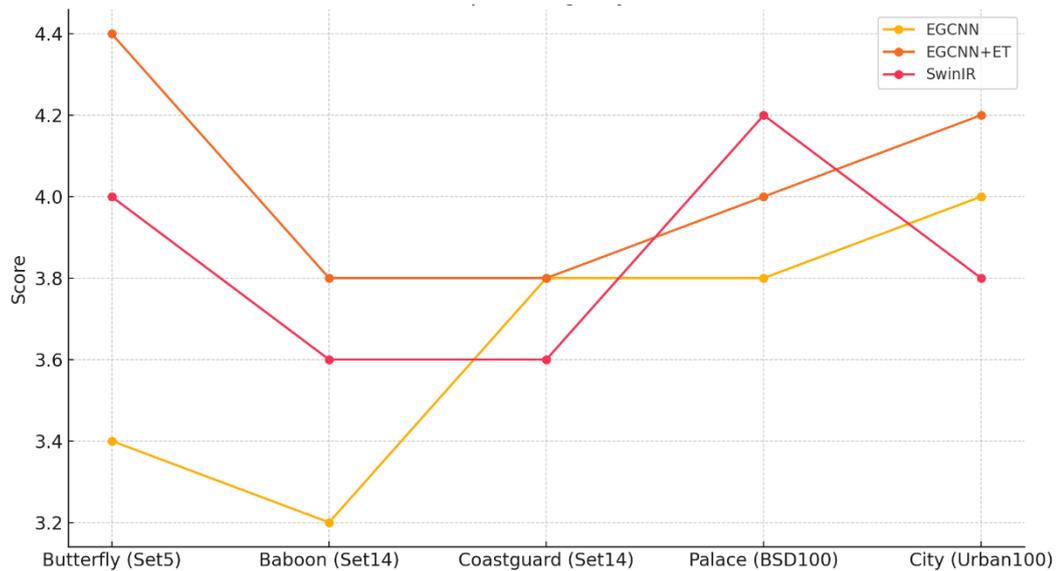
Tabel 4.14 Analisis kualitatif penilaian *upscaling* citra

Skala	Deskripsi
1	Tidak ada pemulihan warna, kontras rendah, distorsi tekstur, artefak tepi, dan visibilitas buruk.
2	Pemulihan warna sebagian, peningkatan kontras, distorsi tekstur, artefak tepi, dan visibilitas buruk.
3	Pemulihan warna, peningkatan kontras, tekstur realistis, artefak tepi lokal, dan visibilitas yang dapat diterima.
4	Pemulihan warna, peningkatan kontras, tekstur realistis, pemulihan artefak tepi yang lebih baik, dan visibilitas yang lebih baik.
5	Pemulihan warna, peningkatan kontras, tekstur realistis, artefak tepi tajam, dan visibilitas yang sangat baik.

Pada pengujian subjektif lima citra, Skor rata-rata per citra menunjukkan EGCNN+ET konsisten tertinggi: 4,4 pada tekstur rumit *Butterfly* dan 4,2 pada pemandangan urban *City*. Peningkatan ini menegaskan kapasitas modul *Performer* menekan artefak sambil menjaga detail. SwinIR unggul tipis di *Palace* (4,2) berkat kekuatan *self-attention* menangani pola tepi berulang, tetapi turun di *City* yang kaya geometri diagonal. EGCNN (*baseline*) stabil namun tidak menonjol, terutama terhambat di *Baboon* (3,2) yang memuat *noise* frekuensi tinggi.

Tabel 4.15 Data mentah penilaian responden berdasarkan citra dan model

Model	Butterfly (Set5)	Baboon (Set14)	Coastguard (Set14)	Palace (BSD100)	City (urban100)
EGCNN (<i>baseline</i>)	3.4	3.2	3.8	3.8	4
EGCNN+ET	4.4	3.8	3.8	4	4.2
SwinIR	4	3.6	3.6	4.2	3.8



Gambar 4.20 Skor per citra berdasarkan model

Tabel 4.16 Mean Opinion Score per model

Model	MOS
EGCNN+ET	4.04
SwinIR	3.84
EGCNN (<i>baseline</i>)	3.64

EGCNN + ET tampil sebagai model paling disukai. Rata-rata skor persepsi (MOS)-nya mencapai 4,04, unggul atas SwinIR (3,84) dan *baseline* EGCNN (3,64). Keunggulan ini konsisten pada citra bertekstur halus (*Butterfly*) dan pemandangan kota bertepi kompleks (*City*) yang menandakan modul ET efektif mempertajam detail sambil menekan artefak. SwinIR sempat memimpin di citra *Palace* (4,2) berkat mekanisme *self-attention* yang kuat untuk pola berulang, tetapi performanya turun pada *City* yang mengindikasikan sensitivitas terhadap struktur diagonal yang

rapat. *Baseline* EGCNN tertinggal di semua citra, paling nyata pada *Baboon* (3,2) yang mengandung frekuensi tinggi acak.

Uji *Friedman* ($\chi^2 = 4,11$; $p = 0,128$) memang belum menunjukkan signifikansi statistik karena jumlah citra yang terbatas, namun pola skor yang konsisten dan selisih MOS $\pm 0,4-0,6$ antarmodel mengisyaratkan perbedaan yang secara praktis bermakna. Dengan demikian, responden menyatakan bahwa mereka merasakan peningkatan kualitas visual pada EGCNN + ET: citra terlihat lebih tajam, warna lebih alami, dan artefak lebih sedikit dibanding dua model perbandingan. Hasil kualitatif ini sejalan dengan nilai PSNR/SSIM pada evaluasi objektif yang memperkuat klaim bahwa integrasi ET ke dalam arsitektur EGCNN benar-benar meningkatkan pengalaman visual pengguna akhir.

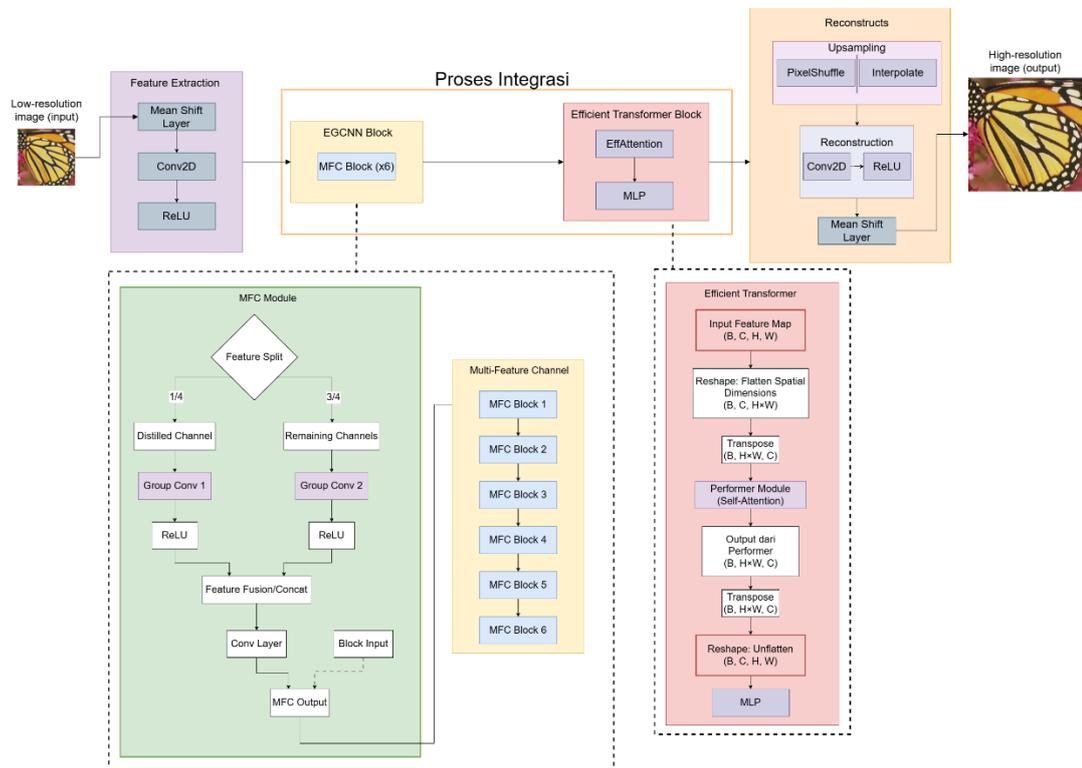
4.4 Pembahasan

Penelitian ini mengacu pada metodologi yang digunakan sebelumnya dengan langkah awal adalah studi literatur yang bertujuan untuk perkembangan terkini dalam *Single Image Super-Resolution* (SISR) dan mengidentifikasi kelemahan model-model yang telah ada. Tahap ini membantu merumuskan masalah dan pertanyaan penelitian yaitu bagaimana mengintegrasikan *Efficient Transformer* (ET) ke dalam *Enhanced Group Convolutional Neural Network* (EGCNN) untuk meningkatkan kinerja SISR. Setelah rumusan penelitian ditetapkan, dilakukan pengumpulan data sekunder melalui pemilihan dataset DF2K (gabungan DIV2K dan Flickr2K) sebagai sumber utama, karena keanekaragaman konten dan resolusinya yang tinggi.

Selanjutnya, data yang terkumpul menjalani *pre-processing* untuk memastikan kualitas dan konsistensi. Proses ini meliputi pembuatan pasangan citra beresolusi rendah (LR) dan beresolusi tinggi (HR), di mana berbagai faktor skala (integer maupun pecahan) digunakan agar model belajar mengatasi beragam skenario degradasi. Hasil *pre-processing* disertai *metadata* yang mencatat skala dan resolusi untuk memudahkan *pipeline* pelatihan membaca pasangan (LR, HR) dengan benar.

Tahap berikutnya adalah desain arsitektur model, di mana penelitian mengembangkan kerangka *hybrid* EGCNN+ET yang ditampilkan pada Gambar 4.21. Dalam arsitektur tersebut, EGCNN menggunakan *group convolution* untuk

mengekstraksi fitur lokal secara efisien melalui serangkaian blok MFC (*Multi-Feature Channel*) yang memanfaatkan mekanisme *distilasi* dan *residual* untuk mengoptimalkan informasi lokal. Sementara itu, modul ET yang diintegrasikan setelah blok EGCNN berfungsi menangkap hubungan jangka panjang antar piksel melalui mekanisme *efficient self-attention* (diimplementasikan menggunakan *Performer*). Transformasi data dilakukan dengan mengubah *tensor* dari format spasial menjadi bentuk sekuensial sehingga ET dapat bekerja secara optimal dan kemudian diubah kembali ke format semula untuk proses rekonstruksi. Selanjutnya, modul *upsampling* yang dimodifikasi dapat menangani skala pembesaran integer maupun pecahan supaya model tetap fleksibel dalam menghadapi degradasi non-integer.



Gambar 4.21 Desain arsitektur *hybrid* EGCNN+ET

Pada penelitian arsitektur *hybrid* lain seperti ESRT yang mengintegrasikan *Efficient Transformer* ke dalam arsitektur RCAN, fokus utamanya adalah pada peningkatan kapasitas global melalui mekanisme *self-attention*, namun dengan overhead parameter yang relatif tinggi. Di sisi lain, penelitian ini menggabungkan efisiensi komputasi yang diperoleh dari *group convolution* pada EGCNN dengan

kekuatan global *self-attention* dari ET, sehingga menghasilkan model yang tidak hanya meningkatkan performa rekonstruksi (PSNR dan SSIM) tetapi juga lebih hemat dalam hal jumlah parameter dan waktu inferensi. Selain itu, pendekatan yang diusulkan juga unggul dari segi fleksibilitas, karena modul *upsampling* yang telah dimodifikasi memungkinkan model untuk menangani berbagai skenario degradasi, baik dengan faktor skala integer maupun pecahan.

Penelitian terdahulu juga telah menyoroti pentingnya memadukan mekanisme *self-attention* dengan metode konvolusi untuk mengatasi kelemahan CNN dalam memahami struktur global. Penelitian ini menekankan penyisipan ET setelah blok MFC untuk memperkaya representasi fitur lokal yang telah diperoleh EGCNN. Pendekatan yang diusulkan menawarkan kompromi antara metode berbasis CNN murni dan *Transformer* penuh sehingga memungkinkan peningkatan kualitas rekonstruksi tanpa menambah kompleksitas model secara drastis. Jika dibandingkan dengan penelitian terdahulu, langkah ini merupakan modifikasi baru variasi arsitektur yang menggabungkan efisiensi *group convolution* dan kemampuan global *self-attention*, sehingga hasil akhirnya memadukan kekuatan lokal dan global dalam satu kerangka model SISR.

Hasil dari Evaluasi kuantitatif, baik untuk skala integer maupun skala pecahan, menegaskan bahwa integrasi ET meningkatkan PSNR dan SSIM secara konsisten dibandingkan EGCNN (*baseline*). Pada skala $\times 2$, EGCNN+ET kerap melampaui SwinIR, sedangkan pada skala yang lebih besar ($\times 3, \times 4$) dan pecahan, SwinIR masih unggul tipis, terutama pada dataset berstruktur kompleks. Hasil kualitatif juga menampilkan peningkatan nyata, di mana EGCNN+ET berhasil menghasilkan rekonstruksi citra dengan detail tekstur yang lebih tajam dan transisi warna yang lebih mulus dibandingkan dengan model *baseline*, walaupun SwinIR menunjukkan kelebihan di beberapa area kritis.

Hasil evaluasi ini sejalan dengan temuan penelitian lain yang menunjukkan bahwa integrasi modul *Transformer* ke dalam arsitektur SISR dapat meningkatkan kualitas rekonstruksi citra secara signifikan. Penelitian mengenai ESRT (Lu et al., 2022) mengungkapkan bahwa penerapan *Efficient Transformer* pada arsitektur RCAN menghasilkan kenaikan PSNR dan SSIM yang konsisten, terutama pada skala pembesaran yang lebih rendah seperti $\times 2$ yang juga mendukung temuan

bahwa EGCNN+ET kerap melampaui model *baseline*. Selain itu, studi oleh Liang et al. (2021) tentang SwinIR menekankan bahwa arsitektur *Transformer* yang lebih besar dapat mengatasi tantangan rekonstruksi pada kondisi degradasi ekstrem, terutama pada dataset dengan struktur kompleks. Hal ini menjelaskan mengapa pada skala pembesaran yang lebih tinggi dan pecahan, SwinIR menunjukkan keunggulan tipis dibandingkan EGCNN+ET. Dengan demikian, penelitian terdahulu mendukung bahwa pendekatan *hybrid* yang menggabungkan efisiensi ekstraksi fitur lokal dari CNN dengan kemampuan global *self-attention* melalui penggunaan *Efficient Transformer* dapat memberikan kompromi optimal dalam meningkatkan performa SISR.

BAB 5

KESIMPULAN DAN SARAN

Bab ini memaparkan kesimpulan yang dapat diambil dari rangkaian penelitian yang telah dilaksanakan, serta memberikan saran untuk pengembangan lebih lanjut.

5.1 Kesimpulan

Pada penelitian ini dikembangkan arsitektur *hybrid* yang mengintegrasikan *Efficient Transformer* (ET) ke dalam arsitektur *Enhanced Group Convolutional Neural Network* (EGCNN). Integrasi dilakukan dengan menempatkan modul ET setelah serangkaian blok EGCNN yang berfungsi mengekstraksi fitur lokal, sehingga fitur tersebut diubah menjadi urutan yang diproses melalui mekanisme *efficient self-attention* dan kemudian direkonstruksi kembali ke bentuk spasial. Pendekatan ini berhasil meningkatkan nilai PSNR dan SSIM secara konsisten di seluruh dataset pengujian, khususnya pada skala pembesaran $\times 2$, di mana peningkatan PSNR mencapai sekitar 0,9 dB dibandingkan dengan model *baseline*. Hal ini menegaskan bahwa ET mampu menangkap hubungan jarak jauh antar piksel dan melengkapi kekurangan ekstraksi fitur lokal EGCNN sehingga menghasilkan rekonstruksi citra yang lebih akurat dan detail.

Selain itu, penelitian ini juga menunjukkan bahwa peningkatan performa yang dicapai oleh EGCNN+ET tidak hanya signifikan jika dibandingkan dengan EGCNN *baseline*, tetapi juga kompetitif dengan model SISR terkini seperti SwinIR. Pada berbagai skenario *upsampling*, baik untuk skala integer ($\times 2$, $\times 3$, $\times 4$) maupun skala pecahan ($\times \frac{5}{4}$, $\times \frac{3}{2}$, $\times \frac{7}{4}$), integrasi ET menghasilkan peningkatan kualitas rekonstruksi citra yang konsisten. Meskipun peningkatan performa relatif lebih menurun pada faktor skala yang lebih tinggi, model EGCNN+ET tetap menunjukkan keunggulan dalam mempertahankan detail dan tekstur yang halus dibandingkan model *baseline*, dengan perbedaan PSNR berkisar antara 0,1 hingga 0,5 dB dan peningkatan SSIM yang selaras. Temuan ini mengindikasikan bahwa pendekatan *hybrid* EGCNN+ET menawarkan solusi yang kompetitif untuk perbaikan kualitas rekonstruksi.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa rekomendasi untuk pengembangan model SISR ke depan sebagai berikut.

1. Eksperimen menunjukkan bahwa menambah blok ET di luar batas tertentu tidak selalu berdampak signifikan pada peningkatan kinerja. Penelitian selanjutnya dapat memfokuskan pada pemilihan jumlah blok ET yang optimal, atau menerapkan strategi adaptif di mana model secara dinamis menyesuaikan jumlah blok *Transformer* sesuai kompleksitas citra input.
2. Meskipun *Efficient Transformer* efektif, penelitian lanjutan bisa mencoba varian *self-attention* lain (misalnya *Linformer* atau *local attention*) untuk menyeimbangkan kapasitas pemodelan global dengan *overhead* komputasi yang minimal, terutama pada skala yang lebih besar.
3. Menerapkan pendekatan EGCNN+ET ke domain video SISR bisa menjadi langkah berikutnya. Konteks *temporal* dan *spatiotemporal self-attention* bisa menambah dimensi tantangan sekaligus potensi peningkatan kualitas, terutama dalam menjaga konsistensi antar *frame*.

DAFTAR PUSTAKA

- Albawi, S., Mohammed, T. A. M., & Alzawi, S. (2017). Layers of a Convolutional Neural Network. *Icet2017*.
- Bashir, S. M. A., Wang, Y., Khan, M., & Niu, Y. (2021). A comprehensive review of deep learningbased single image super-resolution. *PeerJ Computer Science*, 7. <https://doi.org/10.7717/PEERJ-CS.621>
- Brownlee, J. (2019). A Gentle Introduction to Pooling Layers for Convolutional Neural Networks. *Machine Learning Mastery*.
- Bt, I., Bt, I., Itu-r, Q., Itu-r, Q., Itu, T., Itu, T., Assembly, R., & Assembly, R. (2002). RECOMMENDATION ITU-R BT.500-11 Methodology for the subjective assessment of the quality of television pictures. *Methodology*, 12.
- Cai, W., Liu, Y., Chen, Y., Dong, Z., Yuan, H., & Li, N. (2023). A Seabed Terrain Feature Extraction Transformer for the Super-Resolution of the Digital Bathymetric Model. *Remote Sensing*, 15(20). <https://doi.org/10.3390/rs15204906>
- Choromanski, K., Likhoshesterov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., Belanger, D., Colwell, L., & Weller, A. (2020). Performers: Rethinking Attention. *ArXiv*.
- Cruz, C., Mehta, R., Katkovnik, V., & Egiazarian, K. O. (2018). Single image super-resolution based on wiener filter in similarity domain. *IEEE Transactions on Image Processing*, 27(3). <https://doi.org/10.1109/TIP.2017.2779265>
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1.
- Dong, C., Loy, C. C., He, K., & Tang, X. (2014). Learning a deep convolutional network for image super-resolution. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8692 LNCS(PART 4). https://doi.org/10.1007/978-3-319-10593-2_13
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner,

- T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE. *ICLR 2021 - 9th International Conference on Learning Representations*.
- Gankhuyag, G., Huh, J., Kim, M., Yoon, K., Moon, H., Lee, S., Jeong, J., Kim, S., & Choe, Y. (2023). Skip-Concatenated Image Super-Resolution Network for Mobile Devices. *IEEE Access*, 11. <https://doi.org/10.1109/ACCESS.2022.3232258>
- Goodfellow, I. (2015). Deep Feedforward Networks. *Deep Learning Book*, 1.
- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77. <https://doi.org/10.1016/j.patcog.2017.10.013>
- Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., Yang, Z., Zhang, Y., & Tao, D. (2023). A Survey on Vision Transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1). <https://doi.org/10.1109/TPAMI.2022.3152247>
- Hirata, D., & Takahashi, N. (2023). Ensemble Learning in CNN Augmented with Fully Connected Subnetworks. *IEICE Transactions on Information and Systems*, E106.D(7). <https://doi.org/10.1587/transinf.2022EDL8098>
- Hong, J., Lee, B., Ko, K., & Ko, H. (2023). Fast Non-Local Attention network for light super-resolution. *Journal of Visual Communication and Image Representation*, 95. <https://doi.org/10.1016/j.jvcir.2023.103861>
- Horé, A., & Ziou, D. (2010). Image quality metrics: PSNR vs. SSIM. *Proceedings - International Conference on Pattern Recognition*. <https://doi.org/10.1109/ICPR.2010.579>
- Huang, G., Liu, S., Maaten, L. Van Der, & Weinberger, K. Q. (2018). CondenseNet: An Efficient DenseNet Using Learned Group Convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/CVPR.2018.00291>
- Huang, H., Shen, L., He, C., Dong, W., & Liu, W. (2023). Differentiable Neural Architecture Search for Extremely Lightweight Image Super-Resolution. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(6). <https://doi.org/10.1109/TCSVT.2022.3230824>
- Hwang, S., Han, D., & Jeon, M. (2023). Making depthwise convolution SR-friendly

- via kernel attention injection. *Journal of Visual Communication and Image Representation*, 96. <https://doi.org/10.1016/j.jvcir.2023.103930>
- Khalitov, R., Yu, T., Cheng, L., & Yang, Z. (2022). Sparse factorization of square matrices with application to neural attention modeling. *Neural Networks*, 152. <https://doi.org/10.1016/j.neunet.2022.04.014>
- Kim, J., Lee, J. K., & Lee, K. M. (2016). Accurate image super-resolution using very deep convolutional networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-December*. <https://doi.org/10.1109/CVPR.2016.182>
- Lai, W. S., Huang, J. Bin, Ahuja, N., & Yang, M. H. (2019). Fast and Accurate Image Super-Resolution with Deep Laplacian Pyramid Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(11). <https://doi.org/10.1109/TPAMI.2018.2865304>
- Le-Tien, T., Nguyen-Thanh, T., Xuan, H. P., Nguyen-Truong, G., & Ta-Quoc, V. (2020). Deep learning based approach implemented to image super-resolution. *Journal of Advances in Information Technology*, 11(4). <https://doi.org/10.12720/jait.11.4.209-216>
- Lee, Y., Park, J., & Lee, C. O. (2022). Two-level group convolution. *Neural Networks*, 154. <https://doi.org/10.1016/j.neunet.2022.07.024>
- Li, W., Lin, C., Luo, T., Li, H., Xu, H., & Wang, L. (2022). Subjective and Objective Quality Evaluation for Underwater Image Enhancement and Restoration. *Symmetry*, 14(3). <https://doi.org/10.3390/sym14030558>
- Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., & Timofte, R. (2021). SwinIR. *Proceedings of the IEEE International Conference on Computer Vision, 2021-October*.
- Lim, B., Son, S., Kim, H., Nah, S., & Lee, K. M. (2017). Enhanced Deep Residual Networks for Single Image Super-Resolution. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2017-July*. <https://doi.org/10.1109/CVPRW.2017.151>
- Lu, Z., Li, J., Liu, H., Huang, C., Zhang, L., & Zeng, T. (2022). Transformer for Single Image Super-Resolution. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2022-June*. <https://doi.org/10.1109/CVPRW56347.2022.00061>
- Muhammad, W., Aramvith, S., & Onoye, T. (2021). Multi-scale Xception based depthwise separable convolution for single image superresolution. *PLoS ONE*,

16(8 August). <https://doi.org/10.1371/journal.pone.0249278>

- Nan, F., Jing, W., Tian, F., Zhang, J., Chao, K. M., Hong, Z., & Zheng, Q. (2022). Feature super-resolution based Facial Expression Recognition for multi-scale low-resolution images. *Knowledge-Based Systems*, 236. <https://doi.org/10.1016/j.knosys.2021.107678>
- National Instruments Australia. (2013). Peak Signal-to-Noise ratio as an image quality metric. *National Instruments*.
- Pau, D. Pietro, & Aymone, F. M. (2024). Mathematical Formulation of Learning and Its Computational Complexity for Transformers' Layers. *Eng*, 5(1). <https://doi.org/10.3390/eng5010003>
- Sun, Y., Qin, J., Gao, X., Chai, S., & Chen, B. (2022). Attention-enhanced multi-scale residual network for single image super-resolution. *Signal, Image and Video Processing*, 16(5). <https://doi.org/10.1007/s11760-021-02095-x>
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., & Erhan, D. (2016). Going Deeper with Convolutions (GoogleLeNet). *Journal of Chemical Technology and Biotechnology*, 91(8).
- Tai, Y., Yang, J., Liu, X., & Xu, C. (2017). MemNet: A Persistent Memory Network for Image Restoration. *Proceedings of the IEEE International Conference on Computer Vision, 2017-October*. <https://doi.org/10.1109/ICCV.2017.486>
- Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient Transformers: A Survey. *ACM Computing Surveys*, 55(6). <https://doi.org/10.1145/3530811>
- Tian, C., Yuan, Y., Zhang, S., Lin, C. W., Zuo, W., & Zhang, D. (2022). Image super-resolution with an enhanced group convolutional neural network. *Neural Networks*, 153. <https://doi.org/10.1016/j.neunet.2022.06.009>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 2017-December*.
- Viriyavisuthisakul, S., Kaothanthong, N., Sanguansat, P., Nguyen, M. Le, & Haruechaiyasak, C. (2022). Parametric regularization loss in super-resolution reconstruction. *Machine Vision and Applications*, 33(5). <https://doi.org/10.1007/s00138-022-01315-9>
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4). <https://doi.org/10.1109/TIP.2003.819861>

- Wang, Z., Chen, J., & Hoi, S. C. H. (2021). Deep Learning for Image Super-Resolution: A Survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Vol. 43, Issue 10). <https://doi.org/10.1109/TPAMI.2020.2982166>
- Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K., & San Diego, U. (2016). ResNeXt_CVPR2017. *Cvpr2017*.
- Xin, J., Wei, Z., Wang, N., Li, J., & Gao, X. (2022). Face frontalization for uncontrolled scenes. *Journal of Image and Graphics*, 27(9). <https://doi.org/10.11834/jig.211256>
- Xu, Z., Zhang, W., Zhang, T., Yang, Z., & Li, J. (2021). Efficient transformer for remote sensing image segmentation. *Remote Sensing*, 13(18). <https://doi.org/10.3390/rs13183585>
- Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J. H., & Liao, Q. (2019). Deep Learning for Single Image Super-Resolution: A Brief Review. In *IEEE Transactions on Multimedia* (Vol. 21, Issue 12). <https://doi.org/10.1109/TMM.2019.2919431>
- Ye, S., Zhao, S., Hu, Y., & Xie, C. (2023). Single-Image Super-Resolution Challenges: A Brief Review. In *Electronics (Switzerland)* (Vol. 12, Issue 13). <https://doi.org/10.3390/electronics12132975>
- Yu, Q., Liu, F., Xiao, L., Liu, Z., & Yang, X. (2021). Real-time environment monitoring using a lightweight image super-resolution network. *International Journal of Environmental Research and Public Health*, 18(11). <https://doi.org/10.3390/ijerph18115890>
- Zamfir, E., Conde, M. V., & Timofte, R. (2023). Towards Real-Time 4K Image Super-Resolution. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2023-June*. <https://doi.org/10.1109/CVPRW59228.2023.00155>
- Zhang, C., Benz, P., Argaw, D. M., Lee, S., Kim, J., Rameau, F., Bazin, J. C., & Kweon, I. S. (2021). ResNet or DenseNet? Introducing dense shortcuts to ResNet. *Proceedings - 2021 IEEE Winter Conference on Applications of Computer Vision, WACV 2021*. <https://doi.org/10.1109/WACV48630.2021.00359>
- Zhang, L. (2018). Self-Attention Mechanisms in Natural Language Processing. *DZone AI*, 18.
- Zhang, R., Zhao, P., Guo, W., Wang, R., & Lu, W. (2022). Medical named entity

recognition based on dilated convolutional neural network. *Cognitive Robotics*, 2. <https://doi.org/10.1016/j.cogr.2021.11.002>

Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., & Fu, Y. (2018). Image super-resolution using very deep residual channel attention networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11211 LNCS. https://doi.org/10.1007/978-3-030-01234-2_18

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2021). Residual Dense Network for Image Restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7). <https://doi.org/10.1109/TPAMI.2020.2968521>

Zheng, W., Lu, S., Yang, Y., Yin, Z., & Yin, L. (2024). Lightweight transformer image feature extraction network. *PeerJ Computer Science*, 10. <https://doi.org/10.7717/peerj-cs.1755>

Zhu, H., Xie, C., Fei, Y., & Tao, H. (2021). Attention mechanisms in cnn-based single image super-resolution: A brief review and a new perspective. In *Electronics (Switzerland)* (Vol. 10, Issue 10). <https://doi.org/10.3390/electronics10101187>

LAMPIRAN

Lampiran 1. Kuesioner survei penilaian kualitas citra hasil super-resolution

← Preview mode Published [Copy responder link](#)

Survei Penilaian Kualitas Citra Hasil Super-Resolution

Instruksi Penilaian

1. Tiap halaman menampilkan gambar referensi asli (*Ground Truth*) dan gambar hasil super resolusi $\times 2$.
2. Terdapat tiga hasil model *upscaling* dengan urutan gambar A, B, C yang secara acak akan dinilai.
3. Nilaiah setiap hasil pada skala 1 – 5 sesuai aspek pada tabel dibawah ini.
4. Anda bisa men-zoom lebih dekat untuk menilai gambar dengan lebih detail.
5. Klik "Next" setelah menilai setiap gambar per-Section.

Skala	Deskripsi
1	Tidak ada pemulhan warna, kontras rendah, distorsi tekstur, artefak tepi, dan visibilitas buruk.
2	Pemulhan warna sebagian, peningkatan kontras, distorsi tekstur, artefak tepi, dan visibilitas buruk.
3	Pemulhan warna, peningkatan kontras, tekstur realistik, artefak tepi lokal, dan visibilitas yang dapat diterima.
4	Pemulhan warna, peningkatan kontras, tekstur realistik, pemulhan artefak tepi yang lebih baik, dan visibilitas yang lebih baik.
5	Pemulhan warna, peningkatan kontras, tekstur realistik, artefak tepi tajam, dan visibilitas yang sangat baik.

Back Next Page 3 of 8 Clear form

Never submit passwords through Google Forms.
 This content is neither created nor endorsed by Google - [Terms of Service](#) - [Privacy Policy](#)
 Does this form look suspicious? [Report](#)

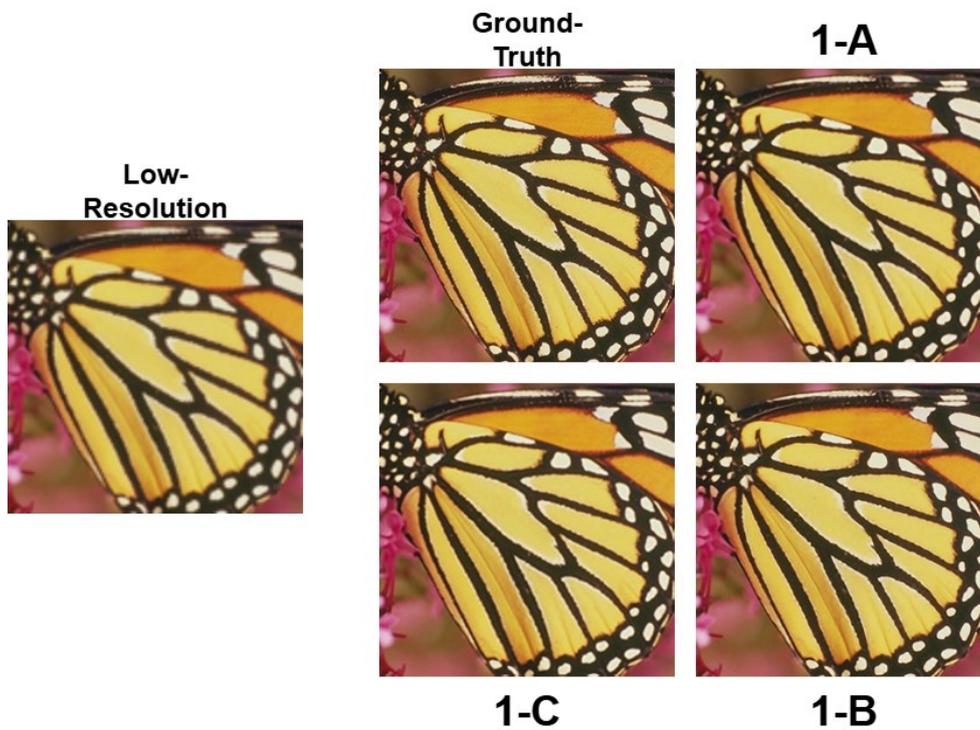
Google Forms

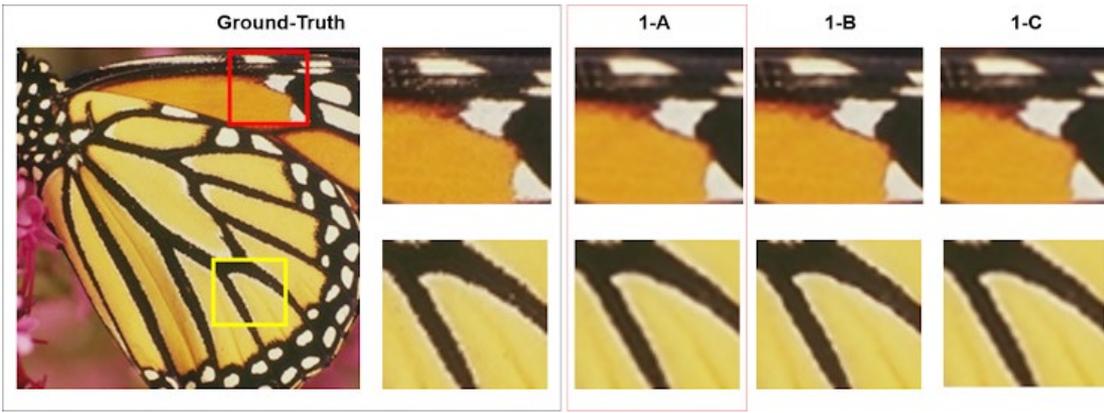
Gambar acak dengan urutan	
Gambar #1 (Dataset Set5 – “Butterfly”)	1-A = EGCNN (<i>baseline</i>) 1-B = EGCNN+ET 1-C = SwinIR
Gambar #2 (Dataset Set14 – “Baboon”)	2-A = EGCNN+ET 2-B = SwinIR 2-C = EGCNN (<i>baseline</i>)
Gambar #3 (Dataset Set14 – “Coastguard”)	3-A = SwinIR 3-B = EGCNN (<i>baseline</i>) 3-C = EGCNN+ET
Gambar #4 (Dataset BSD100 – “Palace”)	4-A = EGCNN+ET 4-B = EGCNN (<i>baseline</i>) 4-C = SwinIR
Gambar #5 (Dataset urban100 – “City”)	5-A = SwinIR 5-B = EGCNN+ET 5-C = EGCNN (<i>baseline</i>)

Tabel skala penilaian

Skala	Deskripsi
1	Tidak ada pemulihan warna, kontras rendah, distorsi tekstur, artefak tepi, dan visibilitas buruk.
2	Pemulihan warna sebagian, peningkatan kontras, distorsi tekstur, artefak tepi, dan visibilitas buruk.
3	Pemulihan warna, peningkatan kontras, tekstur realistis, artefak tepi lokal, dan visibilitas yang dapat diterima.
4	Pemulihan warna, peningkatan kontras, tekstur realistis, pemulihan artefak tepi yang lebih baik, dan visibilitas yang lebih baik.
5	Pemulihan warna, peningkatan kontras, tekstur realistis, artefak tepi tajam, dan visibilitas yang sangat baik.

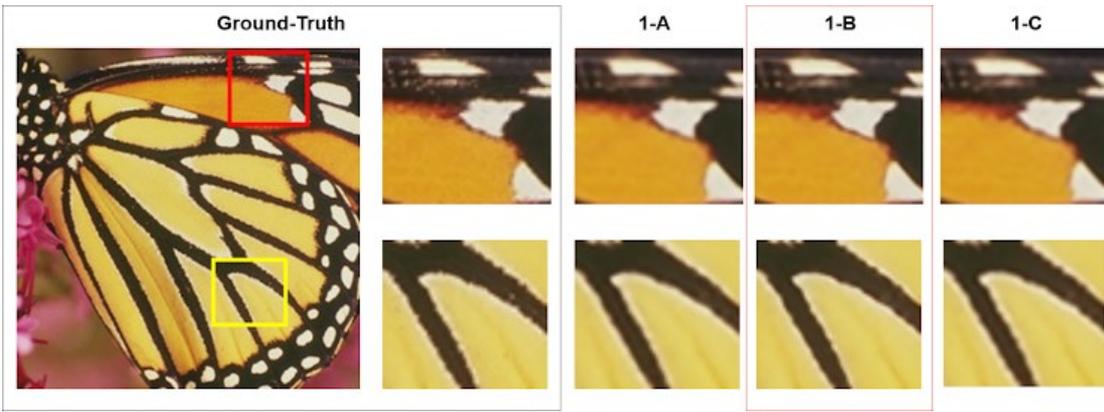
Gambar #1 (Dataset Set5 – “Butterfly”)





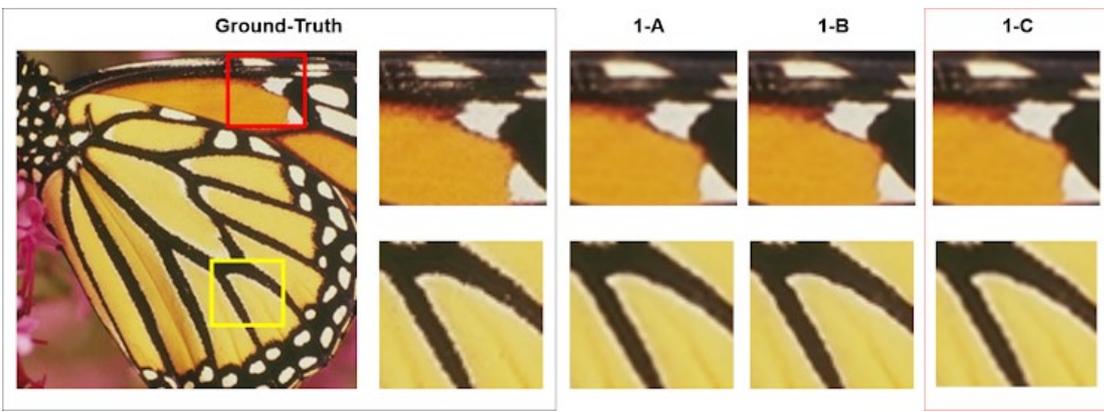
Beri skala 1 - 5 *

1	2	3	4	5
☆	☆	☆	☆	☆



Beri skala 1 - 5 *

1	2	3	4	5
☆	☆	☆	☆	☆



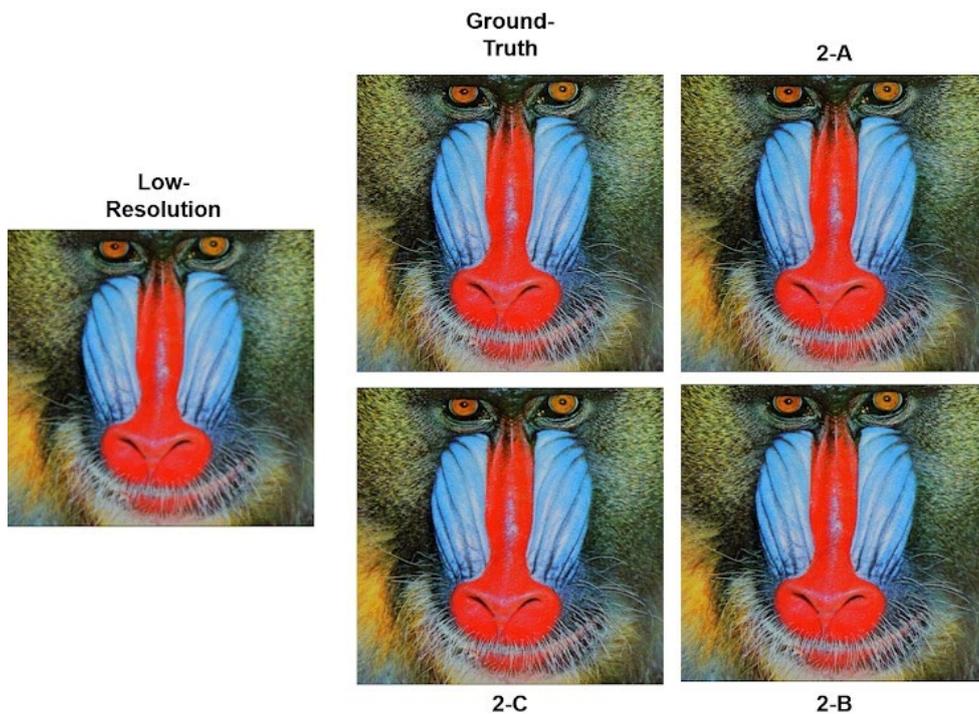
Beri skala 1 - 5 *

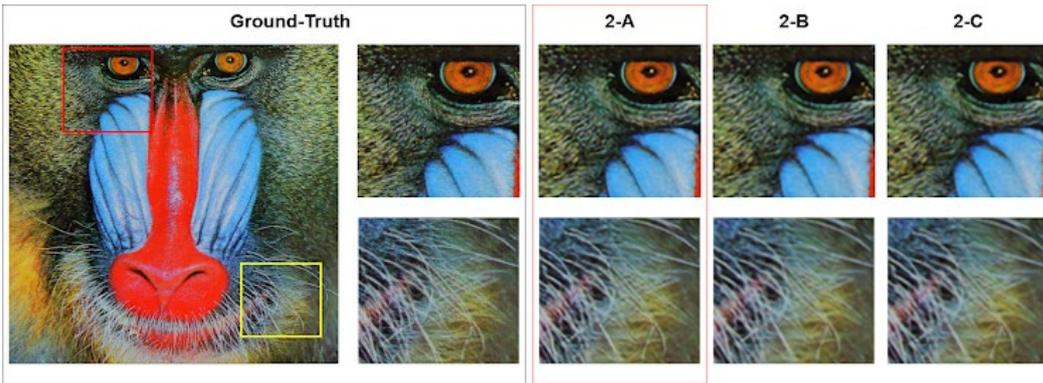
1	2	3	4	5
☆	☆	☆	☆	☆

Tabel skala penilaian

Skala	Deskripsi
1	Tidak ada pemulihan warna, kontras rendah, distorsi tekstur, artefak tepi, dan visibilitas buruk.
2	Pemulihan warna sebagian, peningkatan kontras, distorsi tekstur, artefak tepi, dan visibilitas buruk.
3	Pemulihan warna, peningkatan kontras, tekstur realistis, artefak tepi lokal, dan visibilitas yang dapat diterima.
4	Pemulihan warna, peningkatan kontras, tekstur realistis, pemulihan artefak tepi yang lebih baik, dan visibilitas yang lebih baik.
5	Pemulihan warna, peningkatan kontras, tekstur realistis, artefak tepi tajam, dan visibilitas yang sangat baik.

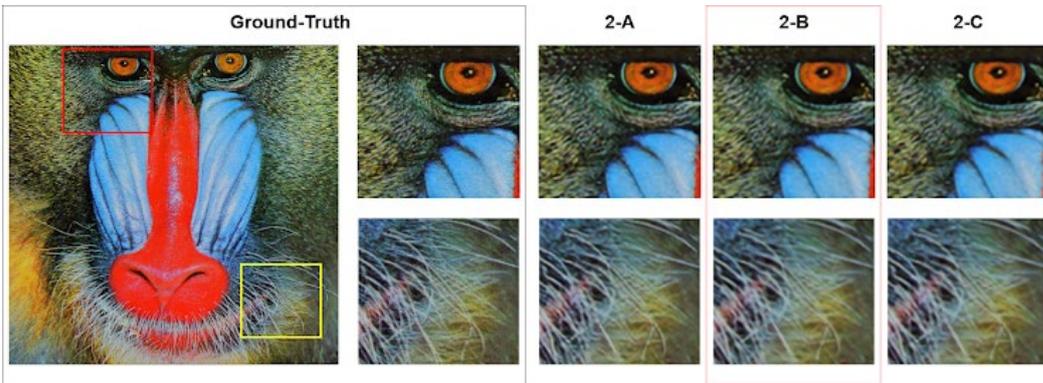
Gambar #2 (Dataset Set14 – “Baboon”)





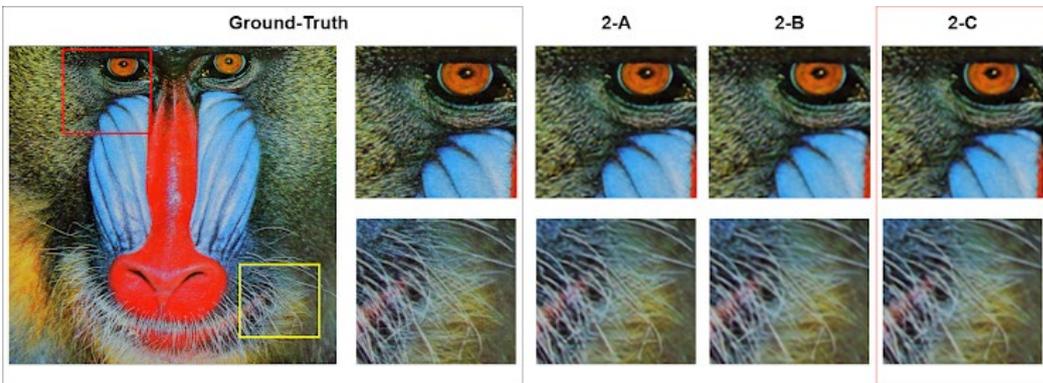
Beri skala 1 - 5 *

1 2 3 4 5
☆ ☆ ☆ ☆ ☆



Beri skala 1 - 5 *

1 2 3 4 5
☆ ☆ ☆ ☆ ☆



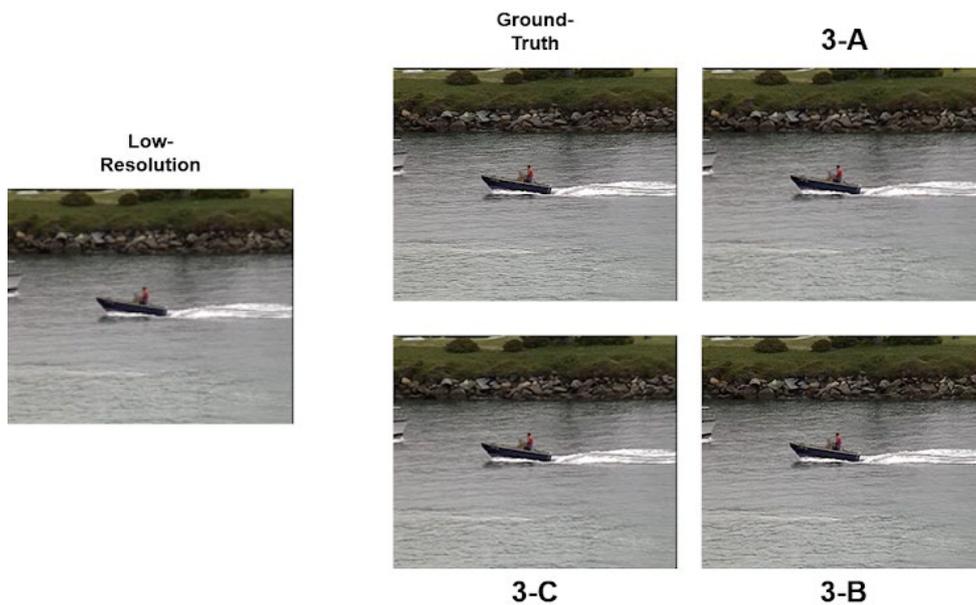
Beri skala 1 - 5 *

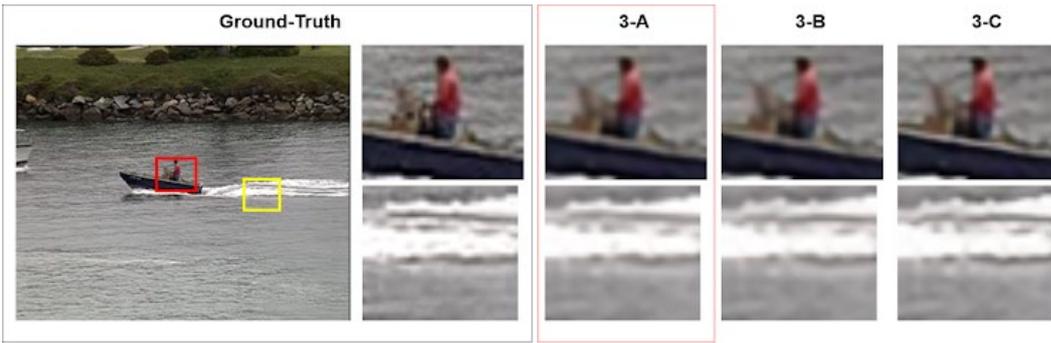
1 2 3 4 5
☆ ☆ ☆ ☆ ☆

Tabel skala penilaian

Skala	Deskripsi
1	Tidak ada pemulihan warna, kontras rendah, distorsi tekstur, artefak tepi, dan visibilitas buruk.
2	Pemulihan warna sebagian, peningkatan kontras, distorsi tekstur, artefak tepi, dan visibilitas buruk.
3	Pemulihan warna, peningkatan kontras, tekstur realistis, artefak tepi lokal, dan visibilitas yang dapat diterima.
4	Pemulihan warna, peningkatan kontras, tekstur realistis, pemulihan artefak tepi yang lebih baik, dan visibilitas yang lebih baik.
5	Pemulihan warna, peningkatan kontras, tekstur realistis, artefak tepi tajam, dan visibilitas yang sangat baik.

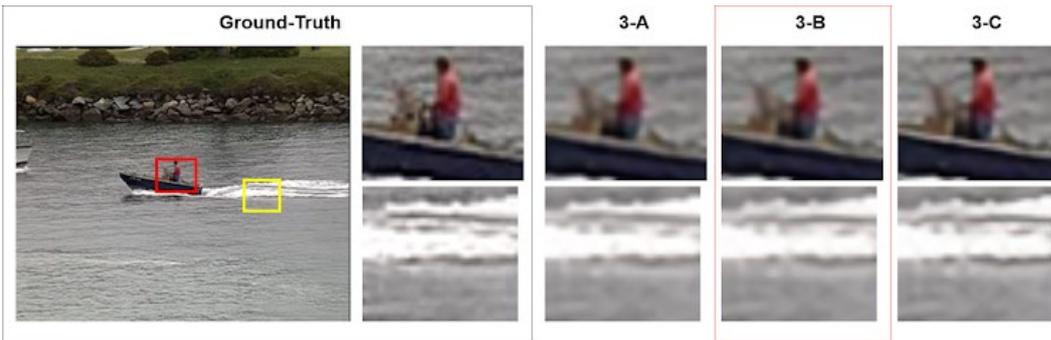
Gambar #3 (Dataset Set14 – “Coastguard”)





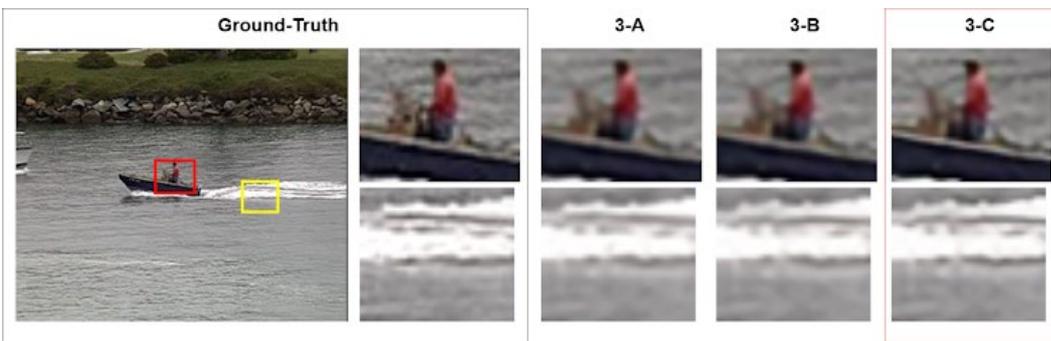
Beri skala 1 - 5 *

1	2	3	4	5
☆	☆	☆	☆	☆



Beri skala 1 - 5 *

1	2	3	4	5
☆	☆	☆	☆	☆



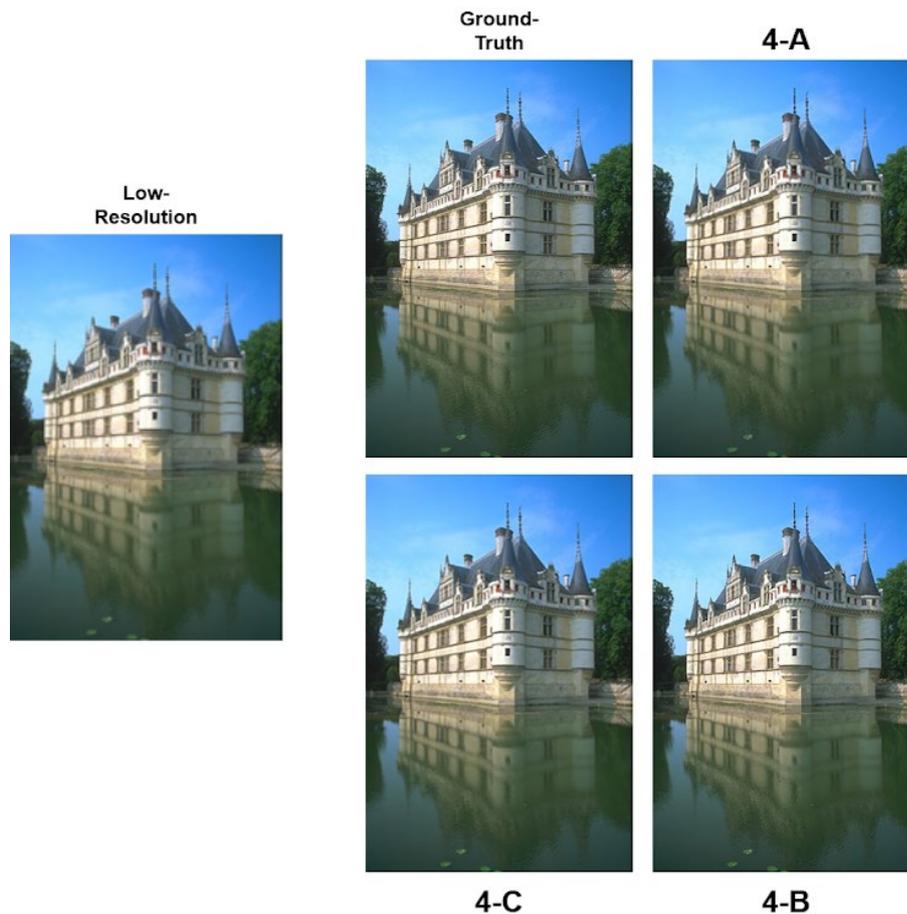
Beri skala 1 - 5 *

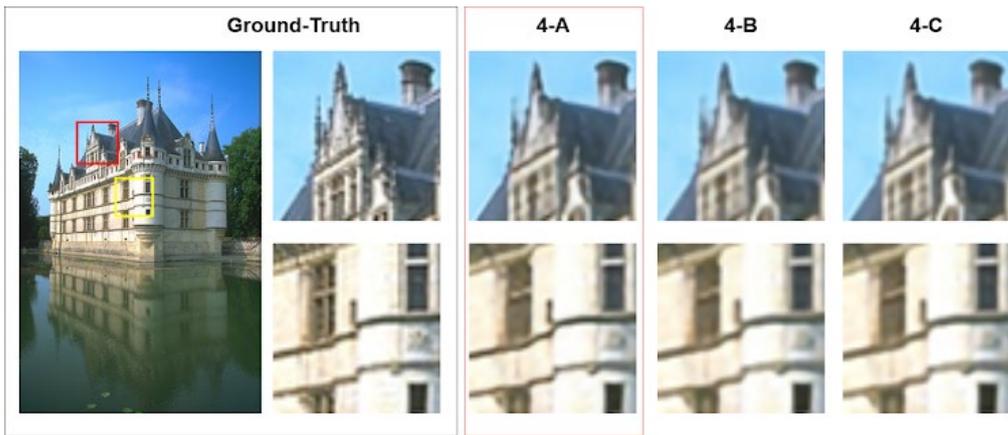
1	2	3	4	5
☆	☆	☆	☆	☆

Tabel skala penilaian

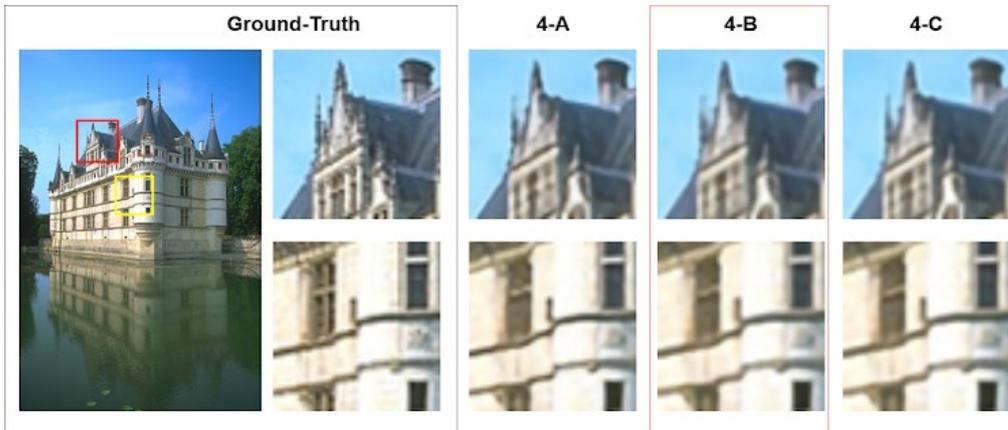
Skala	Deskripsi
1	Tidak ada pemulihan warna, kontras rendah, distorsi tekstur, artefak tepi, dan visibilitas buruk.
2	Pemulihan warna sebagian, peningkatan kontras, distorsi tekstur, artefak tepi, dan visibilitas buruk.
3	Pemulihan warna, peningkatan kontras, tekstur realistis, artefak tepi lokal, dan visibilitas yang dapat diterima.
4	Pemulihan warna, peningkatan kontras, tekstur realistis, pemulihan artefak tepi yang lebih baik, dan visibilitas yang lebih baik.
5	Pemulihan warna, peningkatan kontras, tekstur realistis, artefak tepi tajam, dan visibilitas yang sangat baik.

Gambar #4 (Dataset BSD100 – "Palace")





Beri skala 1 - 5 *



Beri skala 1 - 5 *



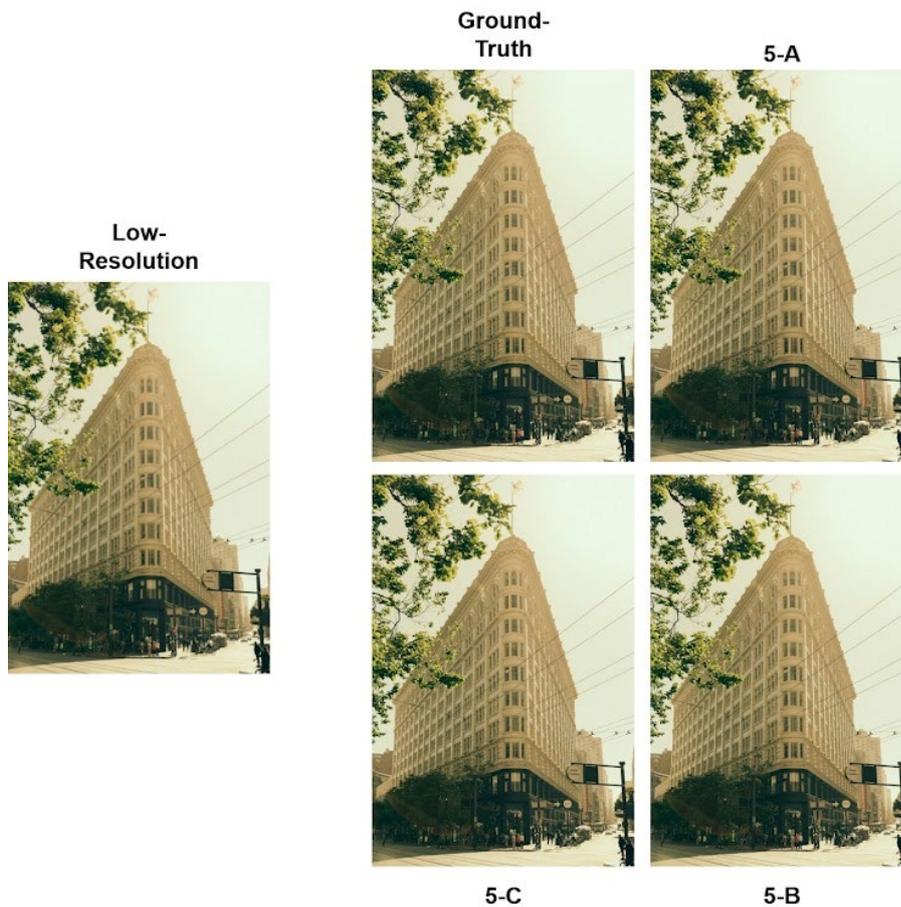
Beri skala 1 - 5 *

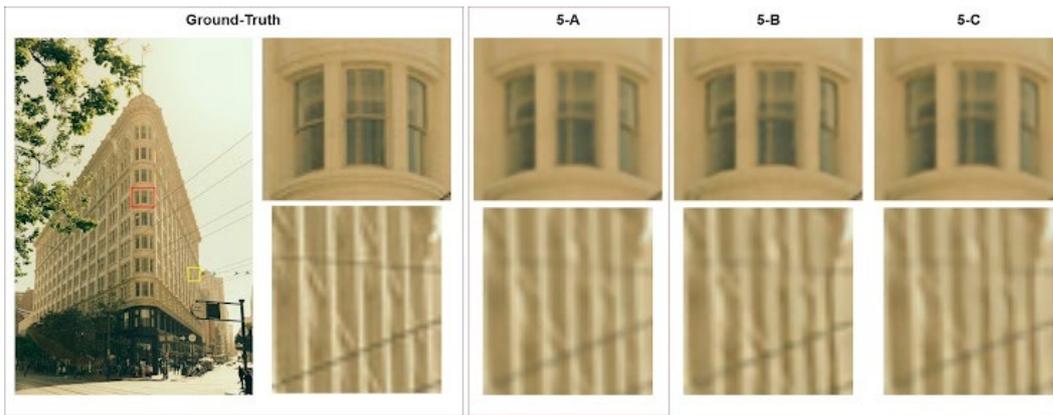


Tabel skala penilaian

Skala	Deskripsi
1	Tidak ada pemulihan warna, kontras rendah, distorsi tekstur, artefak tepi, dan visibilitas buruk.
2	Pemulihan warna sebagian, peningkatan kontras, distorsi tekstur, artefak tepi, dan visibilitas buruk.
3	Pemulihan warna, peningkatan kontras, tekstur realistis, artefak tepi lokal, dan visibilitas yang dapat diterima.
4	Pemulihan warna, peningkatan kontras, tekstur realistis, pemulihan artefak tepi yang lebih baik, dan visibilitas yang lebih baik.
5	Pemulihan warna, peningkatan kontras, tekstur realistis, artefak tepi tajam, dan visibilitas yang sangat baik.

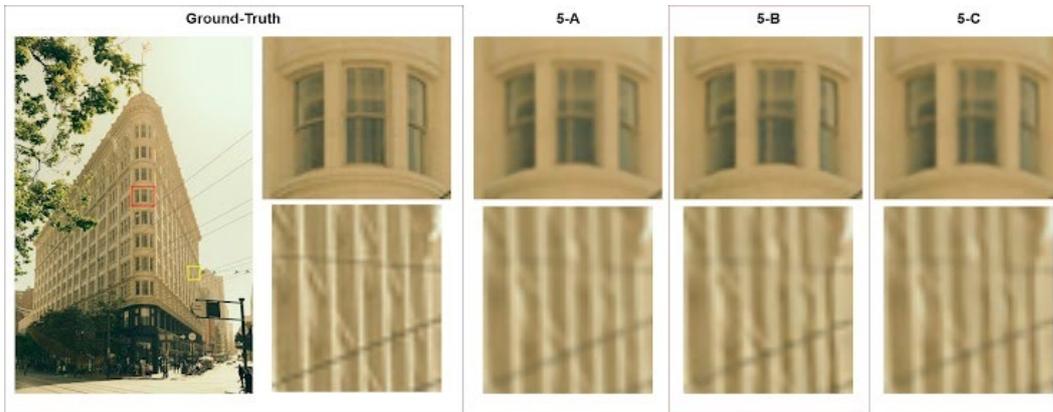
Gambar #5 (Dataset urban100- "City")





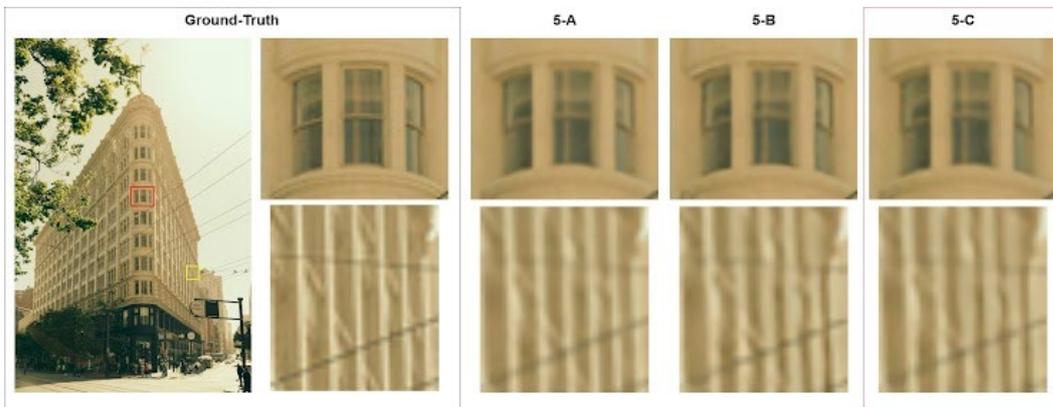
Beri skala 1 - 5 *

1	2	3	4	5
☆	☆	☆	☆	☆



Beri skala 1 - 5 *

1	2	3	4	5
☆	☆	☆	☆	☆



Beri skala 1 - 5 *

1	2	3	4	5
☆	☆	☆	☆	☆

BIODATA PENULIS



Penulis dilahirkan di Sidoarjo pada tanggal 12 April 2002, merupakan anak kedua dari tiga bersaudara. Pendidikan formal penulis dimulai di TK Al Zaitun Sidoarjo, kemudian melanjutkan ke SDN Kepuh Kiriman 1 Waru, SMPN 1 Sidoarjo, dan SMAN 6 Surabaya. Setelah menyelesaikan pendidikan menengah, penulis melanjutkan studi Strata Satu (S1) di Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Airlangga dan lulus dengan predikat *cum laude*. Pada tahun 2024, penulis melanjutkan pendidikan Strata Dua (S2) di Program Studi Sistem Informasi, Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember (ITS). Selama menempuh pendidikan, penulis aktif mengikuti berbagai kegiatan akademik dan pengembangan diri, baik yang diselenggarakan di lingkungan kampus maupun oleh komunitas IT. Penulis juga mengikuti program Bangkit Academy yang diselenggarakan oleh Google, dengan fokus pada *Cloud Computing* dan *Mobile Development*. Saat ini, penulis bekerja sebagai Pegawai Negeri Sipil (PNS) di Badan Pendapatan Daerah Provinsi Jawa Timur (Bapenda Jatim) dengan jabatan sebagai IT Programmer. Minat utama penulis terletak pada pengembangan sistem informasi, rekayasa perangkat lunak, kecerdasan buatan, serta penelitian di bidang *Single Image Super Resolution (SISR)*. Penulis dikenal sebagai pribadi yang disiplin, tekun, dan memiliki semangat belajar yang tinggi.