

## BAB II

### KAJIAN PUSTAKA

#### 2.1 Tinjauan Pustaka

Penelitian yang dilakukan oleh (Üreten dkk., 2020) bertujuan mengembangkan metode diagnosis berbantuan komputer untuk membantu dokter dalam mendiagnosis osteoarthritis pinggul melalui interpretasi radiografi pinggul polos. Metode yang digunakan yaitu CNN dengan arsitektur VGG-16. Penelitian ini dilakukan di Fakultas Kedokteran Universitas Kyrykkale. Variabel yang digunakan yaitu radiografi polos dengan *dataset* yang terdiri dari 221 citra radiografi pinggul normal dan 213 citra radiografi pinggul dengan osteoarthritis (OA). Hasil penelitian menunjukkan akurasi sebesar 90,2%, sensitivitas 97,6%, spesifisitas 83,0%, dan presisi 84,7%.

Penelitian lain yang dilakukan oleh (Mahum dkk., 2021) berfokus pada deteksi dini *Knee Osteoarthritis* (KOA) atau osteoarthritis lutut pada tahap awal melalui ekstraksi dan klasifikasi fitur/ciri berbasis *deep learning*. Metode yang digunakan adalah pendekatan *hybrid* yang memanfaatkan fitur dari *deep CNN*. Fitur/ciri diekstraksi dari citra sinar-X yang menampilkan lebar ruang sendi lutut dengan menggunakan deskriptor fitur/ciri *hybrid* yakni *Convolutional Neural Network* (CNN) di kombinasikan dengan *Local Binary Patterns* (LBP) dan *Histogram of oriented gradient* (HOG). Fitur/ciri tingkat rendah dihitung menggunakan HOG, sementara fitur/ciri tekstur dihitung melalui deskriptor LBP. Selanjutnya, dilakukan klasifikasi multikelas menggunakan algoritma *Random Forest* (RF), *K-Nearest Neighbors* (K-NN) dan *Support Vector Machine* (SVM) untuk mengklasifikasikan KOA berdasarkan skala *Kellgren Lawrence*. Hasil percobaan menunjukkan bahwa deskriptor fitur/ciri HOG memberikan akurasi sekitar 97% dalam deteksi dini dan klasifikasi KOA untuk keempat kelas *Kellgren Lawrence*.

Penelitian yang dilakukan (Jang dkk., 2022) bertujuan memanfaatkan *deep learning* untuk mengembangkan alat estimasi *Hip Joint Center* (HJC) yang cepat

dan objektif pada radiografi *pelvis anteroposterior* (AP). Variabel yang digunakan yaitu radiografi polos dengan metode CNN dan U-Net. Data diperoleh dari *Osteoarthritis Initiative* (OAI), yaitu kumpulan data multisenter mencakup 4.796 pasien yang terdaftar secara prospektif antara tahun 2004 dan 2015. Model yang dikembangkan mampu memprediksi (HJC) dengan kesalahan kurang dari 5 mm pada 80% kasus pinggul menggunakan rasio nonspesifik. Akurasi ini meningkat menjadi 83% ketika menggunakan rasio spesifik berdasarkan jenis kelamin dan mencapai 91% dengan model yang mempertimbangkan rasio tinggi pinggul khusus pasien.

Penelitian yang dilakukan (Magnéli dkk., 2024) mengembangkan model *deep learning* untuk mendiagnosis displasia pinggul melalui radiologi polos serta mengklasifikasikan pinggul displastik berdasarkan tingkat keparahannya. Model yang digunakan adalah jaringan saraf konvolusional (CNN) VGG-16 yang dilatih menggunakan teknik *file-tuning* secara bertahap pada setiap lapisan. Hasil penelitian menunjukkan bahwa model tersebut mampu mencapai akurasi sebesar 92,2% dalam membedakan antara pinggul normal dan pinggul dengan displasia. Temuan ini mengindikasikan bahwa penerapan model *deep learning* memiliki potensi yang kuat dalam meningkatkan akurasi diagnosis displasia pinggul.

Penelitian yang dilakukan oleh (Al-rimy dkk., 2023) berfokus pada deteksi osteoartritis lutut menggunakan DenseNet-169 dengan penerapan teknik *early stooping*. Pendekatan yang di usulkan memungkinkan penentuan jumlah *epoch* pelatihan yang optimal secara efisien sehingga mencegah terjadinya *overfitting*. Teknik yang digunakan dalam penelitian ini menghasilkan alat diagnostik yang lebih akurat dan efisien untuk mendeteksi osteoartritis lutut. Model yang dikembangkan menunjukkan kinerja dengan akurasi sebesar 91,7%, presisi 93,1%, recall 90,5% dan F1 Score 91,3%.

Penelitian yang dilakukan oleh (Gielis dkk., 2020) berfokus pada alur kerja otomatis untuk radiografi pinggul yang difokuskan pada analisis bentuk sendi, serta menguji nilai prognostiknya terhadap resiko OA pinggul di masa mendatang. Studi ini melibatkan 1.002 peserta dari *Cohort Hip* dan *Cohort Knee* (CHECK). Peneliti membangun dan memvalidasi model prediksi secara internal dengan memanfaatkan

data demografi dasar, hasil pemeriksaan fisik, dan skor radiologis, serta menguji nilai prognostik tambahan dari *shape-score* menggunakan *Area under the Curve* (AUC). Hasil penelitian menunjukkan nilai AUC sebesar 0,795, 95%-CI (0,757 – 0,834). setelah penambahan *shape-score*, nilai AUC meningkat menjadi 0,864, 95%-CI (0,833 – 0,895). Temuan ini mengindikasikan bahwa penambahan *shape-score* dapat meningkatkan akurasi prediksi resiko OA pinggul secara signifikan.

Penelitian yang dilakukan oleh (Hirvasniemi dkk., 2019) menggunakan dua metode utama, yaitu ROI (*Region of Interest*) untuk menyeleksi area tulang yang akan di analisis teksturnya, yang kemudian dilanjutkan dengan metode *Logistic regression* menggunakan *Generalized Estimating Equation* (GEE). Penelitian ini memanfaatkan data dari *Cohort Hip and Cohort Knee* (CHECK), sebuah studi kohort prospektif yang melibatkan 1.002 subjek yang dirancang untuk menyelidiki perkembangan awal osteoarthritis lutut dan pinggul. Analisis dilakukan pada radiografi pinggul CHECK *baseline* (987 pinggul), dengan struktur tulang dianalisis menggunakan *Fractal Signature Analysis* (FSA) pada femur dan femur proksimal. Dari 987 pinggul yang tidak memiliki *Radiografi Hip Osteoarthritis* (rHOA) pada *baseline*, sebanyak 435 pinggul (44%) mengalami (rHOA) setelah masa tindak lanjut 10 tahun. Dari 667 pinggul dengan skor *Joint Space Narrowing* (JSN) *baseline* 0, sebanyak 471 (71%) mengalami peningkatan skors JSN  $\geq 1$  dalam 10 tahun. Selain itu, dari 613 pinggul dengan skor (OST) *Osteophyte Score baseline* 0, sebanyak 526 pinggul (86%) mengalami peningkatan skor  $\geq 1$  setelah 10 tahun. Model GEE menunjukkan bahwa usia yang lebih tinggi (OR: 1,06) dan tingkat *Kellgren Lawrence* awal yang lebih tinggi (OR: 3,80) berhubungan signifikan dengan peningkatan resiko signifikan (rHOA) ( $P < 0,05$ ). Sementara itu, jenis kelamin perempuan (OR: 0,54) berhubungan dengan kemungkinan yang lebih rendah untuk mengalami insiden (rHOA) dalam masa tindak lanjut 10 tahun.

Penelitian yang dilakukan (Saber Hosnijeh dkk., 2018) bertujuan mengembangkan dan memvalidasi model prediktif untuk kejadian radiografi osteoarthritis pinggul serta mengevaluasi nilai dari faktor-faktor prediktif yang telah diidentifikasi sebelumnya. Model *Generalized Estimating Equation* (GEE) digunakan untuk memperhitungkan korelasi antara ekstremitas kanan dan kiri pada

setiap individu, menggunakan data dari kohort *Rotterdam Study* (RS-I). Peneliti membangun model dasar yang mencakup data demografi (Usia, Jenis Kelamin, *Body Mass Index* (BMI)), hasil kuesioner, dan variabel pemeriksaan laboratorium klinis, seperti nyeri pinggul, tingkat pendidikan, status merokok, diabetes, riwayat keluarga osteoarthritis dan kadar *C-Reactive Protein* (CRP). Model prediksi yang dikembangkan, yang memadukan data demografis, kuesioner, dan faktor risiko pencitraan, menunjukkan potensi yang menjanjikan untuk memprediksi resiko osteoarthritis pinggul secara akurat.

Penelitian yang dilakukan oleh (Furlough dkk., 2021) menggunakan metode kuisisioner, analisis statistik bivariat dan analisis multivariat. Sampel penelitian terdiri atas 80 wanita (67%) dan 39 pria (33%) dengan usia rata-rata 62 tahun. Sebanyak 91 pasien (76%) di diagnosis menderita osteoarthritis lutut, dan 69% dari mereka mengira bahwa gejala yang dialami disebabkan oleh perubahan terkait usia. Durasi rata-rata gejala adalah 18 bulan (kisaran interkuartil: 8 sampai 60). Dalam analisis bivariat, durasi gejala yang lebih lama ditemukan berhubungan dengan pendapatan tahunan, jenis asuransi (public atau tanpa asuransi), gejala kecemasan yang lebih berat, gejala depresi yang lebih berat, pemikiran katastrofik yang lebih besar, dan tingkat osteoarthritis menurut *Kellgren Lawrence* yang lebih tinggi. Sementara itu, dalam analisis multivariat, durasi gejala yang lebih lama secara signifikan dikaitkan dengan gejala depresi yang lebih besar dan tingkat *Kellgren Lawrence* yang lebih tinggi. Secara keseluruhan, hasil penelitian ini menunjukkan bahwa durasi gejala berhubungan dengan derajat keparahan osteoarthritis dan tingkat gejala depresi yang dialami pasien.

Penelitian yang dilakukan oleh (Klontzas dkk., 2022) memanfaatkan metode pembelajaran mesin, pengelompokan tanpa pengawasan, dan pemodelan multivariat untuk memprediksi penyempitan ruang sendi atau *Joint Space Narrowing* (JSN) yang parah berdasarkan parameter anatomi pinggul, sekaligus mengidentifikasi faktor-faktor yang berhubungan dengan lebar ruang sendi atau *Joint Space Width* (JSW) pada pinggul displastik dan *non-displastik*. Sebanyak 507 pemeriksaan CT pinggul pasien berusia 20-55 tahun dianalisis secara retrospektif, dengan pencatatan parameter JSW, sudut *center edge* (CE), sudut alfa, *Anterior*



*Acetabular Sector Angle* (AASA) dan *neck shaft angel* (NSA). Pada kasus displasia pinggul, kehilangan struktur secara lateral atau anterior saja tidak ditemukan berkorelasi dengan JSN. AASA ( $p < 0,005$ ) dan sudut EC ( $p < 0,032$ ) terbukti sebagai satu-satunya faktor yang secara signifikan berkorelasi dengan JSN pada pinggul displastik. Sementara itu, pada pinggul *non-displastik*, JSW berkorelasi dengan CE, AASA, dan usia serta berkorelasi positif dengan NSA ( $p < 0,001$ ). Model *Random Forest Classifier* mampu memprediksi JSN parah dengan nilai AUC sebesar 69,9% (95% *Convidence Interval*: 47,9 – 91,8%). Selain itu, model kluster *TwoStep* berhasil mengidentifikasi dua kelompok pasien yang berbeda, yaitu kelompok dengan JSW rendah dan kelompok dengan JSW normal yang memiliki karakteristik anatomi berbeda.

Penelitian yang dilakukan (Yanfei Wang dkk., 2020) bertujuan membangun model prognostik untuk memprediksi tingkat keparahan osteoartritis lutut radiografi (KOA) serta mengidentifikasi faktor risiko yang berkontribusi terhadap perkembangan penyakit jangka panjang, guna mendukung intervensi dan pengobatan dini. Peneliti merancang model *Long Short-Term Memory* (LSTM) dengan mekanisme *attention* / perhatian untuk memprediksi tingkat keparahan berdasarkan skala *Kellgren Lawrence* (KL) pada pasien osteoartritis lutut. Hasil penelitian menunjukkan bahwa nilai AUC dari model LSTM untuk masing-masing tingkat KL = 1, 2, 3, dan 4 adalah 0,81, 0,91, 0,99, dan 0,98. Selain itu eksperimen penulis menunjukkan bahwa penerapan LSTM yang dikombinasikan dengan *scaled autoencoder* mampu meningkatkan akurasi sebesar 3%, dari 90% menjadi 93%.

Penelitian yang dilakukan (Almuhammadi dkk., 2022b) menggunakan metode *Decision Tree* dan *Support Vector Machine* (SVM) sebagai algoritma klasifikasi dengan tujuan untuk membedakan pasien OA pinggul dan OA lutut dari subjek penelitian lainnya. Data dalam penelitian ini dikumpulkan di *Sint Maartenskliniek*, Belanda. Sebanyak Lima puluh tiga pasien dengan osteoartritis stadium akhir terdiri atas 26 pasien OA pinggul dan 25 pasien OA lutut, serta 27 subjek kontrol sehat dari kelompok usia yang sama direkrut untuk studi ini. Hasil penelitian menunjukkan bahwa pada pengujian *OA-Pain-Sense*, model *Decision*

*Tree* menghasilkan akurasi rata-rata nilai 86,79%, sedangkan model *Support Vector Machine* (SVM) mendapatkan akurasi rata-rata 83,57%.

## 2.2 Landasan Teori

Berikut ini beberapa penjelasan terkait konsep dan definisi-definisi yang melandasi penelitian yang akan dilakukan:

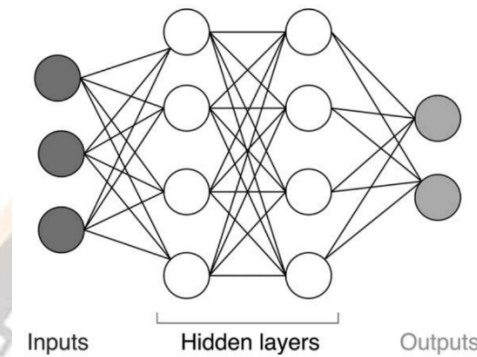
### 2.2.1 *Deep Learning*

Karena otomatisasinya, *Deep Learning* / pembelajaran mendalam yang berasal dari pembelajaran mesin telah mendapatkan popularitas baru-baru ini (Lo & Lai, 2023). Pembelajaran mendalam adalah jenis algoritma mesin (Yifan Li & Liu, 2023; Tanveer dkk., 2023; Sijie Yang dkk., 2021). Pembelajaran mendalam adalah teknik kecerdasan buatan yang membangun jaringan *multi-level* untuk mempelajari fitur/ciri. Kata "mendalam" dan "belajar" masing-masing menunjukkan penggunaan teknik model yang mendalam dan fitur/ciri pembelajaran, sebagai tujuan yang dimaksud (Chenxi Huang dkk., 2023). Pembelajaran mendalam berupaya membangun jaringan saraf yang meniru otak manusia untuk belajar secara analitis. Dalam jaringan saraf, lapisan masukan menerima data, lapisan tersembunyi memprosesnya secara matematis, dan lapisan keluaran mengeluarkan temuan untuk klasifikasi atau prediksi. Dalam pembelajaran mendalam, neuron di setiap lapisan bertindak sebagai blok bangunan penting, dan koneksi antar lapisan didasarkan pada bagaimana neuron saling berhubungan.

Menurut (Lecun dkk., 2015) pembelajaran mendalam berfokus pada jaringan saraf tiruan dengan banyak lapisan tersembunyi yang mempelajari pola kompleks dalam data. Hal ini memungkinkan pemrosesan data yang tidak terstruktur seperti citra, teks, dan suara melalui hierarki representasi yang ditingkatkan pada setiap lapisan. Seperti yang ditunjukkan pada Gambar 2.1 (Chartrand dkk., 2017), jaringan *deep learning* terdiri dari 3 komponen utama yaitu:

- 1) *Input layer*: mempresentasikan data *input* seperti intensitas piksel individual, yang berfungsi menerima data awal dari luar sistem.

- 2) *Hidden Layer*: bagian ini merupakan lapisan tersembunyi yang melakukan transformasi data melalui operasi matematis, seperti fungsi aktivasi
- 3) *Output Layer*: menghasilkan keluaran akhir berupa prediksi atau klasifikasi berdasarkan data yang telah di proses.



Gambar 2.1 Komponen utama *deep learning* (Chartrand dkk., 2017)

Fungsi aktivasi bertujuan untuk memperkenalkan *non-linearitas* ke dalam jaringan, sehingga memungkinkan sebuah jaringan memodelkan hubungan yang kompleks. Fungsi aktivasi yang umum digunakan yaitu seperti berikut ini:

- a) *RELU (Rectified Linier Unit)*: mengatasi masalah *vanishing gradient* dalam jaringan dengan lapisan yang sangat dalam
- b) *Sigmoid*: cocok digunakan untuk data dengan rentang luaran antara 0 dan 1
- c) *Softmax*: Digunakan untuk klasifikasi multikelas.

*Deep learning* terdiri dari beberapa arsitektur yang dapat digunakan untuk melakukan tugas pemrosesan data. Berikut ini merupakan beberapa arsitektur utama dalam pembelajaran mendalam yaitu:

- a. Jaringan saraf tiruan / *Artificial neural Network (ANN)*

Jaringan saraf tiruan adalah dasar dari *deep learning* yang terdiri dari neuron buatan yang terhubung dalam beberapa lapisan. Setiap neuron menerima masukan, mengaplikasikan fungsi aktivasi, dan mengirimkan hasil ke neuron di lapisan berikutnya (Jordan dkk., 2006).

- b. Jaringan saraf konvolusional / *Convolutional Neural Network (CNN)*

Jaringan saraf konvolusional didesain khusus untuk pemrosesan data berbentuk *grid*, seperti citra. Jaringan saraf tiruan menggunakan lapisan konvolusi untuk

mengekstraksi fitur/ciri spasial dari citra, yang diikuti oleh lapisan *pooling* untuk mengurangi dimensi data (Krizhevsky dkk., 2012).

c. Jaringan saraf rekuren / *Recurrent Neural Network* (RNN)

Jaringan saraf rekuren digunakan untuk pemrosesan data yang bersifat sekuensial, seperti sinyal suara dan teks. Jaringan saraf rekuren mempunyai mekanisme memori internal yang memungkinkan informasi dari langkah sebelumnya digunakan pada langkah berikutnya (Hochreiter & Unger Schmidhuber, 1997).

d. Jaringan transformer

Jaringan transformer, yang dikenalkan oleh (Vaswani dkk., 2017), menjadi dasar bagi banyak model terbaru dalam pemrosesan bahasa alami / *Natural Language Processing* (NLP), seperti *Generative Pre-trained Transformer* (GPT) dan *Bidirectional Encoder Representations* (BERT). Transformer menggunakan mekanisme *self-attention* untuk menangkap hubungan dalam data secara efisien.

Pelatihan pembelajaran mendalam dilakukan dengan mengoptimalkan berat jaringan menggunakan algoritma tertentu. Beberapa algoritma terpenting yang digunakan dalam pembelajaran mendalam diantaranya yaitu:

- 1) *Backpropagation* merupakan algoritma yang digunakan untuk memperbarui bobot berdasarkan perbedaan antara prediksi awal dan nilai aktual (Rumelhart dkk., 1986).
- 2) *Gradient Descent* merupakan metode optimasi yang digunakan untuk menemukan bobot optimal dalam jaringan saraf tiruan (Bottou, 2010).
- 3) *Adam Optimizer* merupakan algoritma optimasi yang menggabungkan momentum dan adaptasi *learning rate* untuk meningkatkan efisiensi pelatihan (Kingma & Ba, 2014).

Pembelajaran mendalam memberikan dampak yang besar dalam berbagai aplikasi pengolahan citra seperti:

- a) Segmentasi citra: dapat menggunakan model ROI atau U-Net yang digunakan untuk membedakan bagian dalam suatu citra medis (Ronneberger dkk., 2015).



- b) Klasifikasi citra: dapat menggunakan *convolutional neural network* untuk tugas klasifikasi citra, seperti deteksi penyakit pada citra medis (He dkk., 2015a).
- c) Deteksi objek: dapat menggunakan model seperti *You Only Live Once* (YOLO) dan *faster Region-based Convolutional Neural Networks* (R-CNN) untuk mendeteksi objek dalam citra dengan akurasi tinggi (Redmon dkk., 2016; Ren dkk., 2015).

Pembelajaran mendalam bisa digunakan dalam tiga kategori pembelajaran mesin diantaranya yaitu:

#### 1) *Supervised learning*

*Deep learning* sering digunakan dalam *supervised learning*, dimana model akan dilatih menggunakan *dataset* yang berisi pasangan *input* dan label yang sudah diketahui. Contohnya seperti:

- Klasifikasi citra: CNN digunakan untuk melakukan identifikasi penyakit dari citra medis (Lecun dkk., 2015).
- Pengenalan suara: *speech-to-text* menggunakan RNN atau transformer (Graves, 2013).
- Prediksi harga: menggunakan jaringan saraf untuk regresi (Goodfellow dkk., 2016).

#### 2) *Unsupervised learning*

*Deep learning* juga bisa digunakan dalam *unsupervised learning*, dimana model belajar pola dari data tanpa label eksplisit. Contohnya seperti:

- *Autoencoders*: digunakan untuk reduksi dimensi atau deteksi anomaly (Geoffrey Everest Hinton & Ruslan Rafikovich Salakhutdinov, 2006).
- GANs (*Generative Adversarial Networks*): digunakan untuk pembuatan citra atau augmentasi data medis (Goodfellow dkk., 2016).
- *Clustering* menggunakan *Deep Embeddings*: digunakan untuk pemetaan data ke dalam representasi yang lebih bermakna.

#### 3) *Reinforcement learning*

*Deep learning* sering digunakan juga dalam *deep reinforcement learning*, dimana model belajar dengan mencoba-coba dan mendapatkan *reward* atau *penalty*.

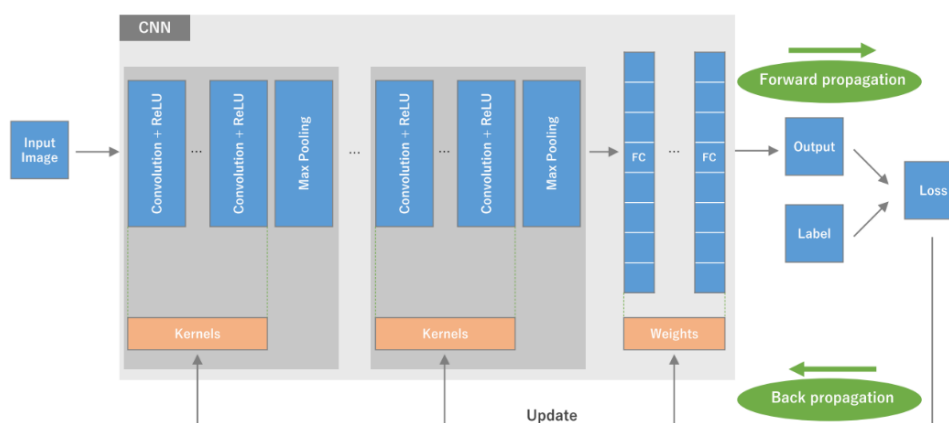
- AlphaGo menggunakan *deep neural network* untuk bermain Go (Silver dkk., 2016).
- Robotika menggunakan *Point of Passage Optimization* (PPO) untuk navigasi robot (Lillicrap dkk., 2015).
- Optimasi pengobatan medis: menggunakan *deep reinforcement learning* untuk rekomendasi dosis obat optimal

### 2.2.2 *Convolutional Neural Network* (CNN)

*Convolutional Neural Network* (CNN) adalah jenis model *deep learning* untuk memproses data yang memiliki pola *grid*, seperti gambar atau citra yang terinspirasi dari bagian korteks visual hewan (Fukushima, 1980; Hubel & Wiesel, 1968) dan dirancang secara otomatis dan adaptif untuk mempelajari hierarki fitur/ciri spasial, dari pola tingkat rendah hingga tingkat tinggi. Arsitektur CNN terdiri dari satu lapisan masukan/*input layer*, lapisan keluaran/*output layer*, dan sejumlah lapisan tersembunyi/*hidden layer*. Lapisan tersembunyi umumnya berisi *convolutional layers*, *pooling layers*, *normalization layers*, *RELU layer*, *fully connected layers*, dan *loss layer* (Zahangir Alom dkk., 2018). Lapisan konvolusi memainkan peran kunci dalam CNN, yang terdiri dari tumpukan operasi matematika, seperti konvolusi, jenis operasi linier khusus. Dalam gambar digital, nilai piksel disimpan dalam kisi dua dimensi (2D), yaitu, susunan angka, dan kisi kecil parameter yang disebut kernel, ekstraktor fitur/ciri yang dapat dioptimalkan, diterapkan pada setiap posisi gambar, yang membuat CNN sangat efisien untuk pemrosesan citra, karena fitur/ciri dapat muncul di mana saja pada citra. Saat satu lapisan memasukkan keluarannya ke lapisan berikutnya, fitur/ciri yang diekstraksi dapat secara hierarkis dan progresif menjadi lebih kompleks. Pelatihan merupakan proses menyempurnakan parameter seperti kernel untuk meminimalkan perbedaan antara keluaran dan label kebenaran dasar. Hal ini dilakukan dengan menggunakan

berbagai algoritma optimasi, termasuk penurunan gradien dan propagasi mundur (Yamashita dkk., 2018b).

Dalam penjelasan lainnya arsitektur CNN mencakup beberapa blok bangunan, seperti *convolution layers*, *pooling layers*, dan *fully connected layers*. Arsitektur umum terdiri dari pengulangan tumpukan beberapa *convolution layers* dan *pooling layers*, diikuti oleh satu atau lebih *fully connected layers*. Tahapan dimana data masukan diubah menjadi data keluaran melalui lapisan-lapisan ini disebut perambatan maju atau *forward propagation* Gambar 2.2.



Gambar 2.2 Arsitektur CNN (Yamashita dkk., 2018b)

Meskipun operasi *convolution* dan *pooling* yang dijelaskan di bagian ini adalah untuk 2D-CNN, operasi serupa juga dapat dilakukan untuk tiga dimensi (3D)-CNN. CNN dianggap sebagai salah satu teknik terbaik untuk mempelajari konten citra dan telah ditampilkan hasil canggih pada pengenalan citra, segmentasi, deteksi, dan tugas terkait pengambilan (Cirean dkk., 2012; Liu dkk., 2019).

CNN digunakan untuk mengekstrak fitur/ciri dari citra *input* atau masukan menggunakan filter atau kernel. Operasi konvolusi memungkinkan model untuk dapat mengenali pola suatu citra seperti tepi, tekstur, dan bentuk tertentu dalam citra (Goodfellow dkk., 2016). *Pooling layer* digunakan untuk mengurangi dimensi fitur/ciri yang dihasilkan dari lapisan konvolusi, sehingga mengurangi jumlah parameter dan meningkatkan proses efisiensi komputasi. *Max pooling* merupakan salah satu teknik yang sering digunakan (Scherer dkk., 2010). *Fully Connected Layer* berfungsi untuk menghubungkan ciri-ciri yang telah diekstraksi dan digunakan untuk melakukan klasifikasi atau regresi. Sedangkan neuron pada *Fully*

*Connected Layer* menerima *input* dari semua neuron pada lapisan sebelumnya dan mengubahnya menjadi *output* akhir (Krizhevsky dkk., 2012).

Fungsi aktivasi memiliki peran penting dalam menentukan *output* dari neuron. Fungsi aktivasi yang sering digunakan dalam CNN yaitu *ReLU (Rectified Linear Unit)*, yang berfungsi merubah nilai negatif menjadi nol untuk meningkatkan *non-linearitas* jaringan (Nair & Hinton, 2010), serta *Softmax* yang digunakan dalam lapisan *output* untuk klasifikasi multi-kelas. Proses pelatihan CNN melibatkan optimasi bobot menggunakan algoritma *backpropagation* dan *stochastic gradient descent*. CNN menggunakan fungsi *loss* seperti *categorical cross-entropy* untuk klasifikasi dan *MSE (Mean Squared Error)* untuk regresi (Rumelhart dkk., 1986).

Arsitektur CNN yang khas umumnya terdiri dari lapisan konvolusi dan penyatuan alternatif diikuti oleh satu atau lebih lapisan yang terhubung sepenuhnya di bagian akhir. Dalam beberapa kasus, lapisan terhubung sepenuhnya diganti dengan lapisan *pooling* rata-rata global. Selain tahapan pembelajaran yang beragam, berbeda pula unit regulasi seperti normalisasi *batch* dan *dropout* juga digabungkan untuk mengoptimalkan kinerja CNN (Khan dkk., 2019). Berikut ini merupakan penjelasan terkait dengan *convolution layers*, *pooling layers*, dan *fully connected layers*.

a) *convolution layers* atau lapisan konvolusional

Lapisan konvolusional terdiri dari satu set kernel konvolusional (setiap neuron bertindak sebagai kernel) Kernel ini dikaitkan dengan area kecil dari gambar yang dikenal sebagai bidang reseptif. Berhasil dengan membagi gambar menjadi blok-blok kecil (bidang reseptif) dan menggabungkannya dengan yang spesifik set bobot (menggandakan elemen filter dengan elemen bidang reseptif yang sesuai) (Khan dkk., 2019). Operasi konvolusi bisa dinyatakan sebagai berikut:

$$F_l^k = (I_{x,y} * K_l^k) \quad (2.1)$$



Persamaan 2.1, gambar masukan diwakili oleh  $I_{x,y}$ ,  $x,y$  menunjukkan lokalitas spasial,  $K_l^k$  mewakili  $l^{th}$  kernel konvolusional dari lapisan  $k^{th}$ . Pembagian gambar menjadi blok-blok kecil membantu dalam mengekstraksi nilai piksel berkorelasi lokal. Informasi yang dikumpulkan secara lokal ini juga dikenal sebagai fitur/ciri motif. Serangkaian fitur/ciri yang berbeda di dalam gambar diekstraksi dengan menggeser kernel konvolusional pada seluruh gambar dengan kumpulan bobot yang sama. Fitur/ciri pembagian berat ini dari operasi konvolusi membuat parameter CNN efisien dibandingkan dengan *Nets* yang terhubung penuh. Operasi konvolusi selanjutnya dapat dikategorikan ke dalam jenis yang berbeda berdasarkan jenis dan ukuran filter, jenis *padding*, dan arah konvolusi (Lecun dkk., 2015)

b) *pooling layer* atau lapisan penyatuan

Motif ciri yang dihasilkan sebagai keluaran dari operasi konvolusi dapat terjadi di berbagai tempat lokasi dalam gambar. Setelah fitur/ciri diekstraksi, lokasi persisnya menjadi kurang penting selama perkiraan posisinya relatif terhadap yang lain dipertahankan. *Pooling* atau *downsampling* seperti konvolusi, merupakan operasi lokal yang menarik. Ini meringkas informasi serupa di lingkungan sekitar dari bidang reseptif dan menghasilkan respons dominan dalam wilayah lokal ini (Lee dkk., 2015).

$$Z_l = f_p(F_{x,y}^l) \quad (2.2)$$

Persamaan (2.2) menampilkan operasi *pooling* dimana  $Z_l$  mempresentasikan  $l^{th}$  peta fitur/ciri keluaran,  $F_{x,y}^l$  menunjukkan  $l^{th}$  masukan peta fitur/ciri, sedangkan  $f_p(.)$  mendefinisikan jenis operasi penggabungan. Penggunaan operasi penyatuan membantu mengekstraksi kombinasi fitur/ciri, yang tidak berbeda dengan terjemahan pergeseran dan distorsi kecil (Ranzato dkk., 2007; Scherer dkk., 2010). Pengurangan ukuran peta fitur/ciri menjadi fitur/ciri invariant set tidak hanya mengatur kompleksitas jaringan tetapi juga membantu dalam meningkatkan generalisasi dengan mengurangi *overfitting*. Berbagai jenis formulasi penyatuan

seperti *maks*, rata-rata, L2, tumpang tindih, penggabungan piramida spasial, dan lain-lain yang digunakan di CNN (Kaiming dkk., 2015b).

c) *fully connected layers* atau lapisan yang terhubung sepenuhnya

Lapisan yang terhubung sepenuhnya sebagian besar digunakan di ujung jaringan untuk tujuan klasifikasi. Tidak seperti penyatuan dan konvolusi, ini adalah operasi global. Dibutuhkan masukan dari lapisan sebelumnya dan secara global menganalisis keluaran dari semua lapisan sebelumnya (Lin dkk., 2013). Ini membuat kombinasi *non-linear* dari fitur/ciri terpilih, yang digunakan untuk klasifikasi data (Rawat & Wang, 2017).

*Supervised learning* merupakan metode pembelajaran mesin di mana model dilatih menggunakan dataset berlabel. Hal ini berarti setiap data memiliki pasangan berupa *input* dan label yang sudah diketahui. Dalam penerapan *supervised learning* pada CNN, langkah pertama yaitu menyiapkan *dataset* yang terdiri dari pasangan *input* berupa citra dan label yang sudah diketahui. Sebagai contoh, dalam identifikasi atau klasifikasi citra medis, *dataset* dapat berupa citra hasil pemindaian seperti citra sinar-X dengan label yang menunjukkan kondisi medis tertentu, seperti penyakit tumor jinak atau paru-paru (Xiaosong Wang dkk., 2017).

Proses pelatihan dimulai dengan *forward propagation*, di mana data citra melewati lapisan-lapisan dalam CNN, seperti lapisan *konvolusi*, *pooling*, dan *fully connected*. CNN akan mengekstrak fitur/ciri dari citra dan menghasilkan prediksi berdasarkan bobot model. Setelah itu, hasil prediksi dibandingkan dengan label atau data aktual menggunakan fungsi *loss*, seperti *cross-entropy loss* yang digunakan untuk tugas klasifikasi. Fungsi *loss* ini digunakan untuk menghitung selisih antara hasil prediksi dan label yang benar, di mana tujuan model yaitu meminimalkan nilai *loss* selama pelatihan.

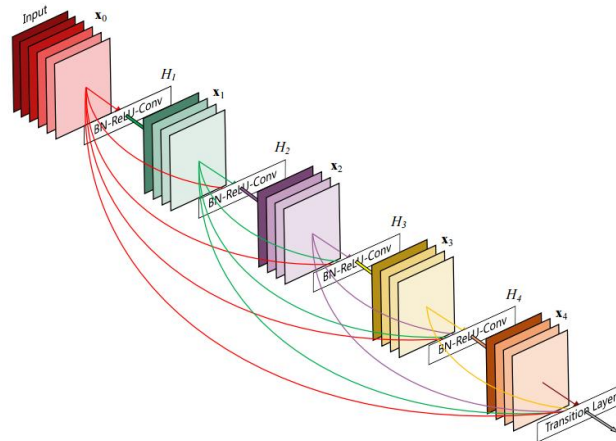
Selanjutnya, model melakukan *backpropagation*, yaitu proses pembaruan bobot berdasarkan *error* yang dihitung dari fungsi *loss*. Dalam tahap ini, algoritma optimasi seperti *Adam* digunakan untuk memperbarui bobot model dengan menghitung *gradient loss* terhadap parameter CNN. Dengan setiap iterasi pembaruan, model diharapkan dapat semakin baik dalam memprediksi *output* atau hasil yang benar.

Setelah proses pelatihan selesai, kinerja model akan dievaluasi menggunakan data validasi atau data uji untuk mengukur performanya dengan berbagai metrik, seperti akurasi, presisi, recall, F1-score. Sebagai contoh penerapan, jika kita ingin mendeteksi penyakit paru-paru menggunakan *dataset Chest X-ray8* (Xiaosong Wang dkk., 2017), CNN dapat dilatih untuk mengenali pola spesifik dari citra sinar-X dan dikalibrasi untuk memprediksi 8 kelas penyakit yang telah dilabeli sebelumnya. Selanjutnya, model akan diuji pada data baru untuk memvalidasi kemampuannya dalam mengenali pola yang relevan.

Penelitian terdahulu telah membuktikan keberhasilan metode CNN ini. Contohnya seperti penelitian yang dilakukan oleh (Krizhevsky dkk., 2012) yang menggunakan CNN untuk klasifikasi *dataset ImageNet*, serta penelitian (Xiaosong Wang dkk., 2017) yang memanfaatkan dataset *Chest X-ray8* untuk mendeteksi penyakit thorax dengan metode *supervised learning* berbasis CNN. Penerapan CNN dalam *supervised learning* juga merujuk pada prinsip dasar pembelajaran berbasis gradien yang dikembangkan dalam penelitian oleh (Lecun dkk., 1998), yang menunjukkan efektivitas algoritma ini dalam pengenalan pola pada berbagai jenis citra.

### 2.2.3 Densely Connected Convolutional Networks (DenseNet)

*Densely Connected Convolutional Networks* (DenseNet) adalah arsitektur *Convolutional Neural Network* (CNN) yang dikembangkan untuk mengatasi masalah hilangnya *gradient* (Gao Huang dkk., 2016; Kaiming dkk., 2015a; Srivastava dkk., 2015). DenseNet menghubungkan setiap lapisan/blok ke setiap lapisan lainnya dengan cara umpan maju, sehingga peta fitur/ciri dari semua lapisan sebelumnya digunakan sebagai *input* ke semua lapisan-lapisan berikutnya. Ini menetapkan  $\frac{l(l+1)}{2}$  koneksi langsung di DenseNet, dibandingkan dengan 1 koneksi antara lapisan dan lapisan sebelumnya di CNN tradisional (Gao Huang dkk., 2016; Khan dkk., 2019). DenseNet memiliki beberapa keunggulan menarik yaitu meringankan masalah gradien, memperkuat penyebaran fitur/ciri, mendorong penggunaan kembali fitur/ciri, dan secara substansial mengurangi jumlah parameter.



Gambar 2.3 Struktur DenseNet dengan 5 lapisan dengan 4 perluasan Setiap lapisan mengambil semua peta fitur/ciri sebelumnya sebagai masukan (Gao Huang dkk., 2016)

Pada Gambar 2.3 untuk setiap komposisi lapisan menggunakan *batch normalization*, *ReLU activation* dan *convolution* dengan filter 3x3. Pada setiap blok ada masukan berupa matriks sesuai dengan pixel citra kemudian masuk ke proses *batch normalization* untuk mengurangi adanya *overfitting* pada saat proses *training*, *ReLU activation* digunakan untuk mengubah nilai  $x$  menjadi 0 jika nilai  $x$  tersebut bernilai negatif, sedangkan sebaliknya untuk nilai  $x$  tetap dipertahankan apabila nilai tidak kurang dari 0, *convolution* dengan filter 3x3 proses citra matriks yang sudah dilakukan operasi *ReLU activation* akan dikalikan dengan matriks *convolution* dengan filter 3x3 dan keluaran yang dihasilkan berupa nilai matriks yang sudah di proses sebelumnya.

| Layers               | Output Size      | DenseNet-121                                                                                 | DenseNet-169                                                                                 | DenseNet-201                                                                                 | DenseNet-264                                                                                 |
|----------------------|------------------|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| Convolution          | $112 \times 112$ | $7 \times 7$ conv, stride 2                                                                  |                                                                                              |                                                                                              |                                                                                              |
| Pooling              | $56 \times 56$   | $3 \times 3$ max pool, stride 2                                                              |                                                                                              |                                                                                              |                                                                                              |
| Dense Block (1)      | $56 \times 56$   | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$  | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$  | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$  | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$  |
| Transition Layer (1) | $56 \times 56$   | $1 \times 1$ conv                                                                            |                                                                                              |                                                                                              |                                                                                              |
|                      | $28 \times 28$   | $2 \times 2$ average pool, stride 2                                                          |                                                                                              |                                                                                              |                                                                                              |
| Dense Block (2)      | $28 \times 28$   | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ |
| Transition Layer (2) | $28 \times 28$   | $1 \times 1$ conv                                                                            |                                                                                              |                                                                                              |                                                                                              |
|                      | $14 \times 14$   | $2 \times 2$ average pool, stride 2                                                          |                                                                                              |                                                                                              |                                                                                              |
| Dense Block (3)      | $14 \times 14$   | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$ |
| Transition Layer (3) | $14 \times 14$   | $1 \times 1$ conv                                                                            |                                                                                              |                                                                                              |                                                                                              |
|                      | $7 \times 7$     | $2 \times 2$ average pool, stride 2                                                          |                                                                                              |                                                                                              |                                                                                              |
| Dense Block (4)      | $7 \times 7$     | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$ |
| Classification Layer | $1 \times 1$     | $7 \times 7$ global average pool                                                             |                                                                                              |                                                                                              |                                                                                              |
|                      |                  | 1000D fully-connected, softmax                                                               |                                                                                              |                                                                                              |                                                                                              |

Gambar 2.4 Arsitektur DenseNet untuk ImageNet. Laju pertumbuhan untuk semua jaringan adalah  $k = 32$ . Perhatikan bahwa setiap lapisan “konvolusi” yang ditampilkan sesuai dengan urutan *BN-ReLU-Conv* (Gao Huang dkk., 2016)



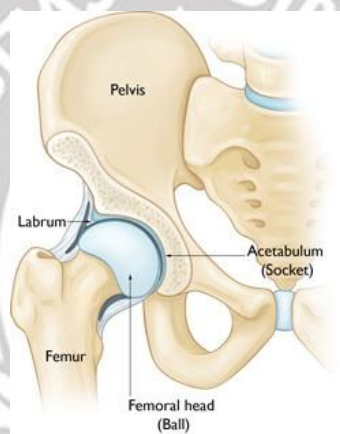
Penelitian yang dilakukan (Gao Huang dkk., 2016) menggunakan struktur DenseNet-BC dengan empat blok padat pada 224x224 citra masukan di ImageNet. Lapisan konvolusi pertama terdiri dari 2k konvolusi, ukuran 77 dengan langkah 2, dan jumlah peta fitur/ciri semua lapisan berikutnya ditentukan dengan menyetel k. Gambar 2.4 menampilkan konfigurasi jaringan persis yang diterapkan pada ImageNet.

DenseNet memiliki beberapa keunggulan dibandingkan dengan model CNN tradisional dan arsitektur lainnya, seperti ResNet. Salah satu keunggulannya yaitu pemanfaatan ulang fitur/ciri / *feature reuse*, dimana setiap lapisan berbagi informasi dengan semua lapisan sebelumnya, sehingga mengurangi kebutuhan untuk belajar ulang fitur/ciri yang sama. Selain itu, peningkatan propagasi gradien membuat pelatihan model menjadi lebih stabil, terutama untuk jaringan yang lebih dalam. Dengan jumlah parameter yang lebih sedikit dibandingkan ResNet, DenseNet menjadi lebih efisien dalam penggunaan parameter, karena fitur/ciri dapat digunakan kembali tanpa harus dipelajari ulang pada setiap lapisan (Gao Huang dkk., 2016). Keunggulan DenseNet lainnya yaitu regularisasi implisit, dimana pola koneksi yang rapat memberikan efek regulasi alami yang membantu mengurangi risiko *overfitting* pada dataset yang berukuran kecil.

Dalam bidang pengolahan citra medis, DenseNet telah banyak diterapkan dalam berbagai tugas seperti, deteksi dan klasifikasi penyakit, segmentasi citra medis, serta peningkatan kualitas citra. Contoh penelitian (Ausawalaithong dkk., 2018) menunjukkan bahwa DenseNet dapat digunakan untuk mendiagnosis kanker paru-paru berdasarkan citra *rontgen* dada dengan tingkat akurasi yang tinggi. Penelitian lainnya yang dilakukan oleh (Xiaoming dkk., 2017) menerapkan DenseNet dalam model H-DenseUNet untuk segmentasi hati dan tumor pada citra CT-Scan, menghasilkan peningkatan kinerja yang signifikan dibandingkan metode tradisional.

### 2.2.4 Struktur Anatomi Sendi Pinggul

Sendi pinggul atau *hip joint* adalah salah satu sendi utama tubuh manusia yang berperan penting dalam mendukung postur dan pergerakan. Sendi ini merupakan jenis sendi bola dan soket (*Ball-and-socket joint*) yang terdiri dari kepala femur (*femoral head*) berbentuk bola yang masuk ke dalam *acetabulum* pada *pelvis* yang dapat dilihat pada Gambar 2.5. Struktur ini memungkinkan Gerakan yang luas, termasuk fleksibel, ekstensi, abduksi, adduksi, serta rotasi internal dan eksternal (Gray\_s\_Anatomy, 2025).



Gambar 2.5 Struktur sendi pinggul (orthoinfo.aaos.org)

Struktur anatomi sendi pinggul yaitu sebagai berikut:

- a) Tulang penyusun:
  - *Acetabulum* merupakan cekungan pada tulang pinggul yang berfungsi sebagai soket bagi kepala femur.
  - Kepala femur (*head femur*) merupakan bagian bulat di ujung proksimal tulang paha yang masuk ke dalam *acetabulum*, membentuk sendi bola dan soket.
- b) Tulang rawan articular merupakan permukaan *acetabulum* dan kepala femur dilapisi oleh tulang rawan hialin yang halus, memungkinkan pergerakan sendi yang lancar dan mengurangi gesekan.
- c) *Labrum acetabular* merupakan struktur *fibrocartilaginous* yang mengelilingi tepi *acetabulum*, memperdalam soket dan meningkatkan stabilitas sendi pinggul.

- d) Kapsul sendi merupakan kapsul fibrosa kuat yang mengelilingi sendi pinggul, melekat dari tepi *acetabulum* hingga leher femur. Kapsul ini memberikan stabilitas tambahan pada sendi
- e) Ligamen:
- *Ligamentum Iliofemorale* merupakan ligament terkuat yang mencegah hiperekstensi sendi pinggul.
  - *Ligamentum Pubofemorale* merupakan ligamen yang membatasi abduksi dan ekstensi berlebihan.
  - *Ligamentum Ischiofemorale* merupakan ligament yang membantu membatasi rotasi internal dan stabilitas posterior.
  - *Ligamentum Capitis femoris* merupakan ligamen yang menghubungkan kepala femur dengan acetabulum dan membawa suplai darah kecil ke kepala femur.
- f) Otot dimana berbagai otot mengelilingi sendi pinggul, termasuk otot gluteal, iliopsoas, dan kelompok otot adduktor yang berperan dalam pergerakan dan stabilitas sendi.

### 2.2.5 Osteoarthritis

Osteoarthritis (OA) adalah penyakit yang berkaitan dengan persendian yang disebabkan oleh proses biologis kompleks yang melibatkan tulang rawan, tulang, otot, ligamen, sinovium, dan meniscus (Hinman dkk., 2023). Dalam penjelasan lain, osteoarthritis juga bisa diartikan sebagai penyakit sendi ortopedi yang paling umum terjadi pada orang tua, dan dapat mempengaruhi jutaan orang di seluruh dunia (Goekoop dkk., 2011). Salah satu ciri umum OA adalah terjadi nyeri pada bagian yang terkena OA (Veronese dkk., 2017). OA dapat mempengaruhi sendi lutut, pinggul, tangan dan bahu (Prieto Alhambra dkk., 2014; Tri- Wahyuningrum dkk., 2019). OA pinggul merupakan jenis OA kedua yang paling sering terjadi setelah lutut (Veronese dkk., 2017).

Ahli ortopedi telah menemukan bahwa pasien OA dengan gangguan, akan mengalami gaya berjalan lebih lambat daripada orang normal dengan langkah yang lebih pendek (Kaufman dkk., 2001). Tren Prevalensi *Osteoarthritis Site-Specific*

dari *Global Burden of Disease Study* 2019 menunjukkan bahwa selama periode sepuluh tahun, prevalensi kasus OA meningkat sebesar 113,25%, dari 247,51 juta pada tahun 1990 menjadi 527,81 juta pada tahun 2019 (Abbafati dkk., 2020; Cross dkk., 2014; Long dkk., 2022; Turkiewicz dkk., 2014). Kerentanan OA dapat meningkat sebagian karena warisan turun-temurun (riwayat keluarga yang positif meningkatkan risiko), usia, ras, faktor makanan, dan jenis kelamin wanita (Felson, 2004).

Istilah degeneratif pada OA tidak sepenuhnya akurat karena sendi yang terkena mengalami *remodeling* yang ditandai dengan pembentukan *osteofit* dan penampakan tulang subkondral yang sklerotik. Ini adalah proses destruksi dan *remodeling* yang dinamis serta mendefinisikan gambaran klasik osteoarthritis (Gerard Sheridan & James Cashman, 2019; Lespasio dkk., 2018). Sendi pinggul merupakan sendi yang dapat bergerak ke berbagai arah karena bentuknya yang menyerupai bola dalam mangkok atau lebih dikenal dengan sebutan *ball joint* (Byrne dkk., 2010; Sangeux, 2019). Karena adanya gerakan yang kompleks ini, dari sudut pandang biomekanik menyebabkan terjadinya proses gesekan antara dua permukaan tulang penyusun sendi panggul, sehingga seiring bertambahnya usia, terjadi proses degeneratif pada sendi pinggul yang lebih dikenal dengan istilah *primary hip osteoarthritis* (Dennison Harris, 2013; Sangeux, 2019). Sendi pinggul merupakan salah satu sendi yang menerima beban terbesar di tubuh, setelah sendi lutut sehingga paling banyak mengalami kasus osteoarthritis.

OA pinggul dapat dibagi menjadi dua kategori yaitu OA pinggul primer atau *primary hip Osteoarthritis* (PHOA) dan OA pinggul sekunder. OA pinggul primer adalah kondisi dimana tidak ditemukan kelainan anatomi atau proses penyakit spesifik yang dapat diidentifikasi. Sementara itu, OA pinggul sekunder disebabkan oleh berbagai lesi sendi seperti osteonekrosis kepala femoralis (*osteonecrosis of the femoral head* / ONFH), displasia perkembangan pinggul, *impingement femoroacetabular*, dan lain sebagainya (Wylie dkk., 2018).

Beberapa faktor risiko diketahui yang berhubungan dengan perkembangan *primary hip osteoarthritis* (PHOA). Pertama adalah usia: risiko OA meningkat seiring bertambahnya usia karena tulang rawan mengalami degenerasi alami seiring

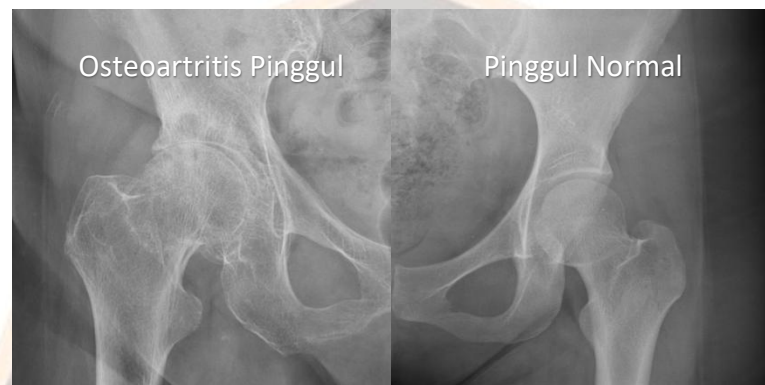


waktu. Kedua adalah faktor genetik: terdapat bukti bahwa faktor genetik dapat mempengaruhi kerentanan seseorang terhadap osteoarthritis. Ketiga adalah jenis kelamin: wanita cenderung memiliki resiko lebih tinggi mengalami OA pinggul dibandingkan pria. Ke empat adalah obesitas: kelebihan berat badan meningkatkan beban mekanis pada sendi pinggul dan mempercepat kerusakan tulang rawan. Kelima adalah faktor mekanis: cedera sebelumnya pada pinggul, kelainan struktural, dan aktivitas fisik yang berlebihan dapat mempengaruhi risiko osteoarthritis pinggul. Pasien dengan osteoarthritis pinggul umumnya mengalami nyeri yang berkembang selama berbulan-bulan hingga bertahun-tahun, bukan terjadi secara tiba-tiba. Pasien mengalami nyeri pangkal paha yang terus-menerus, awalnya sporadis, memburuk di penghujung hari, dan akan mempengaruhi aktivitas fisik, terutama saat berjalan atau menaiki tangga (Woolhead dkk., 2010).

Dalam analisis citra medis, khususnya radiografi pinggul, perbandingan antara citra pinggul dengan osteoarthritis dan pinggul normal menunjukkan perbedaan struktural yang jelas, baik secara morfologi maupun intensitas radiografis. Pada citra pinggul normal, permukaan sendi tampak halus, celah sendi antara *acetabulum* (rongga pada panggul) dan kepala *femur* terjaga dengan baik, serta tidak terlihat adanya pertumbuhan tulang tambahan *osteofit*. Struktur tulang *subkondral* juga tampak normal tanpa adanya peningkatan *densitas* atau *sklerosis* (Aresti dkk., 2016).

Sebaliknya, pada citra radiografi pinggul dengan osteoarthritis, terdapat sejumlah perubahan degeneratif khas yang mudah dikenali. Salah satu tanda utama adalah penyempitan celah sendi akibat hilangnya tulang rawan artikular, yang menjadi ciri khas OA. Selain itu, terdapat pertumbuhan *osteofit*, yaitu tonjolan tulang abnormal di tepi sendi yang tampak sebagai proyeksi radiopak pada citra X-ray. *Sklerosis subkondral*, yaitu penebalan dan peningkatan densitas tulang di bawah tulang rawan yang mengalami stres kronis, juga tampak jelas sebagai area lebih putih pada radiografi. Pada kasus lanjut, dapat pula terlihat kista subkondral (rongga kecil di tulang) dan *deformitas* bentuk kepala *femur* maupun *acetabulum* akibat *remodeling* tulang jangka panjang (Murphy dkk., 2016). Perbedaan lain yang penting adalah kontur permukaan sendi. Pada pinggul normal, kontur *acetabulum*

dan kepala *femur* halus dan bulat, sedangkan pada OA pinggul kontur ini menjadi kasar dan tidak teratur akibat erosi tulang rawan dan *remodeling* tulang. Citra OA pinggul juga sering menunjukkan penurunan jarak antar tulang, sedangkan pada citra normal jarak ini tetap lebar dan merata di seluruh permukaan sendi (Aresti dkk., 2016; Lespasio dkk., 2018; Sophia Fox dkk., 2009). Untuk perbandingan citra pinggul dengan osteoarthritis dan pinggul normal bisa dilihat pada Gambar 2.6



Gambar 2.6 Perbedaan osteoarthritis dengan pinggul normal

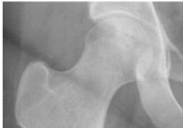
Radiologi polos merupakan metode paling sederhana, paling murah, dan paling umum digunakan untuk menilai osteoarthritis pinggul (Guerhazi dkk., 2009). Metode tersebut tidak dapat memvisualisasikan tulang rawan atau jaringan lunak, tetapi ruang sendi memberikan perkiraan tingkat kerusakan tulang rawan. Radiologi polos telah terbukti memiliki spesifitas yang tinggi (0,76 – 0,90) dan sensitivitas yang bervariasi (0,44 – 0,78) dibandingkan dengan pencitraan resonansi *magnetic* (MRI) dalam mendeteksi OA pinggul (Lixu dkk., 2013). OA pinggul sering kali dapat di diagnosis berdasarkan presentasi klinis, tapi pemeriksaan pencitraan medis melalui radiologi polos dapat berguna untuk memastikan diagnosis dan membantu perkembangan penyakit (Gambar 2.7 A-C) (Lespasio dkk., 2018; Murphy dkk., 2016).

SEKOLAH PASCASARJANA



Gambar 2.7 Radiografi pinggul anteroposterior. A – Penyempitan ruang sendi superomedial terlihat pada daerah yang diberi garis putih. B – Osteoarthritis tulang ke tulang yang ditunjukkan panah lebih tebal dengan pembentukan *osteofit* (panah lebih tipis). C – Osteoarthritis stadium akhir dengan deformasi kepala femoralis (bintang) dan pembentukan kista (lingkaran) (Lespasio dkk., 2018; Murphy dkk., 2016)

*Grade*/derajat atau tingkat keparahan osteoarthritis pinggul primer dapat dikelompokkan berdasarkan gambaran radiologis polos pinggul yang menggambarkan status kedua sendi pinggul yang akan dikelompokkan menggunakan klasifikasi *Kellgren-Lawrence*. Klasifikasi *Kellgren-Lawrence* membagi tingkat keparahan osteoarthritis menjadi lima derajat, yaitu derajat 0, derajat 1, derajat 2, derajat 3 dan derajat 4. Semakin tinggi derajat klasifikasi *Kellgren-Lawrence*, maka semakin parah kerusakannya (Gerard Sheridan & James Cashman, 2019; Lespasio dkk., 2018). Berikut ini untuk klasifikasi *Kellgren-Lawrence* yang dapat di lihat pada Gambar 2.8

|                                                                                     |                                                                                                      |                                                                                                                                             |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
|  | <b><u>Grade 0 – No OA</u></b>                                                                        |                                                                                                                                             |
|  | <b><u>Grade 1 – Doubtful OA</u></b><br>Doubtful joint space narrowing, possible osteophytic clipping |                                                         |
|  | <b><u>Grade 2 – Mild OA</u></b><br>Definite osteophytes, definite narrowing of joint space           |                                                         |
|                                                                                     |                                                                                                      | <b><u>Grade 3 – Moderate OA</u></b><br>Moderate, multiple osteophytes, narrowing of joint, sclerosis and possible deformity of bone contour |
|                                                                                     |                                                                                                      | <b><u>Grade 4 – Severe OA</u></b><br>Large osteophytes, marked narrowing of joint space, severe sclerosis and deformity of bone contour     |

Gambar 2.8 Klasifikasi osteoarthritis pinggul berdasarkan *Kellgren-Lawrence* (Gerard Sheridan & James Cashman, 2019)

Derajat 0 menunjukkan tidak adanya tanda-tanda osteoarthritis, di mana ruang sendi masih normal tanpa *osteofit* atau perubahan lainnya. Kemudian derajat 1 menggambarkan adanya *osteofit* kecil dengan ruang sendi yang masih terlihat normal dan sedikit ada perubahan awal pada struktur sendi, tetapi tidak ada gejala klinis yang signifikan. Derajat 2 menunjukkan adanya *osteofit* yang lebih jelas dengan adanya sedikit penyempitan ruang sendi, dan orang yang mengalami mungkin terasa nyeri ringan. Sedangkan untuk derajat 3 ditandai dengan penyempitan ruang sendi yang jelas, *osteofit* yang lebih besar, *sklerosis subkondral*, serta kemungkinan *deformitas* pada kepala femur atau *acetabulum*, dimana penderita sudah mulai mengalami nyeri dan keterbatasan gerak yang signifikan. Selanjutnya derajat 4 merupakan tahapan yang paling berat, dengan penyempitan ruang sendi yang hampir hilang, *osteofit* besar, *deformitas* sendi yang nyata, *sklerosis subkondral* yang lebih tebal, serta kemungkinan adanya kista subkondral. Pada tahap ini, pasien mengalami nyeri berat serta keterbatasan fungsi yang signifikan (Altman dkk., 1991). Terkait dengan evaluasi radiografi osteoarthritis pinggul menjelaskan bahwa penilaian *Kellgren-Lawrence* dapat disederhanakan menjadi 3 kategori yaitu osteoarthritis normal, ringan dan berat. Dimana untuk derajat 0 dan 1 menjadi tidak ada osteoarthritis, kemudian untuk derajat 2 dan 3 menjadi osteoarthritis ringan dan untuk derajat 4 menjadi osteoarthritis berat.

#### 2.2.6 Pengolahan Citra

Pengolahan citra (*image processing*) merupakan teknik yang digunakan untuk menganalisis, memanipulasi dan meningkatkan citra digital dengan tujuan memperoleh informasi tertentu atau meningkatkan kualitas visual citra (Gonzalez & Woods, 2018). Berdasarkan penjelasan lainnya, pengolahan citra digital adalah suatu teknik yang memproses data citra dengan bantuan komputer untuk menghasilkan citra baru atau informasi yang berguna dari citra tersebut. Pengolahan citra melibatkan transformasi citra dalam domain spasial maupun frekuensi. Menurut (Szeliski, 2011), pengolahan citra dapat digunakan untuk menganalisis pola, mendeteksi perubahan dan menyederhanakan kompleksitas data visual agar lebih mudah dipahami.



Pengolahan citra dapat dikelompokkan menjadi beberapa kategori berdasarkan tujuannya:

- 1) Pengolahan citra geometri: berfokus pada transformasi geometris citra, seperti rotasi, translasi, scaling dan deformasi. Hal ini berguna dalam pemrosesan citra berbasis peta atau citra medis.
- 2) Pengolahan citra radiometrik: berhubungan dengan manipulasi nilai intensitas piksel dalam citra, seperti pengaturan kontras, *histogram equalization*, dan perubahan warna.
- 3) Kompresi citra: bertujuan untuk mengurangi ukuran file citra tanpa mengorbankan kualitas signifikan, seperti pada *Joint Photographic Experts Group* (JPEG) atau *Portable Network Graphics* (PNG).
- 4) Restorasi citra: berfokus pada pemulihan citra dari kondisi rusak dengan teknik seperti *deblurring* atau *noise/derau removal*.
- 5) Pengolahan citra dalam domain frekuensi: transformasi fourier adalah salah satu metode yang digunakan untuk memproses citra dalam domain frekuensi. Teknik ini sering digunakan dalam kompresi citra dan deteksi fitur/ciri tertentu (Gonzalez & Woods, 2018).

Dalam pengolahan citra ada beberapa algoritma penting diantaranya yaitu sebagai berikut:

- 1) Filter linear dan *Non-linear*
  - Filter linear: seperti *convolutional* dengan kernel tertentu (*gaussian*, *sobel* atau *laplasian*)
  - Filter *non-linear*: seperti median filter untuk menghilangkan *noise/derau impulsive*.
- 2) Transformasi Histogram  
Histogram digunakan untuk menganalisis distribusi intensitas citra, dan manipulasi histogram (misal seperti, *histogram equalization*, *clahe*, *gamma*) dapat meningkatkan kualitas visual citra.
- 3) *Thresholding*/Pengembangan

Teknik ini membagi citra menjadi dua bagian (*foreground*/latar depan dan *background*/latar belakang) berdasarkan nilai ambang tertentu. *Thresholding*/Pengambangan sering digunakan dalam segmentasi objek

4) Deteksi tepi (*edge detection*)

Metode seperti *sobel*, *canny* dan *Laplacian* digunakan untuk mendeteksi batas-batas objek dalam citra. Deteksi tepi merupakan langkah awal penting dalam analisis citra lebih lanjut.

5) Segmentasi citra

Segmentasi membagi citra menjadi beberapa wilayah berdasarkan karakteristik tertentu, seperti intensitas, warna, atau tekstur. Metode ini digunakan dalam analisis medis dan sistem pemantauan visual (Zhaobin Wang dkk., 2020).

Dalam konteks pengolahan citra, identifikasi dapat dilakukan beberapa langkah utama:

- 1) Pra-pemrosesan: meningkatkan kualitas citra melalui teknik seperti normalisasi, *filtering*, atau penghapusan derau.
- 2) Segmentasi: memisahkan objek yang ingin diidentifikasi dari latar belakang.
- 3) Ekstraksi fitur/ciri: mengambil karakteristik penting dari citra, seperti tekstur, bentuk, warna, atau tepi.
- 4) Klasifikasi: Menggunakan model pembelajaran mesin atau metode statistik untuk menentukan kategori atau identitas objek dalam citra.

### 2.2.7 Peningkatan kontras Citra

Salah satu bidang terpenting dalam pengembangan teknik pengolahan citra yaitu peningkatan kontras citra. Ini secara efisien memberikan peningkatan visual citra dan efek ketajaman gambar yang bermanfaat untuk pemrosesan komputasi dan analisis lebih lanjut (Muttaqin dkk., 2024; Senthilkumaran & Thimmiaraja, 2014; Yang & Xuan, 2023). Menurut penjelasan lain peningkatan kontras citra adalah proses dalam pengolahan citra digital untuk memperbaiki perbedaan antara daerah terang dan gelap pada citra, sehingga detail pada gambar menjadi lebih jelas. Teknik ini sangat penting dalam berbagai aplikasi seperti medis, pengawasan dan

pengenalan pola, dimana visualisasi yang baik sangat dibutuhkan (Gonzalez & Woods, 2018).

Untuk meningkatkan kualitas citra dalam visi komputer, pengenalan pola, dan pemrosesan citra digital, peningkatan kontras sangat penting. Beberapa faktor, seperti peralatan pencitraan yang tidak memadai dan operator yang tidak berpengalaman, dapat menyebabkan rendahnya kontras pada video atau citra digital. Berkurangnya kualitas kontras juga dapat disebabkan oleh faktor lingkungan yang tidak mendukung citra yang di rekam, seperti kurangnya sinar matahari atau pencahayaan dalam ruangan dan hal lainnya (Chia Huang dkk., 2013).

#### **2.2.7.1 CLAHE (*Contrast Limited Adaptive Histogram Equalization*)**

CLAHE atau *Contrast Limited Adaptive Histogram Equalization* merupakan algoritma pengembangan dari AHE (*Adaptive Histogram Equalization*). CLAHE merupakan salah satu teknik yang digunakan untuk meningkatkan kontras citra terutama pada citra abu-abu (Honda dkk., 2021). Keunggulan dari metode CLAHE yaitu mampu meningkatkan citra yang lebih baik dari citra aslinya khususnya pada citra abu-abu yang sering di gunakan para medis untuk foto sinar-X atau *CT-Scan* yang mengalami banyak gangguan atau derau.

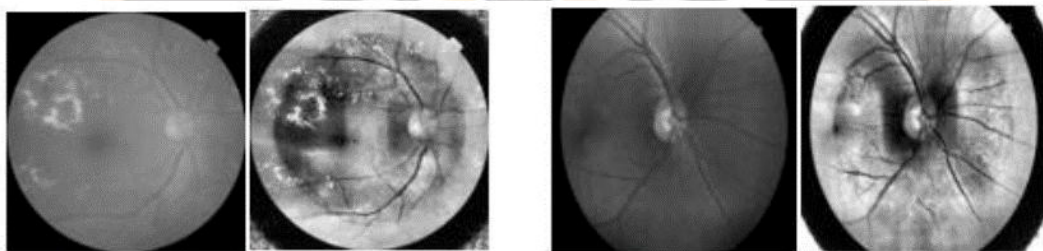
CLAHE membagi citra ke dalam bidang kontekstual (blok kecil) dan menerapkan pemerataan histogram secara lokal pada masing-masing bidang tersebut. Setiap bidang kontekstual dihitung berdasarkan histogram lokal yang kemudian dipangkas sesuai dengan batas yang ditentukan untuk mencegah peningkatan kontras yang berlebihan. Proses ini menghasilkan distribusi nilai abu-abu yang lebih merata, sehingga mampu memperjelas ciri-ciri tersembunyi dalam citra yang sebelumnya kurang terlihat (Pramunendar dkk., 2020).

Permasalahan berupa peningkatan kontras yang berlebihan pada metode AHE dapat diatasi dengan menggunakan *Contrast Limited Adaptive Histogram Equalization* (CLAHE) yang memberikan batasan nilai pada histogram. Proses CLAHE bertujuan untuk mendapatkan citra dengan kontras yang lebih baik tanpa mengurangi kualitas citra itu sendiri (Hayati dkk., 2022).

Metode CLAHE beroperasi pada area kecil pada citra yang biasa disebut ubin atau *tile*. Kontras yang terdapat pada setiap ubin dikoreksi sehingga histogram yang dihasilkan dari area tersebut sesuai dengan bentuk histogram yang ditentukan. Ubin yang berdekatan dihubungkan menggunakan interpolasi bilinear. Cara ini dilakukan agar hasil penggabungan ubin terlihat mulus. Metode CLAHE dapat didefinisikan dengan rumus sebagai berikut (Reza, 2004):

$$\beta = \frac{M}{N} \left( 1 + \frac{\alpha}{100} (S_{\max} - 1) \right) \quad (2.3)$$

Pada persamaan 2.3 dijelaskan bahwa  $\beta$  mewakili nilai batas (*clip limit*), kemudian variabel  $M$  mewakili ukuran area,  $N$  mewakili nilai *grey-level* atau *grayscale* (256),  $\alpha$  mewakili *clip factor* menyatakan penambahan batas histogram dengan nilai 1 sampai 100.  $S_{\max}$  merupakan kemiringan maksimum yang diizinkan. Berikut ini implementasi perbedaan citra sebelum dan sesudah menggunakan CLAHE dapat dilihat pada Gambar 2.9



Gambar 2.9 Perbedaan antara citra asli dengan citra yang telah ditingkatkan dengan CLAHE (Reza, 2004)

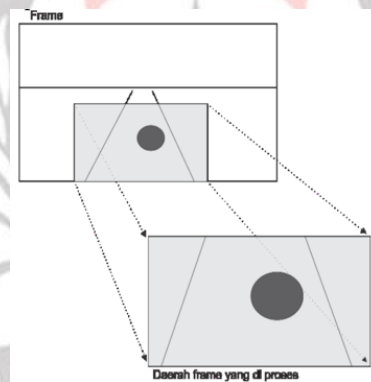
### 2.2.8 ROI (*Region of Interest*)

ROI (*Region of Interest*) merupakan cara yang tepat untuk mengurangi masalah tingginya waktu pemrosesan. ROI mampu memberikan tanda terhadap area tertentu sehingga dapat digunakan untuk mengoptimalkan kinerja sistem. Tanpa penggunaan ROI, pemrosesan pada citra dilakukan pada seluruh piksel citra tanpa terkecuali (Linda, 2010; Long dkk., 2015). ROI adalah salah satu teknik segmentasi sebagai proses pengolahan citra dimana pengguna mampu mengolah citra yang mengandung informasi data citra yang dikehendaki. ROI bekerja dalam

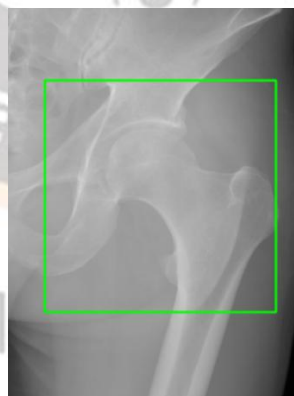


pengkodean secara berbeda pada area tertentu dari citra digital sehingga area citra yang lebih penting akan memiliki kualitas yang lebih baik dari area disekitarnya (Falah dkk., 2016).

ROI yang ditempatkan sebelum proses pengolahan citra berguna agar daerah yang dilakukan proses pengolahan citra berfokus pada frame ROI dan bukan pada seluruh frame (Pratomo dkk., 2020). Menurut (Doukas & Maglogiannis, 2007) dalam dunia medis, fungsi ROI sangat penting karena area tertentu pada citra memiliki nilai diagnostik lebih tinggi daripada area lainnya. Implementasi ROI pada frame dapat ditunjukkan pada Gambar 2.10 Kemudian Gambar 2.11 merupakan frame asli sebelum penambahan proses ROI dan pada Gambar 2.12 merupakan citra yang telah ditambahkan proses ROI.



Gambar 2.10 ROI pada frame (Pratomo dkk., 2020)



Gambar 2.11 Frame asli sebelum ROI



Gambar 2.12 Frame setelah penambahan ROI

### 2.2.9 Matriks Konfusi (*Confusion Matrix*)

*Confusion matrix* adalah tabel khusus yang digunakan dalam bidang pembelajaran mesin untuk tujuan mendeskripsikan dan mengevaluasi kinerja model klasifikasi seperti jaringan syaraf tiruan dengan membandingkan hasil prediksi model dengan nilai yang sebenarnya (Fahmy Amin, 2023). Untuk pengujian kinerja model diukur menggunakan berbagai ukuran seperti *accuracy*, *precision*, *recall* dan *F1-Score*. Empat istilah yang digunakan pada *confusion matrix* adalah pertama, Positif Benar (*True Positif – TP*) merupakan kondisi dimana model prediksi data ada dikelas positif dan yang sebenarnya data memang berada di kelas positif. Kedua, Negatif Benar (*True Negatif – TN*) merupakan kondisi dimana model memprediksi data ada di kelas negatif dan sebenarnya data memang berada di kelas negatif. Ketiga, Positif Salah (*False Positif – FP*) merupakan kondisi dimana model memprediksi data berada di kelas positif, tetapi sebenarnya data ada di kelas negatif. Keempat, Negatif Salah (*False Negatif*) merupakan kondisi dimana model memprediksi data ada di kelas negatif, namun yang sebenarnya data ada di kelas positif.

- a) *Accuracy* merupakan *metric* yang digunakan untuk menghitung rasio prediksi benar (positif dan negatif) dengan keseluruhan data saat melakukan pelatihan. Rumus *Metric accuracy* bisa dilihat pada persamaan (2.4)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.4)$$

- b) *Precision* digunakan untuk menghitung presisi atau data prediksi positif dari keseluruhan hasil yang diprediksi positif. Rumus *Precision* bisa dilihat pada persamaan (2.5)

$$Precision = \frac{TP}{TP+FP} \quad (2.5)$$

- c) *Recall* = *sensitivitas* digunakan untuk menghitung jumlah prediksi pada kelas yang sama. Secara definisi *recall* adalah perbandingan antara *True Positif* (TP) dengan banyaknya data yang sebenarnya positif. Rumus *Recall* bisa dilihat pada persamaan (2.6)

$$Recall = \frac{TP}{TP+FN} \quad (2.6)$$

- d) *F1-Score* merupakan harmonic-mean atau perbandingan rata-rata *precision* dan *recall* yang dibobotkan. Rentang nilai *F-1 Score* adalah 0 sampai 1 dengan nilai terburuk adalah 0 dan nilai terbaik adalah 1. *F1-Score* mengindikasikan bahwa model klasifikasi memiliki *precision* dan *recall* yang baik. Rumus *F1-Score* bisa dilihat pada persamaan (2.7)

$$F1-Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (2.7)$$

#### 2.2.10 SSIM (*Structured Similarity Index Measure*)

SSIM (Sara dkk., 2019) merupakan metrik yang diukur menggunakan tiga aspek utama yaitu *luminance*, *contrast* dan *structural* yang ditunjukkan pada persamaan (2.8) berikut ini:

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (2.8)$$

Pada persamaan (2.13), *l* adalah luminansi yang digunakan untuk membandingkan kecerahan antara dua citra. *c* adalah kontras yang digunakan untuk membedakan

rentang antara wilayah paling terang dan paling gelap dari dua citra.  $s$  adalah struktur yang digunakan untuk membandingkan pola luminasi lokal antara dua citra untuk menemukan kesamaan dan ketidaksamaan citra. Kemudian,  $\alpha$ ,  $\beta$  dan  $\gamma$  adalah konstanta positif. Luminansi, kontras dan struktur citra dapat dinyatakan secara terpisah seperti persamaan (2.9), (2.10) dan (2.11)

$$l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \quad (2.9)$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \quad (2.10)$$

$$s(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_x \sigma_y + c_3} \quad (2.11)$$

Dimana  $\mu_x$  dan  $\mu_y$  adalah rata-rata lokal,  $\sigma_x$  dan  $\sigma_y$  adalah deviasi standar, dan  $\sigma_{xy}$  adalah kovariansi silang untuk citra  $x$  dan  $y$  secara berurutan. Jika  $\alpha = \beta = \gamma = 1$ , maka indeks dapat disederhanakan menggunakan persamaan (2.14) - (2.16) dan hasilnya bisa dilihat pada persamaan (2.12).

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_x\sigma_y + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (2.12)$$

### 2.2.11 Entropi

Entropi merupakan ukuran kompleksitas atau keragaman informasi dalam citra. Semakin tinggi nilai entropi, semakin kompleks dan beragam informasi yang ada pada citra tersebut. Entropi dihitung berdasarkan distribusi intensitas piksel di dalam citra dan digunakan untuk menilai kualitas citra atau untuk pengenalan pola dan segmentasi (Gonzalez & Woods, 2018). Berikut ini rumus terkait entropi:

$$H = - \sum_{i=1}^n p(i) \log p(i) \quad (2.13)$$

Pada persamaan (2.13),  $H$  adalah entropi.  $p(i)$  adalah probabilitas kemunculan intensitas piksel  $i$ .  $n$  adalah jumlah level intensitas dalam citra.



Makna entropi memiliki implikasi luas dalam berbagai aspek pengolahan citra. Salah satu implikasinya yaitu sebagai ukuran kualitas dan kekayaan detail citra. Citra dengan nilai entropi tinggi cenderung memiliki lebih banyak detail, variasi tekstur, serta informasi yang sangat kompleks, dimana hal ini sangat berguna dalam analisis citra medis, seperti identifikasi kelainan pada *sinar-X*, *MRI* atau *CT-scan* (Shannon, 1948). Dalam proses segmentasi dan analisis tekstur, entropi digunakan untuk mengidentifikasi daerah dengan kompleksitas tinggi dibandingkan dengan daerah yang lebih seragam. Salah satu metode yang memanfaatkan konsep ini yaitu *thresholding*/pengembangan berbasis entropi, yang meningkatkan nilai akurasi segmentasi (Kapur dkk., 1985). Dalam pengolahan citra medis, entropi juga digunakan untuk meningkatkan kontras serta mendeteksi struktur abnormal seperti tumor atau jaringan patologis, yang dapat membantu dalam diagnosis berbasis pencitraan medis (Xiaomeng dkk., 2017).

Dalam bidang kriptografi dan keamanan citra, entropi menjadi faktor penting karena citra dengan entropi tinggi sulit untuk dikompresi atau diprediksi, sehingga lebih aman dalam sistem enkripsi (Pareek dkk., 2006). Dengan demikian, entropi merupakan ukuran fundamental yang mencerminkan kompleksitas dan keberagaman informasi dalam suatu citra. Penerapannya dalam berbagai bidang, seperti kompresi, segmentasi, dan analisis tekstur, serta keamanan, menjadikannya alat yang kuat dalam pemrosesan citra digital, termasuk dalam pengolahan citra medis.

#### **2.2.12 CNR (*Contrast-to-Noise Ratio*)**

CNR merupakan ukuran yang digunakan untuk mengevaluasi seberapa baik perbedaan antara area kontras tinggi dan rendah pada citra dapat dibedakan di tengah keberadaan derau dalam citra tersebut. Dalam pemrosesan citra, CNR membantu untuk mengetahui apakah suatu fitur/ciri atau detail di dalam citra dapat dilihat dengan jelas tanpa terganggu oleh derau. CNR yang lebih tinggi menandakan bahwa fitur/ciri pada gambar lebih kontras dengan latar belakang dan lebih mudah untuk dilihat (Igarashi dkk., 2024). Rumus dasar CNR yaitu sebagai berikut:

$$CNR = \frac{|S1-S2|}{\sigma} \quad (2.14)$$

Pada persamaan (2.14),  $S1$  dan  $S2$  adalah intensitas rata-rata di dua area yang berbeda ( $S1$  adalah fitur/ciri yang ingin di lihat dan  $S2$  adalah area latar belakang). Sedangkan untuk  $\sigma$  merupakan simpangan baku dari derau dalam citra.

### 2.2.13 Bayesian Optimization

*Bayesian Optimization* (BO) merupakan metode optimasi berbasis model probabilistik yang digunakan untuk mencari solusi optimal dari fungsi objektif yang besar untuk dievaluasi. BO secara khusus berguna dalam kasus di mana evaluasi langsung dari fungsi objektif membutuhkan waktu atau sumber daya yang signifikan, seperti dalam pengoptimalan *hyperparameter* model pembelajaran mesin (Shahriari dkk., 2016). BO bekerja dengan membangun model probabilistik dari fungsi objektif menggunakan proses Gaussian (*Gaussian Process/GP*) atau model regresi berbasis kernel lainnya. Model ini kemudian digunakan untuk menentukan titik evaluasi berikutnya dengan memaksimalkan suatu fungsi akuisisi (Frazier, 2018).

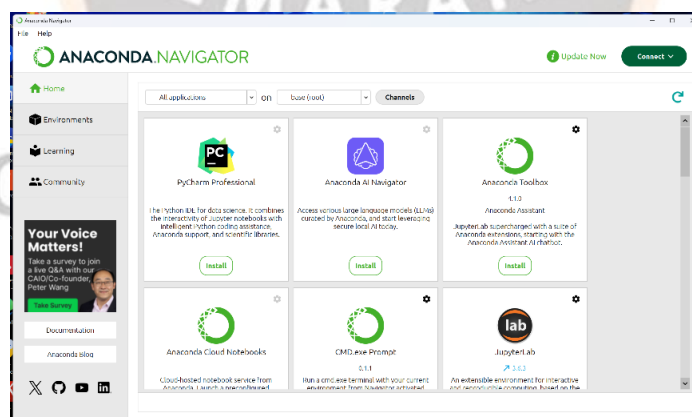
BO memiliki dua komponen utama yaitu *Surrogate Model* dan Fungsi Akuisisi. *Surrogate* model biasanya menggunakan *Gaussian Process Regression* (GPR) untuk membentuk perkiraan distribusi fungsi objektif berdasarkan pengamatan sebelumnya (Rasmussen & Williams, 2006). Fungsi Akuisisi digunakan untuk memilih titik selanjutnya yang akan dievaluasi. Fungsi ini menyeimbangkan eksplorasi (mencari daerah baru) dan eksploitasi (memanfaatkan daerah dengan nilai terbaik saat ini) (Snoek dkk., 2012). Proses BO dimulai dengan menentukan domain pencarian dan fungsi objektif yang akan dioptimalkan. Selanjutnya memilih beberapa titik awal secara acak dan evaluasi fungsi objektifnya. Setelah itu bangun model probabilistik berdasarkan data yang telah diperoleh. Kemudian dilanjutkan dengan menentukan titik berikutnya menggunakan fungsi akuisisi. Selanjutnya melakukan evaluasi fungsi objektif pada

titik yang dipilih dan perbarui model. Terakhir ulangi proses ini sampai kondisi penghentian terpenuhi (misalnya, batas iterasi atau konvergensi telah tercapai).

Keunggulan dari BO yaitu memanfaatkan informasi dari evaluasi sebelumnya untuk membuat estimasi yang lebih baik. BO telah banyak digunakan dalam berbagai hal misalnya seperti pengoptimalan *hyperparameter* dalam pembelajaran mesin (Snoek dkk., 2012), desain eksperimen dalam kimia dan biologi komputasional (Frazier, 2018). BO merupakan pendekatan optimasi berbasis model yang sangat efektif dalam kasus di mana evaluasi fungsi objektif besar dan kompleks. Dengan menggunakan model probabilistik seperti *Gaussian Process*, BO dapat secara efisien memilih titik evaluasi baru, mempercepat proses pencarian solusi optimal dibandingkan dengan metode optimasi tradisional seperti *grid search* atau *random search*.

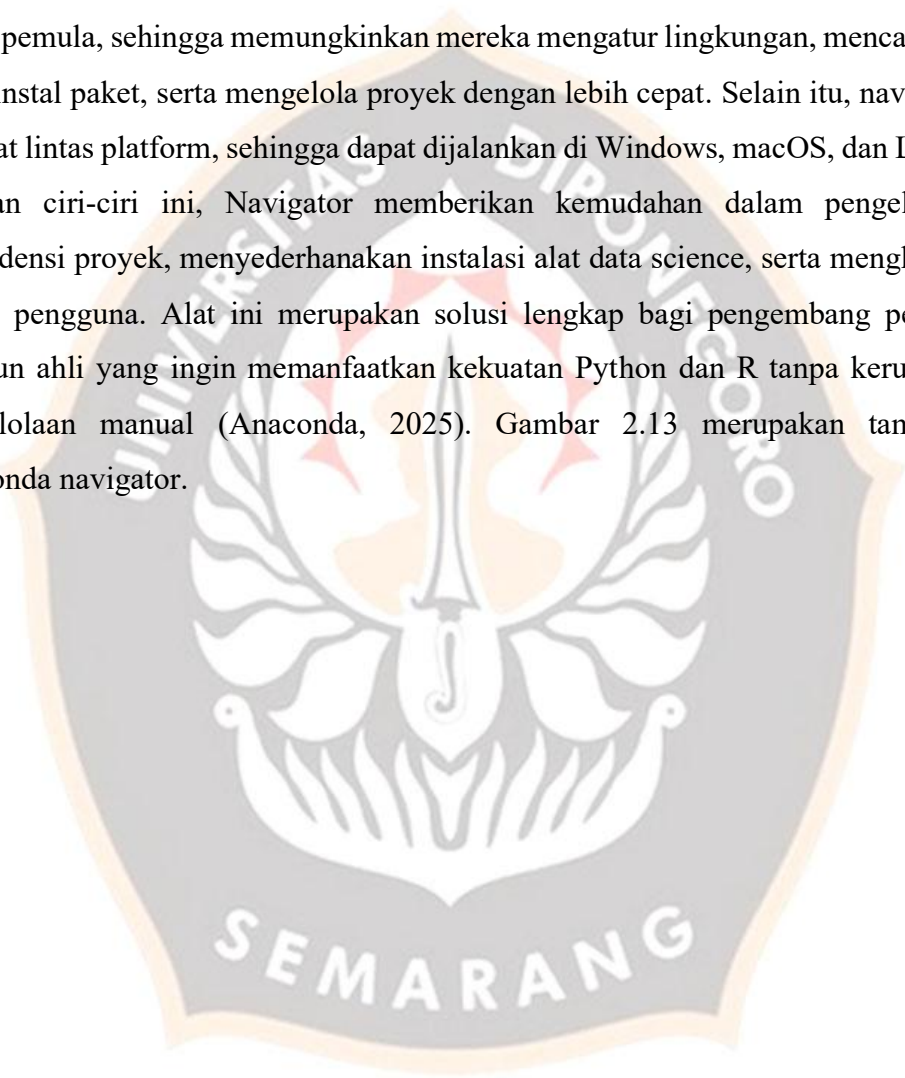
#### 2.2.14 Anaconda Navigator

Anaconda Navigator merupakan *Graphical User Interface* (GUI) yang dirancang untuk mempermudah manajemen lingkungan pengembangan berbasis Python dan R. Navigator merupakan bagian dari distribusi Anaconda, yang sering digunakan oleh pengembang dan peneliti di bidang *data science*, *machine learning*, *deep learning* dan analisis data. Dengan fitur/ciri utama seperti manajemen paket dan lingkungan, Navigator memungkinkan pengguna untuk membuat, mengelola, dan menghapus lingkungan virtual sehingga proyek dengan dependensi berbeda dapat berjalan tanpa konflik. Navigator mendukung pengelolaan paket Python dan R melalui Conda tanpa memerlukan perintah terminal (Anaconda, 2025).



Gambar 2.13 Anaconda Navigator

Selain itu, Anaconda Navigator menyediakan akses mudah ke berbagai *tools* populer lain seperti Jupyter Notebook, JupyterLab, Spyder, RStudio, dan VS Code, yang dapat digunakan dalam pengembangan proyek *data science*. Melalui antarmuka ini, pengguna dapat menginstal dan memperbarui paket atau alat dengan mudah. Antarmuka yang ramah pengguna menjadikan Navigator sangat cocok untuk pemula, sehingga memungkinkan mereka mengatur lingkungan, mencari dan menginstal paket, serta mengelola proyek dengan lebih cepat. Selain itu, navigator bersifat lintas platform, sehingga dapat dijalankan di Windows, macOS, dan Linux. Dengan ciri-ciri ini, Navigator memberikan kemudahan dalam pengelolaan dependensi proyek, menyederhanakan instalasi alat data science, serta menghemat waktu pengguna. Alat ini merupakan solusi lengkap bagi pengembang pemula maupun ahli yang ingin memanfaatkan kekuatan Python dan R tanpa kerumitan pengelolaan manual (Anaconda, 2025). Gambar 2.13 merupakan tampilan Anaconda navigator.



SEKOLAH PASCASARJANA