

# Optimasi Proses Diagnosa Medis pada Analisis Citra MRI melalui Pendekatan Divide and Conquer

Dita Maheswari - 13523125

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: [ditamaheswari05@gmail.com](mailto:ditamaheswari05@gmail.com) , [13523125@std.stei.itb.ac.id](mailto:13523125@std.stei.itb.ac.id)

**Abstrak**—Diagnosis medis yang akurat dan cepat sangat penting dalam perawatan kesehatan modern, terutama dalam analisis citra *Magnetic Resonance Imaging* (MRI) yang menghasilkan volume data besar. Tantangan dalam menginterpretasi citra MRI secara manual seringkali menyebabkan keterlambatan diagnosis dan variabilitas interpretasi. Makalah ini mengusulkan optimasi proses diagnosis medis pada analisis citra MRI melalui pendekatan *Divide and Conquer* (DAC). Metodologi yang diusulkan mencakup pra-pemrosesan citra, pemecahan citra MRI menjadi sub-bagian yang lebih kecil menggunakan strategi pembagian seragam atau adaptif, pemrosesan paralel setiap sub-bagian menggunakan algoritma seperti K-Means untuk segmentasi, GLCM untuk ekstraksi fitur tekstur, dan SVM untuk klasifikasi, serta penggabungan hasil secara cerdas. Analisis kompleksitas waktu menunjukkan bahwa pendekatan DAC dapat mengurangi kompleksitas komputasi secara signifikan dari  $O(N^2)$  menjadi  $O(N \log N)$  untuk masalah tertentu, dan secara substansial mempercepat proses diagnosis melalui paralelisasi. Keunggulan ini menjadikan DAC sangat menjanjikan untuk meningkatkan efisiensi, akurasi, dan skalabilitas sistem diagnosis MRI, meskipun memerlukan desain yang cermat dalam penanganan *overhead* dan penggabungan hasil.

**Kata Kunci**—*Divide and Conquer*, MRI, Diagnosis Medis, Optimasi, Pemrosesan Citra, Segmentasi, Ekstraksi Fitur, Klasifikasi.

## I. PENDAHULUAN

Saat ini, teknologi di bidang kesehatan sudah sangatlah canggih dan cukup akurat dalam mengambil hasil keputusan medis. Diagnosis medis yang tepat dan cepat merupakan dasar penting dalam memberikan layanan kesehatan yang efisien dan tepat waktu. Pada zaman kedokteran kontemporer, teknologi pencitraan medis telah menjadi pondasi dalam proses diagnosis. Salah satu teknik pencitraan yang paling efektif dan informatif adalah *Magnetic Resonance Imaging* (MRI). Alat ini lebih akurat daripada CT Scan yang pernah penulis bahas dalam makalah Aljabar Linier dan Geometri. MRI dapat menghasilkan hasil gambar rinci dari organ, jaringan lunak, tulang, dan hampir seluruh bagian internal tubuh, tanpa memerlukan radiasi pengion[1], yaitu jenis radiasi yang memiliki energi cukup untuk melepaskan elektron dari atom atau molekul dan mengubahnya menjadi ion[2]. Kemampuan ini membuatnya sangat penting untuk mengidentifikasi berbagai kondisi patologis, seperti tumor

otak, lesi pada sumsum tulang belakang, kelainan sendi, serta gangguan vaskular dan neurologis, seringkali pada tahap awal perkembangannya.

Namun, kemampuan luar biasa dari MRI dalam menghasilkan detail juga menghadirkan tantangan besar, yaitu volume data yang sangat besar dan rumit. Satu pemeriksaan MRI dapat menghasilkan ratusan hingga ribuan gambar dua dimensi atau tiga dimensi. Proses manual pemeriksaan dan analisis gambar-gambar ini oleh radiolog atau dokter memerlukan waktu yang sangat lama, keterampilan tinggi, dan fokus terus-menerus. Ini tidak hanya menambah waktu tunggu pasien untuk mendapatkan diagnosis, tetapi juga rentan terhadap kelelahan manusia serta variasi interpretasi di antara peneliti. Variabilitas ini dapat mempengaruhi konsistensi dan ketepatan diagnosis, yang pada akhirnya berpengaruh pada keputusan klinis dan hasil perawatan pasien.

Dengan mempertimbangkan tantangan ini, pengoptimalan proses diagnosis dalam analisis citra MRI menjadi sangat krusial. Optimasi di sini melibatkan usaha untuk mempercepat analisis, meningkatkan ketepatan, dan menjaga konsistensi diagnosis, sembari tetap mempertahankan atau bahkan meningkatkan kualitas interpretasi. Beragam metode komputasi dan teknik pengolahan citra digital telah diteliti untuk mengotomatisasi sebagian atau keseluruhan proses analisis citra MRI. Ini mencakup metode segmentasi gambar untuk memisahkan struktur tertentu, klasifikasi jaringan untuk mengidentifikasi daerah yang tidak normal, dan deteksi objek untuk menemukan anomali tertentu[3][4].

Salah satu paradigma algoritma yang memiliki potensi besar untuk menangani kompleksitas komputasi dalam analisis gambar MRI adalah algoritma *Divide and Conquer*. Strategi *Divide and Conquer* merupakan metode desain algoritma yang membagi masalah besar dan rumit menjadi beberapa sub-masalah yang lebih kecil, terpisah, dan lebih mudah untuk diatasi. Setiap sub-masalah selesai, solusi-solusi dari sub-masalah tersebut digabungkan lagi untuk membentuk solusi akhir dari masalah yang asli[5]. Dalam analisis citra MRI, pendekatan algoritma *Divide and Conquer* dapat diterapkan dengan memecah citra MRI yang besar menjadi beberapa blok atau area yang lebih kecil. Setiap blok ini dapat diproses dan dianalisis secara bersamaan, contohnya untuk segmentasi tumor, deteksi lesi mikroskopis, atau ekstraksi ciri tekstur menggunakan algoritma yang dioptimalkan untuk

ukuran yang lebih kecil. Setelah menganalisis setiap blok secara individual, hasil dari semua blok itu akan digabungkan untuk menghasilkan diagnosis menyeluruh dari seluruh citra MRI.

Penerapan Divide and Conquer dalam analisis citra MRI berpotensi membawa beberapa keuntungan, yaitu:

1. Peningkatan Efisiensi Komputasi

Dengan memproses sub-masalah secara paralel, total waktu komputasi dapat berkurang secara drastis, terutama pada sistem terdistribusi

2. Skalabilitas

Pendekatan ini dapat dengan mudah diskalakan untuk menangani dataset citra yang sangat besar, yang merupakan karakteristik umum dalam pencitraan medis

3. Pengurangan Kompleksitas Masalah

Menganalisis bagian-bagian kecil dari citra secara terpisah dapat menyederhanakan algoritma pemrosesan yang dibutuhkan dan mengurangi kebutuhan memori.

Meskipun demikian, implementasi algoritma Divide and Conquer juga memiliki tantangan, seperti bagaimana menentukan pembagian yang optimal, bagaimana menangani overlap antar sub-masalah jika ada, dan bagaimana mengintegrasikan hasil dari setiap sub-masalah secara koheren dan akurat. Makalah ini akan membahas bagaimana pendekatan algoritma Divide and Conquer dapat dirancang dan dioptimalkan untuk meningkatkan performa proses diagnosis medis pada analisis citra MRI.

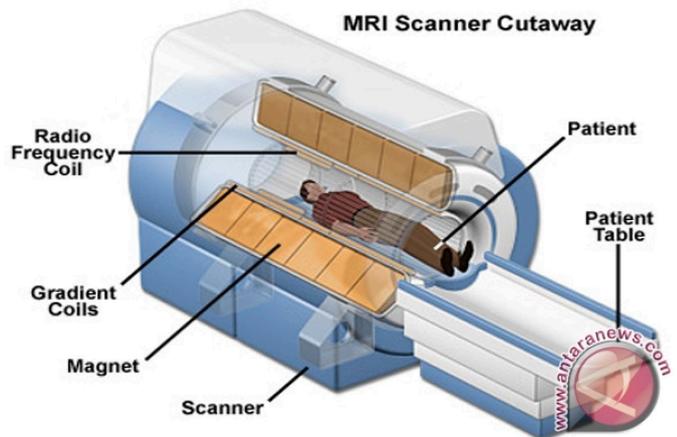
## II. LANDASAN TEORI

### A. Pencitraan Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) merupakan metode pencitraan medis tanpa invasif yang menciptakan gambar dengan rinci dari organ, jaringan lunak, tulang, dan hampir seluruh struktur dalam tubuh. Tidak seperti X-Ray atau CT Scan yang memanfaatkan radiasi pengion, MRI menggunakan medan magnet yang kuat dan gelombang radio untuk menciptakan citra atau gambar [1]. Prinsip mendasar MRI melibatkan hubungan antara medan magnet eksternal yang kuat dan inti atom hidrogen (proton) yang banyak terdapat dalam molekul air di jaringan tubuh.

Saat pasien berada dalam medan magnet MRI, proton-proton di dalam tubuh akan sejalan dengan medan magnet itu, Gelombang radio dipancarkan dan diarahkan untuk memindahkan sementara posisi proton, Setelah pulsa gelombang radio dinonaktifkan, proton akan kembali ke posisi sejajarnya, melepaskan energi dalam bentuk sinyal radio frekuensi yang khas. Gelombang-gelombang ini terdeteksi oleh koil MRI dan selanjutnya diproses oleh komputer untuk menciptakan citra penampang melintang tubuh dengan resolusi tinggi[3].

MRI sangat penting dalam dunia kesehatan karena kemampuannya dalam membedakan berbagai tipe jaringan lunak dengan sangat baik, yang sulit dicapai oleh metode pencitraan lainnya. Ini menjadikannya pilihan utama untuk mendiagnosis kondisi yang melibatkan otak, sumsum tulang belakang, sendi (lutut, bahu, pergelangan tangan, dsb), organ abdomen, serta sistem vaskular[4]. Walaupun begitu, data yang diperoleh dari MRI seringkali berukuran besar dan kompleks, terdiri dari serangkaian gambar dua dimensi atau volume tiga dimensi, yang memerlukan analisis teliti oleh radiolog.



Gambar 2.1 Magnetic Resonance Imaging (<https://images.app.goo.gl/urmKB88cjPoywRCc9>)

### B. Optimasi dalam Diagnosis Medis

Optimasi dalam konteks diagnosis medis berkaitan dengan usaha untuk meningkatkan efektivitas, ketepatan, dan kecepatan proses diagnostik, sering kali memanfaatkan teknologi komputer. Tujuan utamanya adalah untuk mengurangi tekanan pada tenaga kesehatan, meminimalisir kesalahan dalam interpretasi, serta mempercepat proses diagnosis agar penanganan pasien bisa dilakukan dengan lebih cepat dan akurat[5]. Dalam analisis citra medis, pengoptimalan dapat mencakup berbagai aspek, yaitu:

1. Efisiensi Waktu

Mempersingkat durasi yang diperlukan untuk memproses dan menganalisis gambar, dari pengambilan hingga pelaporan diagnosis.

2. Akurasi Diagnosa

Meningkatkan presisi dalam menemukan dan mengenali anomali, serta membedakan antara keadaan normal dan patologis.

3. Konsistensi Interpretasi

Mengurangi variabilitas dalam interpretasi gambar antara radiolog atau dalam berbagai waktu, menjamin standar diagnosis yang konsisten.

4. Pengurangan Beban Kognitif

Membantu radiolog dengan mengotomatisasi tugas-tugas yang sering dilakukan atau menyediakan alat bantu yang menonjolkan area mencurigakan, sehingga mereka dapat berkonsentrasi pada aspek diagnosis yang lebih rumit.

Optimasi ini seringkali dicapai melalui penerapan teknik-teknik pemrosesan citra digital dan kecerdasan buatan, seperti segmentasi otomatis, deteksi objek, klasifikasi, dan pembelajaran mesin. Tujuannya adalah untuk mengubah data citra mentah menjadi informasi diagnostik yang lebih terstruktur dan mudah.

### C. Pemrosesan Citra Digital

Pemrosesan Citra Digital (PCD) merupakan area yang menitikberatkan pada pengolahan citra digital dengan memanfaatkan algoritma komputer[6]. Dalam bidang medis, PCD memiliki peran penting dalam meningkatkan kualitas gambar, mengambil informasi relevan, dan mendukung analisis diagnostik. Langkah-langkah umum dalam PCD adalah sebagai berikut:

#### 1. Peningkatan Citra (*Image Enhancement*)

Metode yang digunakan untuk memperjelas citra agar lebih mudah dipahami oleh mata manusia atau sistem komputer, seperti penyesuaian kontras, pengurangan noise, atau peningkatan ketajaman tepi[7].

#### 2. Restorasi Citra (*Image Restoration*)

Proses untuk mengembalikan gambar yang telah rusak atau menurun ke kondisi semula, sering kali melibatkan model matematis dari kerusakan citra[6].

#### 3. Segmentasi Citra

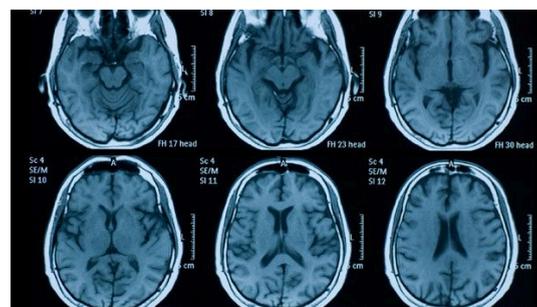
Memisahkan citra menjadi beberapa segmen atau objek yang tidak saling tumpang tindih dan memiliki makna. Pada citra MRI, segmentasi biasanya dipakai untuk memisahkan organ, tumor, atau struktur jaringan tertentu dari latar belakang[8]. Algoritma segmentasi dapat berbeda-beda, mulai dari *thresholding*, metode *edge-based*, pendekatan berbasis wilayah (*region-based*), hingga algoritma *machine learning* seperti K-Means atau jaringan saraf konvolusional [1],[9].

#### 4. Ekstraksi Fitur (*Feature Extraction*)

Menentukan dan mengevaluasi atribut penting dari objek atau area dalam gambar. Fitur dapat terdiri dari tekstur (contohnya *Gray Level Co-occurrence Matrix*), bentuk, ukuran, atau tingkat intensitas piksel. Fitur-fitur ini selanjutnya dapat dimanfaatkan untuk klasifikasi atau analisis lanjutan[10].

#### 5. Klasifikasi Citra (*Image Classification*)

Mengklasifikasikan piksel atau area citra ke dalam kelompok sesuai dengan fitur-fitur yang diambil. Contohnya, mengkategorikan area sebagai jaringan yang sehat, tumor jinak, atau tumor ganas[11]. Metode klasifikasi dapat melibatkan *Support Vector Machine* (SVM), *Artificial Neural Network* (ANN), atau algoritma *deep learning* [3],[10].



Gambar 2.2 Hasil MRI Bagian Otak

(<https://itsbrain.wordpress.com/wp-content/uploads/2019/06/mri.jpg>)

### D. Algoritma Divide and Conquer

Pendekatan algoritma divide and conquer merupakan salah satu paradigma dasar desain algoritma yang paling efisien, mendasar, dan fleksibel dalam bidang komputer. Paradigma ini menawarkan struktur yang kokoh untuk menangani masalah komputasi yang rumit dengan merubahnya menjadi serangkaian masalah yang lebih sederhana[12]. Inti dari filosofi adalah bahwa masalah yang kompleks sering kali bisa diselesaikan dengan lebih baik jika dipecah menjadi bagian-bagian yang lebih kecil. Umumnya, algoritma Divide and Conquer dibagi menjadi tiga langkah utama yang diulang secara rekursif:

#### 1. Divide

Langkah pertama ini mencakup pemecahan masalah utama yang besar menjadi dua atau lebih sub-masalah yang lebih kecil. Kunci keberhasilan di sini adalah memastikan bahwa sub-masalah yang terbentuk memiliki karakteristik yang serupa dengan masalah utama, tetapi dalam ukuran yang lebih kecil. Secara ideal, sub-masalah ini seharusnya independen satu sama lain, atau paling tidak memiliki ketergantungan yang rendah, sehingga dapat diproses secara terpisah. Pembagian ini terjadi secara rekursif sampai sub-masalah mencapai ukuran tertentu yang dikenal sebagai kasus dasar (*base case*). Kasus dasar merupakan kondisi di mana sub-masalah sudah cukup mudah sehingga bisa diselesaikan langsung tanpa memerlukan pemecahan tambahan [5]. Contohnya, dalam konteks citra MRI, sebuah citra volume 3D yang sangat besar bisa dipecahkan menjadi beberapa blok volume 3D yang lebih kecil, atau sekumpulan slice 2D dapat dipecah menjadi sub-slice.

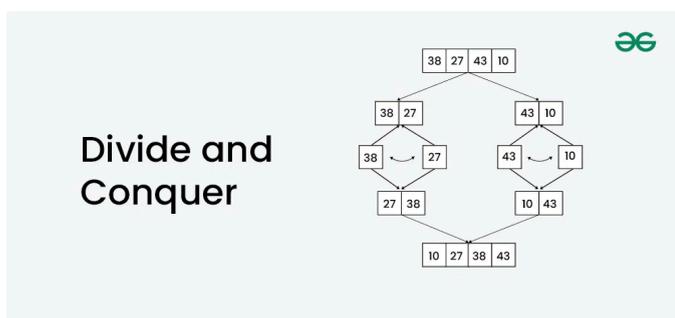
#### 2. Conquer

Setelah masalah utama diuraikan menjadi sub-masalah, langkah ini menitikberatkan pada penyelesaian masing-masing sub-masalah tersebut. Apabila sub-masalah telah mencapai *base case* yang telah ditetapkan, solusi dapat langsung diperoleh dan dilaksanakan. Ini dapat berarti melakukan perhitungan sederhana, seperti mengurutkan dua elemen dalam Merge Sort, atau segera mengklasifikasikan piksel dalam sekumpulan kecil gambar. Akan tetapi, jika sub-masalah masih terlalu

besar dan belum mencapai base case, maka langkah divide akan dipanggil kembali secara rekursif pada sub-masalah itu. Proses rekursif inilah yang menjadi dasar algoritma divide and conquer, memungkinkan algoritma menjelajahi struktur masalah dari yang besar hingga terkecil [13]. Dalam penggunaan citra MRI, hal ini mengindikasikan bahwa setiap blok atau komponen citra akan menjalani analisisnya sendiri, seperti segmentasi area tertentu atau pengambilan fitur tekstur.

### 3. Combine

Setelah seluruh sub-masalah terpecahkan (baik melalui solusi langsung pada base case atau rangkaian panggilan rekursif), langkah terakhir adalah menyatukan solusi-solusi parsial dari masing-masing sub-masalah untuk membentuk solusi akhir dari masalah asli yang lebih besar. Proses penggabungan ini perlu dibuat dengan teliti agar hasil yang diperoleh akurat, koheren, dan mencerminkan solusi yang tepat untuk permasalahan semula. Kompleksitas dari langkah penggabungan mungkin sangat mudah, sedangkan untuk masalah lain, proses penggabungan bisa lebih rumit dan memerlukan algoritma khusus. Dalam analisis citra MRI, dapat diartikan sebagai menggabungkan peta segmentasi dari setiap bagian menjadi peta segmentasi volume secara keseluruhan, atau menyatukan hasil klasifikasi lokal menjadi diagnosis yang menyeluruh.



## Divide and Conquer

Gambar 2.3 Algoritma Divide and Conquer

(<https://media.geeksforgeeks.org/wp-content/uploads/20240403162200/Divide-and-Conquer-banner.webp>)

Analisis kompleksitas waktu dari algoritma divide and conquer seringkali dinyatakan menggunakan persamaan rekurensi. Persamaan ini menggambarkan hubungan antara waktu yang dibutuhkan untuk menyelesaikan masalah berukuran  $N$  dengan waktu yang dibutuhkan untuk menyelesaikan sub-masalah yang lebih kecil. Kompleksitas waktu algoritma divide and conquer pada umumnya adalah  $O(n \log n)$ , meskipun bisa berbeda tergantung pada implementasi spesifiknya. Jika pembagian atau penggabungan memerlukan operasi tambahan yang kompleks, kompleksitas waktunya bisa lebih tinggi dari  $O(n \log n)$ .

Berikut adalah gambaran implementasi sederhana dari algoritma divide and conquer:

```
function DivideAndConquer(input):
```

```
    if base_case(input):
        return solve_base_case(input)

    // 1. Divide: Pecah input menjadi
    bagian-bagian yang lebih kecil
    subproblems = divide(input)

    // 2. Conquer: Selesaikan sub-masalah secara
    rekursif
    results = []
    for sub in subproblems:
        result = DivideAndConquer(sub)
        results.append(result)

    // 3. Combine: Gabungkan hasil dari
    sub-masalah
    return combine(results)
```

## III. PEMBAHASAN

### A. Metodologi Optimasi dengan Divide and Conquer

Optimasi proses diagnosis medis pada analisis citra MRI melalui pendekatan Divide and Conquer melibatkan serangkaian langkah yang terstruktur. Metodologi ini berfokus pada dekomposisi masalah besar (analisis seluruh citra MRI) menjadi sub-masalah yang lebih kecil, memrosesnya secara masing-masing, kemudian menggabungkan hasilnya.

#### 1. Tahap Pra-pemrosesan Citra MRI

Sebelum penerapan strategi divide and conquer yang sesungguhnya, citra MRI mentah umumnya memerlukan proses pra-pemrosesan. Tahapan ini sangat penting untuk memperbaiki kualitas gambar, menstandarkan data, serta menghapus noise atau artefak yang dapat mempengaruhi akurasi dan efektivitas algoritma analisis selanjutnya. Pra-pemrosesan menjamin bahwa data yang diterima di tahap divide sudah dalam keadaan terbaik. Beberapa metode pra-pemrosesan yang umum meliputi:

- Normalisasi Intensitas

Proses pengambilan gambar MRI seringkali menunjukkan perbedaan intensitas piksel/voxel di antara pemindaian yang berlainan, meskipun pada pasien yang sama atau dari perangkat yang berbeda. Normalisasi intensitas bertujuan untuk menyamakan rentang nilai intensitas, sehingga algoritma yang peka terhadap ambang batas atau nilai absolut dapat berfungsi dengan konsisten. Pendekatan seperti normalisasi min-max, normalisasi Z-score, atau pencocokan histogram sering diterapkan untuk mencapai tujuan ini.

- Filter Noise

Citra MRI peka terhadap berbagai jenis noise, seperti noise Gaussian (dari komponen elektronik), noise salt-and-pepper (gangguan impuls), atau noise Rician (ciri

husus MRI yang bergantung pada sinyal). Filter spasial seperti median filter, Gaussian blur, atau filter bilateral efektif dalam mengurangi gangguan sambil tetap mempertahankan rincian tepi. Pengurangan noise ini krusial supaya fitur-fitur penting dalam gambar tidak terdistorsi atau tertutupi

- Koreksi Artefak

Pergerakan pasien saat pemindaian MRI dapat menghasilkan artefak akibat gerakan yang signifikan yang terlihat sebagai kabur atau bayangan. Artefak lainnya bisa muncul karena ketidaksempurnaan mesin MRI atau anomali magnetik. Metode perbaikan artefak yang dapat mencakup algoritma yang berfokus pada registrasi gambar atau teknik dekonvolusi, sangat vital untuk memulihkan ketajaman gambar dan memastikan bahwa struktur anatomi tetap tidak terdistorsi, yang krusial untuk diagnosis yang tepat.

## 2. Tahap Divide : Pemecahan Citra MRI

Tahap divide adalah jantung dari pendekatan divide and conquer. Pada tahap ini, citra MRI yang berukuran besar dipecah menjadi sejumlah sub-citra atau blok yang lebih kecil dan lebih mudah dikelola. Pemilihan strategi pemecahan sangat mempengaruhi efisiensi dan akurasi keseluruhan sistem.

- Pembagian Grid Seragam

Ini merupakan metode pemecahan yang paling sederhana dan paling umum. Citra dibagi menjadi blok-blok dengan ukuran yang sama secara deterministik. Misalnya citra 2D berukuran  $P \times Q$  piksel dapat dibagi menjadi blok-blok berukuran  $p \times q$  piksel, menghasilkan  $(P/p) \times (Q/q)$  blok. Untuk citra volume 3D berukuran  $P \times Q \times R$  voxel, pembagian menjadi blok  $p \times q \times r$  voxel akan menghasilkan  $(P/p) \times (Q/q) \times (R/r)$  blok. Keunggulan metode ini adalah kemudahan implementasi dan keseragaman beban kerja untuk setiap blok, yang sangat ideal untuk paralelisasi. Setiap blok dapat diproses oleh *thread* atau proses yang berbeda dengan asumsi waktu komputasi yang relatif sama.

- Pembagian Adaptif

Berbeda dengan pembagian seragam, metode adaptif ini membagi gambar secara dinamis berdasarkan isi gambar tersebut. Contoh umumnya adalah pemakaian struktur data *quadtree* (pada 2D) atau *octree* (pada 3D). Dalam metode ini, suatu area citra akan terus dipecah menjadi sub-area yang lebih kecil jika area tersebut memiliki variasi

intensitas yang tinggi, tepi yang tajam, atau area yang dicurigai sebagai anomali (berdasarkan kriteria tertentu seperti varian piksel atau adanya peta tepi). Wilayah yang seragam atau kurang memberikan informasi akan tetap berada dalam blok yang lebih besar. Walaupun penerapannya lebih rumit dan memerlukan logika penentuan base case yang cermat, pembagian adaptif dapat meningkatkan efisiensi komputasi dengan memusatkan sumber daya pemrosesan pada wilayah yang paling relevan untuk diagnosis, sehingga menghindari pemrosesan berlebihan di area yang tidak penting.

- Pembagian Berdasarkan Struktur Anatomi

Pendekatan ini memanfaatkan pemahaman di bidang kedokteran. Citra bisa dibagi berdasarkan batas-batas anatomis yang sudah diketahui atau diidentifikasi melalui segmentasi awal. Contohnya, dalam gambar otak, sub-gambar dapat terdiri dari lobus otak yang beragam, batang otak, atau cerebellum. Pembagian seperti ini memungkinkan penerapan algoritma tertentu yang sangat dioptimalkan sesuai dengan karakteristik tiap wilayah anatomi. Tantangannya adalah memerlukan tahap segmentasi yang tepat untuk organ atau struktur awal, yang mungkin meningkatkan kompleksitas pra-pemrosesan.

Dalam ketiga pembagian di atas, penting untuk memperhatikan kemungkinan adanya tumpang tindih (overlap) antara blok-blok. Walaupun memecah blok secara terpisah adalah yang terbaik, tumpang tindih mungkin diperlukan untuk mengatasi anomali atau struktur yang melintasi batas blok secara mulus dan untuk mencegah artefak saat fase *combine*. Sebuah wilayah tumpang tindih kecil dapat menjamin bahwa informasi kontekstual yang diperlukan untuk pemrosesan tepi atau bagian yang terputus tersedia di setiap blok.

## 3. Tahap Conquer : Pemrosesan Sub-Citra Secara Paralel

Saat setiap sub-citra atau blok yang telah dibagi diproses secara terpisah untuk memperoleh informasi diagnostik. Inilah tahap di mana paralelisasi dari divide and conquer memberikan manfaat yang paling besar. Setiap blok dapat diberikan kepada unit pemrosesan yang berbeda (inti CPU, atau GPU, atau bahkan node dalam kluster komputasi), sehingga memungkinkan eksekusi bersamaan.

Algoritma pemrosesan citra yang relevan, segmentasi, ekstraksi fitur atau klasifikasi, diterapkan pada tiap blok. Karena ukuran blok-blok ini jauh lebih kecil daripada gambar aslinya, algoritma yang

mungkin mahal secara komputasi pada gambar penuh menjadi jauh lebih efisien saat diterapkan pada skala lokal.

Berikut merupakan beberapa algoritma dan aplikasi pada penyelesaian sub-citra:

- Segmentasi Tumor/Lesi Lokal  
Memanfaatkan algoritma seperti K-Means untuk mengelompokkan piksel berdasarkan intensitas dan fitur lainnya agar tumor terpisah dari jaringan yang sehat [1] [14]. Atau, metode *Region Growing* bisa dimulai dari titik benih yang dicurigai dalam blok dan diperluas dengan kriteria kesamaan piksel. Untuk kasus yang lebih kompleks, model *machine learning* yang dilatih untuk segmentasi (seperti versi mini U-NET atau arsitektur *fully convolutional network* yang lebih kecil) dapat digunakan pada setiap blok.
- Ekstraksi Fitur  
Untuk diagnosis yang memerlukan analisis tekstur atau bentuk, fitur-fitur seperti GLCM (*Gray Level Co-occurrence Matrix*) [10] dapat dihitung pada setiap blok. GLCM menangkap data mengenai distribusi spasial intensitas piksel dan dapat menunjukkan fitur-fitur kelembutan, kekasaran, atau heterogenitas jaringan, yang merupakan indikator penting dalam identifikasi tumor. Fitur tambahan seperti histogram intensitas, fitur Haralick, atau fitur bentuk (kebulatan dan solidaritas) juga dapat diambil dari objek yang telah tersegmentasi dalam blok.
- Klasifikasi Jaringan dan Anomali  
Setelah fitur diekstraksi atau dilakukan segmentasi, algoritma klasifikasi dapat digunakan pada setiap blok. Ini dapat melibatkan algoritma tradisional seperti *Support Vector Machine* (SVM) [3], untuk mengkategorikan blok sebagai 'normal' atau 'abnormal', atau Jaringan Saraf Tiruan (*Artificial Neural Network*) untuk meramalkan tipe jaringan atau tingkat keparahan penyakit. Dalam *deep learning*, sebuah sub-model atau cabang dari *Convolutional Neural Network* (CNN) [3] yang lebih besar dapat dilatih untuk mengidentifikasi atau mendeteksi anomali dalam setiap blok. Ini mengurangi kebutuhan memori GPU yang besar jika seluruh gambar harus diberikan ke CNN tunggal.

Efisiensi di tahap ini merupakan dampak langsung dari bagian divide atau pemecahan masalah. Lamanya komputasi untuk setiap sub-masalah ( $T(N/b)$ ) secara terpisah lebih minim, dan karena

mereka dapat dikerjakan secara bersamaan, total waktu yang digunakan untuk menyelesaikan semua sub-masalah bisa jauh lebih singkat.

#### 4. Tahap Combine : Penggabungan Hasil Diagnosis

Tahap ini merupakan tahap atau langkah terakhir dan sama pentingnya dengan tahap divide. Setelah semua sub-citra selesai diproses secara independen, hasil parsial dari setiap blok harus digabungkan secara akurat untuk membentuk diagnosis yang komprehensif untuk seluruh citra MRI. Proses penggabungan ini harus cerdas untuk memastikan koherensi spasial dan konsistensi diagnostik.

- Penggabungan Peta Segmentasi

Apabila setiap blok menghasilkan peta segmentasi (contoh: citra biner yang mengindikasikan lokasi tumor), peta-peta tersebut perlu digabungkan kembali ke posisi semula. Apabila terdapat area yang tumpang tindih antara blok, diperlukan strategi untuk menyelesaikan konflik. Ini dapat berupa:

- 1) Voting Mayoritas

Untuk setiap piksel/voxel dalam area yang tumpang tindih, label segmentasi yang paling sering muncul dari semua blok yang bersinggungan akan dipilih

- 2) Averaging/Blending

Apabila hasilnya berupa probabilitas atau nilai kepercayaan, nilai-nilai ini dapat diratakan di wilayah tumpang tindih untuk menghasilkan transisi yang halus

- 3) Prioritas Berdasarkan Keyakinan

Apabila algoritma memberikan nilai keyakinan untuk setiap segmentasi, label dari blok yang memiliki nilai keyakinan tertinggi di wilayah tumpang tindih bisa dipilih. Strategi ini sangat krusial untuk menghindari artefak jahitan atau ketidakcocokan di batas blok.

- Integrasi Hasil Klasifikasi

Jika setiap blok memberikan probabilitas klasifikasi (kalau ada tumor) atau label kelas (jaringan normal, jinak, dan ganas), hasil-hasil ini harus diintegrasikan. Ini bisa mencakup pembuatan peta probabilitas untuk seluruh gambar, di mana setiap piksel/voxel mendapatkan probabilitas untuk menjadi bagian dari kondisi tertentu. Hasil klasifikasi blok dapat dikombinasikan untuk

menghasilkan diagnosis menyeluruh (jika lebih dari batas persentase blok terklasifikasi sebagai 'abnormal', maka seluruh citra dianggap 'abnormal').

- Konsolidasi Deteksi Anomali

Apabila algoritma Conquer berfokus pada identifikasi objek anomali (seperti lesi kecil), daftar anomali yang terindikasi dari setiap blok perlu dikonsolidasikan, Anomali yang teridentifikasi di batas blok yang saling tumpang tindih perlu dianalisis dengan cermat agar tidak terjadi duplikasi entri atau bahkan yang lebih parah lagi, memecah satu anomali besar menjadi beberapa entri kecil yang terpisah. Algoritma non-maksima supresi (*Non-Maximum Suppression*) biasa diterapkan untuk menghapus deteksi yang redundan.

- Pelaporan Komprehensif

Hasil akhir dari proses penggabungan kemudian ditampilkan dalam format yang mudah dipahami oleh radiolog. Ini bisa berupa citra MRI asli yang diberi anotasi dengan batas tumor atau lesi yang terdeteksi, peta panas yang menunjukkan wilayah dengan probabilitas tinggi terhadap penyakit, laporan terstruktur yang merangkum metrik kuantitatif, atau bahkan antarmuka interaktif yang memungkinkan radiolog untuk mengecek detail dari setiap blok jika diperlukan.

### B. Perhitungan Kompleksitas Divide and Conquer

Analisis kompleksitas algoritma adalah aspek krusial untuk memahami mengapa algoritma Divide and Conquer begitu efektif dalam mengoptimalkan pemrosesan citra MRI. Perhitungan ini membantu kita memprediksi performa algoritma saat ukuran masalah (citra MRI) meningkat.

#### 1. Analisis Kompleksitas Waktu Teoritis

Kompleksitas waktu dari algoritma Divide and Conquer umumnya dinyatakan menggunakan persamaan rekurensi. Bentuk umum dari persamaan ini adalah:

$$T(N) = aT(N/b) + f(N)$$

di mana:

- $T(N)$ : Waktu yang dibutuhkan untuk menyelesaikan masalah berukuran  $N$  (jumlah total pixel/voxel dalam citra MRI)
- $a$ : Jumlah sub-masalah yang dihasilkan dari setiap langkah divide
- $b$ : Faktor dimana ukuran masalah dikurangi pada setiap sub-masalah. Jika masalah dibagi dua, maka  $b = 2$ , dan sebagainya

- $T(N/b)$ : Waktu yang dibutuhkan untuk menyelesaikan satu sub-masalah yang berukuran  $N/b$

- $f(N)$ : Waktu yang dibutuhkan untuk tahap divide dan tahap combine pada setiap tingkat rekursi. Ini juga mencakup semua pekerjaan non-rekursif yang dilakukan di luar panggilan rekursif

Untuk menyelesaikan persamaan rekursi ini dan menemukan notasi asimtotik  $O()$  dari  $T(N)$ , kita dapat menggunakan beberapa metode. Metode yang paling umum digunakan adalah Teorema Master. Teorema Master menyediakan solusi untuk persamaan rekurensi bentuk umum ini berdasarkan perbandingan antara  $N^{\log_b a}$  dan  $f(N)$ .

Contoh kasus:

Asumsikan kita membagi citra  $N$  menjadi 4 sub-citra yang sama besar ( $a = 4$ ), dimana setiap sub-citra memiliki ukuran  $N/4$  piksel ( $b = 4$ ). Pada setiap sub-citra ini, kita menerapkan algoritma segmentasi yang sama dengan kompleksitas  $O((N/4)^2)$ .

Untuk langkah divide dan combine, waktu yang dibutuhkan biasanya linear terhadap ukuran citra. Jadi  $f(N) = O(N)$ .

Maka, persamaan rekursi menjadi:

$$T(N) = 4T(N/4) + O(N)$$

Sekarang, mari kita gunakan Teorema Master untuk menyelesaikan persamaan ini. Dalam kasus ini kita memiliki  $a = 4$ ,  $b = 4$ , dan  $f(N) = N^k$  dengan  $k = 1$ . Kita perlu membandingkan  $N^k$  dengan  $N^{\log_b a}$ .

$$\log_b a = \log_4 4 = 1$$

Karena  $k = \log_b a$  (yaitu  $1 = 1$ ), maka kasus di atas dari Teorema Master berlaku

$$T(N) = O(N^k \log N) = O(N^1 \log N) = O(N \log N)$$

Jadi, kompleksitas waktunya adalah  $O(N \log N)$ .

Kita juga bisa menggunakan algoritma brute force untuk menyelesaikan citra MRI berukuran  $N$  piksel, tetapi kompleksitas waktu totalnya adalah  $T(N) = O(N^2)$ , yang dimana kompleksitas tersebut tidak sebaik algoritma divide and conquer. Jika kita ingin membandingkan algoritma brute force dan divide and conquer, maka akan coba digambarkan dengan kasus di bawah ini:

Jika sebuah citra memiliki  
 $N = 1024 \times 1024 = 2^{20} \approx 10^{12}$  piksel, maka:

- Waktu dengan brute force  
 $O(N^2) = (10^6)^2 = 10^{12}$  operasi
- Waktu dengan divide and conquer  
 $O(N \log N) = 10^6 \times \log_2(10^6) \approx 2 \times 10^7$  operasi

Percepatan teoritis yang dicapai sangat besar, sekitar  $10^{12} / (2 \times 10^7) \approx 50.000$  kali lipat. Ini menunjukkan bagaimana pendekatan divide and conquer dapat mengubah masalah yang tidak praktis diselesaikan dalam waktu yang wajar menjadi layak komputasinya, terutama untuk data bervolume tinggi seperti citra MRI.

## 2. Perhitungan pada Implementasi Paralel

Keunggulan algoritma divide and conquer semakin diperkuat ketika diterapkan pada lingkungan komputasi paralel. Jika sub-masalah yang dihasilkan bersifat independen, mereka dapat dieksekusi secara simultan pada berbagai unit pemrosesan. Contoh perhitungan sederhananya adalah sebagai berikut:

Anggap sebuah citra MRI berukuran 1000 x 1000 piksel. Asumsikan sebuah algoritma untuk segmentasi lokal membutuhkan waktu  $T_{blok} = 1$  detik untuk memproses satu blok berukuran 100 x 100 piksel. Bagilah citra utama menjadi 10 x 10 = 100 blok, masing-masing berukuran 100 x 100 piksel

- Waktu pemrosesan sekuensial (tanpa paralel)  
 Jika kita memproses ke-100 blok secara berurutan, total waktu untuk tahap conquer adalah  $100 \times T_{blok} = 100 \times 1$  detik = 100 detik
- Waktu pemrosesan paralel (asumsi dengan 100 prosesor ideal)  
 Jika kita memiliki 100 processor atau thread yang tersedia dan setiap processor mengerjakan satu blok secara paralel, maka waktu total untuk tahap conquer akan menjadi waktu pemrosesan blok terpanjang, yaitu  $T_{blok} = 1$  detik.
- Waktu dengan divide and conquer paralel  
 Misalkan, waktu tahap divide adalah  $T_{divide} = 0.1$  detik dan waktu tahap combine adalah  $T_{combine} = 0.1$  detik. Jadi total waktunya adalah  
 $T_{divide} + T_{conquer\_paralel} + T_{combine}$

$$= 0.1 + 1 + 0.1 = 1.2 \text{ detik.}$$

Dalam contoh di atas, *speed-up* yang dicapai adalah 100 detik / 1.2 detik  $\approx 83$  kali lipat. Ini menggambarkan potensi yang luar biasa dari algoritma divide and conquer dalam konteks komputasi paralel untuk data berukuran besar.

## C. Implementasi dalam Python

Fungsi utama dalam divide and conquer dalam python yang penulis buat adalah *divide\_image\_recursive(..)* dan segmentasi citra akan dipanggil pada fungsi *segment\_image(..)*

```
def divide_image_recursive(self, image, x, y, width, height, depth=0, max_depth=MAX_DEPTH):
    regions = []

    # Base case: jika region terlalu kecil atau mencapai kedalaman maksimum
    if width < self.min_region_size or height < self.min_region_size or depth >= max_depth:
        region = image[y:y+height, x:x+width]
        if region.size > 0:
            regions.append({'region': region, 'bbox': (x, y, width, height), 'depth': depth})
        return regions

    # Analisis homogenitas region
    region = image[y:y+height, x:x+width]
    homogeneity = self.calculate_homogeneity(region)

    # Jika region homogen, tidak perlu dibagi lagi
    if homogeneity > HOMOGENEITY_THRESHOLD:
        regions.append({'region': region, 'bbox': (x, y, width, height), 'depth': depth, 'homogeneity': homogeneity})
        return regions

    # Divide: bagi region menjadi 4 kuadran
    mid_x = width // 2
    mid_y = height // 2

    # Rekursi untuk setiap kuadran
    quadrants = [
        (x, y, mid_x, mid_y), # Top-left
        (x + mid_x, y, width - mid_x, mid_y), # Top-right
        (x, y + mid_y, mid_x, height - mid_y), # Bottom-left
        (x + mid_x, y + mid_y, width - mid_x, height - mid_y) # Bottom-right
    ]

    for qx, qy, qw, qh in quadrants:
        if qw > 0 and qh > 0:
            regions.extend(self.divide_image_recursive(
```

```

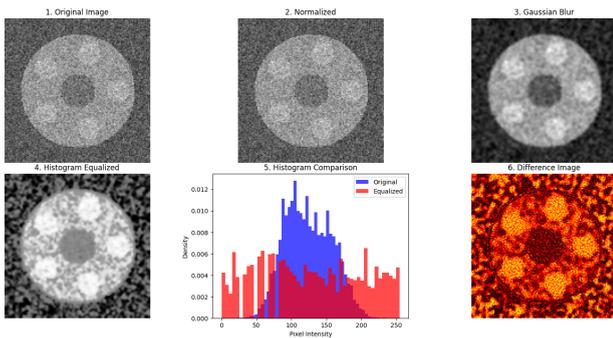
        image, qx, qy, qw, qh, depth
+ 1, max_depth
    ))

    return regions

def segment_image(self, image):
    processed =
self.preprocess_image(image)
    regions =
self.divide_image_recursive(processed, 0, 0,
processed.shape[1], processed.shape[0])
    return regions, processed

```

Untuk implementasi lebih lengkap beserta dengan visualisasinya, sudah tersedia link github yang sudah diberikan di halaman terakhir makalah ini.



Gambar 3.1 Dokumentasi Hasil Implementasi

#### IV. KELEBIHAN DAN KEKURANGAN

Penerapan pendekatan melalui algoritma divide and conquer dalam optimasi proses diagnosis medis pada citra MRI menawarkan berbagai potensi keuntungan, namun juga tidak lepas dari beberapa keterbatasan.

##### 1. Kelebihan Pendekatan Divide and Conquer

Terdapat beberapa kelebihan jika menggunakan algoritma divide dan conquer dalam optimasi proses diagnosis medis pada citra MRI, diantaranya:

###### a. Peningkatan Kecepatan Komputasi

Divide and conquer secara drastis menurunkan waktu pemrosesan dengan membagi analisis citra MRI yang kompleks menjadi sub-masalah yang lebih kecil dan mudah diatur. Ini memungkinkan pelaksanaan algoritma yang lebih cepat, terutama pada gambar beresolusi tinggi atau volume 3D.

###### b. Efisiensi Paralelisasi

Masalah sampingan yang dihasilkan oleh divide and conquer biasanya bersifat mandiri, memungkinkan pemrosesan paralel pada sistem multicore atau terdistribusi. Ini memanfaatkan kemampuan komputasi

terkini dan secara signifikan mempercepat waktu eksekusi keseluruhan.

##### c. Skalabilitas Tinggi

Metode ini sangat dapat diperluas dan ideal untuk mengelola jumlah data MRI yang semakin bertambah. Semakin besar juga kemungkinan keuntungan dari pemecahan masalah.

##### d. Pengelolaan Memori yang Lebih Baik

Dengan memecah citra menjadi bagian-bagian kecil, setiap sub-masalah memerlukan memori yang lebih sedikit. Hal ini dapat mengurangi tekanan pada memori sistem dan memungkinkan pengolahan dataset yang sangat besar yang mungkin tidak dapat dimuat dalam memori secara bersamaan.

##### e. Modularitas dan Kemudahan Debugging

Mengurai masalah yang rumit menjadi modul-modul kecil memungkinkan proses pengembangan, pengujian, dan debugging algoritma menjadi lebih sederhana dan efisien.

##### 2. Kekurangan Pendekatan Divide and Conquer

Terdapat beberapa kekurangan jika menggunakan algoritma divide dan conquer dalam optimasi proses diagnosis medis pada citra MRI, diantaranya:

###### a. Overhead Rekursi dan Penggabungan

Walaupun asimtotik efisien, divide and conquer menambah overhead dari pemanggilan rekursif dan yang lebih krusial adalah tahap penggabungan hasil dari sub-masalah. Jika overhead ini terlalu tinggi atau algoritma penggabungannya sangat rumit, peningkatan kecepatan bisa berkurang, terutama untuk masalah yang berukuran kecil.

###### b. Kompleksitas Desain

Mengembangkan strategi pembagian yang efisien dan algoritma penggabungan yang tepat bisa menjadi sebuah tantangan. Contohnya, penentuan ukuran blok yang benar, pengelolaan area yang tumpang tindih, dan penyelesaian konflik di batas antar sub-masalah membutuhkan perhatian yang mendetail.

###### c. Potensi Kehilangan Konteks Global

Memecah citra secara berlebihan dapat membuat algoritma lokal kehilangan konteks global yang diperlukan untuk diagnosis tertentu. Struktur patologis yang melintasi berbagai batas sub-region bisa

terfragmentasi dan sulit untuk direkonstruksi dengan tepat pada tahap penggabungan.

d. Keseimbangan Beban (*Load Balancing*) yang Sulit

Ketika sub-masalah menunjukkan variasi besar dalam kompleksitas pemrosesan, pembagian tugas yang merata bisa jadi tidak efisien, yang mengakibatkan beberapa unit pemrosesan tidak terpakai menunggu unit lainnya menyelesaikan tugas.

## V. KESIMPULAN

Makalah ini menyimpulkan bahwa pendekatan Divide and Conquer merupakan salah satu solusi yang efektif untuk mengoptimalkan proses diagnosis medis dalam analisis citra MRI. Dengan strategi memecah citra MRI yang besar dan kompleks menjadi sub-bagian yang lebih kecil, Divide and Conquer memungkinkan algoritma pemrosesan citra dijalankan secara paralel, yang secara signifikan mengurangi waktu komputasi yang dibutuhkan. Peningkatan efisiensi dan skalabilitas ini sangat krusial untuk menangani volume data MRI yang terus meningkat di fasilitas kesehatan saat ini. Meskipun demikian, keberhasilan implementasi algoritma Divide and Conquer sangat bergantung dengan desain yang cermat dari strategi pembagian citra, pemrosesan lokal, dan proses penggabungan hasil dimana harus mampu menjaga akurasi dan koherensi diagnostik tanpa menimbulkan *overhead* atau kehilangan hal yang penting. Algoritma Divide and Conquer berpotensi untuk mempercepat diagnosis secara substansial sehingga menjadikan pendekatan dengan algoritma tersebut cukup menjanjikan untuk meningkatkan efisiensi dan kualitas pelayanan kesehatan.

### LINK VIDEO YOUTUBE

<https://youtu.be/mb4DurO6U-M>

### LINK GITHUB

<https://github.com/DitaMaheswari05/mri-divide-conquer.git>

## REFERENSI

- [1] F. A. Wibowo dan A. P. Windarto, "Segmentasi Citra MRI Otak Menggunakan Algoritma K-Means dan *Region Growing*," *Jurnal Informatika*, vol. 12, no. 1, hlm. 1-8, 2018. [Diakses: Jun. 21, 2025]
- [2] World Health Organization, "Ionizing radiation and health effects," *World Health Organization*, Jun. 2022. [Online]. Tersedia: <https://www.who.int/news-room/fact-sheets/detail/ionizing-radiation-and-health-effects>. [Diakses: Jun. 21, 2025].
- [3] Y. Sibaroni dan R. D. Nurmalasari, "Deteksi Tumor Otak pada Citra MRI Menggunakan Metode *Support Vector Machine* (SVM) dengan Ekstraksi Fitur *Gray Level Co-occurrence Matrix* (GLCM)," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 9, no. 2, hlm. 123-130, 2019. [Diakses: Jun. 21, 2025]
- [4] R. M. Sari, M. A. Fikri, dan S. Sumijan, "Klasifikasi Penyakit Alzheimer Berdasarkan Citra MRI Otak Menggunakan *Convolutional Neural Network*," *Jurnal Komputer dan Sistem Informasi*, vol. 6, no. 1, hlm. 1-10, 2021. [Diakses: Jun. 21, 2025]

- [5] [4] A. N. Hakim, M. R. Effendy, dan R. Indriati, "Penerapan Algoritma *Divide and Conquer* pada Permainan Catur," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 2, hlm. 1-8, 2021. [Diakses: Jun. 21, 2025]
- [6] R. T. Wijayanto dan F. C. Setiawan, "Analisis Perbandingan Metode Pengolahan Citra Digital untuk Deteksi Buah Apel," *Jurnal Nasional Komputasi dan Teknologi Informasi*, vol. 3, no. 1, hlm. 1-7, 2020 [Diakses: Jun. 22, 2025].
- [7] I. K. Wibawa dan I. N. S. Kumara, "Peningkatan Kualitas Citra Digital Menggunakan Metode *Histogram Equalization*," *Jurnal Ilmiah Komputer dan Informatika*, vol. 8, no. 2, hlm. 97-103, 2020 [Diakses: Jun. 22, 2025].
- [8] D. Pratiwi dan Y. Yana, "Segmentasi Citra Medis Paru-Paru Menggunakan Metode *Watershed Transformation*," *Jurnal Informatika*, vol. 15, no. 1, hlm. 1-9, 2021 [Diakses: Jun. 22, 2025].
- [9] A. P. Windarto, "Segmentasi Citra Retina Menggunakan Metode K-Means dan *Active Contour Model*," *Jurnal Ilmiah FIFO*, vol. 9, no. 2, hlm. 97-106, 2017 [Diakses: Jun. 22, 2025].
- [10] E. M. Sari, E. M. S., dan D. P. Prihandoko, "Ekstraksi Fitur GLCM dan Klasifikasi Batik Menggunakan Jaringan Saraf Tiruan Backpropagation," *Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)*, vol. 2, no. 1, hlm. 1-8, 2018 [Diakses: Jun. 22, 2025].
- [11] M. Zarkasi dan M. Muslihudin, "Klasifikasi Penyakit Daun Tomat Menggunakan Metode *Convolutional Neural Network*," *Jurnal Komputasi*, vol. 9, no. 1, hlm. 1-9, 2021 [Diakses: Jun. 22, 2025].
- [12] G. Wijaya dan S. Samsinar, "Penerapan Algoritma *Divide and Conquer* untuk Paralelisasi Algoritma *Merge Sort*," *Jurnal Informatika*, vol. 13, no. 2, hlm. 165-172, 2019 [Diakses: Jun. 22, 2025].
- [13] S. A. F. Lubis, R. Rahim, dan A. Siregar, "Implementasi Algoritma *Divide and Conquer* untuk Pemrosesan Citra Digital pada Deteksi Wajah," *Jurnal Mantik*, vol. 5, no. 2, hlm. 1022-1029, 2021 [Diakses: Jun. 22, 2025].
- [14] IBM, "Apa itu K-Means Clustering?" IBM, [Online]. Tersedia: <https://www.ibm.com/id-id/topics/k-means-clustering>. [Diakses: Jun. 23, 2025].

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Juni 2025



Dita Maheswari  
13523125