

**IMPLEMENTASI METODE ResNet-18 UNTUK EKSTRAKSI FITUR TINGKAT  
KERUSAKAN BANGUNAN PASCA BENCANA ALAM**

**SKRIPSI**

**Oleh :  
AL IKHLASUL HAKIMI  
NIM. 210605110059**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2025**

**IMPLEMENTASI METODE ResNet-18 UNTUK EKSTRAKSI FITUR  
TINGKAT KERUSAKAN BANGUNAN PASCA BENCANA ALAM**

**SKRIPSI**

Diajukan kepada:  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :  
**AL IKHLASUL HAKIMI**  
**NIM. 210605110059**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2025**

## HALAMAN PERSETUJUAN

### IMPLEMENTASI METODE ResNet-18 UNTUK EKSTRAKSI FITUR TINGKAT KERUSAKAN BANGUNAN PASCA BENCANA ALAM

#### SKRIPSI

Oleh :  
**AL IKHLASUL HAKIMI**  
NIM. 210605110059

Telah Diperiksa dan Disetujui untuk Diuji:  
Tanggal: 20 Juni 2025

Pembimbing I,



Dr. Agung Teguh Wibowo Almais, M.T  
NIP. 19860301 202321 1 016

Pembimbing II,



Fajar Rohman Hariri, M.Kom  
NIP. 19890515 201801 1 001

Mengetahui,

Ketua Program Studi Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Inchaenrul Kurniawan, M.MT., IPU  
NIP. 19771020 200912 1 001

## HALAMAN PENGESAHAN

### IMPLEMENTASI METODE ResNet-18 UNTUK EKSTRAKSI FITUR TINGKAT KERUSAKAN BANGUNAN PASCA BENCANA ALAM

#### SKRIPSI

Oleh :  
**AL IKHLASUL HAKIMI**  
NIM. 210605110059

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer ( S.Kom )  
Tanggal: 24 Juni 2025

#### Susunan Dewan Penguji

Ketua Penguji : Dr. Totok Chamidy, M.Kom  
NIP. 19691222 200604 1 001

Anggota Penguji I : Roro Inda Melani, M.T., M.Sc  
NIP. 19780925 20050 1 2008

Anggota Penguji II : Dr. Agung Teguh Wibowo Almais, M.T  
NIP. 19860301 202321 1 016

Anggota Penguji III : Fajar Rohman Hariri, M.Kom  
NIP. 19890515 201801 1 001

(  )  
(  )  
(  )  
(  )

Mengetahui dan Mengesahkan,  
Ketua Program Studi Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



  
Dr. Fachrul Kurniawan, M.MT., IPU  
NIP. 19771020 200912 1 001

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Al Ikhlasul Hakimi  
NIM : 210605110059  
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika  
Judul Skripsi : Implementasi Metode ResNet-18 Untuk Ekstraksi  
Fitur  
Tingkat Kerusakan Bangunan Pasca Bencana  
Alam

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 20 Juni 2025

Yang membuat pernyataan,



Al Ikhlasul Hakimi  
NIM. 210605110059

## **MOTTO**

*“Kita bisa membuat rencana untuk sekian tahun ke depan, tapi percuma jika selesai di tengah jalan”-The Adams*

## **HALAMAN PERSEMBAHAN**

Alhamdulillah rabbil ‘alamin. Segala puji bagi Allah SWT atas limpahan rahmat, hidayah, dan kekuatan-Nya sehingga skripsi ini dapat diselesaikan dengan baik.

Shalawat dan salam semoga tercurah kepada Nabi Muhammad SAW, sang pembawa cahaya ilmu dan petunjuk hidup bagi seluruh umat manusia.

Skripsi ini kupersembahkan untuk Ayah dan Ibu tercinta. Terima kasih atas kasih sayang, doa, dan dukungan yang tiada henti. Segala pencapaian ini tak lepas dari pengorbanan Ayah dan Ibu.

Untuk saudara-saudaraku, terima kasih atas semangat dan kebersamaan yang selalu menguatkan.

Kepada dosen pembimbing dan seluruh dosen Teknik Informatika, terima kasih atas ilmu, bimbingan, dan kesabarannya selama proses studi ini.

Untuk teman-teman seperjuangan, terima kasih atas kebersamaan, tawa, dan semangat yang kita bagi selama ini.

Terakhir, untuk diriku sendiri. Terima kasih telah bertahan, terus berjuang, dan tidak menyerah.

## KATA PENGANTAR

*Assalamu'alaikum Warahmatullahi Wabarakatuh*

Segala puji dan rasa syukur penulis panjatkan ke hadirat Allah Subhanahu wa Ta'ala atas limpahan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Shalawat dan salam semoga senantiasa tercurah kepada Nabi Muhammad SAW, semoga kita semua kelak memperoleh syafaat beliau di hari kiamat. Aamiin.

Penulis ucapkan terima kasih yang sebesar-besarnya kepada seluruh pihak yang memberikan dukungan dan motivasi kepada penulis sehingga dapat menyelesaikan skripsi ini. Ucapan terima kasih ini penulis disampaikan kepada:

1. Prof. Dr. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Ir. Fachrul Kurniawan, M.MT., IPU selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Agung Teguh Wibowo Almais, MT, selaku dosen pembimbing 1 serta mentor yang telah memberikan bimbingan, saran, kesabaran dan dukungan yang sangat berarti bagi penulis selama proses penelitian.
5. Fajar Rohman Hariri, M.Kom, selaku dosen pembimbing 2, atas arahan dan pencerahan yang sangat membantu dalam menyempurnakan karya ini.



6. Bapak Dr. Totok Chamidy, M.Kom, selaku dosen Penguji I dan Ibu Roro Inda melani, M.T., M.Sc, selaku dosen Penguji II yang telah memberikan kritik dan saran yang membangun dalam penyusunan skripsi ini.
7. Seluruh staf dan dosen Program Studi Teknik Informatika yang telah memberikan ilmu, dukungan, dan fasilitas kepada penulis selama masa studi.
8. Ayahanda Afrizal (Alm), Ibunda Ermawati, serta saudara-saudara penulis Annajmiyatul Fitria dan Roza Oktavia yang senantiasa menjadi sumber semangat, memberikan dukungan tanpa henti, serta menyertai dengan doa-doa terbaik, sehingga penulisan skripsi ini dapat terselesaikan dengan baik. Semoga Allah SWT senantiasa menganugerahkan kesehatan dan perlindungan, agar senantiasa dapat mendampingi setiap langkah dan pencapaian penulis ke depannya.
9. Teman seperjuangan dan sepembimbingan, yang beranggotakan Charles, Gigih, Najah, Putri, Raka, Roihan, Salsabila, Iza, Annisa, dan Iqlima yang telah membantu banyak dalam proses pengerjaan skripsi ini, semoga selalu diberikan kesehatan.
10. Rekan seperjuangan “Kontrakan Pria Tampan”, yang beranggotakan Rehan, Haikal, Noviansyah, Gianda, Safril, Musa, Khalif, Zufar, Rafi, Fuaidil, Irul, Daffa, Fauzil dan Dika. Terima kasih telah menemani hari-hari penulis dan terima kasih sudah mau menemani suka duka serta berbagi tawa dan air mata. Sukses untuk kalian semua.

11. Shaima Tasya, seseorang yang sangat luar biasa yang senantiasa selalu menjadi sumber semangat, memberikan dukungan tanpa henti, serta menyertai dengan doa-doa terbaik, sehingga penulisan skripsi ini dapat terselesaikan dengan baik.
  12. Seluruh keluarga besar Teknik Informatika, khususnya Angkatan 2021 “ASTER”, atas segala ilmu, semangat, serta pengalaman berharga yang telah dibagikan selama masa perkuliahan. Semoga tali silaturahmi ini senantiasa terjaga, dan kita semua dapat menggapai cita-cita dan impian bersama.
  13. Teman-teman band dari Blessembler dan Zzimplify yaitu Hamzi, Ali, Iqbal, Andini, Shafira, Zacky, Hanif, Zidan, dan Yusuf yang senantiasa menemani penulis dikala mengalami kendala dalam menyusun skripsi, terimakasih karena telah memberikan hiburan disaat-saat penyusunan skripsi ini. Semoga tali silaturrahi ini tetap terjalin dan band kita bisa lebih maju di masa yang akan datang.
  14. DPH Kabinet Menyala yaitu Rozak, Fafa, Yusron, Andini, Minke, dan seluruh jajaran pengurus Kabinet Menyala UKM Kommust, dan seluruh teman-teman dari UKM Kommust yang telah menemani masa-masa kuliah dan memberikan kenangan yang indah selama masa kuliah.
  15. Seluruh pihak yang telah terlibat, baik secara langsung maupun tidak langsung dari awal perkuliahan hingga akhir penulisan skripsi ini
- Penulis berharap semoga skripsi ini dapat memberikan manfaat di masa yang akan datang. Penulis juga menyadari bahwa karya ini masih memiliki

kekurangan. Oleh karena itu, penulis sangat terbuka terhadap segala bentuk masukan dan saran yang membangun untuk pengembangan lebih lanjut.

*Wassalamu'alaikum warahmatullahi wabarakatuh.*

Malang, 16 Juni 2025

Penulis

## DAFTAR ISI

|                                                            |              |
|------------------------------------------------------------|--------------|
| <b>IMPLEMENTASI METODE RESNET-18 UNTUK EKSTRAKSI FITUR</b> |              |
| <b>TINGKAT KERUSAKAN BANGUNAN PASCA BENCANA ALAM.....</b>  | <b>II</b>    |
| <b>HALAMAN PERSETUJUAN.....</b>                            | <b>III</b>   |
| <b>HALAMAN PENGESAHAN.....</b>                             | <b>IV</b>    |
| <b>PERNYATAAN KEASLIAN TULISAN.....</b>                    | <b>V</b>     |
| <b>MOTTO.....</b>                                          | <b>VI</b>    |
| <b>HALAMAN PERSEMBAHAN.....</b>                            | <b>VII</b>   |
| <b>KATA PENGANTAR.....</b>                                 | <b>VIII</b>  |
| <b>DAFTAR ISI.....</b>                                     | <b>XII</b>   |
| <b>DAFTAR GAMBAR.....</b>                                  | <b>XIV</b>   |
| <b>DAFTAR TABEL.....</b>                                   | <b>XV</b>    |
| <b>ABSTRAK.....</b>                                        | <b>XVI</b>   |
| <b>ABSTRACT.....</b>                                       | <b>XVII</b>  |
| <b>المخلص.....</b>                                         | <b>XVIII</b> |
| <b>BAB I PENDAHULUAN.....</b>                              | <b>1</b>     |
| 1.1 LATAR BELAKANG.....                                    | 1            |
| 1.2 RUMUSAN MASALAH.....                                   | 5            |
| 1.3 BATASAN MASALAH.....                                   | 5            |
| 1.4 TUJUAN PENELITIAN.....                                 | 6            |
| 1.5 MANFAAT PENELITIAN.....                                | 6            |
| <b>BAB II STUDI PUSTAKA.....</b>                           | <b>8</b>     |
| 2.1 PENELITIAN TERDAHULU.....                              | 8            |
| 2.2 <i>CONVOLUTIONAL NEURAL NETWORKS</i> .....             | 12           |
| 2.3 <i>TRANSFER LEARNING</i> .....                         | 15           |
| 2.4 RESNET-18.....                                         | 16           |
| 2.5 EKSTRAKSI FITUR.....                                   | 18           |
| 2.6 <i>PRINCIPAL COMPONENT ANALYSIS</i> .....              | 20           |
| 2.7 <i>K-MEANS CLUSTERING</i> .....                        | 21           |
| <b>BAB III METODE PENELITIAN.....</b>                      | <b>23</b>    |
| 3.1 DESAIN SISTEM.....                                     | 23           |
| 3.2 <i>DATA PREPARATION</i> .....                          | 24           |
| <b>3.2.1 Data Akuisisi.....</b>                            | <b>24</b>    |
| <b>3.2.2 Image Preprocessing.....</b>                      | <b>25</b>    |
| 3.3 IMPLEMENTASI METODE RESNET-18.....                     | 27           |
| <b>3.3.1 Arsitektur ResNet-18.....</b>                     | <b>28</b>    |
| <b>3.3.2 Proses Implementasi ResNet-18.....</b>            | <b>38</b>    |
| 3.4 <i>FEATURE ANALYSIS/FEATURE EVALUATION</i> .....       | 39           |
| <b>3.4.1 Principal Component Analysis (PCA).....</b>       | <b>40</b>    |
| <b>3.4.2 K-Means Clustering.....</b>                       | <b>41</b>    |
| <b>3.4.3 Proses Evaluasi.....</b>                          | <b>41</b>    |
| <b>BAB IV.....</b>                                         | <b>43</b>    |
| <b>UJI COBA DAN PEMBAHASAN.....</b>                        | <b>43</b>    |
| 4.1 DATA PENGUJIAN.....                                    | 43           |

|              |                                                                                               |           |
|--------------|-----------------------------------------------------------------------------------------------|-----------|
| 4.2          | HASIL EKSTRAKSI FITUR RESNET-18 .....                                                         | 44        |
| 4.3          | SKENARIO PENGUJIAN.....                                                                       | 63        |
| 4.4          | HASIL UJI COBA.....                                                                           | 64        |
| 4.4.1        | <b>Enam Komponen Utama Hasil PCA .....</b>                                                    | <b>65</b> |
| 4.4.2        | <b>Skenario 1: <i>Clustering</i> Menggunakan Nilai PCA (PC1 dan PC2)</b><br>67                |           |
| 4.4.3        | <b>Skenario 2: <i>Clustering</i> Menggunakan Fitur ResNet-18 Tanpa PCA</b> 68                 |           |
| 4.5          | PEMBAHASAN .....                                                                              | 70        |
| 4.6          | INTEGRASI ISLAM.....                                                                          | 72        |
| 4.6.1        | <b><i>Hablum Minallah</i> (Hubungan Manusia dengan Allah) .....</b>                           | <b>73</b> |
| 4.6.2        | <b><i>Hablum Minannas</i> (Hubungan Manusia dengan Sesama Manusia).....</b>                   | <b>74</b> |
| 4.6.3        | <b><i>Hablum Minal Alam</i> (Hubungan Manusia dengan Alam).....</b>                           | <b>76</b> |
| <b>BAB V</b> | <b>.....</b>                                                                                  | <b>78</b> |
| 5.1          | KESIMPULAN.....                                                                               | 78        |
| 5.2          | SARAN.....                                                                                    | 79        |
|              | <b>DAFTAR PUSTAKA .....</b>                                                                   | <b>81</b> |
|              | <b>LAMPIRAN.....</b>                                                                          | <b>85</b> |
|              | HASIL <i>CLUSTERING</i> FITUR SETELAH DILAKUKAN PROSES REDUKSI DIMENSI<br>MENGUNAKAN PCA..... | 86        |
|              | HASIL <i>CLUSTERING</i> FITUR TANPA DILAKUKAN PROSES REDUKSI DIMENSI<br>MENGUNAKAN PCA.....   | 96        |

## DAFTAR GAMBAR

|                                                                                        |    |
|----------------------------------------------------------------------------------------|----|
| Gambar 2.1 CNN dalam Machine Learning (Chiu et al., 2022).....                         | 12 |
| Gambar 2.2 CNN dalam Deep Learning (Valliani, 2019) .....                              | 13 |
| Gambar 2.3 ResNet dalam Deep Learning (Cao et al., 2018).....                          | 16 |
| Gambar 3.1 Diagram Sistem.....                                                         | 23 |
| Gambar 3.2 Data Gambar Kerusakan Bangunan .....                                        | 25 |
| Gambar 3.3 Flowchart ResNet-18 (Francis & Prakash Verma, 2025) .....                   | 27 |
| Gambar 4.1 Input Proses Konvolusi Pertama .....                                        | 44 |
| Gambar 4.2 <i>Output</i> Konvolusi Pertama .....                                       | 46 |
| Gambar 4.3 <i>Output</i> Proses Max Pooling.....                                       | 47 |
| Gambar 4.4 Konvolusi Pertama Pada Blok Residual Pertama .....                          | 48 |
| Gambar 4.5 Konvolusi Kedua Pada Blok Residual Pertama .....                            | 49 |
| Gambar 4.6 Konvolusi Pertama Pada Blok Residual Kedua .....                            | 50 |
| Gambar 4.7 Konvolusi Kedua Pada Blok Residual Kedua.....                               | 51 |
| Gambar 4.8 Konvolusi Pertama Pada Blok Residual Ketiga.....                            | 52 |
| Gambar 4.9 Konvolusi Kedua Pada Blok Residual Ketiga.....                              | 53 |
| Gambar 4.10 Konvolusi Pertama Pada Blok Residual Keempat .....                         | 54 |
| Gambar 4.11 Konvolusi Kedua Pada Blok Residual Keempat.....                            | 55 |
| Gambar 4.12 Konvolusi Pertama Pada Blok Residual Kelima.....                           | 56 |
| Gambar 4.13 Konvolusi Kedua Pada Blok Residual Kelima .....                            | 57 |
| Gambar 4.14 Konvolusi Pertama Pada Blok Residual Keenam .....                          | 58 |
| Gambar 4.15 Konvolusi Kedua Pada Blok Residual Keenam .....                            | 59 |
| Gambar 4.16 Konvolusi Pertama Pada Blok Residual Ketujuh.....                          | 60 |
| Gambar 4.17 Konvolusi Kedua Pada Blok Residual Ketujuh .....                           | 61 |
| Gambar 4.18 Konvolusi Pertama Pada Blok Residual Terakhir.....                         | 62 |
| Gambar 4.19 Konvolusi Kedua Pada Blok Residual Terakhir .....                          | 62 |
| Gambar 4.20 Hasil Clustering K-Means PC1 dan PC2 Pada Data Gambar 1 .....              | 68 |
| Gambar 4.21 Hasil<br>Clustering K-Means Tanpa Menggunakan PCA dari data gambar 1 ..... | 70 |

## DAFTAR TABEL

|                                                                            |    |
|----------------------------------------------------------------------------|----|
| Tabel 2.1 Penelitian Terdahulu .....                                       | 11 |
| Tabel 3.1 Contoh Input 64x64 Channel R .....                               | 30 |
| Tabel 3.2 Contoh Input 64x64 Channel G .....                               | 30 |
| Tabel 3.3 Contoh Input 64x64 Channel B .....                               | 30 |
| Tabel 3.4 Contoh Kernel 7x7 Channel R.....                                 | 31 |
| Tabel 3.5 Contoh Kernel 7x7 Channel G.....                                 | 31 |
| Tabel 3.6 Contoh Kernel 7x7 Channel B.....                                 | 31 |
| Tabel 3.7 Contoh Hasil Proses Konvolusi .....                              | 34 |
| Tabel 3.8 Contoh Hasil Proses Max Pooling .....                            | 34 |
| Tabel 3.9 Contoh Hasil Proses Downsampling.....                            | 36 |
| Tabel 3.10 Contoh Hasil Residual Block.....                                | 36 |
| Tabel 3.11 Contoh                                                          |    |
| Hasil Proses Residual Block Final Setelah Dilakukan Skip Connection.....   | 37 |
| Tabel 3.12 Contoh                                                          |    |
| Hasil Fitur Setelah Dilakukan Proses Activation Function (ReLU) .....      | 37 |
| Tabel 4.1 Nilai Pixel (R) Sebagai Input ResNet-18.....                     | 45 |
| Tabel 4.2 Nilai Pixel (G) Sebagai Input ResNet-18.....                     | 45 |
| Tabel 4.3 Nilai Pixel (B) Sebagai Input ResNet-18.....                     | 45 |
| Tabel 4.4 Hasil Proses Konvolusi Conv1 - 7×7, 64 filter, stride = 2 .....  | 46 |
| Tabel 4.5 Fitur Hasil Proses Max Pooling - 3×3, stride = 2 .....           | 47 |
| Tabel 4.6 Fitur Hasil Proses Conv2_1a - 3×3, 64 filter, stride = 1 .....   | 48 |
| Tabel 4.7 Fitur Hasil Proses Conv2_1b - 3×3, 64 filter, stride = 1.....    | 49 |
| Tabel 4.8 Fitur Hasil Proses Conv2_2a - 3×3, 64 filter, stride = 1 .....   | 50 |
| Tabel 4.9 Fitur Hasil Proses Conv2_2b - 3×3, 64 filter, stride = 1.....    | 51 |
| Tabel 4.10 Fitur Hasil Proses Conv3_1a - 3×3, 128 filter, stride = 2.....  | 52 |
| Tabel 4.11 Fitur Hasil Proses Conv3_1b - 3×3, 128 filter, stride = 1.....  | 53 |
| Tabel 4.12 Fitur Hasil Proses Conv3_2a - 3×3, 128 filter, stride = 1 ..... | 54 |
| Tabel 4.13 Fitur Hasil Proses Conv3_2b - 3×3, 128 filter, stride = 1.....  | 55 |
| Tabel 4.14 Fitur Hasil Proses Conv4_1a - 3×3, 256 filter, stride = 2.....  | 56 |
| Tabel 4.15 Fitur Hasil Proses Conv4_1b - 3×3, 256 filter, stride = 1.....  | 57 |
| Tabel 4.16 Fitur Hasil Proses Conv4_2a - 3×3, 256 filter, stride = 1 ..... | 58 |
| Tabel 4.17 Fitur Hasil Proses Conv4_2b - 3×3, 256 filter, stride = 1.....  | 59 |
| Tabel 4.18 Fitur Hasil Proses Conv5_1a - 3×3, 512 filter, stride = 2.....  | 60 |
| Tabel 4.19 Fitur Hasil Proses Conv5_1b - 3×3, 512 filter, stride = 1.....  | 61 |
| Tabel 4.20 Fitur Hasil Proses Conv5_2a - 3×3, 512 filter, stride = 1 ..... | 62 |
| Tabel 4.21 Fitur Hasil Proses Conv5_2b - 3×3, 512 filter, stride = 1.....  | 63 |
| Tabel 4.22 Hasil Ekstraksi Fitur.....                                      | 63 |
| Tabel 4.23 Skenario Uji .....                                              | 64 |
| Tabel 4.24 Hasil Analisis PCA .....                                        | 65 |

## ABSTRAK

Hakimi, Al Ikhlasul. 2025. **Implementasi Metode Resnet-18 Untuk Ekstraksi Fitur Tingkat Kerusakan Bangunan Pasca Bencana Alam** Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Agung Teguh Wibowo Almais, MT. (II) Fajar Rohman Hariri, M.Kom.

**Kata Kunci:** *Clustering*, Deteksi kerusakan bangunan, PCA, ResNet-18, *Transfer learning*.

Indonesia adalah negara yang rawan bencana alam, seperti gempa bumi dan banjir, yang sering menyebabkan kerusakan infrastruktur secara masif. Untuk mendukung mitigasi bencana, diperlukan sistem yang mampu mengidentifikasi tingkat kerusakan bangunan secara cepat dan akurat. Penelitian ini menjawab dua rumusan masalah utama, yaitu bagaimana mengimplementasikan metode ResNet-18 untuk ekstraksi fitur citra kerusakan bangunan, serta bagaimana mengevaluasi performa metode tersebut dalam mengenali tingkat kerusakan berdasarkan hasil representasi fitur. Arsitektur ResNet-18 diterapkan sebagai bagian dari transfer learning untuk mengekstraksi fitur visual dari gambar bangunan pascabencana. Fitur berdimensi tinggi yang dihasilkan kemudian direduksi menggunakan PCA agar lebih efisien dan tetap informatif. Hasil dari PCA ini diproses lebih lanjut dengan algoritma K-Means Clustering untuk mengelompokkan gambar berdasarkan pola kerusakan visual yang serupa. Pengujian menunjukkan bahwa kombinasi ResNet-18 dan PCA mampu menghasilkan struktur cluster yang lebih jelas, terpisah, dan stabil secara komputasi dibandingkan dengan pengelompokan langsung pada fitur mentah tanpa reduksi. Reduksi dimensi juga mempercepat proses tanpa mengorbankan kualitas informasi. Kesimpulannya, pendekatan ini terbukti efektif sebagai dasar sistem otomatis dalam mendeteksi tingkat kerusakan bangunan, sehingga dapat membantu pemerintah dan lembaga terkait dalam merespons bencana dengan lebih efisien dan tepat sasaran.



## ABSTRACT

Hakimi, Al Ikhlasul. 2025. **Implementation of Resnet-18 Method for Feature Extraction of Building Damage Levels After Natural Disasters** Thesis. Informatics Engineering Study, Faculty of Science and Technology Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) Dr. Agung Teguh Wibowo Almais, MT. (II) Fajar Rohman Hariri, M.Kom.

**Keyword:** ResNet-18, *Transfer learning*, PCA, *Clustering*, *Building damage detection*.

Indonesia is a country prone to natural disasters, such as earthquakes and floods, which often cause massive infrastructure damage. To support disaster mitigation, a system capable of identifying the level of building damage quickly and accurately is required. This study addresses two main research questions: how to implement the ResNet-18 method for feature extraction of post-disaster building damage images, and how to evaluate the performance of this method in recognizing damage levels based on feature representations. The ResNet-18 architecture is applied as part of transfer learning to extract visual features from post-disaster building images. The high-dimensional features produced are then reduced using PCA to improve efficiency while retaining informative content. The PCA results are further processed using the K-Means Clustering algorithm to group images based on similar visual damage patterns. Testing shows that the combination of ResNet-18 and PCA produces cluster structures that are clearer, more distinct, and computationally stable compared to direct clustering on raw features without reduction. Dimensionality reduction also accelerates the process without compromising information quality. In conclusion, this approach proves effective as the basis for an automatic system in detecting building damage levels, thereby assisting the government and related agencies in responding to disasters more efficiently and accurately.

## الملخص

حكيمي، الإخلاص. 2025. تطبيق أسلوب Resnet-18 لاستخراج خصائص مستويات الضرر في المباني بعد الكوارث الطبيعية أطلوخة . برنامج دراسة هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية في مالانج . مُرشد (I) :لدكتو أغونغ تيغوه ويوو ألميس ،الماجستير (II) .فجر رحمن الحريري ،الماجستير .

الكلمات المفتاحية : شبكة ريسنت ١٨ ، التعلم التحويلي، تحليل المكونات الرئيسية، التجميع، الكشف عن أضرار المباني.

إندونيسيا دولة معرضة للكوارث الطبيعية، مثل الزلازل والفيضانات، والتي غالبًا ما تسبب أضرارًا جسيمة في البنية التحتية. ولدعم التخفيف من آثار الكوارث، يلزم وجود نظام قادر على تحديد مستوى الأضرار التي لحقت بالمباني بسرعة ودقة. تتناول هذه الدراسة مسألتين بحثيتين رئيسيتين: كيفية تنفيذ طريقة ResNet-18 لاستخراج سمات صور الأضرار التي لحقت بالمباني بعد الكوارث، وكيفية تقييم أداء هذه الطريقة في التعرف على مستويات الأضرار بناءً على تمثيلات السمات. تُطبق بنية ResNet-18 كجزء من التعلم التحويلي لاستخراج السمات البصرية من صور المباني بعد الكوارث. ثم يتم تقليل السمات عالية الأبعاد الناتجة باستخدام PCA لتحسين الكفاءة مع الاحتفاظ بالمحتوى الغني بالمعلومات. تتم معالجة نتائج PCA بشكل أكبر باستخدام خوارزمية K-Means Clustering لتجميع الصور بناءً على أنماط الأضرار البصرية المتشابهة. يُظهر الاختبار أن الجمع بين ResNet-18 وتحليل PCA ينتج عنه هياكل عنقودية أكثر وضوحًا وتميزًا واستقرارًا من الناحية الحسابية مقارنةً بالتجميع المباشر على السمات الأولية دون تقليل الأبعاد. يعمل تقليل الأبعاد أيضًا على تسريع العملية دون المساس بجودة المعلومات. في الختام، يثبت هذا النهج فعاليته كأساس لنظام آلي في الكشف عن مستويات الأضرار التي لحقت بالمباني، وبالتالي مساعدة الحكومة والوكالات ذات الصلة في الاستجابة للكوارث بشكل أكثر كفاءة ود

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Indonesia merupakan salah satu negara yang banyak mengalami kejadian bencana alam seperti gempa bumi, banjir, tsunami, gunung meletus, dan sebagainya. Dampak yang ditimbulkan oleh bencana alam tersebut tidak hanya berpengaruh pada aspek sosial dan ekonomi, tetapi juga pada infrastruktur bangunan yang mengalami kerusakan signifikan. Kerusakan bangunan pascabencana alam dapat menyebabkan korban jiwa, gangguan aktivitas masyarakat, serta meningkatnya biaya rehabilitasi dan rekonstruksi. Oleh karena itu, deteksi dini terhadap tingkat kerusakan bangunan sangat penting untuk mendukung upaya mitigasi bencana dan perencanaan rekonstruksi yang lebih efektif.

Dalam menghadapi tantangan tersebut, penting bagi kita untuk merenungkan apa yang diingatkan oleh Al-Qur'an terkait bencana alam yang merupakan ujian dari Allah SWT. Allah berfirman dalam Q.S. Al-Baqarah ayat 155-157.

وَلَنَبْلُوَنَّكُمْ بِشَيْءٍ مِّنَ الْخَوْفِ وَالْجُوعِ وَنَقْصٍ مِّنَ الْأَمْوَالِ وَالْأَنْفُسِ وَالثَّمَرَاتِ وَبَشِّرِ الصَّابِرِينَ (155) الَّذِينَ إِذَا  
أَصَابَتْهُمْ مُصِيبَةٌ قَالُوا إِنَّا لِلَّهِ وَإِنَّا إِلَيْهِ رَاجِعُونَ (156) أُولَٰئِكَ عَلَيْهِمْ صَلَوَاتٌ مِّن رَّبِّهِمْ وَرَحْمَةٌ وَأُولَٰئِكَ هُمُ الْمُهْتَدُونَ  
(157)

*“(155) Kami pasti akan mengujimu dengan sedikit ketakutan dan kelaparan, kekurangan harta, jiwa, dan buah-buahan. Sampaikanlah (wahai Nabi*

*Muhammad,) kabar gembira kepada orang-orang sabar, (156) (yaitu) orang-orang yang apabila ditimpa musibah, mereka mengucapkan: “Innā lillāhi wa innā ilaihi rāji‘ūn” (sesungguhnya kami adalah milik Allah dan sesungguhnya hanya kepada-Nya kami akan kembali). (157) Mereka itulah yang memperoleh ampunan dan rahmat dari Tuhannya dan mereka itulah orang-orang yang mendapat petunjuk.” (QS.Al-Baqarah:155-157)*

Pada penelitiannya Almais et. al., menjelaskan bahwa untuk menentukan tingkat kerusakan bangunan pasca bencana alam dapat menggunakan hasil ekstraksi fitur menggunakan teknik GLCM, kemudian hasilnya dapat di ketahui labelnya menggunakan PCA-Clustering (Almais et al., 2024). Untuk mengetahui label tersebut menggunakan acuan yang ada pada penelitiannya Almais et. al., yang menjelaskan bahwa terdapat standart nilai Tingkat kerusakan bangunan pasca bencana alam. Selain itu juga terdapat teknik baru Clustering data yang dapat digunakan untuk mengelompokkan data berdasarkan karakter yang sama (Almais et al., 2023).

Dalam beberapa tahun terakhir, teknologi kecerdasan buatan (*Artificial Intelligence*) telah mengalami perkembangan pesat, terutama dalam bidang pengolahan citra menggunakan *Convolutional Neural Network* (CNN) (Purwono et al., 2023). CNN telah terbukti efektif dalam berbagai tugas analisis citra, termasuk dalam mendeteksi kerusakan bangunan pasca bencana alam. Salah satu pendekatan yang menjanjikan adalah metode *transfer learning*, yang memungkinkan model yang telah dilatih sebelumnya untuk diterapkan pada dataset baru dengan pelatihan minimal. Model *Residual Network-18* (ResNet-18) merupakan salah satu arsitektur CNN yang dikenal memiliki kinerja tinggi dalam ekstraksi fitur gambar (Gurjar & Voditel, 2023), menjadikannya kandidat ideal untuk mendeteksi kerusakan

bangunan dengan proses komputasi yang ringan sehingga tidak memakan waktu yang lama dalam proses komputasi.

Meskipun banyak penelitian telah dilakukan dalam bidang deteksi kerusakan bangunan pasca bencana alam menggunakan metode *deep learning*, tantangan masih ada dalam hal generalisasi model pada berbagai jenis kerusakan serta efisiensi pemrosesan data dalam skenario dunia nyata. Penelitian yang akan dilakukan ini berfokus pada penerapan metode *transfer learning* menggunakan model ResNet-18 untuk ekstraksi fitur gambar kerusakan bangunan. (Vasantdada Patil Pratishtan's College of Engineering and Visual Arts, India et al., 2024). Dengan demikian, penelitian yang dilakukan ini bertujuan untuk menganalisis hasil dari ekstraksi fitur tersebut yang nantinya hasil analisis tersebut bisa digunakan dalam penelitian lebih lanjut.

Penelitian ini memiliki urgensi yang tinggi mengingat beberapa wilayah di Indonesia merupakan daerah yang rentan terkena bencana alam salah satunya ada pada Kota Malang. Pada penelitian ini, identifikasi dan klasifikasi kerusakan bangunan secara cepat dan akurat dapat menjadi dasar bagi pemerintah serta lembaga terkait dalam menentukan langkah-langkah penanganan yang tepat. Dengan memanfaatkan teknologi deep learning, khususnya metode transfer learning pada model ResNet-18, diharapkan penelitian ini dapat memberikan kontribusi yang signifikan dalam mendukung upaya mitigasi bencana alam serta meningkatkan kesiapsiagaan terhadap dampak dari bencana alam di Indonesia (Lin et al., 2022).

Beberapa penelitian terdahulu telah membuktikan bahwa metode *transfer learning* berbasis ResNet-18 dapat diterapkan secara efektif dalam analisis citra kerusakan bangunan akibat bencana alam. Studi yang dilakukan oleh Gustavo et al. (2021) menunjukkan bahwa penggunaan *transfer learning* dengan model ResNet-18 dapat meningkatkan akurasi klasifikasi citra makanan/non-makanan hingga 0.978 lebih tinggi dibandingkan metode AlexNet (0,977) (Universitas Brawijaya et al., 2021). Selain itu, penelitian oleh Cahyati et al. (2024) mengungkapkan bahwa Model CNN dengan ResNet-50 mencapai akurasi 96% dalam klasifikasi motif Songket Palembang. CNN lebih unggul dibandingkan metode klasifikasi lain dalam akurasi pengenalan motif (Cahyati & Devella, 2024). Berdasarkan hasil penelitian ini, implementasi ResNet-18 dalam penelitian ini diharapkan dapat memberikan hasil yang optimal dalam mendeteksi kerusakan bangunan di Kota Malang.

Dengan demikian, penelitian yang dilakukan ini akan memberikan kontribusi dalam pengembangan teknologi deteksi kerusakan bangunan pasca bencana alam yang lebih akurat dan efisien. Hasil dari penelitian ini diharapkan dapat dimanfaatkan oleh berbagai pihak, termasuk pemerintah, tim penanggulangan bencana, serta masyarakat luas, dalam rangka meningkatkan respons dan mitigasi terhadap bencana alam di masa depan.

## 1.2 Rumusan Masalah

Dalam melakukan penelitian ini, rumusan masalah yang dikemukakan adalah seperti berikut:

1. Bagaimana mengimplementasikan metode ResNet-18 untuk ekstraksi fitur gambar kerusakan bangunan pasca bencana alam?
2. Bagaimana mengetahui performa metode ResNet-18 untuk mengetahui Tingkat kerusakan bangunan pasca bencana alam?

## 1.3 Batasan Masalah

Agar penelitian ini lebih terfokus dan terarah, beberapa batasan yang diterapkan dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini hanya akan menggunakan model ResNet-18 untuk ekstraksi fitur gambar kerusakan bangunan pasca bencana alam, tanpa membandingkannya dengan arsitektur *deep learning* lainnya.
2. Dataset yang digunakan terdiri dari gambar kerusakan bangunan yang terjadi di Kota Malang yang diambil dari BPBD (Badan Penanggulangan Bencana Daerah) sehingga hasil penelitian ini terbatas pada karakteristik data dari daerah tersebut.
3. Resolusi gambar yang digunakan gambar dengan rasio 224 x 224 pixel.
4. Penelitian ini hanya fokus pada tahap ekstraksi fitur dan tidak mencakup pengembangan sistem deteksi secara *end-to-end* atau integrasi dengan sistem tanggap darurat.

#### 1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Mengimplementasikan metode ResNet-18 dalam ekstraksi fitur citra kerusakan bangunan pasca bencana alam di Kota Malang.
2. Menganalisis performa model dalam hal akurasi, efisiensi pemrosesan, dan generalisasi pada dataset gambar kerusakan bangunan.

#### 1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi Penulis: Penelitian ini memberikan kesempatan bagi penulis untuk menerapkan teori dan metode *deep learning* dalam analisis citra, serta mengembangkan keterampilan dalam pemrosesan data dan evaluasi model kecerdasan buatan.
2. Bagi Pembaca: Hasil penelitian ini dapat menjadi referensi bagi akademisi, mahasiswa, dan peneliti lain yang tertarik dalam pengembangan metode *transfer learning* untuk deteksi kerusakan bangunan akibat bencana alam.
3. Bagi Pemerintah dan Instansi Terkait: Penelitian ini dapat memberikan data dan metode yang dapat digunakan untuk mendukung pengambilan keputusan dalam mitigasi bencana, perencanaan rekonstruksi, serta peningkatan kesiapsiagaan dalam menghadapi bencana alam. Penelitian ini juga bisa di manfaatkan oleh pemerintah atau instansi terkait yang juga rentan mengalami bencana alam.



Dengan adanya tujuan dan manfaat penelitian ini, diharapkan penelitian dapat memberikan dampak positif yang luas bagi berbagai sektor yang berkaitan dengan mitigasi bencana dan pengelolaan infrastruktur pasca bencana alam.

## BAB II

### STUDI PUSTAKA

#### 2.1 Penelitian Terdahulu

Penelitian yang dilakukan oleh Gustavo Thiodorus, Anugrah Prasetya, Luthfi Afrizal Ardhani, Novanto Yudistira (2021) dengan judul “Klasifikasi Citra Makanan/Nonmakanan Menggunakan Metode *Transfer learning* dengan Model ResNet-18 mengembangkan model klasifikasi otomatis yang mampu membedakan citra makanan dan nonmakanan menggunakan pendekatan *Transfer learning* dengan model ResNet-18. Model ini dipilih karena kemampuannya dalam menangani *vanishing gradient* serta memanfaatkan bobot dari pelatihan sebelumnya untuk meningkatkan akurasi dan mengurangi waktu komputasi. Penelitian ini membandingkan model ResNet-18 dan AlexNet dengan teknik *fine tuning* serta *freeze layer* guna meningkatkan kinerja model. Dari 3.000 citra pelatihan dan 1.000 citra pengujian, ditemukan bahwa metode *freeze layer* pada ResNet-18 memberikan akurasi tertinggi sebesar 0,988, sementara AlexNet hanya mencapai 0,978. Hasil ini menunjukkan bahwa ResNet-18 lebih unggul dalam klasifikasi citra makanan serta penggunaan metode *transfer learning* dengan *fine tuning* dan *freeze layer* dapat meningkatkan kinerja model secara signifikan (Universitas Brawijaya et al., 2021).

Penelitian yang dilakukan oleh Cahyati, Devella, dan Yohannes (2024) berjudul “Pengenalan Motif Songket Palembang Menggunakan CNN dengan Arsitektur ResNet-50”. Studi ini bertujuan untuk mengembangkan sistem klasifikasi motif kain Songket Palembang menggunakan CNN dengan arsitektur

ResNet-50. Dataset yang digunakan mencakup berbagai motif Songket Palembang, seperti Bintang Berantai, Naga Besaung, Nampan Perak, dan Pulir, dengan hasil menunjukkan akurasi hingga 96% dalam klasifikasi motif. Penelitian ini berkontribusi dalam pelestarian budaya dengan memanfaatkan teknologi kecerdasan buatan untuk mengenali pola motif songket secara otomatis serta membandingkan efektivitas metode klasifikasi lainnya, seperti SURF, SIFT, HOG, dan histogram HSV, yang menunjukkan keunggulan CNN dalam meningkatkan akurasi pengenalan motif (Cahyati & Devella, 2024).

Setiaji dan Harintaka (2019) dalam penelitiannya yang berjudul “Ekstraksi Fitur Bangunan Secara Cepat pada Foto UAV Menggunakan Metode *Deep Residual Neural Network* Berbasis FCN” mengembangkan metode untuk mengekstraksi fitur bangunan dari citra udara yang diperoleh menggunakan *Unmanned Aerial Vehicle* (UAV). Teknologi *deep learning* berbasis *Fully Convolutional Network* (FCN) dengan arsitektur ResNet digunakan untuk meningkatkan efisiensi ekstraksi fitur bangunan. Hasil penelitian menunjukkan bahwa metode ini dapat mencapai akurasi hingga 95% dalam skala pemetaan 1:5000 dan lebih efisien dibandingkan interpretasi manual oleh operator manusia. Temuan ini berpotensi digunakan dalam berbagai aplikasi, seperti perencanaan tata kota, pemantauan Izin Mendirikan Bangunan (IMB), serta analisis perubahan tutupan lahan (Setiaji & Harintaka, 2019).

Sementara itu, penelitian yang dilakukan oleh Hendri Candra Mayana dan Desmarita Leni (2023) dengan judul “Deteksi Kerusakan Ban Mobil Menggunakan CNN dengan Arsitektur ResNet-34” bertujuan untuk meningkatkan akurasi dalam

mengidentifikasi kondisi ban mobil menggunakan metode CNN berbasis ResNet-34. Model dilatih dengan *optimizer Adam* menggunakan *learning rate* 0.0001, *batch size* 32, dan selama 50 *epoch*. Dataset penelitian mencakup 762 citra yang dikategorikan ke dalam dua kelas, yaitu ban dalam kondisi baik dan ban rusak. Untuk meningkatkan performa model, dilakukan *preprocessing data*, seperti normalisasi, konversi gambar ke format *Red, Green, Blue* (RGB), serta pembagian dataset menjadi data latih dan data uji. Berdasarkan evaluasi menggunakan *Confusion Matrix*, model ini mencapai akurasi 91,6%, dengan *presisi* 90,7%, *recall* 92,7%, dan *F1-score* 91,7%. Hasil tersebut menunjukkan bahwa arsitektur ResNet-34 efektif dalam mendeteksi kerusakan ban secara otomatis (Hendri & Desmarita, 2023).

Penelitian lainnya yang juga dilakukan oleh Ami Rahmawati, Ita Yulianti, Siti Nurajizah, Taufik Hidayatulloh, dan Ani Oktarini Sari (2025) berjudul Analisis Performa Model ResNet-50 Pada Diagnosis Pneumonia Balita Berdasarkan Citra Radiografi *Thorax*. Penelitian ini bertujuan untuk mengembangkan model klasifikasi citra medis menggunakan arsitektur ResNet-50 untuk mendiagnosis pneumonia pada balita berdasarkan citra radiografi *thorax*. Model ini dibangun dengan pendekatan CNN dan menggunakan data *augmentation* untuk meningkatkan keanekaragaman citra serta mencegah *overfitting*. Proses pelatihan dilakukan dengan Adam optimizer, *learning rate* 0.0001, dan *batch size* 32 selama 50 *epoch*. Dataset yang digunakan terdiri dari 1.349 citra paru-paru normal dan 3.883 citra paru-paru dengan pneumonia, yang diperoleh dari *Kaggle Repository*. Hasil evaluasi model menunjukkan akurasi sebesar 85%, yang menandakan

kemampuan model dalam membedakan antara citra paru-paru normal dan yang mengalami pneumonia dengan tingkat akurasi yang cukup tinggi, sehingga dapat membantu proses diagnosis penyakit ini secara lebih efisien (*Rahmawati et al. - 2025*).

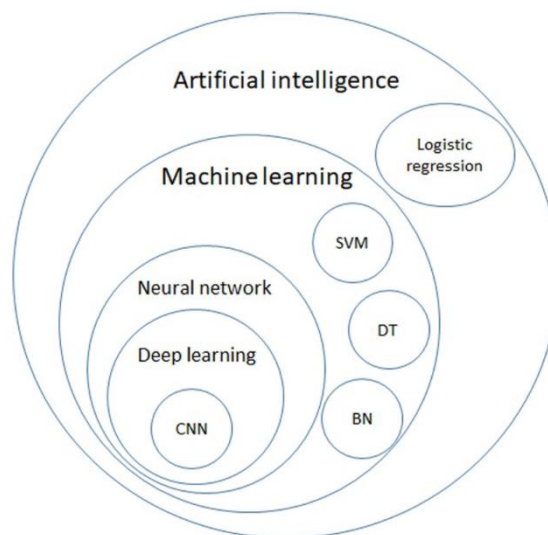
Tabel 2.1 Penelitian Terdahulu

| No | Nama dan Tahun Peneliti | Judul Penelitian                                                                                                         | Metode, Penggalan Data, Pendekatan, dan Analisis Data                                                                                                                                                                                                                        | Hasil Penelitian                                                                                                             |
|----|-------------------------|--------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| 1. | Gustavo (2021)          | Klasifikasi Citra Makanan/Nonmakanan<br>Menggunakan Metode <i>Transfer learning</i> dengan Model <i>Residual Network</i> | Membandingkan model ResNet-18 dan AlexNet dengan teknik <i>fine tuning</i> serta <i>freeze layer</i> . Data yang digunakan 3.000 citra pelatihan dan 1.000 citra pengujian.                                                                                                  | ResNet-18 lebih unggul dari AlexNet yang memberikan akurasi tertinggi sebesar 0,988, sementara AlexNet hanya mencapai 0,978. |
| 2. | Cahyati dkk (2019)      | Pengenalan Motif Songket Palembang<br>Menggunakan Convolutional Neural Network dengan Arsitektur ResNet-50               | Metode yang digunakan yaitu ResNet-50 dengan dataset mencakup berbagai motif Songket Palembang, seperti Bintang Berantai, Naga Besaung, Nampun Perak, dan Pulir.                                                                                                             | Hasilnya menunjukkan akurasi hingga 96%.                                                                                     |
| 3. | Setiaji dkk (2019)      | Ekstraksi Fitur Bangunan Secara Cepat pada Foto UAV<br>Menggunakan Metode Deep Residual Neural Network Berbasis FCN      | Metode yang digunakan adalah <i>Deep Residual Neural Network</i> Berbasis FCN. Dataset berupa citra udara yang diperoleh menggunakan <i>Unmanned Aerial Vehicle</i> (UAV).                                                                                                   | Hasil penelitian menunjukkan bahwa metode ini dapat mencapai akurasi hingga 95% dalam skala pemetaan 1:5000                  |
| 4. | Hendri dkk (2021)       | Deteksi Kerusakan Ban Mobil<br>Menggunakan Convolutional Neural Network dengan Arsitektur ResNet-34                      | Metode yang digunakan adalah ResNet-34. Model dilatih dengan <i>optimizer Adam</i> menggunakan <i>learning rate</i> 0.0001, <i>batch size</i> 32, dan selama 50 <i>epoch</i> . Dataset mencakup 762 citra yang dibagi dalam dua kelas, yaitu ban kondisi baik dan ban rusak. | Hasilnya model ini mencapai akurasi 91,6%, dengan <i>presisi</i> 90,7%, <i>recall</i> 92,7%, dan <i>F1-score</i> 91,7%.      |
| 5. | Rahmawati dkk (2019)    | Analisis Performa Model ResNet-50 Pada Diagnosis Pneumonia Balita                                                        | Metode yang digunakan adalah ResNet-50. Dataset yang digunakan terdiri dari 1.349 citra paru-paru normal dan                                                                                                                                                                 | Hasil evaluasi model menunjukkan akurasi sebesar 85%.                                                                        |

|  |  |                                            |                                                                                                                                                                          |  |
|--|--|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
|  |  | Berdasarkan Citra Radiografi <i>Thorax</i> | 3.883 citra paru-paru dengan pneumonia. Proses pelatihan dilakukan dengan Adam optimizer, <i>learning rate</i> 0.0001, dan <i>batch size</i> 32 selama 50 <i>epoch</i> . |  |
|--|--|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

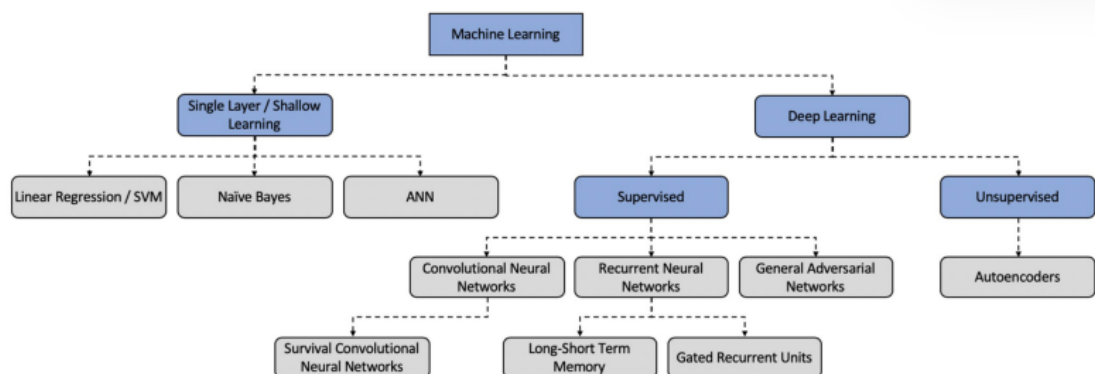
## 2.2 Convolutional Neural Networks

*Convolutional Neural Networks* (CNN) merupakan salah satu jenis arsitektur *deep learning* yang dirancang khusus untuk menangani data berbentuk citra (Zafar et al., 2024). CNN dikembangkan dengan tujuan meniru cara kerja otak manusia dalam mengenali pola visual, dengan memanfaatkan konvolusi sebagai teknik utama dalam ekstraksi fitur. CNN memiliki keunggulan dalam mengenali struktur spasial suatu gambar, yang membuatnya lebih efektif dibandingkan model *machine learning* konvensional dalam tugas klasifikasi gambar, segmentasi, dan deteksi objek (Mienye et al., 2024).



Gambar 2.1 CNN dalam *Machine Learning* (Chiu et al., 2022)

Gambar di atas merupakan hubungan bagaimana CNN menjadi salah satu arsitektur dari *deep learning* di dalam AI. AI mencakup berbagai konsep seperti *Machine Learning* (ML), *neural network*, dan *deep learning*, yang masing-masing memiliki peran dalam pemrosesan data dan pengambilan keputusan. AI secara umum merujuk pada sistem yang dapat memprediksi solusi suatu permasalahan, dengan salah satu metode klasiknya adalah *logistic regression*. ML memungkinkan sistem untuk belajar dari data tanpa pemrograman eksplisit melalui berbagai algoritma seperti *Decision Tree* (DT), *Support Vector Machine* (SVM), dan *Bayesian Network* (BN). Salah satu pengembangan dalam *machine learning* terhadap *neural network* buatan yaitu *Artificial Neural Network* (ANN), yang meniru cara kerja otak manusia dengan menghubungkan *neuron* buatan melalui beberapa *input* dan satu *output*. Lebih lanjut, *deep learning* merupakan bentuk *neural network* yang memiliki banyak lapisan tersembunyi untuk meningkatkan kapabilitas pemrosesan data, dengan CNN sebagai salah satu arsitektur yang paling dikenal dalam analisis citra dan pemrosesan visual (Chiu et al., 2022).



Gambar 2.2 CNN dalam *Deep Learning* (Valliani, 2019)

Gambar di atas memberikan gambaran lebih rinci mengenai pembagian dalam *machine learning*, yang terbagi menjadi dua kategori utama, yaitu *single layer/shallow learning* dan *deep learning*. *Shallow learning* mencakup algoritma seperti *linear regression*, SVM, *Naïve Bayes*, serta ANN yang masih menggunakan lapisan terbatas. Sementara itu, *deep learning* terbagi menjadi dua pendekatan utama, yaitu *supervised* dan *unsupervised*. Dalam *supervised learning*, terdapat beberapa arsitektur jaringan seperti CNN, *Recurrent Neural Networks* (RNN) yang mencakup model seperti *Long Short-Term Memory* (LSTM) dan *Gated Recurrent Units* (GRU), serta jaringan *General Adversarial Networks* (GAN). Sementara itu, dalam *unsupervised learning*, *autoencoder* menjadi salah satu metode yang banyak digunakan untuk merepresentasikan data dalam bentuk yang lebih efisien. Gambar ini memperjelas bagaimana *machine learning* berkembang dari algoritma sederhana hingga model yang lebih kompleks dengan struktur yang lebih dalam untuk menangani tugas yang lebih kompleks (Valliani, 2019).

Struktur dasar CNN terdiri dari beberapa lapisan utama, yaitu *convolutional layer*, *pooling layer*, *fully connected layer*, dan *activation function*. Lapisan konvolusi bertugas mengekstrak fitur dari gambar dengan menerapkan kernel atau filter yang bergerak melintasi seluruh citra, sedangkan *pooling layer* bertindak sebagai mekanisme downsampling untuk mengurangi dimensi data tanpa menghilangkan informasi penting. Setelah melewati proses ekstraksi fitur, nilai-nilai yang diperoleh kemudian diproses melalui *fully connected layer*, yang akan mengubah fitur tersebut menjadi vektor prediksi. Aktivasi non-linear seperti ReLU



(*Rectified Linear Unit*) sering digunakan untuk meningkatkan performa jaringan dengan menambahkan kompleksitas representasi fitur (C. Wang & Wang, 2024).

CNN memiliki banyak keunggulan dalam pemrosesan citra, seperti reduksi kompleksitas komputasi, otomatisasi ekstraksi fitur, dan peningkatan akurasi dibandingkan metode konvensional (Z. Wu, 2024). Karena keunggulannya, CNN telah diterapkan dalam berbagai bidang, termasuk pengolahan citra medis, pengenalan wajah, kendaraan otonom, serta klasifikasi motif kain dan tekstil.

### 2.3 *Transfer learning*

*Transfer learning* merupakan teknik dalam *Deep learning* yang memungkinkan model yang telah dilatih pada satu tugas digunakan kembali untuk tugas lain dengan penyesuaian tertentu (Zorić et al., 2024). Konsep ini didasarkan pada pemahaman bahwa fitur-fitur yang telah dipelajari oleh model pada dataset besar dapat berguna untuk dataset yang lebih kecil atau berbeda, sehingga dapat menghemat waktu pelatihan dan meningkatkan akurasi.

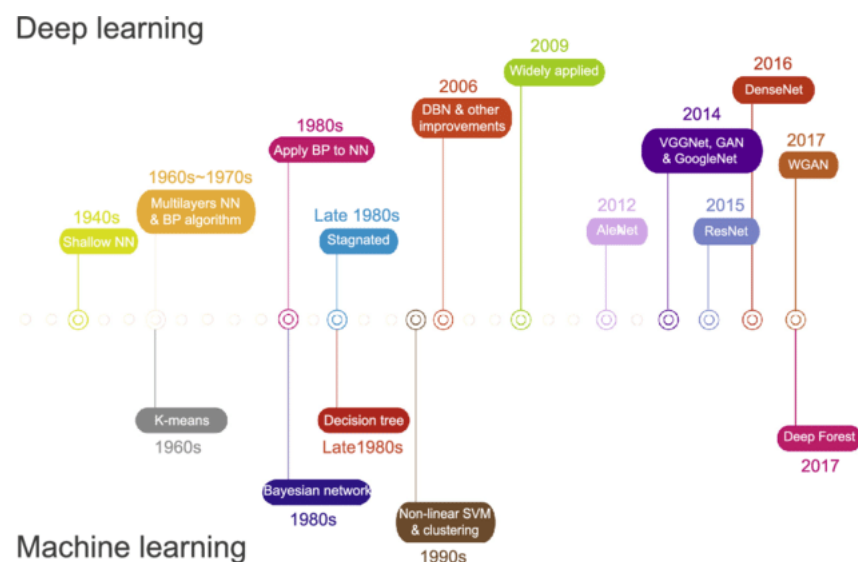
Dalam *Transfer learning*, model yang telah dilatih sebelumnya pada dataset besar, seperti *ImageNet*, digunakan sebagai basis untuk menyelesaikan tugas baru dengan melakukan *fine-tuning* atau *Feature extraction*. *Fine-tuning* dilakukan dengan menyesuaikan bobot dari beberapa lapisan dalam model, sedangkan *Feature extraction* hanya menggunakan lapisan awal model untuk mengekstrak fitur tanpa melatih ulang seluruh jaringan (Mohammadi et al., 2024).

Pendekatan *Transfer learning* sangat berguna dalam kasus di mana dataset yang tersedia terbatas, seperti dalam penelitian ini yang berfokus pada klasifikasi pola gambar tertentu dengan jumlah sampel yang tidak sebesar dataset umum

seperti *ImageNet*. Dengan memanfaatkan *Transfer learning*, model dapat belajar lebih cepat, menghindari overfitting, dan menghasilkan prediksi yang lebih akurat tanpa perlu pelatihan dari nol (Chen et al., 2024).

## 2.4 ResNet-18

ResNet-18 adalah salah satu varian dari *Residual Network* (ResNet), yang merupakan arsitektur CNN yang dikembangkan untuk mengatasi degradasi akurasi akibat semakin dalamnya jaringan. ResNet diperkenalkan oleh Kaiming He dan rekan-rekannya pada tahun 2015 yang memiliki konsep utama yang disebut *residual learning*, di mana terdapat *shortcut connections* atau *skip connections* yang memungkinkan informasi dapat diteruskan langsung ke lapisan yang lebih dalam tanpa mengalami transformasi berulang (Abdelhamid Mahri University, Algeria et al., 2022).



Gambar 2.3 ResNet dalam *Deep Learning* (Cao et al., 2018)

Gambar di atas menggambarkan perkembangan *machine learning* dan pembelajaran *deep learning* dari tahun 1940-an hingga 2017, menunjukkan berbagai kemajuan dalam algoritma dan arsitektur *neural network*. Awalnya, *machine learning* berkembang dengan *shallow neural networks* pada 1940-an, diikuti dengan algoritma *K-Means* pada 1960-an, serta *multilayer Neural Networks* (NN) dan algoritma *backpropagation* (BP) pada 1960-an hingga 1970-an. Pada 1980-an, algoritma *backpropagation* diterapkan pada *neural network*, meskipun mengalami stagnasi di akhir dekade tersebut, sementara dalam *machine learning* muncul metode seperti *decision tree*, *bayesian network*, serta SVM non-linear dan teknik *Clustering* pada 1990-an. Perkembangan signifikan dalam *deep learning* terjadi pada 2006 dengan *Deep Belief Networks* (DBN) dan mulai diterapkan secara luas pada 2009. Kemajuan ini berlanjut dengan munculnya arsitektur AlexNet pada 2012, diikuti oleh VGGNet, GAN, dan GoogleNet pada 2014. Salah satu inovasi penting adalah ResNet yang diperkenalkan pada 2015, yang menjadi solusi terhadap permasalahan *vanishing gradient* dalam *neural network* yang sangat dalam dengan memperkenalkan konsep *skip connections*, memungkinkan propagasi informasi tanpa degradasi. Pendekatan ini meningkatkan efisiensi pelatihan *neural network* dalam, menjadikannya arsitektur yang sangat populer dalam berbagai tugas pemrosesan citra seperti deteksi objek dan segmentasi gambar. Setelah ResNet, pengembangan *deep learning* terus berlanjut dengan munculnya DenseNet pada 2016 serta WGAN dan *Deep Forest* pada 2017 (Cao et al., 2018).

Arsitektur ResNet-18 terdiri dari 18 lapisan jaringan, yang mencakup konvolusi, *batch normalization*, ReLU *activation*, dan *shortcut connections*. Dengan adanya *shortcut connections*, jaringan dapat menghindari masalah *vanishing gradient*, sehingga model tetap dapat belajar secara efektif meskipun memiliki kedalaman yang lebih tinggi dibandingkan model CNN konvensional (Saini & Rawat, 2022).

Keunggulan utama dari ResNet-18 dibandingkan arsitektur lain seperti VGG-16 atau AlexNet adalah efisiensi komputasi, akurasi yang lebih tinggi, serta kemampuannya dalam menangani dataset yang lebih kompleks. Oleh karena itu, ResNet-18 sering digunakan dalam berbagai aplikasi klasifikasi citra, deteksi objek, dan pengenalan pola, termasuk dalam penelitian ini untuk mengklasifikasikan pola motif berdasarkan fitur yang telah dipelajari (Nagpal et al., 2022).

## 2.5 Ekstraksi Fitur

Ekstraksi fitur merupakan salah satu langkah penting dalam proses pengolahan citra, yang bertujuan untuk mengidentifikasi dan mengisolasi karakteristik penting dari suatu gambar agar dapat digunakan dalam analisis lebih lanjut. Dalam konteks pengenalan pola dan klasifikasi gambar, fitur yang diekstraksi dari suatu citra dapat berupa tepi, tekstur, bentuk, atau warna, yang kemudian digunakan sebagai representasi numerik dalam model pembelajaran mesin atau *deep learning* (Gupta & Gupta, 2024).

Dalam ekstraksi fitur, terdapat dua pendekatan utama, yaitu ekstraksi fitur berbasis metode konvensional dan ekstraksi fitur berbasis *deep learning*. Metode konvensional mencakup teknik seperti *Histogram of Oriented Gradients* (HOG),

*Scale-Invariant Feature Transform* (SIFT), *Speeded-Up Robust Features* (SURF), dan *Local Binary Patterns* (LBP). Teknik-teknik ini mengekstrak fitur berdasarkan karakteristik visual spesifik yang ada dalam citra dan sering digunakan dalam aplikasi pengenalan wajah, pengenalan objek, dan klasifikasi tekstur (D. Wu, 2024).

Seiring dengan perkembangan teknologi *deep learning*, ekstraksi fitur kini lebih banyak dilakukan secara otomatis menggunakan CNN. CNN secara otomatis mengekstrak fitur dari gambar melalui serangkaian lapisan konvolusi tanpa memerlukan rekayasa fitur manual. Lapisan konvolusi bekerja dengan menggunakan kernel (filter) yang berjalan melintasi gambar untuk menangkap pola yang muncul pada berbagai skala dan orientasi. Setiap lapisan dalam CNN bertugas mengekstrak fitur yang semakin kompleks, mulai dari tepi sederhana pada lapisan awal hingga pola yang lebih kompleks pada lapisan yang lebih dalam (Z. Wu, 2024).

Dalam penelitian ini, metode ekstraksi fitur menggunakan pendekatan *deep learning* dengan CNN dan arsitektur ResNet-18, yang mampu mengekstrak fitur secara lebih efisien dibandingkan metode konvensional. Dengan bantuan *Transfer learning*, fitur yang telah dipelajari oleh model yang dilatih sebelumnya dapat digunakan untuk dataset baru dengan sedikit penyesuaian, sehingga memungkinkan peningkatan akurasi dalam klasifikasi citra dengan jumlah data yang lebih terbatas.

Ekstraksi fitur memiliki peran penting dalam meningkatkan kinerja model klasifikasi karena fitur yang diperoleh menjadi dasar bagi proses pengambilan keputusan oleh *neural network* tiruan. Oleh karena itu, teknik ekstraksi fitur yang

digunakan dalam penelitian ini akan berfokus pada pendekatan *deep learning* untuk memperoleh representasi fitur yang lebih kaya dan bermakna guna mendukung akurasi klasifikasi yang optimal.

## 2.6 *Principal Component Analysis*

*Principal Component Analysis* (PCA) merupakan teknik statistik multivariat yang digunakan untuk mereduksi dimensi data dengan tetap mempertahankan sebanyak mungkin variasi dalam data tersebut. PCA bekerja dengan cara mentransformasikan data awal yang mungkin terdiri dari banyak variabel yang saling berkorelasi menjadi serangkaian variabel baru yang tidak berkorelasi satu sama lain, yang disebut sebagai *Principal Components*. Komponen-komponen ini diurutkan berdasarkan besar variansi yang dikandungnya, sehingga komponen pertama (PC1) mengandung informasi paling besar, dan komponen selanjutnya mengandung informasi yang semakin kecil. (Khalid et al., 2014)

Secara matematis, PCA melibatkan proses dekomposisi *eigen* terhadap matriks kovarians dari data, yang menghasilkan pasangan *eigenvalue* dan *eigenvector*. *Eigenvector* menunjukkan arah dari *principal component*, sedangkan *eigenvalue* menunjukkan jumlah variasi yang dijelaskan oleh masing-masing komponen. Dengan memilih sejumlah komponen utama teratas (misalnya dua atau tiga), dimensi data dapat dikurangi secara signifikan tanpa kehilangan informasi penting secara drastis. Hal ini sangat berguna dalam pengolahan citra atau dataset berdimensi tinggi untuk menghindari masalah *overfitting*, mengurangi beban komputasi, serta meningkatkan performa model klasifikasi atau *Clustering*. (Abdi & Williams, 2010)

Dalam konteks pembelajaran mesin dan pengolahan citra, PCA sering digunakan sebagai langkah praproses sebelum tahap klasifikasi atau *Clustering*. Misalnya, dalam ekstraksi fitur, PCA dapat membantu mereduksi kompleksitas fitur yang diekstrak dari CNN atau metode lain agar lebih efisien dan representatif. PCA juga digunakan untuk visualisasi data berdimensi tinggi ke dalam bentuk dua atau tiga dimensi agar lebih mudah dianalisis secara visual. Dengan demikian, PCA menjadi salah satu teknik fundamental yang penting dalam pipeline pengolahan data dan pembelajaran mesin modern. (Aslam & Rabie, 2023)

## 2.7 *K-Means Clustering*

*K-Means Clustering* adalah salah satu metode *unsupervised learning* yang paling populer untuk melakukan *Clustering* data berdasarkan kemiripan fitur. Metode ini bertujuan untuk membagi sekumpulan data ke dalam  $k$  kelompok (*cluster*) yang berbeda, sehingga data dalam satu *cluster* memiliki kemiripan maksimum, sedangkan data antar *cluster* memiliki perbedaan maksimum. Proses *Clustering* ini sangat berguna untuk menemukan struktur atau pola tersembunyi dalam data yang tidak memiliki label kelas. (Patel et al., 2022)

Secara umum, algoritma *K-Means* bekerja dalam dua tahap utama secara iteratif: pertama, penempatan awal  $k$  *centroid* secara acak atau dengan teknik khusus seperti *K-Means++*; kedua, pengelompokan setiap titik data ke *centroid* terdekat berdasarkan jarak *Euclidean*. Setelah semua titik dialokasikan ke *cluster*, *centroid* setiap *cluster* dihitung ulang sebagai rata-rata posisi semua titik dalam *cluster* tersebut. Proses ini diulang hingga konvergensi, yaitu ketika posisi *centroid* tidak lagi berubah secara signifikan atau jumlah iterasi maksimum tercapai.

Namun, *K-Means* memiliki beberapa keterbatasan. Algoritma ini sangat sensitif terhadap nilai awal *centroid* dan pemilihan jumlah *cluster* *k*. Jika pemilihan nilai *k* tidak sesuai, hasil *Clustering* dapat menjadi kurang representatif. Oleh karena itu, diperlukan metode tambahan seperti *Elbow Method* atau *Silhouette Analysis* untuk menentukan jumlah *cluster* yang optimal. Selain itu, algoritma ini juga tidak bekerja dengan baik untuk data yang memiliki bentuk *cluster* non-linier atau memiliki ukuran dan densitas yang sangat bervariasi. (Lakshmi et al., 2019)

Dalam aplikasi pengolahan citra, *K-Means* digunakan secara luas untuk segmentasi gambar, di mana *pixel* dikelompokkan berdasarkan kesamaan warna, tekstur, atau fitur lain. Selain itu, *K-Means* juga dapat diterapkan setelah ekstraksi fitur (misalnya dari CNN) untuk mengelompokkan citra ke dalam kategori tertentu tanpa memerlukan label eksplisit. Kombinasi PCA dan *K-Means* sering digunakan secara bersamaan, di mana PCA terlebih dahulu mereduksi dimensi fitur sebelum dilakukan *Clustering* dengan *K-Means*, guna meningkatkan efisiensi dan efektivitas pemisahan *cluster*. (Sun & Li, 2019)

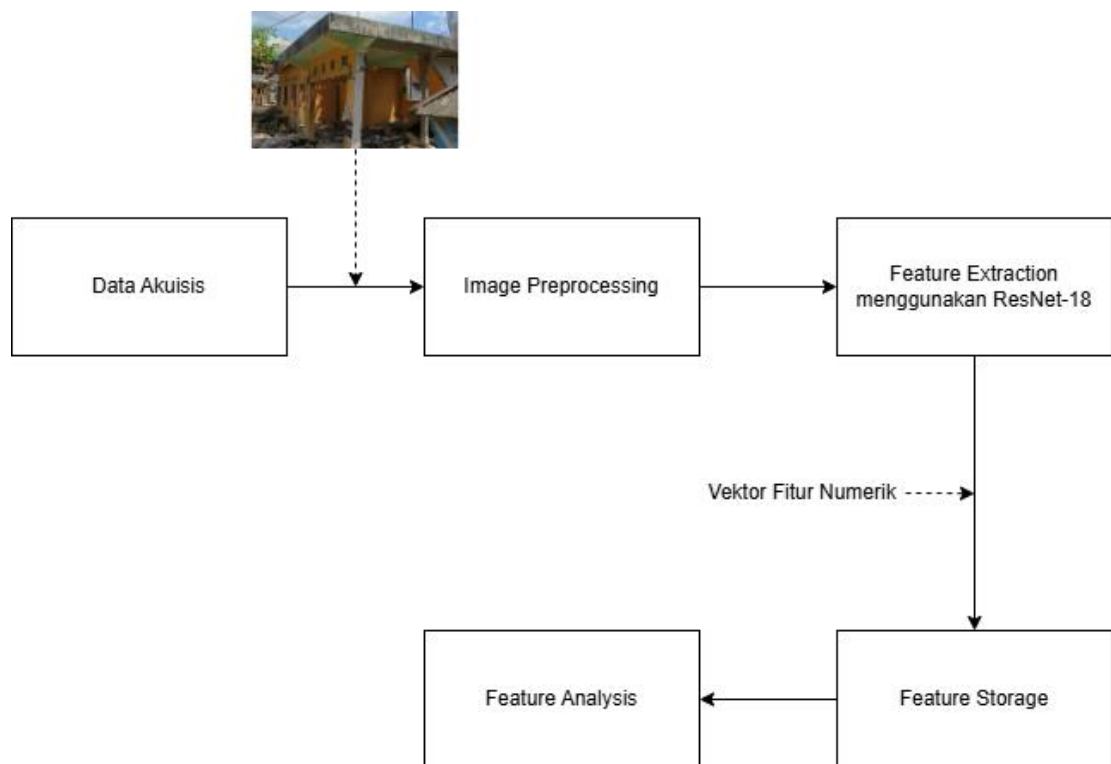


## BAB III

### METODE PENELITIAN

#### 3.1 Desain Sistem

Proses ekstraksi fitur tingkat kerusakan bangunan pasca bencana alam menggunakan ResNet-18 dimulai dengan Data Akuisisi, di mana gambar dikumpulkan dari BPBD Kota Malang.



Gambar 3.1 Diagram Sistem

Pada gambar 3.1 di atas menjelaskan tentang alur dari ekstraksi fitur menggunakan ResNet-18 yang di mulai dari proses Data Akuisisi seperti yang di jelaskan sebelumnya. Setelah itu, gambar mengalami *Image Preprocessing*, dengan

*preprocess* yang di alami adalah menormalisasi nilai *pixel* sesuai dengan skema ResNet-18 dan proses *centering* pada data untuk meningkatkan kualitas *input* sebelum ekstraksi fitur. Kemudian, gambar akan memasuki pada tahap *Feature extraction*, di mana model ResNet-18 digunakan untuk mengekstrak fitur penting yang mewakili tingkat kerusakan bangunan. Hasil dari ekstraksi fitur ini disimpan dalam *Feature Storage*, yang memungkinkan penyimpanan dan pemrosesan lebih lanjut. Akhirnya, fitur yang telah disimpan akan dievaluasi dalam tahap *Model Evaluation* untuk mengukur *Clustering* data berdasarkan kemiripan fitur.

### **3.2 Data Preparation**

Tahapan *Data Preparation* dalam penelitian ini dilakukan untuk memastikan bahwa dataset yang digunakan memiliki kualitas yang baik sebelum dilakukan proses ekstraksi fitur menggunakan ResNet-18. Pada penelitian ini, proses *Data Preparation* terdiri dari dua tahap utama, yaitu Data Akuisisi dan *Image Preprocessing*, yang berfungsi untuk menyediakan data dalam format yang optimal sebelum proses ekstraksi fitur gambar kerusakan bangunan pasca bencana.

#### **3.2.1 Data Akuisisi**

Tahapan Data Akuisisi merupakan langkah awal dalam penelitian ini, di mana dataset gambar bangunan yang mengalami kerusakan akibat bencana alam dikumpulkan. Dataset yang digunakan gambar bangunan dalam berbagai tingkat kerusakan, seperti ringan, sedang, dan berat. Dataset yang digunakan dalam penelitian ini telah di filter berdasarkan kriteria tertentu, yaitu:

1. Resolusi gambar: gambar yang akan digunakan menggunakan gambar  $224 \times 224$  pixel.
2. Format RGB: Seluruh gambar harus memiliki format RGB (*Red, Green, Blue*) agar model dapat memproses warna dengan optimal.
3. Minim *Noise* dan Distorsi: Gambar yang memiliki terlalu banyak *Noise* atau distorsi dikeluarkan dari dataset untuk meningkatkan kualitas ekstraksi fitur.



Gambar 3.2 Data Gambar Kerusakan Bangunan

Gambar 3.2 di atas merupakan contoh data yang akan digunakan pada penelitian ini. Setelah data dikumpulkan, langkah selanjutnya adalah *Image Preprocessing*, yang bertujuan untuk meningkatkan kualitas gambar sebelum diekstraksi fiturnya.

### 3.2.2 *Image Preprocessing*

*Image Preprocessing* dilakukan untuk meningkatkan kualitas gambar sebelum diekstraksi fitur menggunakan ResNet-18 (Stojnev & Stojnev Ilić, 2020). Pada tahap ini, beberapa langkah utama diterapkan untuk memastikan bahwa gambar memiliki format yang sesuai dengan kebutuhan model *deep learning*. Langkah-langkah *preprocessing* yang dilakukan dalam penelitian ini meliputi:

## 1. Normalisasi *Pixel*

- a) Nilai *pixel* dalam gambar dinormalisasi agar sesuai dengan skema *preprocessing* ResNet-18. Normalisasi dilakukan dengan rumus :

$$X' = \frac{X - \mu}{\sigma} \quad (3.1)$$

**Dimana:**

$X$  = nilai *pixel* asli dalam gambar.

$\mu$  = rata-rata nilai *pixel* dalam dataset.

$\sigma$  = standar deviasi nilai *pixel*.

- b) Normalisasi ini bertujuan untuk menyamakan distribusi nilai *pixel* dalam gambar, sehingga model dapat menangkap pola fitur dengan lebih optimal.

## 2. *Centering Data*

- a) Setiap gambar mengalami proses *mean subtraction*, yaitu mengurangi rata-rata nilai *pixel* dari setiap *pixel* dalam gambar.
- b) Proses ini bertujuan untuk menghilangkan *noise* akibat variasi pencahayaan dan memastikan bahwa fitur yang diekstraksi lebih representatif terhadap karakteristik tingkat kerusakan bangunan. Proses ini dilakukan dengan rumus :

$$X_c = X - \mu \quad (3.2)$$

**Dimana :**

$X_c$  adalah nilai *pixel* setelah *mean subtraction*.

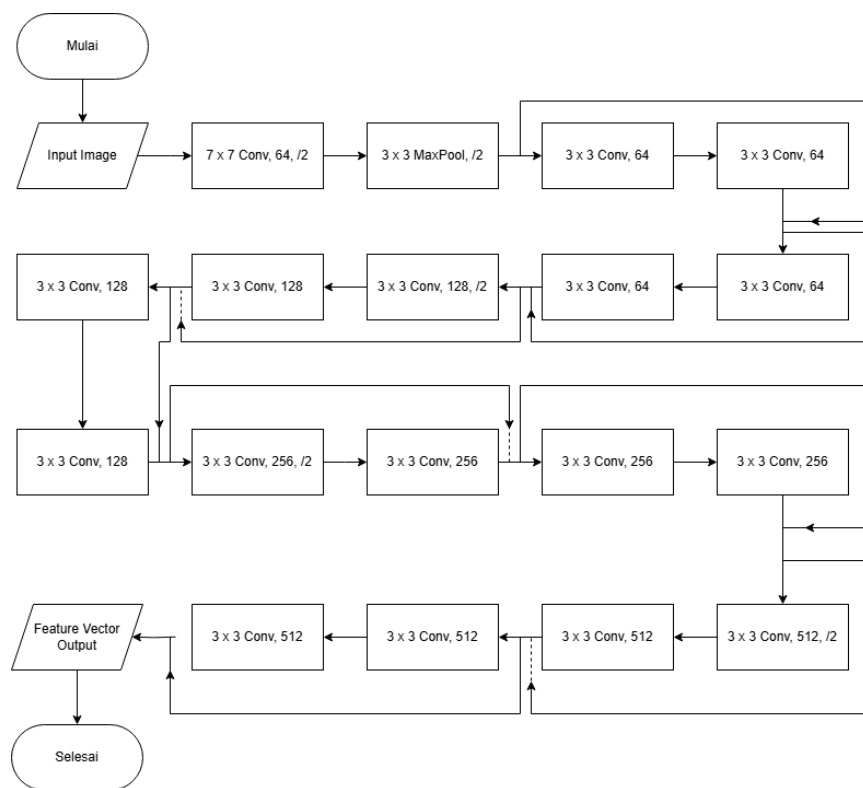
$X$  adalah nilai *pixel* asli dalam gambar.

$\mu$  adalah rata-rata nilai *pixel* dalam seluruh dataset.

Setelah tahap *Image Preprocessing*, gambar yang telah diproses akan langsung masuk ke tahap *Feature extraction* menggunakan ResNet-18.

### 3.3 Implementasi Metode ResNet-18

Pada tahap ini, metode ResNet-18 diimplementasikan untuk mengekstraksi fitur dari gambar bangunan pasca bencana alam. ResNet-18 merupakan salah satu arsitektur CNN yang dikembangkan untuk mengatasi permasalahan *vanishing gradient* dalam *neural network* ((Abdelhamid Mahri University, Algeria et al., 2022).



Gambar 3.3 Flowchart ResNet-18 (Francis & Prakash Verma, 2025)

Gambar 3.3 di atas mengutip dari (Francis & Prakash Verma, 2025) merupakan *flowchart* dari model ResNet-18. Model ini terdiri dari 18 lapisan, yang mencakup *convolutional layers*, *batch normalization*, *activation functions* (ReLU), dan *shortcut connections* yang memungkinkan informasi dari lapisan sebelumnya dapat langsung diteruskan ke lapisan berikutnya tanpa transformasi yang kompleks.

Berbeda dengan implementasi ResNet-18 yang digunakan dalam klasifikasi, penelitian ini hanya menggunakan ResNet-18 sebagai *feature extractor*, yang berarti model hanya digunakan untuk menghasilkan representasi fitur dari gambar tanpa melakukan proses klasifikasi.

### 3.3.1 Arsitektur ResNet-18

ResNet-18 dirancang dengan konsep *residual learning*, di mana model ini menggunakan *shortcut connections* atau *skip connections* untuk memungkinkan informasi melewati beberapa lapisan tanpa mengalami perubahan. Dengan adanya pendekatan ini, ResNet-18 mampu mempertahankan akurasi tinggi tanpa mengalami degradasi performa meskipun jumlah lapisannya lebih banyak dibandingkan CNN konvensional.

Arsitektur ResNet-18 dalam penelitian ini terdiri dari beberapa tahapan utama, sebagai berikut:

1. *Convolutional layer* (Conv1):
  - a) Lapisan pertama dalam arsitektur ini menggunakan kernel 7x7 dengan 64 filter dan stride 2.
  - b) Lapisan ini berfungsi untuk mengekstraksi fitur dasar dari gambar, seperti tepi dan tekstur, serta mengurangi dimensi spasial dengan stride 2.
2. *Max pooling layer*:
  - a) Setelah *convolutional layer*, dilakukan *max pooling* dengan kernel 3x3 dan stride 2.

- b) *Max pooling* ini bertujuan untuk mengurangi resolusi gambar secara signifikan sambil mempertahankan fitur penting.

3. Residual *blocks*:

- a) Model ini terdiri dari empat tahap *residual block*, masing-masing dengan jumlah filter sebagai berikut:

- *Residual block 1*: Dua lapisan konvolusi 3x3 dengan 64 filter.
- *Residual block 2*: Dua lapisan konvolusi 3x3 dengan 128 filter (stride 2 untuk downsampling).
- *Residual block 3*: Dua lapisan konvolusi 3x3 dengan 256 filter (stride 2 untuk downsampling).
- *Residual block 4*: Dua lapisan konvolusi 3x3 dengan 512 filter (stride 2 untuk downsampling).

- b) Setiap blok residual menggunakan *skip connection* untuk mempertahankan informasi penting dari fitur sebelumnya.

- c) Setiap gambar masukan diproses melalui lapisan konvolusi yang memiliki kernel filter dengan ukuran tertentu. Operasi konvolusi didefinisikan sebagai:

$$Y(i,j) = \sum_m \sum_n X(i+m, j+n) \cdot K(m,n) \quad (3.3)$$

**Dimana :**

$Y(i,j)$  = hasil konvolusi pada posisi  $i,j$ .

$X(i+m, j+n)$  = nilai *pixel* pada gambar *input*.

$K(m,n)$  = kernel filter pada jaringan konvolusi.

Proses ini memungkinkan model untuk menangkap pola tekstur dan struktur dalam gambar yang kemudian digunakan untuk analisis lebih lanjut.

Sebagai ilustrasi, misalkan terdapat *input* citra berukuran 64x64 piksel, kemudian diproses melalui operasi konvolusi menggunakan kernel berukuran 7x7.

Jika *input* dan kernel-nya adalah sebagai berikut:

Tabel 3.1 Contoh *Input* 64x64 Channel R

|      |      |      |      |      |      |      |      |      |      |      |      |     |      |
|------|------|------|------|------|------|------|------|------|------|------|------|-----|------|
| 2.04 | 2.04 | 2.04 | 2.04 | 3.58 | 3.58 | 3.58 | 3.58 | 1.84 | 1.84 | 1.84 | 1.84 | ... | 2.32 |
| 2.04 | 2.04 | 2.04 | 2.04 | 3.58 | 3.58 | 3.58 | 3.58 | 1.84 | 1.84 | 1.84 | 1.84 | ... | 2.32 |
| 2.04 | 2.04 | 2.04 | 2.04 | 3.58 | 3.58 | 3.58 | 3.58 | 1.84 | 1.84 | 1.84 | 1.84 | ... | 2.32 |
| 2.04 | 2.04 | 2.04 | 2.04 | 3.58 | 3.58 | 3.58 | 3.58 | 1.84 | 1.84 | 1.84 | 1.84 | ... | 2.32 |
| 1.98 | 1.98 | 1.98 | 1.98 | 2.06 | 2.06 | 2.06 | 2.06 | 3.02 | 3.02 | 3.02 | 3.02 | ... | 3.2  |
| 1.98 | 1.98 | 1.98 | 1.98 | 2.06 | 2.06 | 2.06 | 2.06 | 3.02 | 3.02 | 3.02 | 3.02 | ... | 3.2  |
| 1.98 | 1.98 | 1.98 | 1.98 | 2.06 | 2.06 | 2.06 | 2.06 | 3.02 | 3.02 | 3.02 | 3.02 | ... | 3.2  |
| 1.98 | 1.98 | 1.98 | 1.98 | 2.06 | 2.06 | 2.06 | 2.06 | 3.02 | 3.02 | 3.02 | 3.02 | ... | 3.2  |
| 4.06 | 4.06 | 4.06 | 4.06 | 1.14 | 1.14 | 1.14 | 1.14 | 0.42 | 0.42 | 0.42 | 0.42 | ... | 0.28 |
| 4.06 | 4.06 | 4.06 | 4.06 | 1.14 | 1.14 | 1.14 | 1.14 | 0.42 | 0.42 | 0.42 | 0.42 | ... | 0.28 |
| 4.06 | 4.06 | 4.06 | 4.06 | 1.14 | 1.14 | 1.14 | 1.14 | 0.42 | 0.42 | 0.42 | 0.42 | ... | 0.28 |
| 4.06 | 4.06 | 4.06 | 4.06 | 1.14 | 1.14 | 1.14 | 1.14 | 0.42 | 0.42 | 0.42 | 0.42 | ... | 0.28 |
| 3.78 | 3.78 | 3.78 | 3.78 | 3.78 | 3.78 | 3.78 | 3.78 | 3.48 | 3.48 | 3.48 | 3.48 | ... | 1.44 |
| 3.78 | 3.78 | 3.78 | 3.78 | 3.78 | 3.78 | 3.78 | 3.78 | 3.48 | 3.48 | 3.48 | 3.48 | ... | 1.44 |
| ...  | ...  | ...  | ...  | ...  | ...  | ...  | ...  | ...  | ...  | ...  | ...  | ... | ...  |
| 4.98 | 4.98 | 4.98 | 4.98 | 4.92 | 4.92 | 4.92 | 4.92 | 1.82 | 1.82 | 1.82 | 1.82 | ... | 0.04 |

Tabel 3.2 Contoh *Input* 64x64 Channel G

|      |      |      |      |      |      |      |      |      |      |      |      |     |      |
|------|------|------|------|------|------|------|------|------|------|------|------|-----|------|
| 2.04 | 2.04 | 2.04 | 2.04 | 3.94 | 3.94 | 3.94 | 3.94 | 3.98 | 3.98 | 3.98 | 3.98 | ... | 3.18 |
| 2.04 | 2.04 | 2.04 | 2.04 | 3.94 | 3.94 | 3.94 | 3.94 | 3.98 | 3.98 | 3.98 | 3.98 | ... | 3.18 |
| 2.04 | 2.04 | 2.04 | 2.04 | 3.94 | 3.94 | 3.94 | 3.94 | 3.98 | 3.98 | 3.98 | 3.98 | ... | 3.18 |
| 2.04 | 2.04 | 2.04 | 2.04 | 3.94 | 3.94 | 3.94 | 3.94 | 3.98 | 3.98 | 3.98 | 3.98 | ... | 3.18 |
| 1.9  | 1.9  | 1.9  | 1.9  | 4.3  | 4.3  | 4.3  | 4.3  | 4.64 | 4.64 | 4.64 | 4.64 | ... | 2    |
| 1.9  | 1.9  | 1.9  | 1.9  | 4.3  | 4.3  | 4.3  | 4.3  | 4.64 | 4.64 | 4.64 | 4.64 | ... | 2    |
| 1.9  | 1.9  | 1.9  | 1.9  | 4.3  | 4.3  | 4.3  | 4.3  | 4.64 | 4.64 | 4.64 | 4.64 | ... | 2    |
| 1.9  | 1.9  | 1.9  | 1.9  | 4.3  | 4.3  | 4.3  | 4.3  | 4.64 | 4.64 | 4.64 | 4.64 | ... | 2    |
| 2.24 | 2.24 | 2.24 | 2.24 | 3.66 | 3.66 | 3.66 | 3.66 | 1.6  | 1.6  | 1.6  | 1.6  | ... | 2.5  |
| 2.24 | 2.24 | 2.24 | 2.24 | 3.66 | 3.66 | 3.66 | 3.66 | 1.6  | 1.6  | 1.6  | 1.6  | ... | 2.5  |
| 2.24 | 2.24 | 2.24 | 2.24 | 3.66 | 3.66 | 3.66 | 3.66 | 1.6  | 1.6  | 1.6  | 1.6  | ... | 2.5  |
| 2.24 | 2.24 | 2.24 | 2.24 | 3.66 | 3.66 | 3.66 | 3.66 | 1.6  | 1.6  | 1.6  | 1.6  | ... | 2.5  |
| 2.58 | 2.58 | 2.58 | 2.58 | 1.04 | 1.04 | 1.04 | 1.04 | 3.42 | 3.42 | 3.42 | 3.42 | ... | 0.74 |
| 2.58 | 2.58 | 2.58 | 2.58 | 1.04 | 1.04 | 1.04 | 1.04 | 3.42 | 3.42 | 3.42 | 3.42 | ... | 0.74 |
| ...  | ...  | ...  | ...  | ...  | ...  | ...  | ...  | ...  | ...  | ...  | ...  | ... | ...  |
| 1.96 | 1.96 | 1.96 | 1.96 | 3.74 | 3.74 | 3.74 | 3.74 | 4.48 | 4.48 | 4.48 | 4.48 | ... | 4.66 |

Tabel 3.3 Contoh *Input* 64x64 Channel B

|      |      |      |      |      |      |      |      |      |      |      |      |     |      |
|------|------|------|------|------|------|------|------|------|------|------|------|-----|------|
| 4.44 | 4.44 | 4.44 | 4.44 | 1.06 | 1.06 | 1.06 | 1.06 | 1.14 | 1.14 | 1.14 | 1.14 | ... | 4.66 |
| 4.44 | 4.44 | 4.44 | 4.44 | 1.06 | 1.06 | 1.06 | 1.06 | 1.14 | 1.14 | 1.14 | 1.14 | ... | 4.66 |
| 4.44 | 4.44 | 4.44 | 4.44 | 1.06 | 1.06 | 1.06 | 1.06 | 1.14 | 1.14 | 1.14 | 1.14 | ... | 4.66 |
| 4.44 | 4.44 | 4.44 | 4.44 | 1.06 | 1.06 | 1.06 | 1.06 | 1.14 | 1.14 | 1.14 | 1.14 | ... | 4.66 |
| 2.58 | 2.58 | 2.58 | 2.58 | 4.34 | 4.34 | 4.34 | 4.34 | 0.32 | 0.32 | 0.32 | 0.32 | ... | 0.98 |
| 2.58 | 2.58 | 2.58 | 2.58 | 4.34 | 4.34 | 4.34 | 4.34 | 0.32 | 0.32 | 0.32 | 0.32 | ... | 0.98 |



|      |      |      |      |      |      |      |      |      |      |      |      |     |      |
|------|------|------|------|------|------|------|------|------|------|------|------|-----|------|
| 2.58 | 2.58 | 2.58 | 2.58 | 4.34 | 4.34 | 4.34 | 4.34 | 0.32 | 0.32 | 0.32 | 0.32 | ... | 0.98 |
| 2.58 | 2.58 | 2.58 | 2.58 | 4.34 | 4.34 | 4.34 | 4.34 | 0.32 | 0.32 | 0.32 | 0.32 | ... | 0.98 |
| 3.04 | 3.04 | 3.04 | 3.04 | 3.02 | 3.02 | 3.02 | 3.02 | 0.24 | 0.24 | 0.24 | 0.24 | ... | 2.82 |
| 3.04 | 3.04 | 3.04 | 3.04 | 3.02 | 3.02 | 3.02 | 3.02 | 0.24 | 0.24 | 0.24 | 0.24 | ... | 2.82 |
| 3.04 | 3.04 | 3.04 | 3.04 | 3.02 | 3.02 | 3.02 | 3.02 | 0.24 | 0.24 | 0.24 | 0.24 | ... | 2.82 |
| 3.04 | 3.04 | 3.04 | 3.04 | 3.02 | 3.02 | 3.02 | 3.02 | 0.24 | 0.24 | 0.24 | 0.24 | ... | 2.82 |
| 4.06 | 4.06 | 4.06 | 4.06 | 2.28 | 2.28 | 2.28 | 2.28 | 4.28 | 4.28 | 4.28 | 4.28 | ... | 3.78 |
| 4.06 | 4.06 | 4.06 | 4.06 | 2.28 | 2.28 | 2.28 | 2.28 | 4.28 | 4.28 | 4.28 | 4.28 | ... | 3.78 |
| ...  | ...  | ...  | ...  | ...  | ...  | ...  | ...  | ...  | ...  | ...  | ...  | ... | ...  |
| 3.06 | 3.06 | 3.06 | 3.06 | 4.02 | 4.02 | 4.02 | 4.02 | 1.78 | 1.78 | 1.78 | 1.78 | ... | 2.38 |

Tabel 3.4 Contoh Kernel 7x7 *Channel R*

|   |    |    |    |    |    |    |
|---|----|----|----|----|----|----|
| 1 | 2  | 0  | 2  | 2  | -1 | 0  |
| 0 | 0  | 2  | 1  | 0  | 2  | -1 |
| 1 | -1 | 1  | 2  | -2 | 1  | -1 |
| 2 | 1  | -2 | -2 | 0  | 0  | -1 |
| 1 | 1  | 0  | 1  | 1  | -2 | 0  |
| 2 | 0  | 2  | -2 | -1 | 1  | -2 |
| 1 | -1 | -1 | -2 | -1 | 2  | -1 |

Tabel 3.5 Contoh Kernel 7x7 *Channel G*

|    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|
| 1  | 1  | 1  | 1  | 2  | 0  | -2 |
| 1  | -1 | 1  | -1 | -1 | 1  | 2  |
| -1 | -1 | 1  | -1 | -1 | 1  | 1  |
| -2 | 2  | 2  | -1 | 2  | -1 | -2 |
| 1  | 1  | 1  | 2  | -2 | 2  | 2  |
| -2 | -2 | -2 | -2 | 1  | 0  | 0  |
| -2 | 0  | 0  | -2 | 0  | 2  | -1 |

Tabel 3.6 Contoh Kernel 7x7 *Channel B*

|    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|
| -1 | -2 | 1  | -2 | 1  | -1 | -2 |
| 2  | 0  | 1  | 0  | 0  | -2 | 0  |
| 2  | 0  | -2 | 2  | -1 | 0  | -2 |
| -1 | -1 | 1  | 2  | 0  | -2 | 1  |
| 2  | 1  | 2  | 2  | 0  | 2  | 1  |
| 2  | 0  | 0  | 1  | -1 | -1 | 2  |
| -2 | 2  | 1  | 1  | 1  | 1  | 1  |

Langkah konvolusi diambil dengan menggeser kernel dua piksel setiap langkah (stride = 2), lalu menjumlahkan hasil perkalian elemen-elemen yang saling berhadapan. Sebagai contoh, hasil konvolusi di posisi awal (pojok kiri atas) untuk channel R adalah:

***Channel R***

$$\begin{aligned}
&= (2,04x1)+(2,04x1)+(2,04x1)+(2,04x1)+(3,58x2)+(3,58x0)+(3,58x-2) \\
&+(2,04x0)+(2,04x0)+(2,04x2)+(2,04x1)+(3,58x0)+(3,58x2)+(3,58x-1) \\
&+(2,04x1)+(2,04x-1)+(2,04x1)+(2,04x2)+(3,58x-2)+(3,58x1)+(3,58x-1) \\
&+(2,04x2)+(2,04x1)+(2,04x-1)+(2,04x-2)+(3,58x0)+(3,58x0)+(3,58x-2) \\
&+(1,98x1)+(1,98x1)+(1,98x0)+(1,98x1)+(2,06x1)+(2,06x-2)+(2,06x0) \\
&+(1,98x2)+(1,98x0)+(1,98x-2)+(1,98x-2)+(2,06x-1)+(2,06x1)+(2,06x-2) \\
&+(1,98x1)+(1,98x-1)+(1,98x-1)+(1,98x-2)+(2,06x-1)+(2,06x2)+(2,06x-1) = 11,92
\end{aligned}$$

Setelah proses konvolusi pojok kiri pada channel R telah dilakukan, selanjutnya dilakukan hal yang sama pada channel G dan B kemudian ketiga nilai tersebut di jumlahkan, sehingga nilai fitur hasil konvolusi paling pojok kiri di temukan. Berikut contoh hasil konvolusi di posisi awal (pojok kiri atas) untuk channel G dan B:

***Channel G***

$$\begin{aligned}
&= (2,04x1)+(2,04x1)+(2,04x1)+(2,04x1)+(3,94x2)+(3,94x0)+(3,94x-2) \\
&+(2,04x1)+(2,04x-1)+(2,04x1)+(2,04x-1)+(3,94x-1)+(3,94x1)+(3,94x2) \\
&+(2,04x-1)+(2,04x-1)+(2,04x1)+(2,04x-1)+(3,94x-1)+(3,94x1)+(3,94x1) \\
&+(2,04x-2)+(2,04x2)+(2,04x2)+(2,04x-1)+(3,94x2)+(3,94x-1)+(3,94x-2) \\
&+(1,9x1)+(1,9x1)+(1,9x1)+(1,9x2)+(4,3x-2)+(4,3x2)+(4,3x2)
\end{aligned}$$

$$+(1.9x-2)+(1.9x-2)+(1.9x-2)+(1.9x-2)+(4.3x1)+(4.3x0)+(4.3x0)$$

$$+(1.9x-2)+(1.9x0)+(1.9x0)+(1.9x-2)+(4.3x0)+(4.3x2)+(4.3x-1) = -8.92$$

### ***Channel B***

$$= (4.44x-1)+(4.44x-2)+(4.44x1)+(4.44x-2)+(1.06x1)+(1.06x-1)+(1.06x-2)$$

$$+(4.44x2)+(4.44x0)+(4.44x1)+(4.44x0)+(1.06x0)+(1.06x-2)+(1.06x0)$$

$$+(4.44x2)+(4.44x0)+(4.44x-2)+(4.44x2)+(1.06x-1)+(1.06x0)+(1.06x-2)$$

$$+(4.44x-1)+(4.44x-1)+(4.44x1)+(4.44x2)+(1.06x0)+(1.06x-2)+(1.06x1)$$

$$+(2.58x2)+(2.58x1)+(2.58x2)+(2.58x2)+(4.34x0)+(4.34x2)+(4.34x1)$$

$$+(2.58x2)+(2.58x0)+(2.58x0)+(2.58x1)+(4.34x-1)+(4.34x-1)+(4.34x-2)$$

$$+(2.58x-2)+(2.58x2)+(2.58x1)+(2.58x1)+(4.34x1)+(4.34x1)+(4.34x1) = 56,4$$

Dari ketiga proses di atas, hasil konvolusi masing-masing *channel* di lakukan penjumlahan, sehingga nilai fitur hasil konvolusi pertama di dapatkan:

$$11,92+(-8.92)+ 56,4=59,4$$

Setelah proses konvolusi dilakukan ke seluruh wilayah gambar, hasilnya akan melalui proses *Max Pooling* untuk mereduksi dimensi sambil mempertahankan fitur terpenting. Sebagai contoh, jika hasil proses konvolusi sebelumnya adalah:

Tabel 3.7 Contoh Hasil Proses Konvolusi

|       |       |       |       |       |       |       |       |       |     |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-----|-------|
| 59.4  | 15.24 | 2.38  | -1.6  | 18.26 | 53.9  | 43.66 | 46.68 | 42.42 | ... | 70.22 |
| 57.96 | 15.48 | 20.26 | 19.6  | 22.5  | 62.34 | 70.46 | 59.72 | 36.92 | ... | 58.16 |
| 46.78 | 33.3  | 28.62 | 23.66 | 33.4  | 58.22 | 63.58 | 69.88 | 35    | ... | 34.5  |
| 27.58 | 39.68 | 39.62 | 35.54 | 45.2  | 57.58 | 50.58 | 58.72 | 48.76 | ... | 22.2  |
| 29.36 | 34.06 | 46.64 | 52.48 | 33.14 | 6.28  | 41.34 | 58.1  | 68.72 | ... | 27.9  |
| 51.8  | 51.98 | 44.74 | 41.06 | 44.7  | -0.6  | 8.78  | 30.68 | 47.7  | ... | 49.08 |
| 56.32 | 50.7  | 46.44 | 31.58 | 54.68 | 33.16 | 28.08 | 31.76 | 28.04 | ... | 64.88 |
| 30.6  | 44.42 | 66.18 | 38.8  | 20.3  | -3.6  | 14.08 | 21.52 | 20.48 | ... | 56.24 |
| 32.78 | 62.6  | 70.82 | 22.86 | 6     | 0.74  | 16.6  | 1.82  | 11.44 | ... | 47.84 |
| 34.86 | 60.36 | 52.74 | 1.48  | 12.82 | 31.26 | 51.14 | 24.44 | 2.14  | ... | 14.12 |
| 21.96 | 38.78 | 40.14 | 21.26 | 34.92 | 56.32 | 43.72 | 14.08 | 0.36  | ... | -0.62 |
| 19.8  | 29.88 | 36.72 | 37.94 | 67.04 | 82.54 | 41.34 | 22.82 | 34.1  | ... | 47.78 |
| ...   | ...   | ...   | ...   | ...   | ...   | ...   | ...   | ...   | ... | ...   |
| 37.38 | 2.38  | -0.5  | 37.24 | 57.02 | 70.94 | 34.22 | 18.2  | 47.38 | ... | 20.62 |

Dari nilai fitur di atas dilakukan proses *Max Pooling* 3x3, sebagai contoh :

|       |       |       |
|-------|-------|-------|
| 59.4  | 15.24 | 2.38  |
| 57.96 | 15.48 | 20.26 |
| 46.78 | 33.3  | 28.62 |

Maka hasil *Max Pooling* adalah nilai tertinggi: 59,4. Proses ini dilakukan ke seluruh fitur dengan stride = 2. Jadi untuk hasil *Max Pooling* diambil nilai tertingginya adalah sebagai berikut:

Tabel 3.8 Contoh Hasil Proses Max Pooling

|       |           |       |       |       |       |            |            |       |        |       |
|-------|-----------|-------|-------|-------|-------|------------|------------|-------|--------|-------|
| 59.4  | 33.4      | 70.46 | 70.46 | 42.42 | 69.1  | 110.9<br>6 | 103.2<br>2 | 64.1  | 104.92 | 116.8 |
| 46.78 | 52.4<br>8 | 63.58 | 69.88 | 68.72 | 69.1  | 95.86      | 99.78      | 74.08 | 104.92 | 116.8 |
| 56.32 | 54.6<br>8 | 54.68 | 68.72 | 68.72 | 56.96 | 54.16      | 94.56      | 85.24 | 51.48  | 44.58 |
| 70.82 | 70.8<br>2 | 54.68 | 31.76 | 79.48 | 79.48 | 39.64      | 80.62      | 80.62 | 51.48  | 61.68 |
| 70.82 | 70.8<br>2 | 56.32 | 51.14 | 79.54 | 79.54 | 34.28      | 41.66      | 76.76 | 76.76  | 61.68 |
| 48.34 | 67.0<br>4 | 82.54 | 43.72 | 79.54 | 79.54 | 85.52      | 64.42      | 76.76 | 76.76  | 45.14 |
| 95.52 | 56.3<br>8 | 58.36 | 44.86 | 49.18 | 69.92 | 85.64      | 83.12      | 79.22 | 36.86  | 26.48 |
| 95.52 | 51.0<br>2 | 58.36 | 87.86 | 85.02 | 69.56 | 64.6       | 83.12      | 79.22 | 110.72 | 82.72 |

|       |           |            |       |       |            |            |       |            |        |            |
|-------|-----------|------------|-------|-------|------------|------------|-------|------------|--------|------------|
| 73.4  | 60.3<br>8 | 46.16      | 87.86 | 85.02 | 86.28      | 76.42      | 41.2  | 75.06      | 120.82 | 120.8<br>2 |
| 91.98 | 97.3<br>4 | 107.5<br>2 | 79.46 | 79.54 | 111.6<br>2 | 111.6<br>2 | 47.66 | 56.94      | 103.54 | 103.5<br>4 |
| 79.66 | 97.3<br>4 | 107.5<br>2 | 79.26 | 79.54 | 100.2<br>2 | 100.2<br>2 | 97.54 | 104.0<br>6 | 104.06 | 71.9       |
| 75.34 | 50.2<br>8 | 40.1       | 59.14 | 66.9  | 74.46      | 66.28      | 97.54 | 112.2<br>2 | 104.06 | 71.9       |
| 75.34 | 44.7      | 37.92      | 43.02 | 83.02 | 83.02      | 56.64      | 85.54 | 112.2<br>2 | 89.38  | 54.98      |
| 61.2  | 57.0<br>2 | 70.94      | 48.4  | 83.02 | 83.02      | 85.16      | 96.38 | 103.0<br>8 | 73.32  | 54.5       |

Setelah proses *Max Pooling* di lakukan dan di dapat hasil seperti pada tabel 3.8 di atas, selanjutnya di lakukan proses *Residual Blocks* dengan kernel filter 3x3 sebanyak 4 kali. Sebagai contoh *input* dari hasil *Max Pooling* di atas akan dilakukan proses *Residual Blocks* dengan kernel seperti berikut

|   |   |    |
|---|---|----|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

Langkahnya diambil dengan menggeser kernel dua piksel setiap langkah (stride = 2), lalu menjumlahkan hasil perkalian elemen-elemen yang saling berhadapan. Sebagai contoh, hasil proses *Residual Block* di posisi awal (pojok kiri atas) adalah:

$$\begin{aligned}
 &= (59,5 \times 1) + (33,4 \times 0) + (70,46 \times -1) \\
 &\quad + (46,78 \times 1) + (52,48 \times 0) + (63,58 \times -1) \\
 &\quad + (56,32 \times 1) + (54,68 \times 0) + (54,68 \times -1) = -26.22
 \end{aligned}$$

Proses *Residual Block* dilakukan ke seluruh fitur hasil *Max Pooling* di atas. Hasilnya akan dilakukan proses *skip connection* yaitu menjumlahkan hasil proses

*Residual Block* dengan *input*-nya. Tetapi karena nilai fitur hasil *Max Pooling* memiliki ukuran yang berbeda dengan nilai fitur hasil *Residual Block*, maka dilakukan proses *Downsappling* pada nilai fitur hasil *Max Pooling* agar ukurannya sama dengan nilai fitur hasil *Residual Block*. Untuk melakukannya yaitu dengan mengambil nilai rata-rata dari fitur dalam blok 2x2 untuk seluruh dimensi fitur nya.

Contoh dari tabel 3.8 untuk nilai awal (pojok kiri atas):

$$= (59,5+33,4+46,78+52,48)/4 = 48.015$$

Proses ini dilakukan ke seluruh dimensi fitur sehingga ukurannya sama seperti nilai fitur hasil *Residual Block*.

Tabel 3.9 Contoh Hasil Proses *Downsappling*

|        |        |        |         |        |        |        |
|--------|--------|--------|---------|--------|--------|--------|
| 48.015 | 68.595 | 62.335 | 102.455 | 87.005 | 106.03 | 74.965 |
| 63.16  | 52.46  | 71.16  | 67.245  | 67.205 | 68.685 | 72.955 |
| 64.255 | 58.43  | 79.54  | 56.47   | 76.76  | 57.66  | 62.81  |
| 74.61  | 62.36  | 68.42  | 79.12   | 76.505 | 59.195 | 79.7   |
| 80.775 | 80.25  | 90.615 | 69.225  | 89.09  | 93.47  | 75.695 |
| 75.655 | 71.505 | 80.28  | 90.395  | 106.1  | 77.705 | 80.715 |
| 59.565 | 50.07  | 83.02  | 80.93   | 94.5   | 68.68  | 64.875 |
| 48.015 | 68.595 | 62.335 | 102.455 | 87.005 | 106.03 | 74.965 |
| 63.16  | 52.46  | 71.16  | 67.245  | 67.205 | 68.685 | 72.955 |
| 64.255 | 58.43  | 79.54  | 56.47   | 76.76  | 57.66  | 62.81  |
| 74.61  | 62.36  | 68.42  | 79.12   | 76.505 | 59.195 | 79.7   |
| 80.775 | 80.25  | 90.615 | 69.225  | 89.09  | 93.47  | 75.695 |
| 75.655 | 71.505 | 80.28  | 90.395  | 106.1  | 77.705 | 80.715 |
| 59.565 | 50.07  | 83.02  | 80.93   | 94.5   | 68.68  | 64.875 |

Tabel 3.10 Contoh Hasil *Residual Block*

|        |        |        |         |        |        |        |
|--------|--------|--------|---------|--------|--------|--------|
| -26.22 | 8.86   | -81.12 | 37.56   | -54.76 | 17.02  | 65.72  |
| 32.28  | -62.06 | 99.66  | -114.54 | 74.68  | -58.28 | 27.14  |
| 17.46  | -11.04 | 2.82   | -27.3   | 99.44  | -58.44 | -27.16 |
| 101.56 | -56.34 | -7.44  | -6.84   | 3.48   | 0.62   | 5.36   |
| -16.16 | 17.1   | -44.16 | 52.2    | -60.2  | 71.06  | -15.68 |
| 44.8   | -43.92 | 6.32   | -105.36 | 129.72 | -95.06 | 105.26 |
| 27.68  | -57.18 | 24.24  | -73.5   | 105.82 | -55.08 | 69.62  |
| -26.22 | 8.86   | -81.12 | 37.56   | -54.76 | 17.02  | 65.72  |
| 32.28  | -62.06 | 99.66  | -114.54 | 74.68  | -58.28 | 27.14  |
| 17.46  | -11.04 | 2.82   | -27.3   | 99.44  | -58.44 | -27.16 |
| 101.56 | -56.34 | -7.44  | -6.84   | 3.48   | 0.62   | 5.36   |
| -16.16 | 17.1   | -44.16 | 52.2    | -60.2  | 71.06  | -15.68 |
| 44.8   | -43.92 | 6.32   | -105.36 | 129.72 | -95.06 | 105.26 |
| 27.68  | -57.18 | 24.24  | -73.5   | 105.82 | -55.08 | 69.62  |

Setelah proses *Downsampling* di atas, Maka untuk proses *skip connection* nya yaitu :

$$48.015 + (-26.22) = 21.795$$

Proses ini dilakukan ke seluruh dimensi nilai fitur. Setelah proses *skip connection*, selanjutnya dilakukan proses *activation function (ReLU)* dengan tujuan untuk menghilangkan nilai negatif menjadi 0 sehingga nilai fitur dengan informasi penting tetap di pertahankan.

Tabel 3.11 Contoh Hasil Proses *Residual Block* Final Setelah Dilakukan *Skip Connection*

|         |        |         |         |         |         |         |
|---------|--------|---------|---------|---------|---------|---------|
| 21.795  | 77.455 | -18.785 | 140.015 | 32.245  | 123.05  | 140.685 |
| 95.44   | -9.6   | 170.82  | -47.295 | 141.885 | 10.405  | 100.095 |
| 81.715  | 47.39  | 82.36   | 29.17   | 176.2   | -0.78   | 35.65   |
| 176.17  | 6.02   | 60.98   | 72.28   | 79.985  | 59.815  | 85.06   |
| 64.615  | 97.35  | 46.455  | 121.425 | 28.89   | 164.53  | 60.015  |
| 120.455 | 27.585 | 86.6    | -14.965 | 235.82  | -17.355 | 185.975 |
| 87.245  | -7.11  | 107.26  | 7.43    | 200.32  | 13.6    | 134.495 |
| 21.795  | 77.455 | -18.785 | 140.015 | 32.245  | 123.05  | 140.685 |
| 95.44   | -9.6   | 170.82  | -47.295 | 141.885 | 10.405  | 100.095 |
| 81.715  | 47.39  | 82.36   | 29.17   | 176.2   | -0.78   | 35.65   |
| 176.17  | 6.02   | 60.98   | 72.28   | 79.985  | 59.815  | 85.06   |
| 64.615  | 97.35  | 46.455  | 121.425 | 28.89   | 164.53  | 60.015  |
| 120.455 | 27.585 | 86.6    | -14.965 | 235.82  | -17.355 | 185.975 |
| 87.245  | -7.11  | 107.26  | 7.43    | 200.32  | 13.6    | 134.495 |

Dari hasil proses *Residual Block* seperti pada tabel 3.11 di atas, dilakukan proses *activation function (ReLU)* sehingga nilainya menjadi :

Tabel 3.12 Contoh Hasil Fitur Setelah Dilakukan Proses *Activation Function (ReLU)*

|         |        |        |         |         |        |         |
|---------|--------|--------|---------|---------|--------|---------|
| 21.795  | 77.455 | 0      | 140.015 | 32.245  | 123.05 | 140.685 |
| 95.44   | 0      | 170.82 | 0       | 141.885 | 10.405 | 100.095 |
| 81.715  | 47.39  | 82.36  | 29.17   | 176.2   | 0      | 35.65   |
| 176.17  | 6.02   | 60.98  | 72.28   | 79.985  | 59.815 | 85.06   |
| 64.615  | 97.35  | 46.455 | 121.425 | 28.89   | 164.53 | 60.015  |
| 120.455 | 27.585 | 86.6   | 0       | 235.82  | 0      | 185.975 |
| 87.245  | 0      | 107.26 | 7.43    | 200.32  | 13.6   | 134.495 |
| 21.795  | 77.455 | 0      | 140.015 | 32.245  | 123.05 | 140.685 |
| 95.44   | 0      | 170.82 | 0       | 141.885 | 10.405 | 100.095 |
| 81.715  | 47.39  | 82.36  | 29.17   | 176.2   | 0      | 35.65   |
| 176.17  | 6.02   | 60.98  | 72.28   | 79.985  | 59.815 | 85.06   |
| 64.615  | 97.35  | 46.455 | 121.425 | 28.89   | 164.53 | 60.015  |

|         |        |        |      |        |      |         |
|---------|--------|--------|------|--------|------|---------|
| 120.455 | 27.585 | 86.6   | 0    | 235.82 | 0    | 185.975 |
| 87.245  | 0      | 107.26 | 7.43 | 200.32 | 13.6 | 134.495 |

Proses ini dilakukan sebanyak 4 kali sesuai dengan *flowchart* dari ResNet-18 seperti pada gambar 3.3 di atas. Selanjutnya hasil dari proses ini akan masuk ke dalam *Feature Vector Output*, yang mana dalam penelitian ini akan disimpan dalam file excel (.xlsx).

#### 4. *Feature Vector Output*

- 1) *Output* dari *residual block 4* adalah *feature vector*, yang merupakan representasi numerik dari fitur dalam gambar.
- 2) Vektor ini digunakan sebagai *input* dalam analisis lebih lanjut, yaitu *Clustering* data berdasarkan kemiripan fitur menggunakan model lain.

### 3.3.2 Proses Implementasi ResNet-18

Implementasi ResNet-18 dalam penelitian ini dilakukan dengan beberapa tahapan utama, yaitu pemanggilan model, ekstraksi fitur, dan penyimpanan vektor fitur.

#### 1. Pemanggilan Model

- a) Model ResNet-18 yang telah dilatih sebelumnya (*pretrained model*) digunakan untuk ekstraksi fitur.
- b) Model ini diambil dari dataset *ImageNet* tanpa melakukan *fine-tuning*, sehingga semua bobot model tetap tidak berubah.

#### 2. Proses Ekstraksi Fitur

- a) Gambar masukan yang telah melalui tahap *preprocessing* dimasukkan ke dalam ResNet-18.



- b) Gambar melewati *Convolutional layer*, *Max Pooling*, dan *Residual blocks* untuk menghasilkan fitur tingkat tinggi.
- c) Hasil dari proses ini adalah *feature vector*, yang merupakan representasi dari gambar *input* awal.

### 3. Penyimpanan Vektor Fitur

- a) Vektor fitur yang dihasilkan dari *Residual blocks* step terakhir yang disimpan dalam bentuk numerik.
- b) Vektor ini digunakan dalam proses *Clustering* data berdasarkan kemiripan fitur.
- c) Pada penelitian ini, vektor fitur akan dianalisis lebih lanjut untuk melihat efektivitas ekstraksi fitur menggunakan ResNet-18.

### 3.4 Feature Analysis/Feature Evaluation

Tahap evaluasi dilakukan untuk menganalisis kualitas representasi fitur yang dihasilkan oleh ResNet-18 dari gambar kerusakan bangunan pasca bencana. Tujuan utama dari tahap ini adalah untuk mengetahui apakah vektor fitur yang telah diekstraksi mampu membedakan gambar berdasarkan tingkat kerusakannya. Dalam penelitian ini, proses evaluasi tidak dilakukan melalui metrik klasifikasi seperti akurasi, presisi, atau *recall*, karena tidak ada proses klasifikasi eksplisit.

Sebagai gantinya, digunakan pendekatan berbasis *unsupervised learning* dengan metode *Principal Component Analysis* (PCA) dan *K-Means Clustering*. PCA digunakan untuk mereduksi dimensi data fitur agar pola distribusi dapat lebih mudah dianalisis secara visual maupun numerik. Selanjutnya, hasil dari PCA

digunakan dalam proses *Clustering* menggunakan *K-Means* untuk mengelompokkan gambar berdasarkan kemiripan fitur.

Metode ini memungkinkan analisis struktur data tanpa label eksplisit dan memberikan gambaran sejauh mana fitur hasil ekstraksi mampu membentuk pengelompokan yang bermakna.

### 3.4.1 *Principal Component Analysis (PCA)*

*Principal Component Analysis (PCA)* merupakan teknik statistik yang digunakan untuk mereduksi dimensi data sambil mempertahankan informasi yang paling signifikan. Dalam penelitian ini, PCA digunakan untuk menyederhanakan vektor fitur berdimensi tinggi yang dihasilkan oleh ResNet-18 menjadi dimensi yang lebih rendah, seperti 2D, sehingga mempermudah proses analisis dan visualisasi.

Secara matematis, PCA menghitung *eigenvalue* dan *eigenvector* dari matriks kovarians

$$\Sigma = \frac{1}{n} \sum_{\{i=1\}}^{\{n\}} (x_i - \mu)(x_i - \mu)^T \quad (3.3)$$

Dimana:

$x_i$ : vektor fitur ke- $i$

$\mu$ : rata-rata semua vektor fitur

$\Sigma$ : matriks kovarians

$v$ : *eigenvector (principal component)*

$\lambda$ : *eigenvalue* (besarnya variasi yang dijelaskan)

Komponen utama dipilih berdasarkan nilai *eigenvalue* tertinggi untuk menghasilkan representasi data yang tetap informatif meskipun dimensinya dikurangi.

### 3.4.2 *K-Means Clustering*

Setelah fitur direduksi dengan PCA, tahap selanjutnya adalah proses *Clustering* menggunakan algoritma *K-Means*. Metode ini bertujuan untuk mengelompokkan gambar berdasarkan kedekatan fitur ke dalam sejumlah *cluster*  $k$ , yang merepresentasikan kemungkinan kategori tingkat kerusakan bangunan.

Algoritma *K-Means* bekerja dengan iterasi sebagai berikut:

1. Inisialisasi *centroid*  $\mu_1, \mu_2, \dots, \mu_k$  secara acak.
2. Alokasikan setiap data  $x_i$  ke *cluster* dengan *centroid* terdekat berdasarkan rumus jarak *Euclidean*:

$$\mu_j = \frac{1}{|C_j|} \sum_{\{x_i \in C_j\}} x_i \quad (3.4)$$

3. Hitung ulang *centroid* berdasarkan rata-rata titik dalam setiap *cluster*:
4. Ulangi langkah 2 dan 3 hingga *konvergen*.

Pemilihan nilai  $k$  dalam penelitian ini mengacu pada kualitas fitur yang telah diekstrak, agar hasil *Clustering* dapat merepresentasikan kategori kerusakan dengan lebih akurat.

### 3.4.3 Proses Evaluasi

Proses evaluasi dilakukan melalui gabungan antara PCA dan *K-Means*. Langkah-langkah evaluasi fitur yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Ekstraksi Fitur

Dataset gambar kerusakan bangunan diproses melalui ResNet-18 untuk menghasilkan vektor fitur. Vektor ini disimpan dalam bentuk numerik sebagai representasi karakteristik dari masing-masing gambar.

2. Reduksi Dimensi dengan PCA

Vektor fitur berdimensi tinggi kemudian diproses menggunakan PCA untuk mereduksi dimensi menjadi tiga dimensi utama. Reduksi ini bertujuan agar data dapat divisualisasikan dan diproses lebih lanjut dengan efisien.

3. Proses *Clustering* dengan *K-Means*

Hasil reduksi fitur dari PCA menjadi *input* bagi algoritma *K-Means*. Data di*cluster* berdasarkan kedekatan fitur dalam ruang hasil reduksi PCA.

4. Visualisasi dan Analisis *Cluster*

Visualisasi hasil *Clustering* dilakukan dalam bentuk *scatter plot* untuk melihat distribusi data dalam ruang 2D. Apabila fitur berkualitas baik, maka data dengan tingkat kerusakan serupa akan berada dalam *cluster* yang sama atau berdekatan.

## BAB IV

### UJI COBA DAN PEMBAHASAN

#### 4.1 Data Pengujian

Pada tahap awal penelitian ini, data yang digunakan berasal dari Badan Penanggulangan Bencana Daerah (BPBD) Kota Malang. Dataset yang dipilih terdiri dari 36 gambar bangunan yang mengalami kerusakan pasca bencana alam. Data citra ini menjadi dasar untuk proses ekstraksi fitur menggunakan metode ResNet-18. Sebelum dilakukan ekstraksi fitur, gambar mengalami proses *preprocessing* yang meliputi *resizing*, *centering data*, dan normalisasi *pixel*. Gambar diubah ukurannya menjadi 224 x 224 piksel agar sesuai dengan kebutuhan *input* pada model ResNet-18. Selanjutnya, dilakukan proses centering data untuk menyeimbangkan intensitas *pixel*, serta normalisasi *pixel* untuk menstandarisasi nilai *pixel* agar berada dalam rentang yang sesuai, sehingga dapat meningkatkan performa model dalam mengekstrak fitur yang representatif. Setelah melewati tahap *preprocessing*, dilakukan ekstraksi fitur menggunakan model ResNet-18. Hasil ekstraksi fitur yang dilakukan berupa matriks data dengan total 49 baris dan 512 kolom, di mana baris mewakili hasil akhir dari ekstraksi fitur nya dengan bentuk matriks 7x7 yang di jadikan 49 baris dan setiap kolom merepresentasikan fitur yang dihasilkan dari model. Dimensi fitur yang besar ini merepresentasikan karakteristik visual dari data citra kerusakan bangunan. Untuk memahami dan mengolah data fitur yang memiliki dimensi tinggi tersebut, dilakukan analisis dengan menggunakan metode PCA. PCA berfungsi untuk mereduksi dimensi data

fitur yang kompleks menjadi beberapa komponen utama yang paling signifikan, sehingga dapat memudahkan visualisasi dan pemahaman pola dalam data, sekaligus mengurangi redundansi dan mempercepat proses pengolahan selanjutnya.

#### 4.2 Hasil Ekstraksi Fitur ResNet-18

Proses ekstraksi fitur dari citra bangunan rusak dalam penelitian ini dilakukan dengan memanfaatkan arsitektur jaringan konvolusional ResNet-18. Model ini terdiri dari serangkaian layer konvolusi, *pooling*, dan *residual blocks* yang dirancang untuk mengekstraksi fitur secara bertahap, dari yang paling dasar hingga abstrak. Setiap blok bertugas menangkap karakteristik visual yang semakin kompleks.

Berikut adalah uraian detail untuk setiap tahapan konvolusi berdasarkan *flowchart* ResNet-18:

1. Conv1 -  $7 \times 7$ , 64 filter, stride = 2



Gambar 4.1 *Input* Proses Konvolusi Pertama

Gambar 4.1 menunjukkan citra *input* awal yang masuk ke dalam ResNet-

18. *Input* ini berupa citra RGB berukuran  $224 \times 224$  piksel.

Tabel 4.1 Nilai Pixel (R) Sebagai *Input* ResNet-18

|           | Pixel 1  | Pixel 2  | Pixel 3 | Pixel 4 | Pixel 5  | Pixel n | Pixel 224 |
|-----------|----------|----------|---------|---------|----------|---------|-----------|
| Pixel 1   | -1.25431 | -1.25431 | -1.1567 | -1.1323 | -1.1323  | ...     | -0.35147  |
| Pixel 2   | -1.25431 | -1.25431 | -1.1079 | -1.1811 | -1.20551 | ...     | -0.66868  |
| Pixel 3   | -1.20551 | -1.22991 | -1.0835 | -1.1567 | -1.22991 | ...     | -0.96149  |
| Pixel 4   | -1.22991 | -1.20551 | -1.1567 | -1.0835 | -1.25431 | ...     | -1.0103   |
| Pixel 5   | -1.30311 | -1.22991 | -1.1811 | -1.1079 | -1.20551 | ...     | -1.0591   |
| Pixel n   | ...      | ...      | ...     | ...     | ...      | ...     | ...       |
| Pixel 224 | 2.601071 | 2.57667  | 2.57667 | 2.57667 | 2.57667  | ...     | 2.503467  |

Tabel 4.2 Nilai Pixel (G) Sebagai *Input* ResNet-18

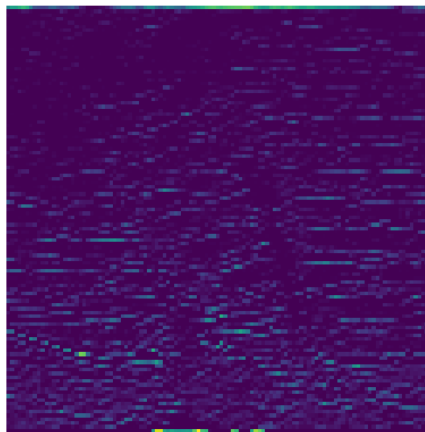
|           | Pixel 1  | Pixel 2  | Pixel 3  | Pixel 4  | Pixel 5  | Pixel n | Pixel 224 |
|-----------|----------|----------|----------|----------|----------|---------|-----------|
| Pixel 1   | -1.63243 | -1.6612  | -1.60365 | -1.57487 | -1.48855 | ...     | -0.39506  |
| Pixel 2   | -1.63243 | -1.6612  | -1.5461  | -1.6612  | -1.57487 | ...     | -0.74037  |
| Pixel 3   | -1.57487 | -1.63243 | -1.51732 | -1.63243 | -1.63243 | ...     | -1.11446  |
| Pixel 4   | -1.60365 | -1.63243 | -1.63243 | -1.57487 | -1.63243 | ...     | -1.17201  |
| Pixel 5   | -1.68998 | -1.6612  | -1.6612  | -1.60365 | -1.6612  | ...     | -1.17201  |
| Pixel n   | ...      | ...      | ...      | ...      | ...      | ...     | ...       |
| Pixel 224 | 2.568859 | 2.540083 | 2.540083 | 2.540083 | 2.540083 | ...     | 1.734357  |

Tabel 4.3 Nilai Pixel (B) Sebagai *Input* ResNet-18

|           | Pixel 1  | Pixel 2  | Pixel 3  | Pixel 4  | Pixel 5  | Pixel n | Pixel 224 |
|-----------|----------|----------|----------|----------|----------|---------|-----------|
| Pixel 1   | -0.24346 | -0.27488 | -0.27488 | -0.43194 | -0.40053 | ...     | -0.5576   |
| Pixel 2   | -0.24346 | -0.27488 | -0.21205 | -0.49477 | -0.49477 | ...     | -0.96597  |
| Pixel 3   | -0.18064 | -0.24346 | -0.18064 | -0.43194 | -0.52618 | ...     | -1.2801   |
| Pixel 4   | -0.18064 | -0.18064 | -0.21205 | -0.36912 | -0.52618 | ...     | -1.34292  |
| Pixel 5   | -0.24346 | -0.21205 | -0.24346 | -0.36912 | -0.52618 | ...     | -1.37434  |
| Pixel n   | ...      | ...      | ...      | ...      | ...      | ...     | ...       |
| Pixel 224 | 2.39524  | 2.363827 | 2.363827 | 2.363827 | 2.363827 | ...     | 0.227733  |

Tabel 4.1, 4.2, dan 4.3 masing-masing menampilkan nilai normalisasi piksel untuk channel *Red (R)*, *Green (G)*, dan *Blue (B)*. Tampak bahwa nilai pixel bervariasi di tiap posisi, menandakan intensitas warna yang berbeda dalam satu gambar. Pada tahap ini, dilakukan operasi konvolusi menggunakan 64 buah filter berukuran  $7 \times 7$  dan stride sebesar 2. Hasilnya adalah *feature map* berukuran  $112 \times 112 \times 64$ . Seperti yang di tunjukkan pada tabel 4.4 di bawah. Proses ini

bertujuan untuk menangkap fitur global dari gambar seperti kontur besar dan tepi utama, serta mengurangi ukuran spasial gambar agar efisien untuk tahap selanjutnya. Karena ukuran kernel cukup besar, informasi visual awal seperti bentuk-bentuk utama dapat langsung dikenali. Tampilan visual Gambar 4.2 memperlihatkan respons awal jaringan terhadap pola-pola besar seperti batas dinding atau tepi struktur.



Gambar 4.2 *Output* Konvolusi Pertama

Tabel 4.4 Hasil Proses Konvolusi Conv1 -  $7 \times 7$ , 64 filter, stride = 2

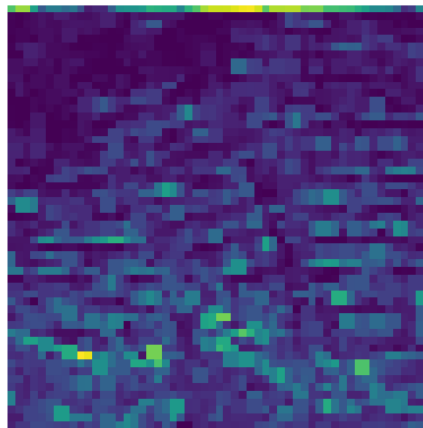
| ch1      | ch2      | ch3 | ch4      | ch5      | ... | ch64     |
|----------|----------|-----|----------|----------|-----|----------|
| 3.218376 | 3.446103 | 0   | 0.722503 | 0        | ... | 0        |
| 4.008309 | 5.418897 | 0   | 0.53376  | 0        | ... | 0.557797 |
| 3.986556 | 6.096347 | 0   | 0.604136 | 0        | ... | 0        |
| 4.379931 | 6.716126 | 0   | 0.733955 | 0        | ... | 0.091494 |
| 4.657507 | 7.171926 | 0   | 0.897489 | 0        | ... | 0.510254 |
| ...      | ...      | ... | ...      | ...      | ... | ...      |
| 0        | 0        | 0   | 0        | 4.928942 | ... | 0        |

## 2. Max Pooling - $3 \times 3$ , stride = 2

Tahap selanjutnya adalah *Max Pooling* untuk menyaring informasi penting dari hasil konvolusi awal. Setelah proses konvolusi awal, hasilnya diproses dengan *Max Pooling* menggunakan kernel  $3 \times 3$  dan stride 2, sehingga ukuran spasial diperkecil menjadi  $56 \times 56 \times 64$ . *Pooling* berfungsi menyaring informasi dominan



dari masing-masing area kecil, menjaga fitur paling signifikan, dan membantu menstabilkan hasil ekstraksi fitur terhadap variasi translasi kecil pada *input* gambar. Tabel 4.5 menampilkan nilai maksimum yang dipilih dalam setiap area kecil. Nilai ini akan digunakan untuk proses selanjutnya. Gambar 4.3 secara visual menunjukkan bahwa area dengan fitur kuat tetap dipertahankan, sedangkan detail kecil atau noise diabaikan.



Gambar 4.3 *Output Proses Max Pooling*

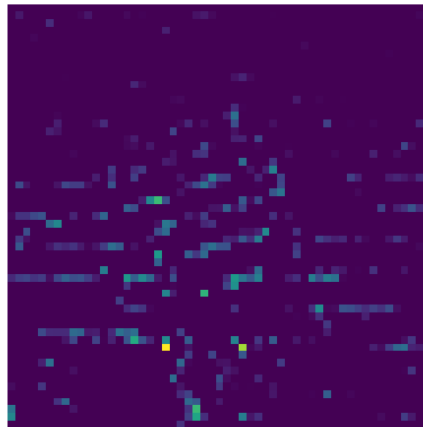
Tabel 4.5 Fitur Hasil Proses Max Pooling -  $3 \times 3$ , stride = 2

| ch1      | ch2      | ch3      | ch4      | ch5      | ... | ch64     |
|----------|----------|----------|----------|----------|-----|----------|
| 4.008309 | 6.096347 | 0        | 2.22513  | 0        | ... | 0.66005  |
| 4.657507 | 7.171926 | 0        | 2.500003 | 0        | ... | 0.846095 |
| 4.657507 | 7.171926 | 0        | 2.500003 | 0        | ... | 2.046725 |
| 3.130242 | 5.34165  | 0        | 1.318245 | 0        | ... | 1.980518 |
| 1.555522 | 2.528114 | 0        | 1.070778 | 0        | ... | 0.488656 |
| ...      | ...      | ...      | ...      | ...      | ... | ...      |
| 1.487683 | 1.645705 | 6.773022 | 0        | 8.224032 | ... | 1.400359 |

### 3. Conv2\_1a - $3 \times 3$ , 64 filter, stride = 1

Lapisan ini adalah bagian pertama dari blok residual pertama. Menggunakan 64 filter dengan ukuran  $3 \times 3$ , konvolusi dilakukan tanpa perubahan dimensi spasial (*stride* = 1). Fitur yang dihasilkan mulai menggambarkan pola-pola kecil seperti garis-garis tipis, sudut, atau batas tekstur dari objek dalam gambar. Aktivasi ReLU

digunakan untuk menambahkan non-linearitas terhadap keluaran. Gambar 4.4 menunjukkan visualisasi hasil dari proses ini. Serta tabel 4.6 merupakan nilai *feature map* nya. Nilai dari tabel ini akan digunakan untuk proses selanjutnya.



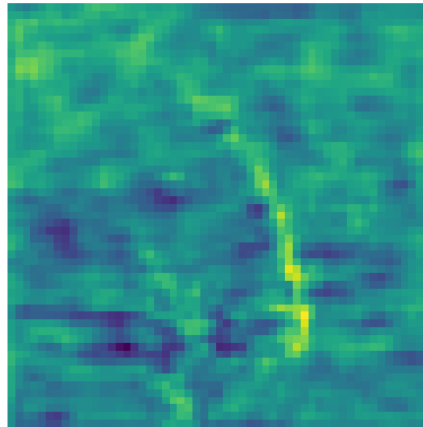
Gambar 4.4 Konvolusi Pertama Pada Blok Residual Pertama

Tabel 4.6 Fitur Hasil Proses Conv2\_1a -  $3 \times 3$ , 64 filter, stride = 1

| ch1 | ch2 | ch3      | ch4 | ch5 | ... | ch64     |
|-----|-----|----------|-----|-----|-----|----------|
| 0   | 0   | 4.742303 | 0   | 0   | ... | 1.558197 |
| 0   | 0   | 0        | 0   | 0   | ... | 2.471303 |
| 0   | 0   | 0        | 0   | 0   | ... | 3.133738 |
| 0   | 0   | 0        | 0   | 0   | ... | 0        |
| 0   | 0   | 0.356962 | 0   | 0   | ... | 0        |
| ... | ... | ...      | ... | ... | ... | ...      |
| 0   | 0   | 0.492859 | 0   | 0   | ... | 0.456199 |

#### 4. Conv2\_1b - $3 \times 3$ , 64 filter, stride = 1

Ini adalah lapisan kedua dari *residual block* pertama. *Output* dari lapisan ini dijumlahkan dengan *input* awal blok sebagai *shortcut connection* (tanpa transformasi). *Residual learning* ini membantu mencegah degradasi akurasi saat kedalaman jaringan meningkat, serta memastikan bahwa informasi penting dari *input* awal tidak hilang selama proses konvolusi berlapis. Gambar 4.5 menunjukkan visualisasi hasil dari proses ini. Dan tabel 4.7 merupakan nilai *feature map* nya. Nilai dari tabel ini akan digunakan untuk proses selanjutnya.



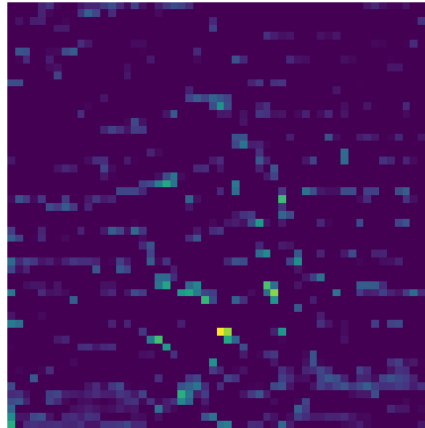
Gambar 4.5 Konvolusi Kedua Pada Blok Residual Pertama

Tabel 4.7 Fitur Hasil Proses Conv2\_1b -  $3 \times 3$ , 64 filter, stride = 1

| ch1      | ch2      | ch3      | ch4      | ch5      | ... | ch64     |
|----------|----------|----------|----------|----------|-----|----------|
| -2.50353 | -4.10194 | 4.128051 | 6.339822 | -0.14596 | ... | -1.93756 |
| -3.32307 | -5.07643 | 4.169655 | -0.29256 | -1.33267 | ... | -2.62577 |
| -3.80991 | -2.96414 | 3.072607 | -6.7209  | -1.00182 | ... | -3.00512 |
| -3.03878 | -2.78169 | 2.780026 | -8.95783 | -0.18315 | ... | -1.80862 |
| -2.1544  | -2.37722 | 2.342352 | -3.02827 | 0.119461 | ... | -1.37798 |
| ...      | ...      | ...      | ...      | ...      | ... | ...      |
| -1.65736 | -1.6687  | 3.772396 | 1.198152 | -1.27541 | ... | -0.20739 |

##### 5. Conv2\_2a - $3 \times 3$ , 64 filter, stride = 1

Blok residual kedua masih menggunakan filter  $3 \times 3$  dan 64 saluran. Operasi konvolusi ini memperdalam pemahaman model terhadap fitur lokal yang lebih rumit, seperti pola kerusakan permukaan atau detail tekstur pada objek. Ukuran spasial tetap  $56 \times 56$ , mempertahankan konteks lokal dengan cakupan area tetap. Gambar 4.6 menunjukkan visualisasi hasil dari proses ini. Serta tabel 4.8 merupakan nilai *feature map* nya. Nilai dari tabel ini akan digunakan untuk proses selanjutnya.



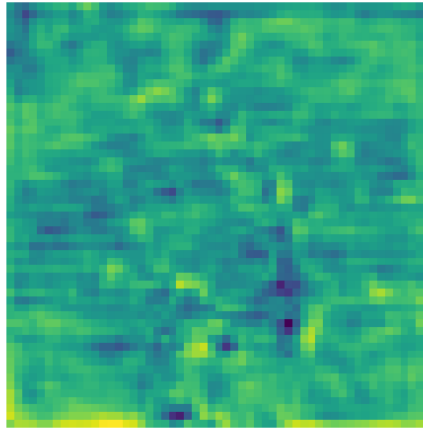
Gambar 4.6 Konvolusi Pertama Pada Blok Residual Kedua

Tabel 4.8 Fitur Hasil Proses Conv2\_2a -  $3 \times 3$ , 64 filter, stride = 1

| ch1      | ch2      | ch3 | ch4      | ch5      | ... | ch64     |
|----------|----------|-----|----------|----------|-----|----------|
| 0        | 2.026416 | 0   | 0.588327 | 1.705362 | ... | 0        |
| 0        | 0        | 0   | 4.082889 | 0.663873 | ... | 0        |
| 0        | 0        | 0   | 3.973372 | 2.205078 | ... | 0        |
| 0        | 0        | 0   | 2.305793 | 0.921132 | ... | 0        |
| 0.881315 | 0        | 0   | 0.427956 | 0        | ... | 0        |
| ...      | ...      | ... | ...      | ...      | ... | ...      |
| 0.666901 | 0        | 0   | 1.594062 | 0        | ... | 0.171261 |

#### 6. Conv2\_2b - $3 \times 3$ , 64 filter, stride = 1

Lapisan kedua dari blok ini mempertahankan ukuran fitur dan menyelesaikan proses *residual* block dengan *shortcut connection*. Penambahan *shortcut* memungkinkan model belajar fungsi identitas bila diperlukan, sehingga mencegah hilangnya fitur penting. Gambar 4.7 menunjukkan visualisasi hasil dari proses ini. Serta tabel 4.9 merupakan nilai *feature map* nya. Nilai dari tabel ini akan digunakan untuk proses selanjutnya.



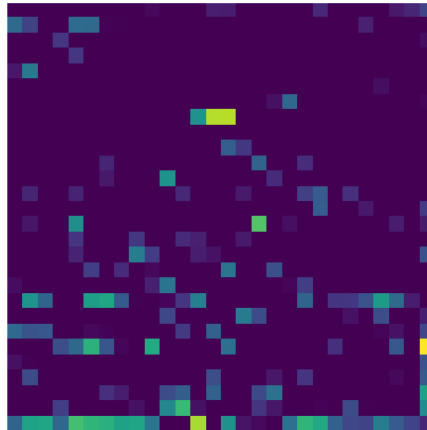
Gambar 4.7 Konvolusi Kedua Pada Blok Residual Kedua

Tabel 4.9 Fitur Hasil Proses Conv2 2b -  $3 \times 3$ , 64 filter, stride = 1

| ch1      | ch2      | ch3      | ch4      | ch5      | ... | ch64     |
|----------|----------|----------|----------|----------|-----|----------|
| -1.32566 | -11.5892 | 8.265839 | 0.476646 | 3.771027 | ... | 0.659162 |
| -2.28857 | -9.67023 | 7.00917  | 1.225109 | 2.963255 | ... | -0.06902 |
| -2.65396 | -3.88158 | 3.966646 | 1.591029 | 1.035461 | ... | -1.08974 |
| -1.69126 | -0.91238 | 2.44692  | 0.07171  | 1.049251 | ... | -0.65365 |
| -1.07294 | 1.883359 | 1.255    | 0.723886 | 0.83021  | ... | 0.362136 |
| ...      | ...      | ...      | ...      | ...      | ... | ...      |
| 0.945179 | -0.9905  | -0.6273  | 2.84399  | -1.21471 | ... | 0.39795  |

#### 7. Conv3\_1a - $3 \times 3$ , 128 filter, stride = 2

Ini adalah awal dari blok residual ketiga. Dilakukan *downsampling* dengan stride = 2, yang menurunkan ukuran spasial menjadi  $28 \times 28$  dan meningkatkan jumlah saluran menjadi 128. *Shortcut connection* juga mengalami transformasi agar ukuran *output* sesuai. Filter yang lebih banyak memungkinkan penangkapan fitur visual yang lebih kompleks. Visualisasi Gambar 4.8 menunjukkan hasil dari proses ini dan tabel 4.10 dibawah merupakan nilai *feature map* yang akan digunakan untuk proses selanjutnya.



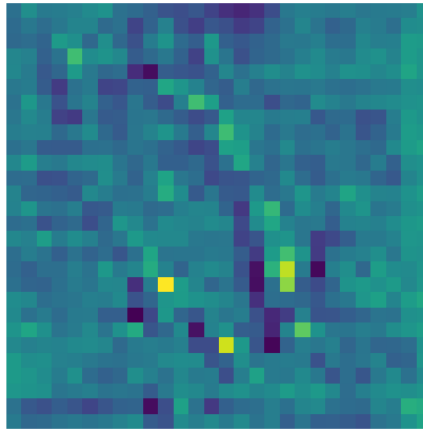
Gambar 4.8 Konvolusi Pertama Pada Blok Residual Ketiga

Tabel 4.10 Fitur Hasil Proses Conv3\_1a -  $3 \times 3$ , 128 filter, stride = 2

| ch1      | ch2      | ch3      | ch4      | ch5      | ... | Ch128    |
|----------|----------|----------|----------|----------|-----|----------|
| 0        | 0        | 5.919079 | 0        | 0        | ... | 0        |
| 0        | 0        | 3.563867 | 0        | 1.948008 | ... | 0        |
| 0        | 0        | 0.602099 | 0        | 0        | ... | 0        |
| 0        | 0.349955 | 2.373247 | 0        | 0        | ... | 0        |
| 0        | 0        | 2.983922 | 0        | 1.561401 | ... | 1.787314 |
| ...      | ...      | ...      | ...      | ...      | ... | ...      |
| 0.389485 | 0        | 0        | 1.292535 | 2.438053 | ... | 0        |

#### 8. Conv3\_1b - $3 \times 3$ , 128 filter, stride = 1

Lapisan konvolusi kedua dalam blok ini memperkuat fitur spasial yang sudah disaring sebelumnya. Dengan ukuran fitur tetap  $28 \times 28 \times 128$ , jaringan mulai mengenali objek atau pola struktural seperti garis batas dinding atau area kerusakan. Visualisasi Gambar 4.9 menunjukkan hasil dari proses ini dan tabel 4.11 dibawah merupakan nilai *feature map* yang akan digunakan untuk proses selanjutnya.



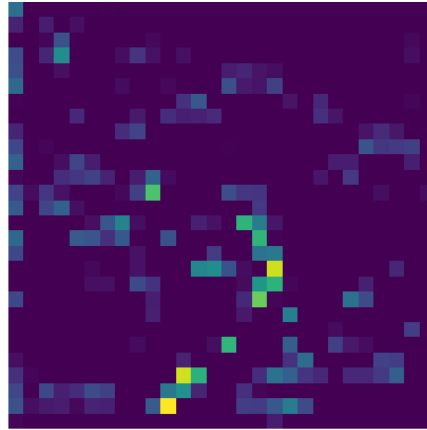
Gambar 4.9 Konvolusi Kedua Pada Blok Residual Ketiga

Tabel 4.11 Fitur Hasil Proses Conv3\_1b -  $3 \times 3$ , 128 filter, stride = 1

| ch1      | ch2      | ch3      | ch4      | ch5      | ... | Ch128    |
|----------|----------|----------|----------|----------|-----|----------|
| -1.87532 | -3.14436 | 0.033302 | -1.4274  | 0.754987 | ... | 0.554744 |
| -0.4411  | -4.22694 | -0.00598 | -3.08018 | 2.308056 | ... | -0.51352 |
| -1.41086 | -2.63179 | -0.93016 | -2.13812 | 1.025874 | ... | -0.04408 |
| -1.28728 | -2.67685 | -0.17768 | -2.62183 | 0.968891 | ... | 0.454751 |
| -1.13393 | -4.30131 | -1.27968 | -2.4015  | 1.534096 | ... | 0.962305 |
| ...      | ...      | ...      | ...      | ...      | ... | ...      |
| 0.562485 | -2.99095 | -0.20253 | -0.21764 | -3.1068  | ... | -0.8695  |

#### 9. Conv3\_2a - $3 \times 3$ , 128 filter, stride = 1

Lapisan ini melanjutkan konvolusi pada resolusi  $28 \times 28$ . Tujuannya adalah mempertajam fitur spasial yang mulai terbentuk di blok sebelumnya, menambahkan konteks tambahan terhadap fitur tekstur atau pola-pola besar dalam gambar. Visualisasi Gambar 4.10 menunjukkan hasil dari proses ini dan tabel 4.12 dibawah merupakan nilai *feature map* yang akan digunakan untuk proses selanjutnya



Gambar 4.10 Konvolusi Pertama Pada Blok Residual Keempat

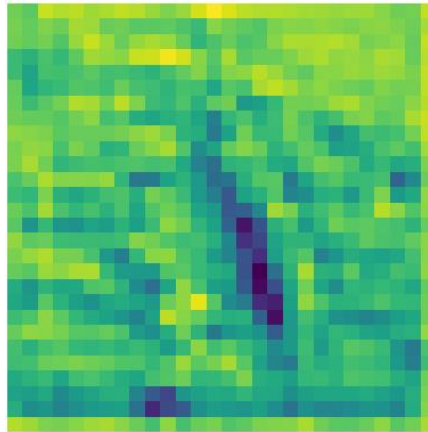
Tabel 4.12 Fitur Hasil Proses Conv3\_2a -  $3 \times 3$ , 128 filter, stride = 1

| ch1      | ch2      | ch3 | ch4      | ch5 | ... | Ch128    |
|----------|----------|-----|----------|-----|-----|----------|
| 2.602841 | 0        | 0   | 0        | 0   | ... | 0        |
| 0        | 0        | 0   | 0        | 0   | ... | 1.511608 |
| 0        | 0        | 0   | 1.061116 | 0   | ... | 0        |
| 0        | 0        | 0   | 0        | 0   | ... | 0        |
| 0        | 0        | 0   | 0        | 0   | ... | 0.344688 |
| ...      | ...      | ... | ...      | ... | ... | ...      |
| 0        | 0.307987 | 0   | 0        | 0   | ... | 1.76331  |

#### 10. Conv3\_2b - $3 \times 3$ , 128 filter, stride = 1

Konvolusi terakhir pada blok ini menyelesaikan tahap dengan *shortcut connection* ke *input* awal blok. Fitur spasial yang diperoleh sudah jauh lebih deskriptif, mencerminkan elemen-elemen besar dari gambar. Visualisasi Gambar 4.11 menunjukkan hasil dari proses ini dan tabel 4.13 dibawah merupakan nilai *feature map* yang akan digunakan untuk proses selanjutnya





Gambar 4.11 Konvolusi Kedua Pada Blok Residual Keempat

Tabel 4.13 Fitur Hasil Proses Conv3 2b -  $3 \times 3$ , 128 filter, stride = 1

| ch1      | ch2      | ch3      | ch4      | ch5      | ... | Ch128    |
|----------|----------|----------|----------|----------|-----|----------|
| -0.10688 | 0.247051 | 0.154733 | -1.20781 | -0.98174 | ... | 0.859202 |
| -0.42067 | -0.60796 | 1.280489 | -0.25816 | -1.8305  | ... | 1.219229 |
| 0.923424 | 0.188132 | 0.729997 | -0.55137 | -2.33535 | ... | -0.10056 |
| 0.674798 | 0.084007 | 1.694582 | -0.42424 | -1.90251 | ... | -0.91687 |
| 0.808666 | -0.90945 | 1.630566 | -0.16451 | -2.1952  | ... | -0.40392 |
| ...      | ...      | ...      | ...      | ...      | ... | ...      |
| 0.586592 | -0.40409 | -1.14365 | -0.28642 | -1.51608 | ... | -0.99428 |

#### 11. Conv4\_1a - $3 \times 3$ , 256 filter, stride = 2

*Downsampling* ketiga dilakukan pada tahap ini, mengecilkan ukuran spasial menjadi  $14 \times 14$  dan memperluas kedalaman fitur menjadi 256 *channel*. Konvolusi ini memungkinkan pengenalan fitur kompleks seperti bentuk keruntuhan besar, tiang patah, atau dinding retak yang tersegmentasi jelas. Visualisasi Gambar 4.12 menunjukkan hasil dari proses ini dan tabel 4.1 dibawah merupakan nilai *feature map* yang akan digunakan untuk proses selanjutnya



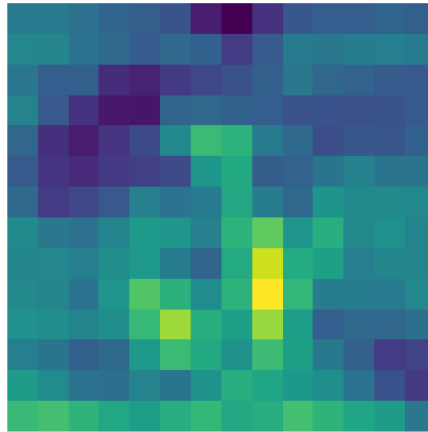
Gambar 4.12 Konvolusi Pertama Pada Blok Residual Kelima

Tabel 4.14 Fitur Hasil Proses Conv4\_1a -  $3 \times 3$ , 256 filter, stride = 2

| ch1      | ch2 | ch3 | ch4      | ch5      | ... | Ch256 |
|----------|-----|-----|----------|----------|-----|-------|
| 0        | 0   | 0   | 1.91957  | 0.540801 | ... | 0     |
| 0        | 0   | 0   | 2.306746 | 0        | ... | 0     |
| 0        | 0   | 0   | 2.278572 | 0.897366 | ... | 0     |
| 0        | 0   | 0   | 3.046667 | 0.047246 | ... | 0     |
| 0        | 0   | 0   | 2.054678 | 0.204793 | ... | 0     |
| ...      | ... | ... | ...      | ...      | ... | ...   |
| 0.194622 | 0   | 0   | 0.128092 | 1.53851  | ... | 0     |

#### 12. Conv4\_1b - $3 \times 3$ , 256 filter, stride = 1

Konvolusi ini mempertahankan ukuran dan memperkuat fitur struktural dari lapisan sebelumnya. Dalam konteks citra kerusakan bangunan, model mulai memahami bagian-bagian kerusakan utama berdasarkan komposisi visualnya. Visualisasi Gambar 4.13 menunjukkan hasil dari proses ini dan tabel 4.15 dibawah merupakan nilai *feature map* yang akan digunakan untuk proses selanjutnya



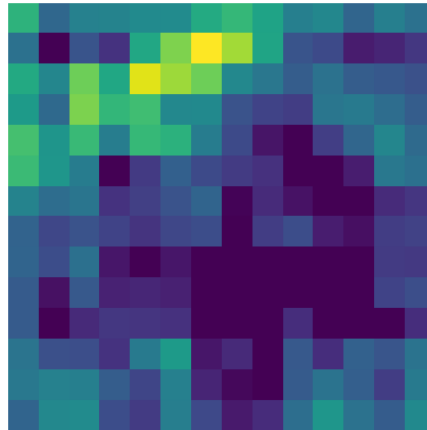
Gambar 4.13 Konvolusi Kedua Pada Blok Residual Kelima

Tabel 4.15 Fitur Hasil Proses Conv4\_1b -  $3 \times 3$ , 256 filter, stride = 1

| ch1      | ch2      | ch3      | ch4      | ch5      | ... | Ch256    |
|----------|----------|----------|----------|----------|-----|----------|
| -0.0747  | -3.12932 | 2.18037  | 2.68431  | 0.323808 | ... | -1.58979 |
| -0.15735 | -4.09445 | 1.829879 | 1.877675 | -2.87612 | ... | -2.65522 |
| -0.57135 | -6.25932 | 4.989129 | 0.355488 | -0.8832  | ... | -2.24307 |
| -1.1484  | -8.61068 | 4.345156 | -0.84444 | -2.40526 | ... | -2.4607  |
| -1.45191 | -8.23281 | 3.097188 | -0.4136  | -3.99676 | ... | -2.27165 |
| ...      | ...      | ...      | ...      | ...      | ... | ...      |
| -0.29557 | -3.02699 | -0.23084 | -0.19868 | -3.34687 | ... | 0.920863 |

### 13. Conv4\_2a - $3 \times 3$ , 256 filter, stride = 1

Lapisan ini berfungsi untuk mendalami konteks spasial secara lebih detail. Dengan tetap menjaga dimensi, konvolusi ini menstabilkan representasi spasial dari kerusakan kompleks yang ada dalam gambar. Visualisasi Gambar 4.14 menunjukkan hasil dari proses ini dan tabel 4.16 dibawah merupakan nilai *feature map* yang akan digunakan untuk proses selanjutnya



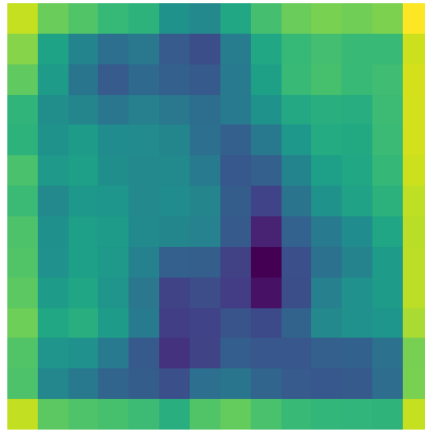
Gambar 4.14 Konvolusi Pertama Pada Blok Residual Keenam

Tabel 4.16 Fitur Hasil Proses Conv4\_2a -  $3 \times 3$ , 256 filter, stride = 1

| ch1      | ch2      | ch3      | ch4 | ch5 | ... | Ch256 |
|----------|----------|----------|-----|-----|-----|-------|
| 5.096756 | 0.180114 | 2.833754 | 0   | 0   | ... | 0     |
| 2.620627 | 0        | 3.863058 | 0   | 0   | ... | 0     |
| 3.467071 | 4.329097 | 4.07559  | 0   | 0   | ... | 0     |
| 3.513173 | 4.905761 | 6.166411 | 0   | 0   | ... | 0     |
| 3.780504 | 3.693038 | 6.068897 | 0   | 0   | ... | 0     |
| ...      | ...      | ...      | ... | ... | ... | ...   |
| 3.749651 | 0        | 2.806734 | 0   | 0   | ... | 0     |

#### 14. Conv4\_2b - $3 \times 3$ , 256 filter, stride = 1

Konvolusi terakhir di blok ini. Shortcut connection menjaga agar gradien tetap mengalir dari lapisan atas ke bawah tanpa kehilangan informasi penting, dan *output* dari blok ini mencerminkan fitur spasial mendalam. Visualisasi Gambar 4.15 menunjukkan hasil dari proses ini dan tabel 4.17 dibawah merupakan nilai *feature map* yang akan digunakan untuk proses selanjutnya



Gambar 4.15 Konvolusi Kedua Pada Blok Residual Keenam

Tabel 4.17 Fitur Hasil Proses Conv4\_2b -  $3 \times 3$ , 256 filter, stride = 1

| ch1      | ch2      | ch3      | ch4      | ch5      | ... | Ch256    |
|----------|----------|----------|----------|----------|-----|----------|
| -4.35366 | -4.46329 | -2.35706 | -6.29389 | -3.7883  | ... | 0.6109   |
| -6.8889  | -6.15622 | -3.94443 | -9.35086 | -4.84327 | ... | 1.086893 |
| -7.62784 | -7.94996 | -3.47644 | -9.75502 | -5.59489 | ... | 0.547957 |
| -8.65862 | -7.91909 | -4.25364 | -10.6093 | -5.29693 | ... | 0.57353  |
| -9.09636 | -7.22366 | -4.92611 | -11.3544 | -5.35187 | ... | 0.956968 |
| ...      | ...      | ...      | ...      | ...      | ... | ...      |
| -4.36806 | -4.21379 | -1.95362 | -5.83791 | -1.84679 | ... | 1.096944 |

#### 15. Conv5\_1a - $3 \times 3$ , 512 filter, stride = 2

Downsampling terakhir dilakukan, mengubah ukuran spasial menjadi  $7 \times 7$  dan menaikkan kedalaman fitur menjadi 512. Fitur di tahap ini sangat abstrak dan mengandung informasi semantik tingkat tinggi, seperti klasifikasi tipe-tipe kerusakan atau bentuk struktur dominan dari bangunan. Visualisasi Gambar 4.6 menunjukkan hasil dari proses ini dan tabel 4.18 dibawah merupakan nilai *feature map* yang akan digunakan untuk proses selanjutnya



Gambar 4.16 Konvolusi Pertama Pada Blok Residual Ketujuh

Tabel 4.18 Fitur Hasil Proses Conv5\_1a -  $3 \times 3$ , 512 filter, stride = 2

| ch1      | ch2      | ch3      | ch4 | ch5      | ... | Ch512 |
|----------|----------|----------|-----|----------|-----|-------|
| 0        | 1.766601 | 0.086775 | 0   | 0.161869 | ... | 0     |
| 1.408085 | 2.153302 | 0        | 0   | 0        | ... | 0     |
| 1.148634 | 2.646274 | 0        | 0   | 0        | ... | 0     |
| 0        | 3.830735 | 0        | 0   | 0        | ... | 0     |
| 0.22618  | 2.909461 | 0        | 0   | 0        | ... | 0     |
| ...      | ...      | ...      | ... | ...      | ... | ...   |
| 0        | 2.914895 | 0.817139 | 0   | 5.476203 | ... | 0     |

#### 16. Conv5\_1b - $3 \times 3$ , 512 filter, stride = 1

Dengan mempertahankan ukuran  $7 \times 7 \times 512$ , konvolusi ini memperkuat abstraksi yang telah dibentuk sebelumnya. *Shortcut connection* tetap aktif untuk menjaga kesinambungan fitur dari awal. Visualisasi Gambar 4.17 menunjukkan hasil dari proses ini dan tabel 4.19 dibawah merupakan nilai *feature map* yang akan digunakan untuk proses selanjutnya



Gambar 4.17 Konvolusi Kedua Pada Blok Residual Ketujuh

Tabel 4.19 Fitur Hasil Proses Conv5\_1b -  $3 \times 3$ , 512 filter, stride = 1

| ch1      | ch2      | ch3      | ch4      | ch5      | ... | Ch512    |
|----------|----------|----------|----------|----------|-----|----------|
| 0.329746 | -0.05382 | -1.62791 | 0.253719 | -3.51153 | ... | -1.78018 |
| -0.42112 | -1.41244 | -1.8116  | 0.136003 | -5.56796 | ... | -3.06932 |
| -0.29014 | -1.43358 | -1.87302 | 0.033603 | -5.937   | ... | -3.37953 |
| -0.4169  | -1.11036 | -1.35813 | -0.1631  | -5.39884 | ... | -3.39021 |
| -1.2819  | -0.63254 | -1.04946 | -0.94312 | -4.34548 | ... | -3.46124 |
| ...      | ...      | ...      | ...      | ...      | ... | ...      |
| -0.42112 | -0.52979 | 2.450505 | -0.02667 | -3.06831 | ... | -4.34473 |

#### 17. Conv5\_2a - $3 \times 3$ , 512 filter, stride = 1

Konvolusi lanjutan dalam blok akhir, menangkap korelasi spasial kompleks dan relasi antar fitur yang lebih tinggi. Layer ini sangat penting dalam membangun pemahaman mendalam terhadap konteks gambar secara menyeluruh. Visualisasi Gambar 4.18 menunjukkan hasil dari proses ini dan tabel 4.20 dibawah merupakan nilai *feature map* yang akan digunakan untuk proses selanjutnya



Gambar 4.18 Konvolusi Pertama Pada Blok Residual Terakhir

Tabel 4.20 Fitur Hasil Proses Conv5\_2a -  $3 \times 3$ , 512 filter, stride = 1

| ch1      | ch2 | ch3 | ch4 | ch5 | ... | Ch512 |
|----------|-----|-----|-----|-----|-----|-------|
| 0        | 0   | 0   | 0   | 0   | ... | 0     |
| 0        | 0   | 0   | 0   | 0   | ... | 0     |
| 0        | 0   | 0   | 0   | 0   | ... | 0     |
| 0        | 0   | 0   | 0   | 0   | ... | 0     |
| 0        | 0   | 0   | 0   | 0   | ... | 0     |
| ...      | ... | ... | ... | ... | ... | ...   |
| 0.166894 | 0   | 0   | 0   | 0   | ... | 0     |

#### 18. Conv5\_2b - $3 \times 3$ , 512 filter, stride = 1

Lapisan terakhir dari seluruh jaringan konvolusi ResNet-18. *Output* di tahap ini adalah representasi akhir gambar dalam bentuk *feature map*  $7 \times 7 \times 512$ , yang siap diringkas dalam tahap berikutnya. Gambar 4.19 menjadi representasi final visual dari ekstraksi fitur.



Gambar 4.19 Konvolusi Kedua Pada Blok Residual Terakhir



Tabel 4.21 Fitur Hasil Proses Conv5 2b - 3×3, 512 filter, stride = 1

| ch1      | ch2      | ch3      | ch4      | ch5      | ... | Ch512    |
|----------|----------|----------|----------|----------|-----|----------|
| -0.33577 | -0.60771 | 0.094459 | -0.07463 | -0.40283 | ... | -0.02382 |
| -0.51584 | -0.94558 | 0.088845 | -0.09913 | -0.54683 | ... | -0.00469 |
| -0.45245 | -1.16143 | 0.042126 | -0.12973 | -0.5012  | ... | 0.020266 |
| -0.34151 | -1.05426 | -0.10703 | -0.08791 | -0.50156 | ... | -0.00375 |
| -0.13841 | -0.97512 | -0.21172 | -0.05517 | -0.56864 | ... | -0.2717  |
| ...      | ...      | ...      | ...      | ...      | ... | ...      |
| -0.04885 | -1.05333 | -0.42078 | -0.45016 | -0.40868 | ... | -0.88145 |

Dengan melalui setiap proses di atas, ResNet-18 berhasil membentuk representasi fitur yang sangat kaya dari gambar *input*. Tiap layer bertugas menyusun hierarki fitur dari rendah hingga tinggi, mulai dari pola piksel dasar hingga konsep abstrak, yang sangat berguna dalam analisis kerusakan bangunan pasca bencana. Tabel 4.22 merangkum hasil akhir dari proses ekstraksi fitur pada satu gambar. Terdapat 512 fitur per patch gambar dengan 36 data gambar.

Tabel 4.22 Hasil Ekstraksi Fitur

| fitur 1 | fitur 2 | fitur 3  | fitur 4  | fitur 5 | ... | fitur 512 |
|---------|---------|----------|----------|---------|-----|-----------|
| 0       | 0       | 0.094459 | 0.179085 | 0       | ... | 0         |
| 0       | 0       | 0.088845 | 0.036873 | 0       | ... | 0         |
| 0       | 0       | 0.042126 | 0        | 0       | ... | 0.020266  |
| 0       | 0       | 0        | 0        | 0       | ... | 0         |
| 0       | 0       | 0        | 0        | 0       | ... | 0         |
| ...     | ...     | ...      | ...      | ...     | ... | ...       |
| 0       | 0       | 2.02972  | 0        | 0       | ... | 0         |

### 4.3 Skenario Pengujian

Skenario pengujian pada penelitian ini dirancang untuk mengevaluasi efektivitas metode pengelompokan data menggunakan algoritma *K-Means*, yang diaplikasikan terhadap dua pendekatan fitur berbeda. Pendekatan pertama menggunakan fitur hasil ekstraksi dari jaringan ResNet-18 yang direduksi dimensinya dengan metode PCA. Pendekatan kedua menggunakan fitur ResNet-18 tanpa proses reduksi dimensi sebelumnya.

Untuk pendekatan pertama, dilakukan analisis PCA terhadap vektor fitur hasil ekstraksi ResNet-18, yang terdiri atas 512 fitur dari setiap patch citra. Reduksi dimensi dilakukan dengan mengambil dua komponen utama berdasarkan nilai eigen tertinggi. Proyeksi dua dimensi menggunakan PC1 dan PC2 digunakan untuk proses *K-Means Clustering* dan visualisasi hasil *Clustering*.

Pada pendekatan kedua, fitur hasil ekstraksi ResNet-18 digunakan secara langsung tanpa melalui proses PCA. Vektor berdimensi 512 tersebut diklasifikasikan menggunakan *K-Means* untuk menguji bagaimana hasil *Clustering* berubah ketika data tidak direduksi terlebih dahulu.

Visualisasi hasil *Clustering* pada masing-masing skenario dibandingkan untuk menentukan skenario dengan pemisahan *cluster* paling optimal. Visualisasi terbaik dipilih sebagai dasar pembahasan pada bagian berikutnya.

Tabel 4.23 Skenario Uji

| No. | Skenario   | Komponen yang Digunakan               |
|-----|------------|---------------------------------------|
| 1   | Skenario 1 | PC1 dan PC2                           |
| 2   | Skenario 2 | Nilai Fitur dengan variansi tertinggi |

Setiap skenario menghasilkan *scatter plot* visualisasi hasil *Clustering*, yang kemudian dibandingkan secara visual untuk menentukan distribusi *cluster* terbaik. Skenario dengan pemisahan *cluster* yang paling jelas dan proporsional akan dipilih untuk dibahas lebih lanjut pada bagian pembahasan.

#### 4.4 Hasil Uji Coba

Setelah dilakukan proses ekstraksi fitur dengan ResNet-18 dan reduksi dimensi menggunakan PCA, tahap selanjutnya adalah melakukan pengujian terhadap hasil representasi data menggunakan algoritma *K-Means*. Lalu pengujian

selanjutnya yaitu melakukan *Clustering* terhadap fitur data gambar tanpa direduksi terlebih dahulu menggunakan PCA. Pengujian ini bertujuan untuk mengetahui kombinasi komponen utama yang mampu memberikan visualisasi dan *Clustering* terbaik terhadap fitur yang dihasilkan.

#### 4.4.1 Enam Komponen Utama Hasil PCA

Dari proses PCA yang dilakukan terhadap data hasil ekstraksi fitur, diperoleh beberapa komponen dengan nilai dari PC1 dan PC2 sebagai dua komponen utama dengan nilai *eigenvalue* tertinggi. Komponen ini menjadi dasar representasi fitur dalam dimensi yang lebih rendah untuk memudahkan visualisasi dan pengelompokan.

Tabel 4.24 Hasil Analisis PCA

| No. | Gambar    | Komponen Utama | <i>Eigenvalue</i> | <i>Explained variance (%)</i> |
|-----|-----------|----------------|-------------------|-------------------------------|
| 1.  | Gambar 1  | PC1            | 40.81             | 15.19%                        |
|     |           | PC2            | 33.68             | 12.54%                        |
| 2.  | Gambar 2  | PC1            | 41.18             | 14.05%                        |
|     |           | PC2            | 38.42             | 13.11%                        |
| 3.  | Gambar 3  | PC1            | 45.96             | 14.80%                        |
|     |           | PC2            | 35.52             | 11.44%                        |
| 4.  | Gambar 4  | PC1            | 38.98             | 14.03%                        |
|     |           | PC2            | 36.61             | 13.18%                        |
| 5.  | Gambar 5  | PC1            | 51.90             | 17.00%                        |
|     |           | PC2            | 34.00             | 11.14%                        |
| 6.  | Gambar 6  | PC1            | 42.01             | 15.07%                        |
|     |           | PC2            | 29.17             | 10.46%                        |
| 7.  | Gambar 7  | PC1            | 40.05             | 12.65%                        |
|     |           | PC2            | 32.29             | 10.20%                        |
| 8.  | Gambar 8  | PC1            | 37.91             | 13.21%                        |
|     |           | PC2            | 34.01             | 11.85%                        |
| 9.  | Gambar 9  | PC1            | 40.60             | 12.20%                        |
|     |           | PC2            | 32.60             | 9.79%                         |
| 10. | Gambar 10 | PC1            | 44.14             | 15.66%                        |
|     |           | PC2            | 37.54             | 13.32%                        |
| 11. | Gambar 11 | PC1            | 37.01             | 12.99%                        |
|     |           | PC2            | 32.31             | 11.34%                        |
| 12. | Gambar 12 | PC1            | 35.35             | 13.79%                        |
|     |           | PC2            | 30.81             | 12.02%                        |
| 13. | Gambar 13 | PC1            | 44.68             | 18.00%                        |
|     |           | PC2            | 34.30             | 13.82%                        |
| 14. | Gambar 14 | PC1            | 43.15             | 14.62%                        |

|     |           |     |       |        |
|-----|-----------|-----|-------|--------|
|     |           | PC2 | 36.68 | 12.43% |
| 15. | Gambar 15 | PC1 | 41.31 | 13.26% |
|     |           | PC2 | 30.71 | 9.86%  |
| 16. | Gambar 16 | PC1 | 42.01 | 13.49% |
|     |           | PC2 | 35.04 | 11.25% |
| 17. | Gambar 17 | PC1 | 39.77 | 15.27% |
|     |           | PC2 | 29.44 | 11.31% |
| 18. | Gambar 18 | PC1 | 42.16 | 14.96% |
|     |           | PC2 | 29.45 | 10.45% |
| 19. | Gambar 19 | PC1 | 36.83 | 13.12% |
|     |           | PC2 | 32.72 | 11.65% |
| 20. | Gambar 20 | PC1 | 37.58 | 13.43% |
|     |           | PC2 | 33.71 | 12.05% |
| 21. | Gambar 21 | PC1 | 40.91 | 14.95% |
|     |           | PC2 | 31.16 | 11.39% |
| 22. | Gambar 22 | PC1 | 55.75 | 19.93% |
|     |           | PC2 | 28.91 | 10.33% |
| 23. | Gambar 23 | PC1 | 40.76 | 15.12% |
|     |           | PC2 | 32.14 | 11.92% |
| 24. | Gambar 24 | PC1 | 39.22 | 14.07% |
|     |           | PC2 | 34.21 | 12.27% |
| 25. | Gambar 25 | PC1 | 40.46 | 13.12% |
|     |           | PC2 | 32.49 | 10.54% |
| 26. | Gambar 26 | PC1 | 40.46 | 13.12% |
|     |           | PC2 | 32.49 | 10.54% |
| 27. | Gambar 27 | PC1 | 52.50 | 18.36% |
|     |           | PC2 | 31.41 | 10.99% |
| 28. | Gambar 28 | PC1 | 33.21 | 12.32% |
|     |           | PC2 | 29.27 | 10.86% |
| 29. | Gambar 29 | PC1 | 59.35 | 20.25% |
|     |           | PC2 | 33.20 | 11.33% |
| 30. | Gambar 30 | PC1 | 43.69 | 15.50% |
|     |           | PC2 | 32.51 | 11.53% |
| 31. | Gambar 31 | PC1 | 42.71 | 14.13% |
|     |           | PC2 | 32.63 | 10.80% |
| 32. | Gambar 32 | PC1 | 40.43 | 15.05% |
|     |           | PC2 | 33.70 | 12.55% |
| 33. | Gambar 33 | PC1 | 40.32 | 13.34% |
|     |           | PC2 | 32.39 | 10.72% |
| 34. | Gambar 34 | PC1 | 45.96 | 14.80% |
|     |           | PC2 | 35.52 | 11.44% |
| 35. | Gambar 35 | PC1 | 42.80 | 15.94% |
|     |           | PC2 | 31.24 | 11.63% |
| 36. | Gambar 36 | PC1 | 51.54 | 16.71% |
|     |           | PC2 | 27.96 | 9.07%  |
| 37. | Gambar 37 | PC1 | 38.98 | 14.03% |
|     |           | PC2 | 36.61 | 13.18% |

Berdasarkan tabel di atas, terlihat bahwa PC1 memberikan kontribusi terbesar terhadap representasi data, disusul oleh PC2.

#### 4.4.2 Skenario 1: *Clustering* Menggunakan Nilai PCA (PC1 dan PC2)

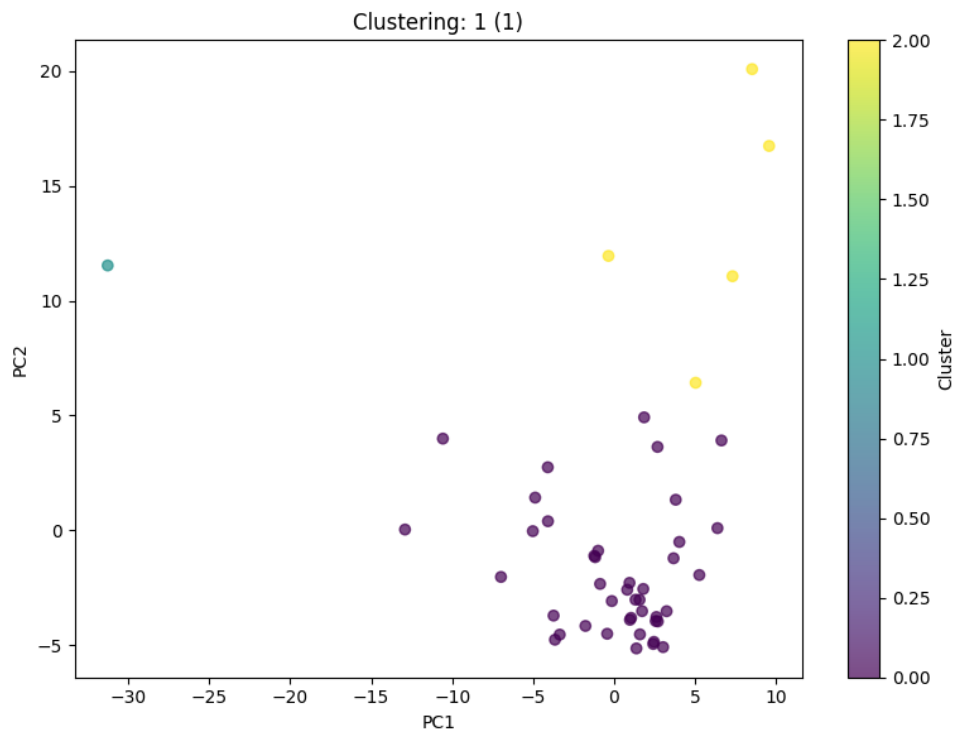
Pada skenario ini, proses *Clustering* dilakukan terhadap data fitur hasil ekstraksi ResNet-18 yang telah direduksi dimensinya menggunakan PCA, dengan memilih dua komponen utama yaitu PC1 dan PC2. Alasan pemilihan dua komponen ini adalah karena keduanya memiliki kontribusi terbesar terhadap variansi data dan dianggap mewakili karakteristik paling penting dari fitur visual yang diekstraksi. Kombinasi PC1 dan PC2 dianggap cukup representatif untuk keperluan visualisasi dua dimensi.

Proyeksi fitur ke dalam ruang dua dimensi menggunakan PC1 dan PC2 kemudian digunakan sebagai *input* untuk algoritma *K-Means Clustering* dengan jumlah *cluster* ditentukan sebanyak tiga ( $k = 3$ ). Visualisasi hasil *clustering* ditampilkan dalam bentuk *scatter plot* untuk mempermudah evaluasi pola penyebaran dan pemisahan antar *i*.

Hasil visualisasi menunjukkan pembentukan tiga *i* yang cukup jelas, dengan penyebaran titik-titik data yang relatif seimbang. Masing-masing *cluster* tampak memiliki batas yang cukup tegas, meskipun terdapat beberapa titik data yang berada di area transisi, menandakan adanya kemiripan karakteristik antar *cluster* di wilayah tersebut. Namun secara umum, distribusi data yang terbentuk menunjukkan struktur yang cukup alami, yang mengindikasikan bahwa kombinasi PC1 dan PC2 berhasil mengungkapkan dimensi laten dari fitur citra dengan baik.

Keseimbangan visual antar *cluster* juga menjadi indikator bahwa pemisahan yang dilakukan tidak bersifat bias terhadap satu kelompok data tertentu. Ini penting dalam pengelompokan citra kerusakan bangunan yang beragam, agar

tiap jenis kerusakan dapat direpresentasikan dengan adil dan proporsional dalam ruang fitur. Gambar 4.20 di bawah merupakan visualisasi dari data gambar nomor 1 yang terlampir pada Lampiran 1 yang digunakan untuk proses *clustering*. Untuk 35 data gambar lainnya terdapat pada bagian lampiran.



Gambar 4.20 Hasil *Clustering K-Means* PC1 dan PC2 Pada Data Gambar 1

#### 4.4.3 Skenario 2: *Clustering* Menggunakan Fitur ResNet-18 Tanpa

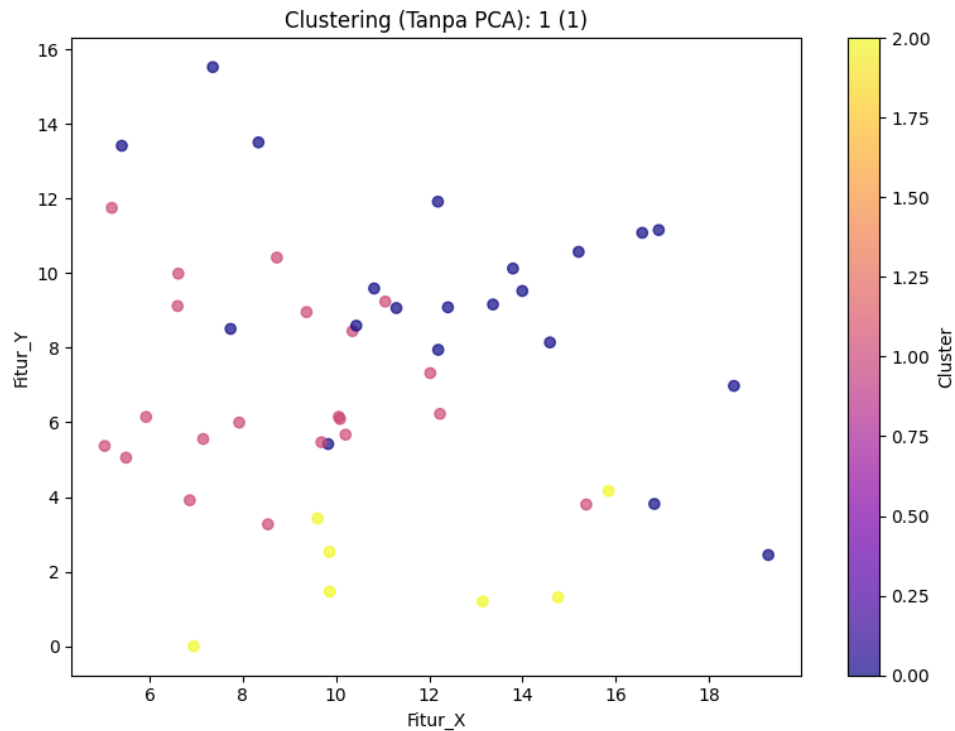
##### PCA

Pada skenario ini, proses *Clustering* dilakukan secara langsung terhadap fitur citra hasil ekstraksi dari ResNet-18 tanpa melalui proses reduksi dimensi menggunakan PCA. Setiap fitur memiliki dimensi 512, yang mencerminkan representasi sangat kompleks dari masing-masing patch citra. Dengan kata lain, vektor fitur ini mengandung banyak informasi, namun tidak seluruhnya relevan atau penting dalam proses pengelompokan data.

*K-Means Clustering* diaplikasikan langsung terhadap fitur-fitur ini dengan tetap menggunakan parameter yang sama, yaitu tiga *cluster* ( $k = 3$ ). Namun karena data berdimensi tinggi tidak dapat divisualisasikan secara langsung dalam dua dimensi, diperlukan teknik visualisasi seperti t-SNE atau PCA sekunder hanya untuk *plotting*, bukan untuk pengolahan *Clustering* itu sendiri. Dalam hal ini, visualisasi digunakan sebagai alat bantu untuk memahami struktur hasil *Clustering*, bukan sebagai dasar pemisahan fitur.

Hasil dari skenario ini menunjukkan bahwa tanpa reduksi dimensi, proses *Clustering* menjadi kurang efektif secara visual. *Scatter plot* hasil visualisasi memperlihatkan tumpang tindih yang cukup signifikan antar titik data, dengan batas *cluster* yang tidak jelas dan susah untuk diinterpretasikan. Sebaran data cenderung acak, dan tidak terbentuk struktur *cluster* yang konsisten sebagaimana yang terlihat pada skenario dengan PCA.

Kondisi ini menunjukkan bahwa dimensi tinggi tidak selalu memberikan keunggulan dalam proses cluster. Sebaliknya, redundansi informasi dalam dimensi tinggi justru dapat menyamarkan pola yang ada, sehingga proses pengelompokan menjadi tidak optimal. Oleh karena itu, penting untuk melakukan proses reduksi dimensi terlebih dahulu agar fitur yang digunakan dalam *Clustering* benar-benar mencerminkan struktur alami dari data visual. Gambar 4.21 di bawah merupakan visualisasi dari data gambar nomor 1 yang terlampir pada Lampiran 1 yang digunakan untuk proses *clustering*. Untuk 35 data gambar lainnya terdapat pada bagian lampiran.



Gambar 4.21 Hasil *Clustering K-Means* Tanpa Menggunakan PCA dari data gambar 1

#### 4.5 Pembahasan

Pembahasan pada bagian ini difokuskan pada analisis dan perbandingan terhadap dua pendekatan utama yang telah dilakukan dalam uji coba sebelumnya, yakni:

1. *Clustering* menggunakan algoritma *K-Means* terhadap fitur hasil ekstraksi ResNet-18 yang terlebih dahulu direduksi dimensinya menggunakan PCA, khususnya pada kombinasi dua komponen utama: PC1 dan PC2.
2. *Clustering* langsung terhadap fitur hasil ekstraksi ResNet-18 tanpa dilakukan reduksi dimensi sebelumnya.

Kedua pendekatan tersebut dievaluasi untuk mengkaji efektivitas penggunaan PCA sebagai tahap praproses dalam sistem pengelompokan fitur citra



kerusakan bangunan berbasis pembelajaran mesin tanpa pengawasan (*unsupervised learning*).

Dari hasil visualisasi yang ditampilkan pada bagian gambar 4.20, *Clustering* berdasarkan proyeksi PCA pada komponen PC1 dan PC2 menunjukkan distribusi data yang cukup terorganisir. Tiga *cluster* utama berhasil terbentuk dengan penyebaran titik data yang relatif proporsional. Meskipun masih terdapat sejumlah titik yang berada di wilayah transisi antar *cluster*, secara keseluruhan batas-batas antar kelompok data dapat dikenali dengan baik.

Hal ini menunjukkan bahwa proyeksi dua dimensi dari PCA berhasil menangkap struktur laten dari data fitur hasil ekstraksi. PC1 dan PC2 yang memiliki kontribusi variansi paling besar dari seluruh komponen, terbukti mampu merepresentasikan karakteristik penting dari fitur citra, dan memungkinkan pembentukan pola *Clustering* yang jelas. Ini menjadikan proses klasifikasi tidak hanya lebih mudah divisualisasikan, tetapi juga lebih stabil secara komputasional, karena dimensi data telah dikurangi secara signifikan tanpa menghilangkan informasi penting.

Sebaliknya, hasil uji coba pada skenario *Clustering* tanpa PCA yang dijelaskan pada gambar 4.21 memperlihatkan kelemahan signifikan dalam struktur hasil pengelompokan. Karena data fitur memiliki dimensi asli yang sangat tinggi (512 dimensi), maka pola distribusi antar titik menjadi sulit untuk dipisahkan. Visualisasi *scatter plot* yang dihasilkan meskipun menggunakan teknik tambahan seperti t-SNE atau PCA sekunder hanya untuk *plotting*, memperlihatkan bahwa titik-titik data tersebar secara acak dan tidak menunjukkan pemisahan yang berarti.

antar *cluster*. Banyak titik yang saling bertumpuk dan tidak terdapat batas visual yang dapat diidentifikasi dengan mudah.

Penggunaan PCA juga berdampak pada efisiensi proses komputasi. Dengan mereduksi fitur dari 512 dimensi menjadi beberapa komponen dengan 2 komponen yang memiliki nilai *eigen value* tertinggi, beban komputasi berkurang secara signifikan. Proses *K-Means* menjadi lebih cepat dan lebih konsisten, karena bekerja pada ruang vektor yang lebih sederhana. Ini sangat penting dalam konteks sistem berbasis citra dengan jumlah data yang besar, seperti dalam kasus pemetaan kerusakan bangunan setelah bencana.

Model *Clustering* dengan PCA juga cenderung menghasilkan representasi yang lebih mudah ditafsirkan oleh manusia. Visualisasi dua dimensi dari proyeksi PCA tidak hanya membantu dalam evaluasi kualitatif terhadap hasil *Clustering*, tetapi juga memungkinkan validasi manual terhadap kelompok-kelompok yang terbentuk, terutama dalam konteks aplikasi yang tidak memiliki label *ground truth*.

Sebaliknya, *Clustering* tanpa PCA meskipun secara teori mampu menangkap seluruh kompleksitas fitur, justru menghasilkan struktur yang tidak terdefinisi dengan baik. Hal ini berdampak pada keandalan sistem, terutama ketika hasil *Clustering* digunakan sebagai dasar untuk pengambilan keputusan lebih lanjut dalam sistem berbasis AI.

#### 4.6 Integrasi Islam

Penelitian ini tidak hanya berlandaskan pada pendekatan ilmiah dan teknologi, tetapi juga mengintegrasikan nilai-nilai Islam yang menekankan keseimbangan dan tanggung jawab manusia dalam berbagai aspek kehidupan.

Dalam Islam, hubungan manusia dengan Allah, sesama manusia, dan alam merupakan bagian fundamental yang dikenal dengan konsep *Hablum Minallah*, *Hablum Minannas*, dan *Hablum Minal Alam*. Berikut penjelasannya beserta ayat-ayat yang relevan.

#### 4.6.1 *Hablum Minallah* (Hubungan Manusia dengan Allah)

Konsep *Hablum Minallah* atau hubungan manusia dengan Allah mencakup kesadaran spiritual dan ketundukan atas segala takdir dan ketetapan-Nya, termasuk ketika manusia diuji melalui musibah atau bencana. Hal ini ditegaskan dalam QS. Al-Baqarah ayat 155–157:

وَلَنَبْلُوَنَّكُمْ بِشَيْءٍ مِّنَ الْخَوْفِ وَالْجُوعِ وَنَقْصٍ مِّنَ الْأَمْوَالِ وَالْأَنْفُسِ وَالثَّمَرَاتِ وَبَشِّرِ الصَّابِرِينَ ﴿١٥٥﴾ الَّذِينَ إِذَا أَصَابَتْهُمُ مُصِيبَةٌ قَالُوا إِنَّا لِلَّهِ وَإِنَّا إِلَيْهِ رَاجِعُونَ ﴿١٥٦﴾ أُولَٰئِكَ عَلَيْهِمْ صَلَوَاتٌ مِّن رَّبِّهِمْ وَرَحْمَةٌ وَأُولَٰئِكَ هُمُ الْمُفْتَخِرُونَ ﴿١٥٧﴾

“(155) Kami pasti akan mengujimu dengan sedikit ketakutan dan kelaparan, kekurangan harta, jiwa, dan buah-buahan. Sampaikanlah (wahai Nabi Muhammad,) kabar gembira kepada orang-orang sabar, (156)(yaitu) orang-orang yang apabila ditimpa musibah, mereka mengucapkan: “*Innā lillāhi wa innā ilaihi rāji‘ūn*” (sesungguhnya kami adalah milik Allah dan sesungguhnya hanya kepada-Nya kami akan kembali). (157) Mereka itulah yang memperoleh ampunan dan rahmat dari Tuhannya dan mereka itulah orang-orang yang mendapat petunjuk.” (QS. Al-Baqarah: 155-157)

Ayat di atas menyatakan bahwa Allah akan menguji manusia dengan rasa takut, kelaparan, kehilangan harta benda, jiwa, dan hasil pertanian, serta memberikan kabar gembira bagi mereka yang sabar. Dalam Tafsir Ibnu Katsir yang dikutip dari jurnal oleh Maulana, Lutni (2020), dijelaskan bahwa ujian dalam bentuk bencana merupakan bagian dari *sunnatullah* (hukum ketetapan Allah) sebagai sarana penyucian jiwa dan peningkatan derajat keimanan manusia. Namun, kesabaran yang dimaksud bukanlah kesabaran pasif, melainkan harus disertai

dengan kesadaran akan tanggung jawab untuk merespons musibah secara aktif, demi kemaslahatan yang lebih luas (Maulana, 2020).

Dalam konteks penelitian ini, implementasi metode ResNet-18 untuk ekstraksi fitur tingkat kerusakan bangunan pasca bencana alam merupakan wujud ikhtiar ilmiah dalam menjawab ujian dari Allah SWT. Teknologi ini memungkinkan untuk membantu proses *clustering* yang mana proses ini dapat dilanjutkan sampai pada tahap klasifikasi tingkat kerusakan bangunan pascabencana sehingga membantu proses tanggap darurat dan pemulihan pascabencana. Upaya ini mencerminkan hubungan erat dengan Allah, karena di balik setiap musibah terdapat perintah untuk tetap bersabar, berserah diri, sekaligus berusaha secara maksimal. Maka, penggunaan kecerdasan buatan dalam konteks ini bukan hanya bentuk pengembangan teknologi, tetapi juga cerminan dari ketaatan kepada Allah yakni merespons takdir dengan usaha, doa, dan kontribusi nyata bagi kemanusiaan. Penelitian ini memperlihatkan bahwa iman kepada Allah dan pengembangan ilmu pengetahuan bukanlah hal yang terpisah, melainkan saling melengkapi dalam mewujudkan misi pengabdian kepada-Nya.

#### **4.6.2 Hablum Minannas (Hubungan Manusia dengan Sesama Manusia)**

Hubungan baik antar manusia atau Hablum Minannās merupakan prinsip mendasar dalam ajaran Islam, yang mendorong umatnya untuk saling membantu dalam urusan dunia dan akhirat. Hal ini ditegaskan dalam QS. Al-Mā'idah ayat 2 yang berbunyi:

يَا أَيُّهَا الَّذِينَ آمَنُوا لَا تَحْلُوا شَعَائِرَ اللَّهِ وَلَا الشَّهْرَ الْحَرَامَ وَلَا الْهَدْيَ وَلَا الْقَلَائِدَ وَلَا آمِينَ الْبَيْتِ الْحَرَامَ يَبْتَغُونَ فَضْلًا مِّن رَّبِّهِمْ وَرِضْوَانًا وَإِذَا حَلَلْتُمْ فَاصْطَادُوا وَلَا يَجْرِمَنَّكُمْ شَنَا نُ قَوْمٍ أَن صَدُّوكُمْ عَنِ الْمَسْجِدِ الْحَرَامِ أَن تَعْتَدُوا وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ وَاتَّقُوا اللَّهَ إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ ﴿٢﴾

*“Wahai orang-orang yang beriman, janganlah kamu melanggar syiar-syiar (kesucian) Allah, jangan (melanggar kehormatan) bulan-bulan haram, jangan (mengganggu) hadyu (hewan-hewan kurban) dan qalā'id (hewan-hewan kurban yang diberi tanda), dan jangan (pula mengganggu) para pengunjung Baitulharam sedangkan mereka mencari karunia dan rida Tuhannya! Apabila kamu telah bertahalul (menyelesaikan ihram), berburulah (jika mau). Janganlah sekali-kali kebencian(-mu) kepada suatu kaum, karena mereka menghalang-halangi dari Masjidilharam, mendorongmu berbuat melampaui batas (kepada mereka). Tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan permusuhan. Bertakwalah kepada Allah, sesungguhnya Allah sangat berat siksaan-Nya.”(QS.Al-Ma'idah:2)*

Menurut Tafsir Ibnu Katsir yang di kutip dari jurnal oleh Puspitasari (2022), ayat ini mengandung beberapa perintah, salah satunya yaitu perintah agar kaum Muslimin saling bahu-membahu dalam semua bentuk kebaikan (al-birr) dan takwa (at-taqwa), termasuk dalam kegiatan sosial, kemanusiaan, dan profesional yang bermanfaat bagi sesama. Menolong orang yang tertimpa musibah, termasuk korban bencana alam, merupakan salah satu bentuk kebaikan yang utama dalam Islam (Puspitasari, 2022).

Dalam konteks penelitian ini, implementasi metode ResNet-18 untuk ekstraksi fitur tingkat kerusakan bangunan pasca bencana alam merupakan wujud ikhtiar ilmiah dalam menjawab ujian dari Allah SWT. Teknologi ini memungkinkan untuk membantu proses *clustering* yang mana proses ini dapat di lanjutkan sampai pada tahap klasifikasi tingkat kerusakan bangunan pascabencana sehingga membantu proses tanggap darurat dan pemulihan pascabencana. Upaya ini mencerminkan hubungan erat dengan Allah, karena di balik setiap musibah

terdapat perintah untuk tetap bersabar, berserah diri, sekaligus berusaha secara maksimal. Maka, penggunaan kecerdasan buatan dalam konteks ini bukan hanya bentuk pengembangan teknologi, tetapi juga cerminan dari ketaatan kepada Allah yakni merespons takdir dengan usaha, doa, dan kontribusi nyata bagi kemanusiaan. Penelitian ini memperlihatkan bahwa iman kepada Allah dan pengembangan ilmu pengetahuan bukanlah hal yang terpisah, melainkan saling melengkapi dalam mewujudkan misi pengabdian kepada-Nya.

#### 4.6.3 Hablum Minal Alam (Hubungan Manusia dengan Alam)

Islam mengajarkan bahwa hubungan manusia dengan alam (hablum minal ‘ālam) merupakan bagian dari tanggung jawab spiritual dan moral dalam menjaga keseimbangan bumi. Hal ini ditegaskan dalam QS. Ar-Rūm ayat 41:

ظَهَرَ الْفَسَادُ فِي الْبَرِّ وَالْبَحْرِ بِمَا كَسَبَتْ أَيْدِي النَّاسِ لِيُذِيقَهُمْ بَعْضَ الَّذِي عَمِلُوا لَعَلَّهُمْ يَرْجِعُونَ ﴿٤١﴾

*“Telah tampak kerusakan di darat dan di laut disebabkan karena perbuatan tangan manusia, supaya Allah merasakan kepada mereka sebagian dari (akibat) perbuatan mereka, agar mereka kembali (ke jalan yang benar).” (QS. Ar-Rūm: 41).*

Berdasarkan Tafsir al-Lubāb karya Quraish Shihab yang di kutip dari jurnal oleh Rifzikka (2024), ayat ini menjelaskan bahwa kerusakan yang terjadi di bumi, baik berupa bencana alam, degradasi lingkungan, maupun kehancuran infrastruktur, merupakan bentuk konsekuensi dari kelalaian manusia dalam menjaga alam. Kerusakan tersebut juga dimaknai sebagai bentuk peringatan dari Allah agar manusia merenung, kembali pada jalan yang benar, dan memperbaiki perilaku mereka terhadap lingkungan (Rifzikka, 2024).

Dalam konteks penelitian ini, implementasi metode ResNet-18 untuk ekstraksi fitur tingkat kerusakan bangunan pasca bencana alam merupakan bagian dari upaya manusia untuk memahami, memantau, dan merespons kerusakan yang terjadi di bumi. Dengan memanfaatkan teknologi kecerdasan buatan untuk membantu proses *clustering* yang mana proses ini dapat di lanjutkan sampai pada tahap klasifikasi tingkat kerusakan bangunan pascabencana sehingga membantu proses tanggap darurat dan pemulihan pascabencana, penelitian ini berkontribusi pada sistem mitigasi bencana yang lebih efisien dan tanggap. Hal ini mencerminkan bentuk tanggung jawab ilmiah sekaligus spiritual dalam menjaga amanah Allah atas bumi. Maka, penelitian ini dapat dipandang sebagai perwujudan nilai *Hablum Minal ‘Ālam* dalam Islam, yakni memelihara dan memakmurkan bumi melalui pemanfaatan ilmu pengetahuan dan teknologi untuk kemaslahatan alam dan manusia.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Penelitian ini dilakukan untuk menjawab dua rumusan masalah, yaitu bagaimana mengimplementasikan metode ResNet-18 untuk ekstraksi fitur gambar kerusakan bangunan pasca bencana alam, dan bagaimana mengetahui performa metode ResNet-18 dalam mengidentifikasi tingkat kerusakan melalui hasil representasi fiturnya. Berdasarkan hasil pengujian pada Bab sebelumnya yang dapat dilihat pada subbab 4.2, diperoleh kesimpulan bahwa arsitektur ResNet-18 berhasil diimplementasikan dengan baik sebagai alat ekstraksi fitur. Melalui serangkaian proses konvolusi dan *Residual Block*, ResNet-18 mampu menghasilkan vektor fitur berdimensi tinggi yang secara efektif merepresentasikan elemen visual penting dari citra. Representasi fitur ini kemudian digunakan untuk proses *Clustering* guna menilai kemampuannya dalam mengelompokkan citra berdasarkan kesamaan karakteristik kerusakan.

Evaluasi terhadap performa metode dilakukan melalui dua skenario *Clustering* menggunakan algoritma *K-Means*: pertama, dengan fitur yang telah direduksi dimensinya menggunakan PCA, dan kedua, menggunakan fitur asli tanpa reduksi. Hasil pengujian menunjukkan bahwa *Clustering* dengan fitur yang telah direduksi menggunakan PCA (khususnya kombinasi PC1 dan PC2) menghasilkan visualisasi *cluster* yang lebih jelas, terstruktur, dan proporsional seperti yang dapat dilihat pada gambar 4.20 di Bab sebelumnya, serta pada gambar yang terlampir dalam Lampiran 1. Sementara itu, *Clustering* langsung pada fitur berdimensi tinggi



menghasilkan sebaran data yang acak, tumpang tindih, dan tidak membentuk *cluster* yang bermakna seperti yang dapat di lihat pada gambar 4.21 di Bab sebelumnya, serta pada gambar yang terlampir dalam Lampiran 2. Hal ini menunjukkan bahwa penggunaan PCA tidak hanya menyederhanakan kompleksitas data, tetapi juga meningkatkan kejelasan struktur data dan efektivitas pengelompokan. Reduksi dimensi juga memberikan keuntungan dari sisi efisiensi komputasi, karena proses *Clustering* menjadi lebih cepat dan stabil tanpa kehilangan informasi penting.

Secara keseluruhan, dapat disimpulkan bahwa metode ResNet-18 mampu membentuk representasi fitur yang relevan untuk citra bangunan rusak, dan jika dikombinasikan dengan PCA sebagai tahap praproses, fitur tersebut dapat digunakan secara efektif dalam sistem *Clustering* untuk membedakan tingkat atau jenis kerusakan secara tidak langsung. Hasil penelitian ini memberikan dasar yang kuat untuk pengembangan sistem klasifikasi atau pemetaan kerusakan otomatis berbasis citra dalam konteks pasca bencana.

## 5.2 Saran

Berdasarkan hasil dan kesimpulan dari penelitian ini, beberapa saran yang dapat diberikan untuk pengembangan dan penelitian selanjutnya adalah sebagai berikut:

1. Gunakan Lebih Banyak Komponen PCA untuk Eksperimen Lanjutan

Penelitian ini hanya menggunakan dua komponen utama (PC1 dan PC2) untuk proses *Clustering*. Disarankan agar dilakukan pengujian lanjutan dengan

menambahkan lebih banyak komponen (misalnya 3–6) untuk mengevaluasi apakah ada peningkatan akurasi atau struktur *cluster* yang lebih baik.

2. Lakukan Evaluasi Kuantitatif terhadap Hasil *Clustering*

Penilaian hasil *Clustering* dalam penelitian ini bersifat kualitatif melalui visualisasi *scatter plot*. Penelitian berikutnya sebaiknya melibatkan metrik evaluasi kuantitatif seperti *Silhouette Score*, *Davies-Bouldin Index*, atau *Calinski-Harabasz Index* agar hasil pengelompokan dapat dibuktikan secara statistik dan lebih objektif.

3. Gunakan Dataset Lebih Luas dan Beragam

Dataset yang lebih besar dengan variasi jenis kerusakan dan lingkungan bangunan akan membantu menguji generalisasi metode yang dikembangkan. Hal ini penting agar model yang dibangun dapat digunakan dalam situasi nyata yang lebih kompleks.

4. Kembangkan Model Menuju Klasifikasi Kerusakan

Karena hasil *Clustering* menunjukkan bahwa fitur mampu membedakan kelompok gambar dengan baik, saran berikutnya adalah membangun model klasifikasi berbasis supervised learning (misalnya SVM, *Random Forest*, atau CNN bertingkat) yang mampu memberikan label tingkat kerusakan secara otomatis.

## DAFTAR PUSTAKA

- Abdelhamid Mahri University, Algeria, Sabba, S., Smara, M., Abdelhamid Mahri University, Algeria, Benhacine, M., Abdelhamid Mahri University, Algeria, Terra, L., Abdelhamid Mahri University, Algeria, Eddine Terra, Z., & Abdelhamid Mahri University, Algeria. (2022). Residual Neural Network in Genomics. *Computer Science Journal of Moldova*, 30(3(90)), 308–334. <https://doi.org/10.56415/csjm.v30.17>
- Abdi, H., & Williams, L. J. (2010). Principal component analysis. *WIREs Computational Statistics*, 2(4), 433–459. <https://doi.org/10.1002/wics.101>
- Almais, A. T. W., Susilo, A., Naba, A., Sarosa, M., Crysdian, C., Tazi, I., Hariyadi, M. A., Muslim, M. A., Basid, P. M. N. S. A., Arif, Y. M., Purwanto, M. S., Parwatiningtyas, D., Supriyono, & Wicaksono, H. (2023). Principal Component Analysis-Based Data Clustering for Labeling of Level Damage Sector in Post-Natural Disasters. *IEEE Access*, 11, 74590–74601. <https://doi.org/10.1109/ACCESS.2023.3275852>
- Almais, A. T. W., Susilo, A., Naba, A., Sarosa, M., Juwono, A. M., Crysdian, C., Muslim, M. A., & Wicaksono, H. (2024). Characterization of Structural Building Damage in Post-Disaster Using GLCM-PCA Analysis Integration. *IEEE Access*, 12, 146190–146201. <https://doi.org/10.1109/ACCESS.2024.3469637>
- Aslam, S., & Rabie, T. F. (2023). Principal Component Analysis in Image Classification: A review. *2023 Advances in Science and Engineering Technology International Conferences (ASET)*, 1–7. <https://doi.org/10.1109/ASET56582.2023.10180847>
- Cahyati, I. D., & Devella, S. (2024). *Pengenalan Motif Songket Palembang Menggunakan Convolutional Neural Network dengan Arsitektur ResNet-50*. 5(1).
- Cao, C., Liu, F., Tan, H., Song, D., Shu, W., Li, W., Zhou, Y., Bo, X., & Xie, Z. (2018). Deep Learning and Its Applications in Biomedicine. *Genomics, Proteomics & Bioinformatics*, 16(1), 17–32. <https://doi.org/10.1016/j.gpb.2017.07.003>
- Chen, H., Huang, Z., Hong, Y., Wang, Y., Lyu, Z., Xu, Z., Lan, J., & Gu, Z. (2024). *Efficient Transfer Learning for Video-language Foundation Models* (Version 2). arXiv. <https://doi.org/10.48550/ARXIV.2411.11223>
- Chiu, H.-Y., Chao, H.-S., & Chen, Y.-M. (2022). Application of Artificial Intelligence in Lung Cancer. *Cancers*, 14(6), 1370. <https://doi.org/10.3390/cancers14061370>

- Francis, S. B., & Prakash Verma, J. (2025). Deep CNN ResNet-18 based model with attention and transfer learning for Alzheimer's disease detection. *Frontiers in Neuroinformatics*, 18, 1507217. <https://doi.org/10.3389/fninf.2024.1507217>
- Gupta, S., & Gupta, S. (2024). Feature Extraction and Feature Selection Procedures for Medical Image Analysis: In A. Sinha, P. Saxena, S. K. Singh, & H. Singh (Eds.), *Advances in Medical Technologies and Clinical Practice* (pp. 221–280). IGI Global. <https://doi.org/10.4018/979-8-3693-5226-7.ch010>
- Gurjar, A., & Voditel, P. (2023). Incorporating Transfer Learning in CNN Architecture. *International Journal of Next-Generation Computing*. <https://doi.org/10.47164/ijngc.v14i1.1052>
- Khalid, Md., Pal, N., & Arora, K. (2014). Clustering of Image Data Using K-Means and Fuzzy K-Means. *International Journal of Advanced Computer Science and Applications*, 5(7). <https://doi.org/10.14569/IJACSA.2014.050724>
- Lakshmi, M. A., Victor Daniel, G., & Srinivasa Rao, D. (2019). Initial Centroids for K-Means Using Nearest Neighbors and Feature Means. In J. Wang, G. R. M. Reddy, V. K. Prasad, & V. S. Reddy (Eds.), *Soft Computing and Signal Processing* (Vol. 900, pp. 27–34). Springer Singapore. [https://doi.org/10.1007/978-981-13-3600-3\\_3](https://doi.org/10.1007/978-981-13-3600-3_3)
- Lin, Q., Ci, T., Wang, L., Mondal, S. K., Yin, H., & Wang, Y. (2022). Transfer Learning for Improving Seismic Building Damage Assessment. *Remote Sensing*, 14(1), 201. <https://doi.org/10.3390/rs14010201>
- Maulana, L. (2020). *UJIAN DALAM AL-QUR'AN (Studi Komparatif Tafsir Ibnu Katsir Dan M. Quraish Shihab Atas Surat Al Baqarah Ayat 155-157)*. <http://e-repository.perpus.iainsalatiga.ac.id/id/eprint/9841>
- Mienye, I. D., Swart, T. G., Obaido, G., Jordan, M., & Ilono, P. (2024). *Deep Convolutional Neural Networks: A Comprehensive Review*. <https://doi.org/10.20944/preprints202408.1288.v1>
- Mohammadi, A., Kelhe, T., Carreiro, D., Van Dine, A., & Peebles, J. (2024). *Transfer Learning for Passive Sonar Classification using Pre-trained Audio and ImageNet Models* (Version 1). arXiv. <https://doi.org/10.48550/ARXIV.2409.13878>
- Nagpal, P., Bhinge, S. A., & Shitole, A. (2022). A Comparative Analysis of ResNet Architectures. *2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, 1–8. <https://doi.org/10.1109/SMARTGENCON56628.2022.10083966>

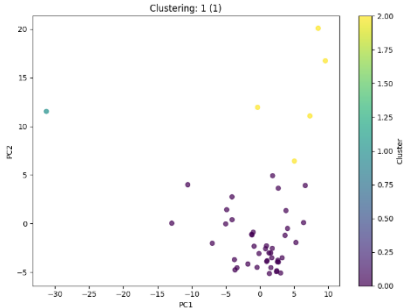
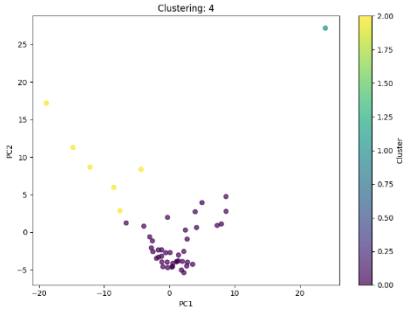
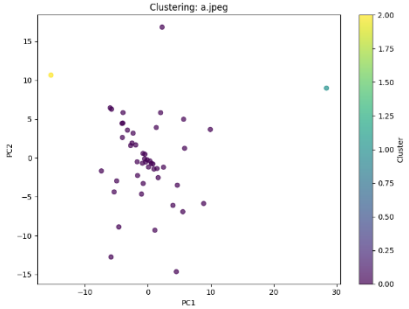
- Patel, S. S., Kumar, N., Aswathy, J., Vaddadi, S. K., Akbar, S. A., & Panchariya, P. C. (2022). K-Means Algorithm: An Unsupervised Clustering Approach Using Various Similarity/Dissimilarity Measures. In J. S. Raj, R. Palanisamy, I. Perikos, & Y. Shi (Eds.), *Intelligent Sustainable Systems* (Vol. 213, pp. 805–813). Springer Singapore. [https://doi.org/10.1007/978-981-16-2422-3\\_63](https://doi.org/10.1007/978-981-16-2422-3_63)
- Purwono, P., Ma'arif, A., Rahmانيar, W., Fathurrahman, H. I. K., Frisky, A. Z. K., & Haq, Q. M. U. (2023). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems*, 2(4), 739–748. <https://doi.org/10.31763/ijrcs.v2i4.888>
- Puspitasari, M. (2022). KERJASAMA DALAM LEMBAGA PENDIDIKAN BERDASARKAN TAFSIR AL-QUR'AN SURAT AL-MAIDAH AYAT 2. *LEARNING : Jurnal Inovasi Penelitian Pendidikan dan Pembelajaran*, 2(3), 209–221. <https://doi.org/10.51878/learning.v2i3.1521>
- Rahmawati et al. - 2025—*Analisis Performa Model ResNet-50 Pada Diagnosis P.pdf*. (n.d.).
- Rifzikka, S. A. (2024). STUDI ANALISIS TAFSIR SURAH AR-RUM AYAT 41 TENTANG KERUSAKAN LINGKUNGAN. *Journal of Islamic Studies and Humanities*, 9(2), 254–298. <https://doi.org/10.21580/jish.v9i2.23659>
- Saini, S. S., & Rawat, P. (2022). Deep Residual Network for Image Recognition. *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, 1–4. <https://doi.org/10.1109/ICDCECE53908.2022.9792645>
- Stojnev, D., & Stojnev Ilić, A. (2020). Preprocessing Image Data for Deep Learning. *Proceedings of the International Scientific Conference - Sinteza 2020*, 312–317. <https://doi.org/10.15308/Sinteza-2020-312-317>
- Sun, H., & Li, Q. (2019). Research on Application of PCA and K-Means Clustering in Enterprise Human Resources. In G. Q. Huang, C.-F. Chien, & R. Dou (Eds.), *Proceeding of the 24th International Conference on Industrial Engineering and Engineering Management 2018* (pp. 695–705). Springer Singapore. [https://doi.org/10.1007/978-981-13-3402-3\\_73](https://doi.org/10.1007/978-981-13-3402-3_73)
- Universitas Brawijaya, Thiodorus, G., Prasetya, A., Universitas Brawijaya, Ardhani, L. A., Universitas Brawijaya, Yudistira, N., & Universitas Brawijaya. (2021). Klasifikasi citra makanan/non makanan menggunakan metode Transfer Learning dengan model Residual Network. *Teknologi*, 11(2), 74–83. <https://doi.org/10.26594/teknologi.v11i2.2402>
- Valliani, A. A.-A. (2019). *Deep Learning and Neurology: A Systematic Review*.

- Vasantdada Patil Pratishthan's College of Engineering and Visual Arts, India, Wagh, K., K.S., S., Rajeswari Vedachalam Government Arts College, India, Sayed, M., Ajeenkya DY Patil University, India, Shahane, S., & Ajeenkya DY Patil University, India. (2024). EXPLORING TRANSFER LEARNING IN IMAGE ANALYSIS USING FEATURE EXTRACTION WITH PRE-TRAINED MODELS. *ICTACT Journal on Image and Video Processing*, 14(3), 3203–3208. <https://doi.org/10.21917/ijivp.2024.0456>
- Wang, C., & Wang, M. (2024). Application Research of Image Classification Based on Convolutional Neural Network Model. *2024 IEEE 2nd International Conference on Image Processing and Computer Applications (ICIPCA)*, 993–999. <https://doi.org/10.1109/ICIPCA61593.2024.10709279>
- Wu, D. (2024). Statistical Feature Extraction and Classification of Complex Data Based on Deep Learning. *2024 3rd International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS)*, 755–760. <https://doi.org/10.1109/AIARS63200.2024.00142>
- Wu, Z. (2024). Application of CNN Classic Model in Modern Image Processing. *Journal of Advances in Engineering and Technology*, 1(3), 1–6. <https://doi.org/10.62177/jaet.v1i3.25>
- Zafar, A., Saba, N., Arshad, A., Alabrah, A., Riaz, S., Suleman, M., Zafar, S., & Nadeem, M. (2024). Convolutional Neural Networks: A Comprehensive Evaluation and Benchmarking of Pooling Layer Variants. *Symmetry*, 16(11), 1516. <https://doi.org/10.3390/sym16111516>
- Zorić, M., Štula, M., Markić, I., & Braović, M. (2024). Transfer Learning in Building Neural Network Model Case Study. *2024 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 1–6. <https://doi.org/10.23919/SoftCOM62040.2024.10721924>

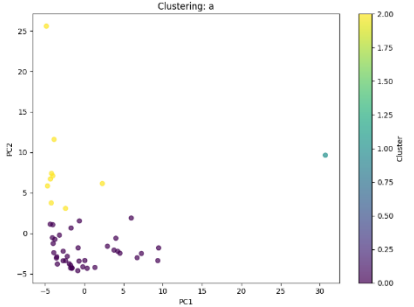
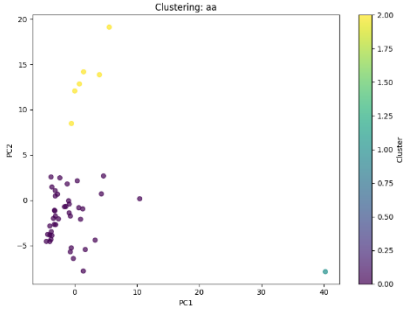
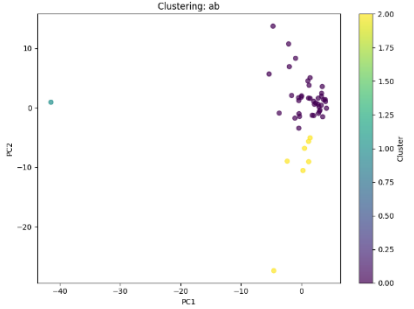
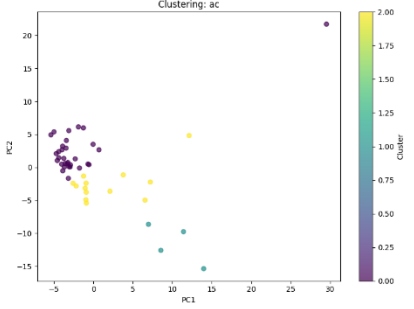
# LAMPIRAN

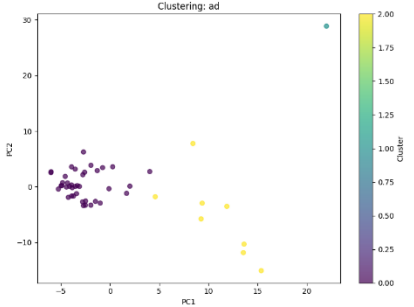
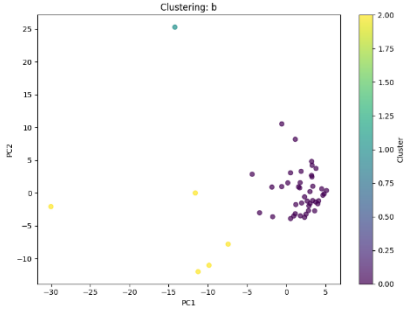
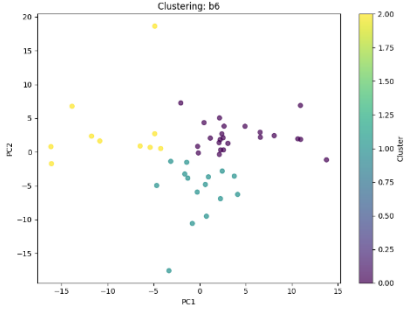
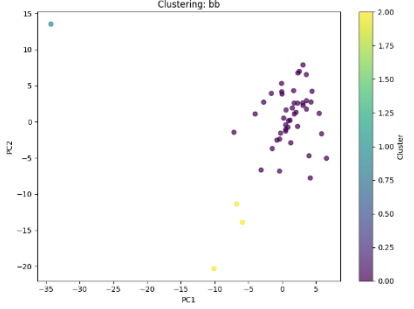
Lampiran 1

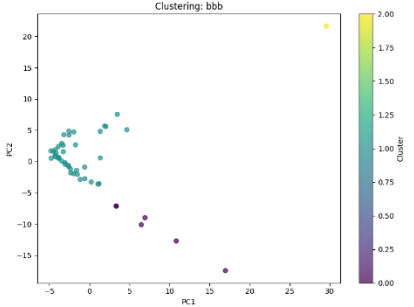
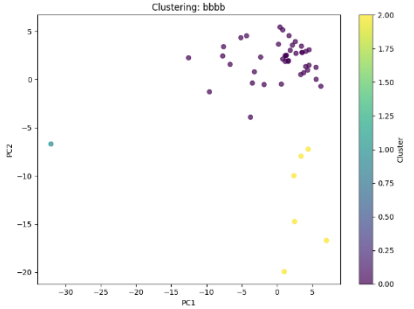
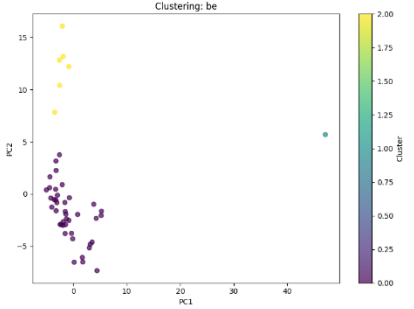
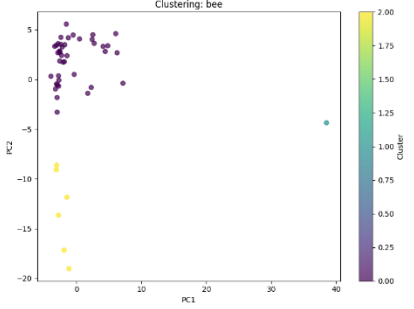
Hasil *Clustering* Fitur setelah dilakukan proses reduksi dimensi menggunakan PCA

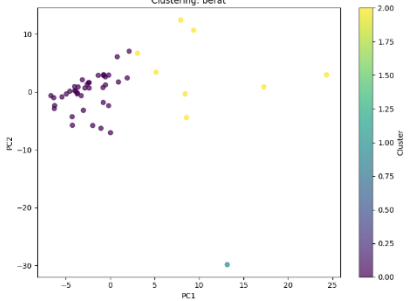
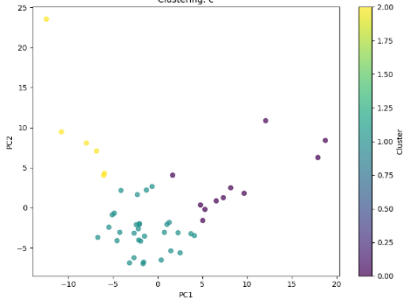
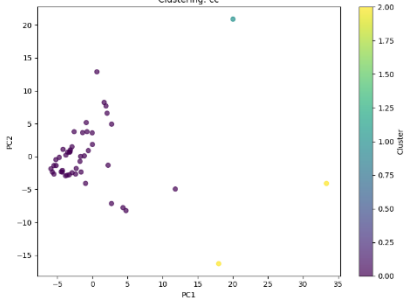
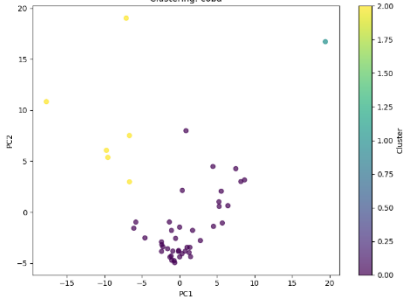
| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 1.  |    |    |
| 2.  |   |   |
| 3.  |  |  |

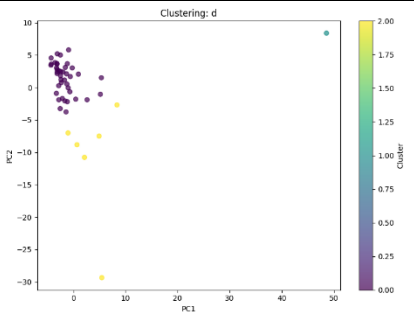
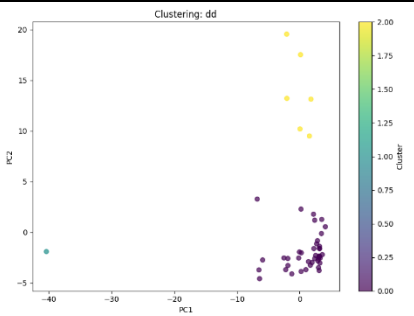
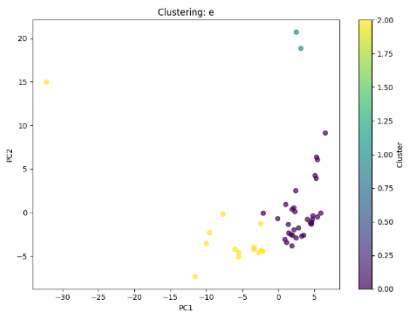
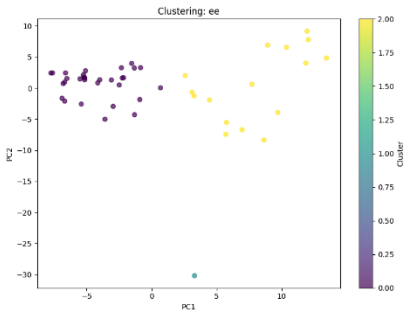


| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 4.  |    |    |
| 5.  |    |   |
| 6.  |  |  |
| 7.  |  |  |

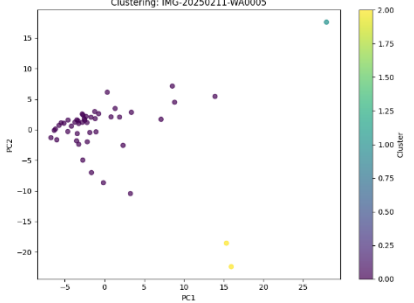
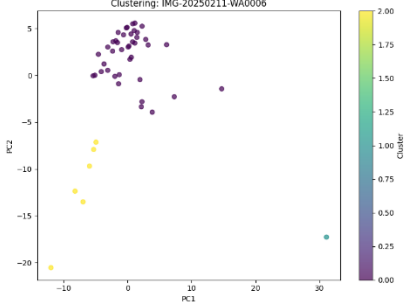
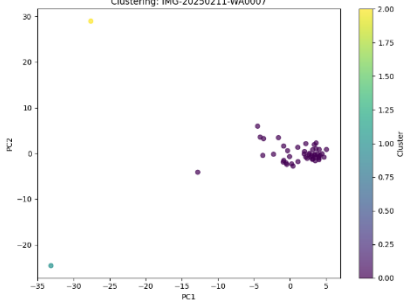
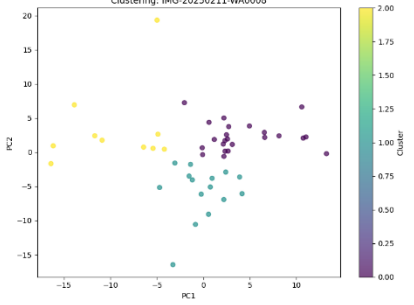
| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 8.  |    |    |
| 9.  |    |   |
| 10. |  |  |
| 11. |  |  |

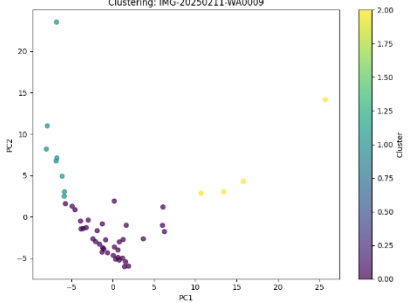
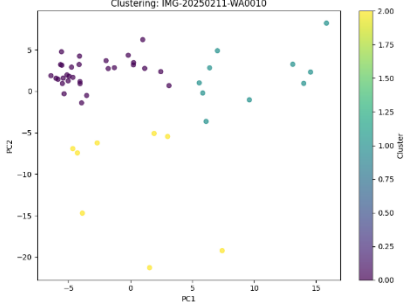
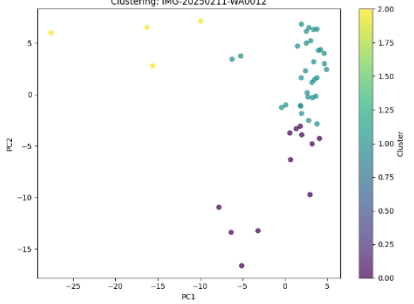
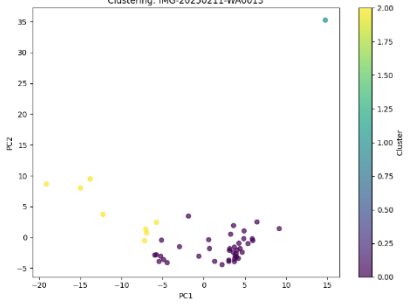
| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 12. |    |    |
| 13. |    |   |
| 14. |  |  |
| 15. |  |  |

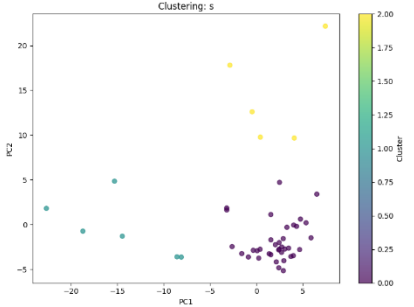
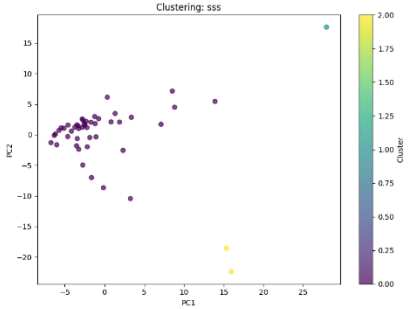
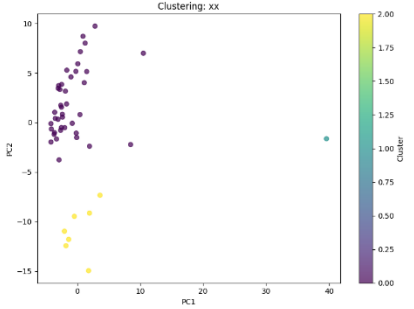
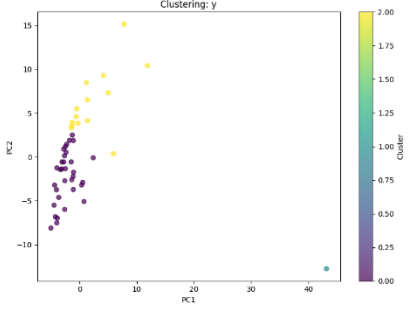
| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 16. |    |    |
| 17. |    |   |
| 18. |  |  |
| 19. |  |  |

| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 20. |    |    |
| 21. |    |   |
| 22. |  |  |
| 23. |  |  |

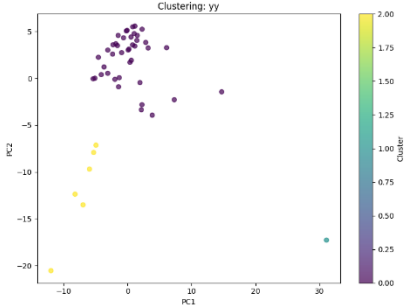


| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 24. |    |    |
| 25. |    |   |
| 26. |  |  |
| 27. |  |  |

| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 28. |    |    |
| 29. |    |   |
| 30. |  |  |
| 31. |  |  |

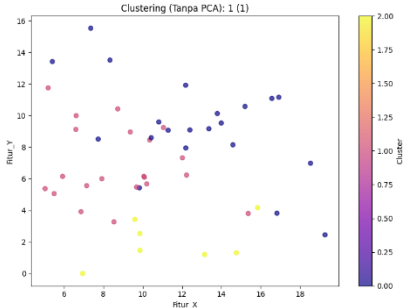
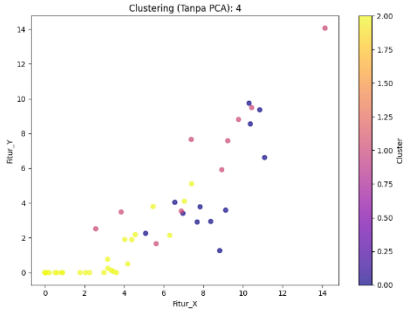
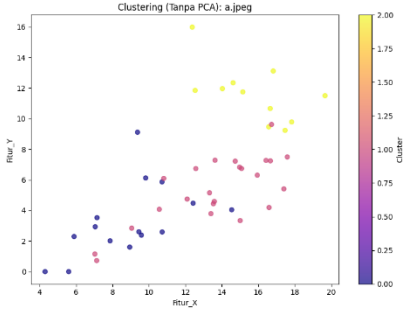
| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 32. |    |    |
| 33. |    |   |
| 34. |  |  |
| 35. |  |  |

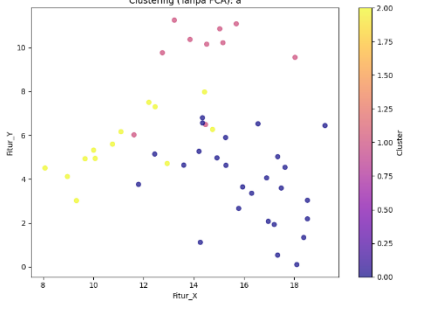
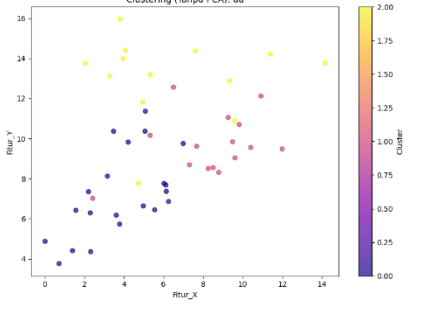
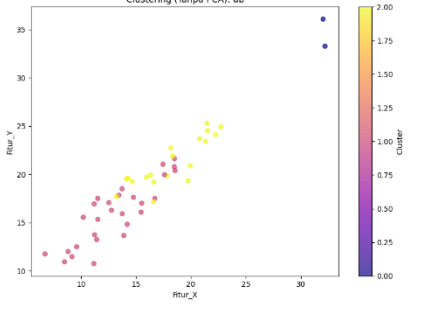
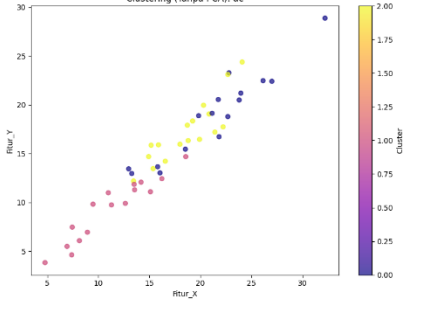


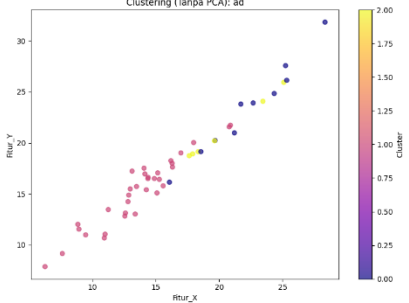
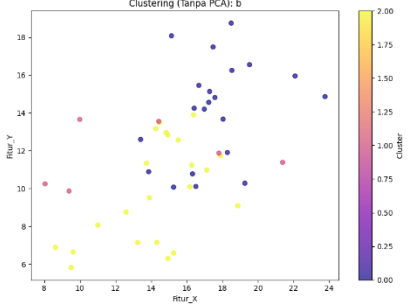
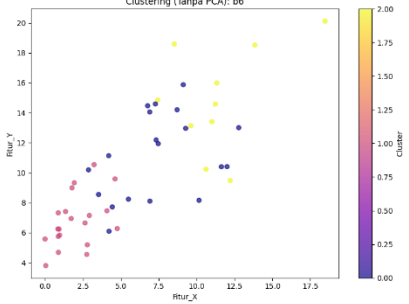
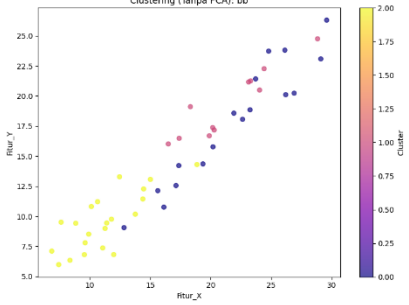
| No. | Data Gambar                                                                       | Hasil                                                                              |
|-----|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| 36. |  |  |

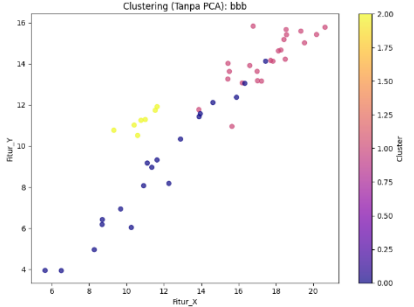
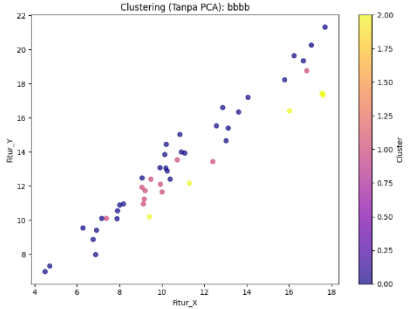
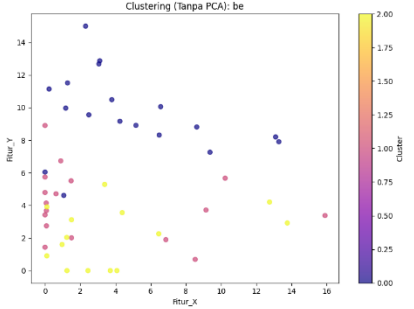
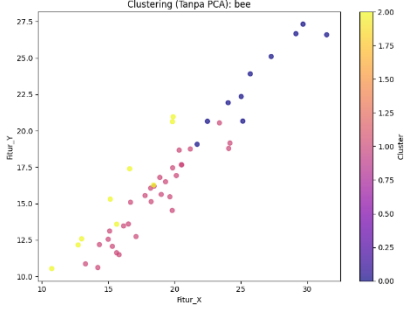
Lampiran 2

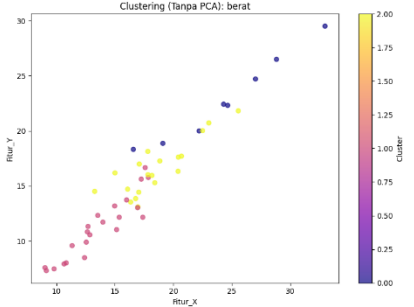
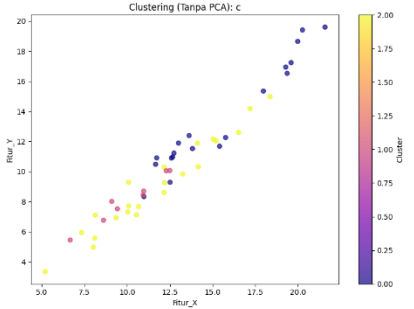
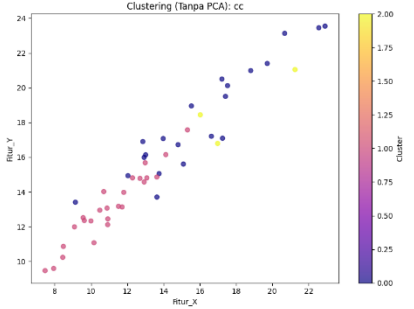
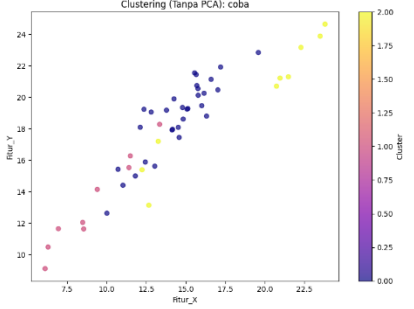
Hasil *Clustering* Fitur tanpa dilakukan proses reduksi dimensi menggunakan PCA

| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 1.  |    |    |
| 2.  |   |   |
| 3.  |  |  |

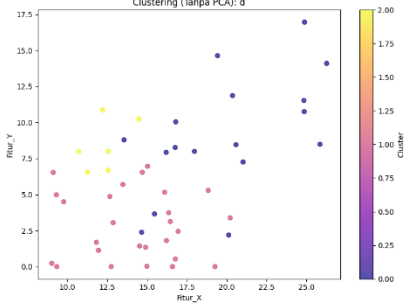
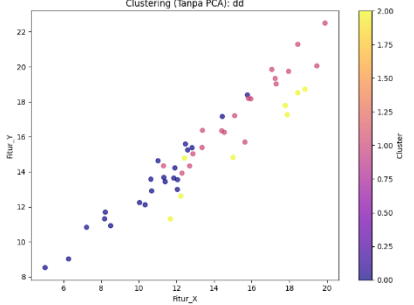
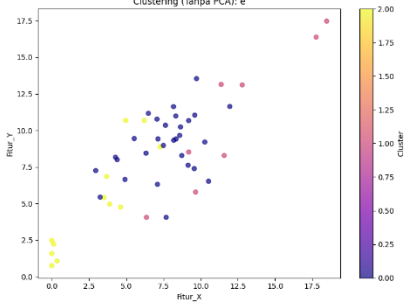
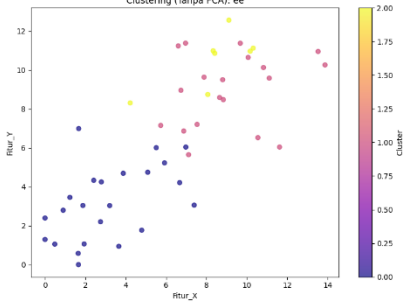
| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 4.  |    |    |
| 5.  |    |   |
| 6.  |  |  |
| 7.  |  |  |

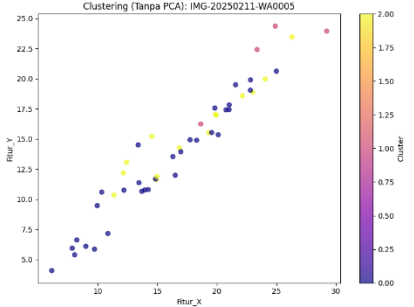
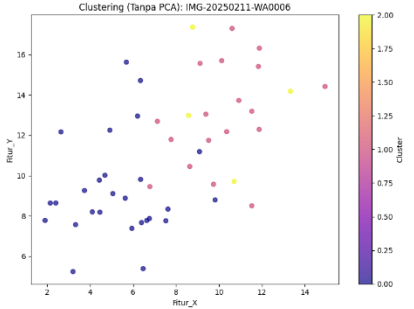
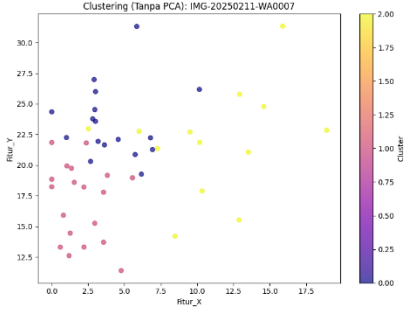
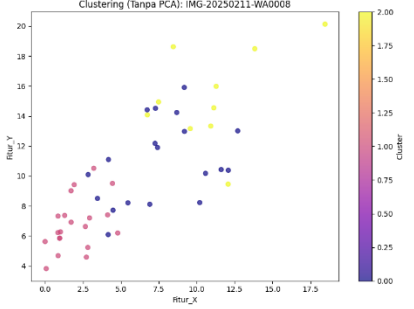
| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 8.  |    |    |
| 9.  |    |   |
| 10. |  |  |
| 11. |  |  |

| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 12. |    |    |
| 13. |    |   |
| 14. |  |  |
| 15. |  |  |

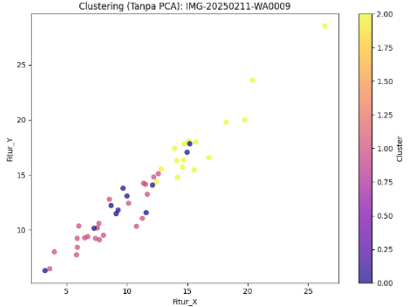
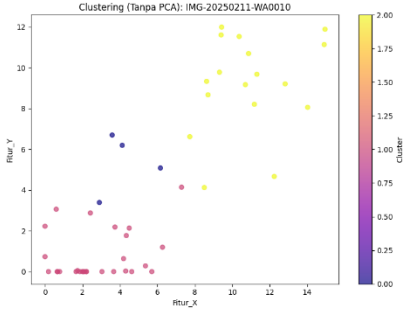
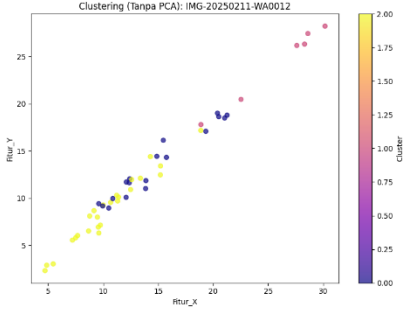
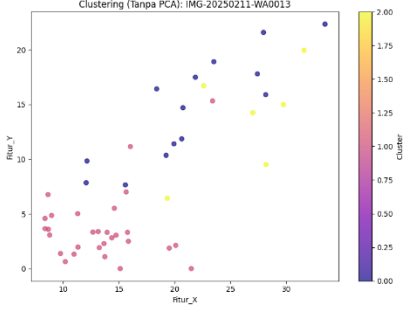
| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 16. |    |    |
| 17. |    |   |
| 18. |  |  |
| 19. |  |  |

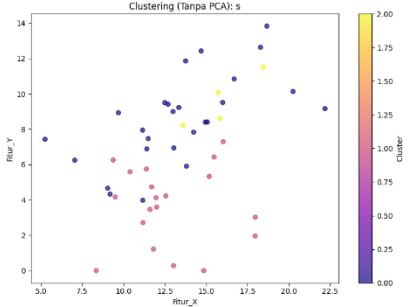
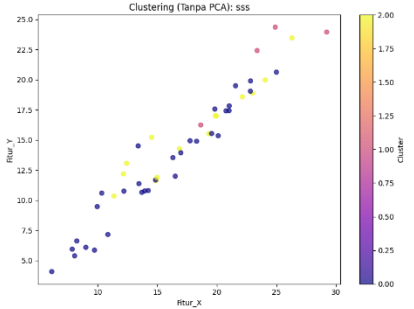
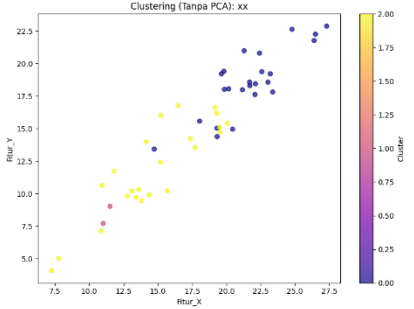
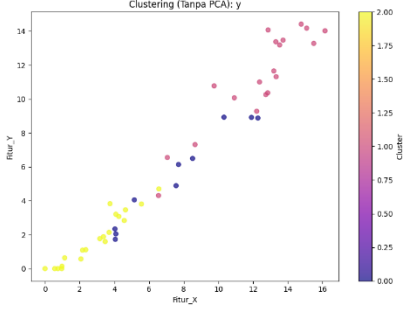


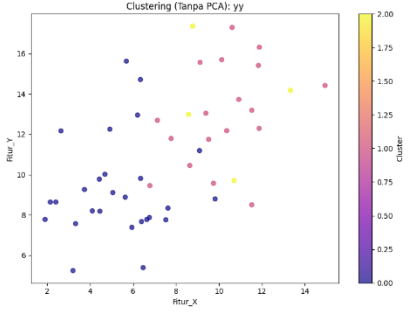
| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 20. |    |    |
| 21. |    |   |
| 22. |  |  |
| 23. |  |  |

| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 24. |    |    |
| 25. |    |   |
| 26. |  |  |
| 27. |  |  |



| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 28. |    |    |
| 29. |    |   |
| 30. |  |  |
| 31. |  |  |

| No. | Data Gambar                                                                         | Hasil                                                                                |
|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 32. |    |    |
| 33. |    |   |
| 34. |  |  |
| 35. |  |  |

| No. | Data Gambar                                                                       | Hasil                                                                              |
|-----|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| 36. |  |  |