

BAB III

METODE PENELITIAN

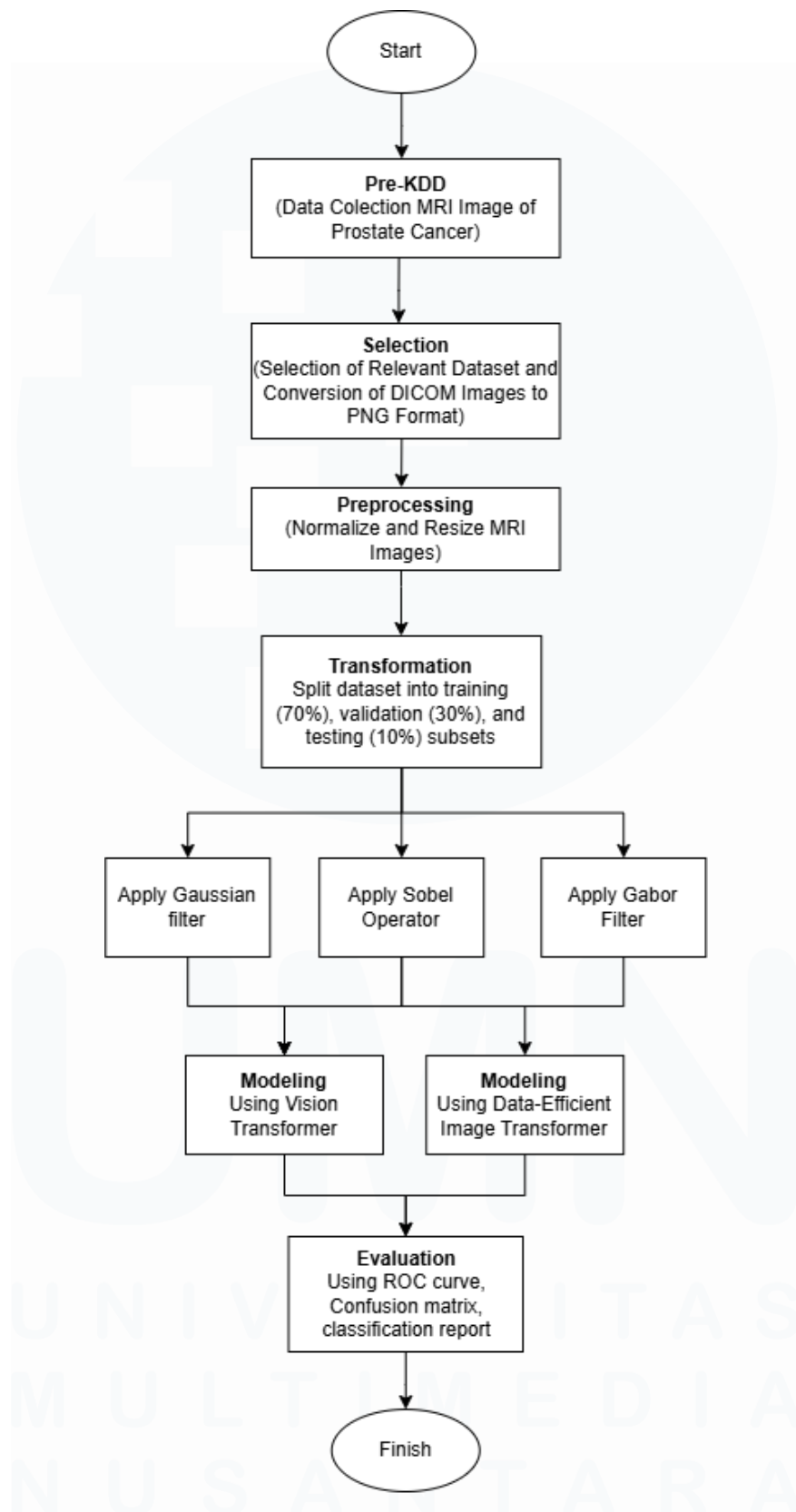
3.1. Metode Penelitian

Metode yang digunakan dalam penelitian ini mengadopsi metode pengembangan sistem *Knowledge Discovery in Databases* (KDD). Metode KDD dipilih karena kemampuannya dalam mengekstraksi informasi yang berharga dari kumpulan data besar, yang sering kali tersembunyi dan tidak terstruktur. KDD merupakan proses sistematis yang mencakup beberapa tahap, mulai dari pemilihan data hingga evaluasi hasil, yang memungkinkan peneliti untuk menemukan pola dan hubungan yang signifikan dalam data [46].

3.1.1 Alur Penelitian

Penelitian ini dilakukan berdasarkan pendekatan *Knowledge Discovery in Databases* (KDD) yang diadopsi untuk mendeteksi metastasis kanker prostat menggunakan data citra MRI. Tahapan KDD yang diterapkan dalam penelitian ini disesuaikan dengan karakteristik dataset citra medis, yang memerlukan pemrosesan khusus untuk memastikan data dapat digunakan secara efektif dalam model pembelajaran mesin. Setiap langkah pada proses KDD dirancang untuk memastikan kualitas data dan efektivitas model dalam mendeteksi pola metastasis kanker prostat dalam citra MRI.

Metode yang digunakan pada penelitian dirangkum dalam *flowchart* pada Gambar 3.1. *Flowchart* pada Gambar 3.1 menunjukkan alur penelitian yang meliputi pengumpulan data, pemrosesan citra, pemodelan dengan metode *Vision Transformer* (ViT) dan *Data-Efficient Image Transformer* (DeiT), serta evaluasi kinerja model menggunakan metrik yang komprehensif. Tahapan-tahapan ini dilakukan secara berurutan dan sistematis untuk mendukung tujuan penelitian, yaitu mendeteksi metastasis kanker prostat secara akurat dan efisien.



Gambar 3.1 Alur Penelitian

Tahapan pertama adalah Pre-KDD, yang melibatkan pengumpulan data citra MRI dari sumber terpercaya, dimana pada penelitian ini adalah dari sebuah rumah sakit di Jawa Timur, sehingga kualitas dan relevansi data yang diperoleh tetap terjaga dengan baik. Selanjutnya, dilakukan proses *Selection*, yaitu memilih *dataset* yang sesuai dan mengonversi file dalam format DICOM (*Digital Imaging and Communications in Medicine*) menjadi format PNG agar kompatibel dengan perangkat analisis citra. Proses ini diikuti oleh *Preprocessing*, di mana citra MRI dinormalisasi untuk menyelaraskan nilai pikselnya dan diubah ukurannya menjadi dimensi seragam, sehingga data siap untuk digunakan dalam model pembelajaran mesin.

Tahapan berikutnya adalah *Transformation*, yaitu membagi *dataset* menjadi tiga subset, yaitu data pelatihan (70%), validasi (20%), dan pengujian (10%). Pembagian ini dilakukan secara stratifikasi untuk memastikan distribusi data yang merata di setiap kelas label, sehingga meningkatkan keandalan model yang akan dibangun. Setelah pembagian *dataset*, dilakukan proses *Image Processing* seperti *Gaussian filter* untuk mengurangi noise, *Sobel operator* untuk mendeteksi tepi, dan *Gabor filter* untuk menangkap pola tekstur serta frekuensi spesifik yang relevan dengan karakteristik metastasis.

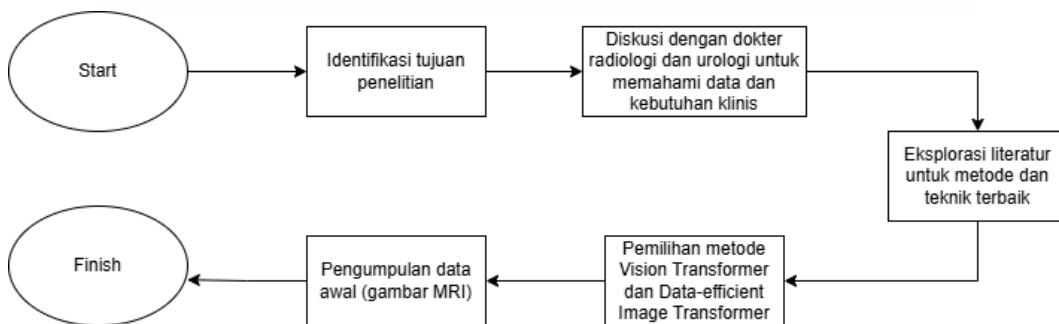
Tahap inti penelitian adalah *Modeling*, di mana dua model berbasis *transformer* diterapkan, yaitu *Vision Transformer* (ViT) dan *Data-Efficient Image Transformer* (DeiT). ViT digunakan untuk memahami hubungan spasial pada citra, sementara DeiT dirancang untuk bekerja secara efisien dengan data yang terbatas. Model dilatih untuk mengenali pola metastasis kanker prostat berdasarkan label kelas yang tersedia. Setelah model dilatih, dilakukan *Evaluation* menggunakan berbagai metrik performa, termasuk kurva ROC untuk mengukur kemampuan diskriminasi, *confusion matrix* untuk mengevaluasi prediksi pada masing-masing kelas, serta *classification report* yang mencakup akurasi, presisi, *recall*, dan *F1-score*. Tahapan ini diakhiri dengan analisis hasil untuk menjawab pertanyaan penelitian, mengidentifikasi model terbaik, dan menyimpulkan kontribusi penelitian terhadap

deteksi metastasis kanker prostat. Pendekatan ini dirancang untuk memastikan setiap langkah penelitian dilakukan secara sistematis dan komprehensif.

3.2 Tahapan Penelitian

3.2.1 Pre-KDD

Penelitian ini dimulai dengan tahapan Pre-KDD, yang bertujuan untuk mengidentifikasi dan memahami konteks data serta tujuan penelitian. Pada tahap ini, dilakukan diskusi dengan dokter radiologi dan urologi dari Rumah Sakit XYZ di Jawa Timur untuk memahami karakteristik data MRI prostat serta kebutuhan klinis terkait deteksi metastasis kanker prostat. Selain itu, dilakukan eksplorasi literatur untuk menentukan metode dan teknik terbaik dalam pengolahan data citra medis, terutama yang berhubungan dengan *Vision Transformer* (ViT) dan *Data-efficient Image Transformer* (DeiT). Tahapan ini juga mencakup pengumpulan data awal, yaitu gambar MRI pasien yang mencakup berbagai tingkat metastasis (M0, M1a, M1b, dan M1c), dan peninjauan kualitas data untuk memastikan kesesuaiannya dengan tujuan penelitian. Proses yang dilakukan selama tahap Pre-KDD ditunjukkan pada Gambar 3.2.

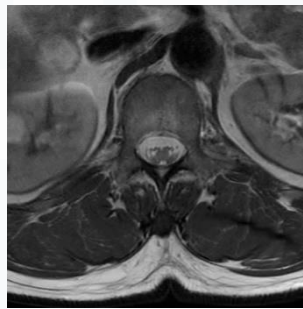


Gambar 3.2 Tahapan Pre-KDD

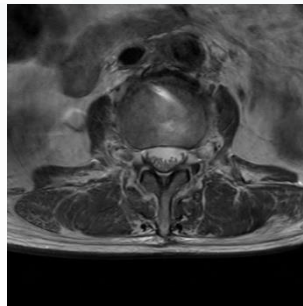
3.2.2 Selection

Tahapan pertama dalam metode KDD adalah *Selection*, yang mencakup proses pemilihan *dataset* yang relevan dari sumber yang tersedia. Dalam penelitian ini, *dataset* yang digunakan berupa gambar MRI prostat dengan format *Digital Imaging and Communications in Medicine* (DICOM) yang diperoleh dari Rumah Sakit XYZ di Jawa Timur. *Dataset* ini berisi gambar MRI pasien dengan berbagai tingkat penyebaran kanker prostat, termasuk M0, M1a, M1b, dan M1c. Gambar-gambar ini

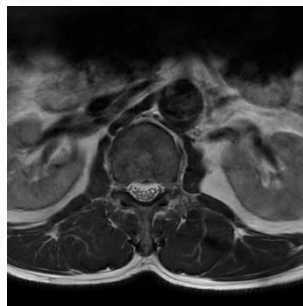
kemudian dikonversi ke format Portable Network Graphics (PNG) untuk mempermudah pengolahan data, dan dilakukan pengelompokan gambar ke dalam *folder* berdasarkan kelas metastasis (M0, M1a, M1b, dan M1c), dengan total gambar akhir adalah 4986 gambar. Tahap ini memastikan bahwa data yang digunakan relevan dengan tujuan penelitian dan mencakup representasi kelas yang sesuai. Hasil konversi gambar dalam format PNG untuk setiap kelas metastasis ditunjukkan pada Gambar 3.2 sampai Gambar 3.5.



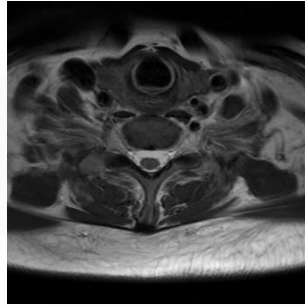
Gambar 3.3 Contoh Gambar MRI Metastasis Kanker Prostat dengan Kelas M0



Gambar 3.4 Contoh Gambar MRI Metastasis Kanker Prostat dengan Kelas M1a



Gambar 3.5 Contoh Gambar MRI Metastasis Kanker Prostat dengan Kelas M1b



Gambar 3.6 Contoh Gambar MRI Metastasis Kanker Prostat dengan Kelas M1c

3.2.3 *Preprocessing*

Tahap berikutnya adalah *Preprocessing*, yang berfokus pada pembersihan dan penyesuaian data untuk memastikan kualitas data yang optimal. Salah satu langkah utama dalam tahap ini adalah normalisasi citra MRI untuk menyelaraskan nilai piksel pada setiap *channel* RGB berdasarkan *mean* dan standar deviasi yang dihitung dari *dataset ImageNet*. Nilai *mean* yang digunakan adalah [0.485, 0.456, 0.406], sedangkan standar deviasi adalah [0.229, 0.224, 0.225]. Normalisasi ini bertujuan untuk memastikan distribusi piksel gambar berada dalam skala yang seragam, mempermudah proses pembelajaran model, dan menjaga kesesuaian dengan data yang digunakan dalam pelatihan model sebelumnya (*pre-trained*). Selain itu, citra diubah ukurannya menjadi 224x224 piksel agar kompatibel dengan arsitektur *Vision Transformer (ViT)* dan *Data-efficient Image Transformer (DeiT)*. Langkah-langkah *preprocessing* ini memastikan bahwa *dataset* siap digunakan dalam pelatihan model dengan kualitas dan konsistensi yang optimal.

3.2.4 *Transformation*

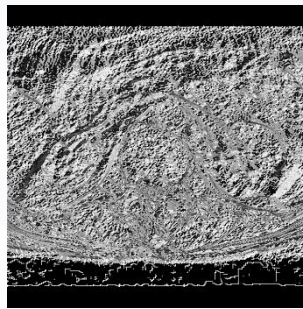
Tahapan selanjutnya adalah *Transformation*, yang bertujuan untuk mengubah data mentah menjadi format yang dapat digunakan oleh algoritma pemodelan. Dalam penelitian ini, proses transformasi dimulai dengan membagi *dataset* menjadi tiga subset: *training*, *validation*, dan *testing*. Sebanyak 10% dari total *dataset* dipisahkan untuk *testing*, sementara sisanya dibagi menjadi 70% untuk *training* dan 30% untuk *validation*. Pembagian ini dilakukan untuk memastikan bahwa model

dapat belajar secara optimal dari data pelatihan serta divalidasi pada data yang tidak pernah dilihat sebelumnya.

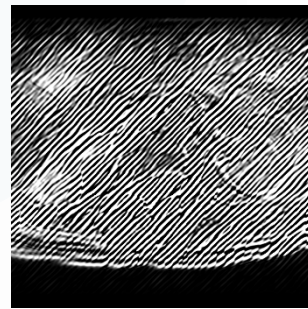
Selain pembagian *dataset*, pada tahap ini juga diterapkan metode *image processing* untuk meningkatkan kualitas data dan mengekstraksi fitur visual yang relevan. Proses ini mencakup penerapan tiga teknik, yaitu *Gaussian filter*, *Sobel operator*, dan *Gabor filter*, di mana setiap teknik menghasilkan representasi citra yang berbeda. Hasil dari proses ini kemudian disimpan dalam empat *folder* terpisah: satu *folder* berisi citra asli (tanpa pemrosesan gambar), dan tiga *folder* lainnya berisi citra hasil pemrosesan. Hasil pemrosesan gambar menggunakan *Gaussian filter*, *Sobel operator* atau *Sobel filter*, dan *Gabor filter* ditunjukkan pada Gambar 3.6 sampai Gambar 3.8.



Gambar 3.7 Contoh Gambar yang Telah Diproses Menggunakan *Gaussian Filter*



Gambar 3.8 Contoh Gambar yang Telah Diproses Menggunakan *Sobel Filter*



Gambar 3.9 Contoh Gambar yang Telah Diproses Menggunakan *Gabor Filter*

3.2.5 Modeling

Tahap Data Mining dilakukan untuk membangun model prediktif menggunakan algoritma *Vision Transformer* (ViT) dan *Data-efficient Image Transformer* (DeiT). Model ViT menggunakan arsitektur ViT Base 16 dengan bobot *pre-trained* ImageNet1K_V1, sedangkan model DeiT menggunakan arsitektur DeiT Base yang dilatih dengan teknik *distillation*. Kedua model dimodifikasi pada lapisan keluaran (*output head*) untuk mendukung klasifikasi ke dalam empat kelas metastasis kanker prostat.

3.2.6 Evaluation

Tahap terakhir adalah *Evaluation*, di mana kinerja model dinilai menggunakan berbagai metrik evaluasi seperti *confusion matrix*, ROC curve, dan nilai AUC. *Confusion matrix* digunakan untuk menganalisis distribusi kesalahan prediksi pada setiap kelas, sementara ROC curve dan AUC mengukur kemampuan model untuk membedakan antara kelas-kelas positif dan negatif. Selain itu, metrik lainnya seperti akurasi, *recall*, *precision*, dan *F1 score* juga digunakan untuk memberikan gambaran lebih lengkap tentang kinerja model. Akurasi mengukur seberapa banyak prediksi yang benar dari keseluruhan prediksi, sementara *recall* menunjukkan seberapa baik model dalam mengidentifikasi kelas positif. *Precision* mengukur proporsi prediksi positif yang benar, dan *F1 score* memberikan rata-rata harmonis antara *precision* dan *recall*. Evaluasi ini memastikan bahwa model yang dikembangkan memiliki akurasi dan generalisasi yang baik dalam mendeteksi metastasis kanker prostat, serta mampu memberikan hasil yang seimbang antara identifikasi kelas positif dan negatif.



BAB IV

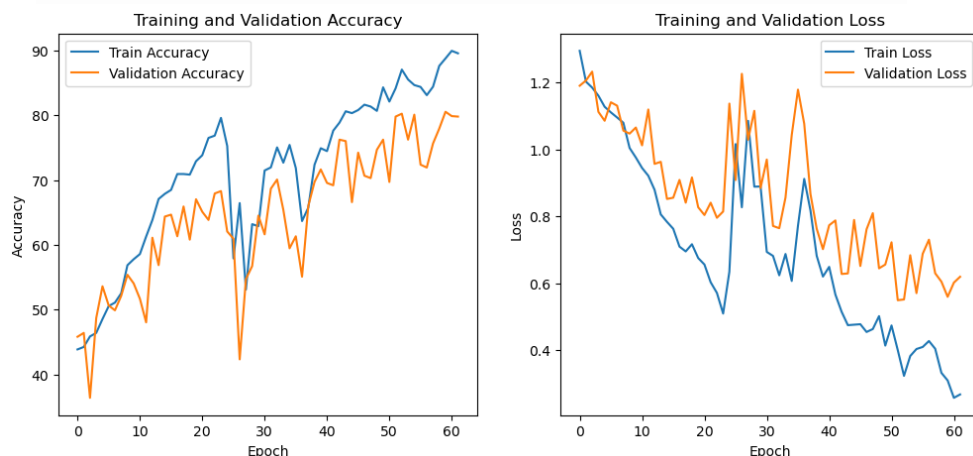
HASIL DAN PEMBAHASAN

4.1 Hasil Perancangan

4.1.1 Vision Transformer

Model untuk deteksi metastasis kanker prostat dibangun menggunakan arsitektur *Vision Transformer* (ViT) dengan parameter *default*. Proses pelatihan menggunakan metode *early stopping* dengan nilai *patience* 10 untuk menentukan jumlah *epoch* yang optimal. Optimizer Adam digunakan dengan *learning rate* 0.001, yang merupakan nilai *default*, dan *batch size* diatur sebesar 32. Masing-masing skenario pelatihan ViT akan menggunakan empat jenis data, yaitu data asli (original), data dengan *filter Gaussian*, data dengan *operator Sobel*, dan data dengan *filter Gabor*. Hasil pelatihan dicatat dalam grafik akurasi dan *loss*, serta model dievaluasi dengan menggunakan *confusion matrix* dan kurva ROC, serta metrik *precision*, *recall*, dan *f1-score* untuk mengukur performa secara menyeluruh.

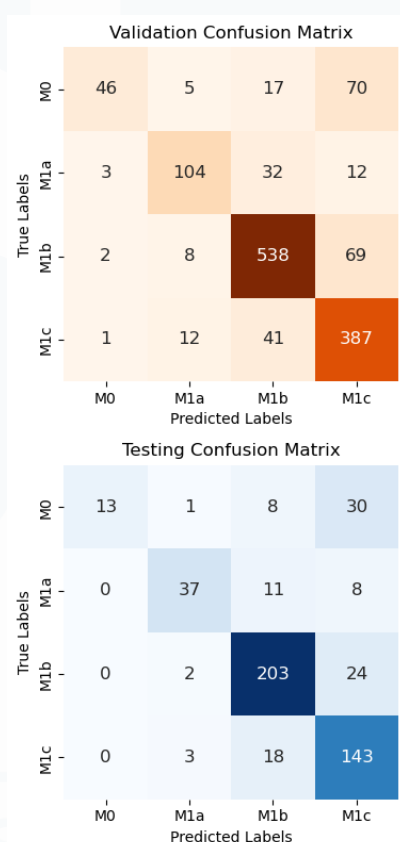
4.1.1.1 Pengujian Model ViT dengan Data Tanpa Image Processing



Gambar 4.1 Grafik Akurasi dan *Loss* Model ViT Menggunakan *Dataset* Tanpa Pemrosesan Citra

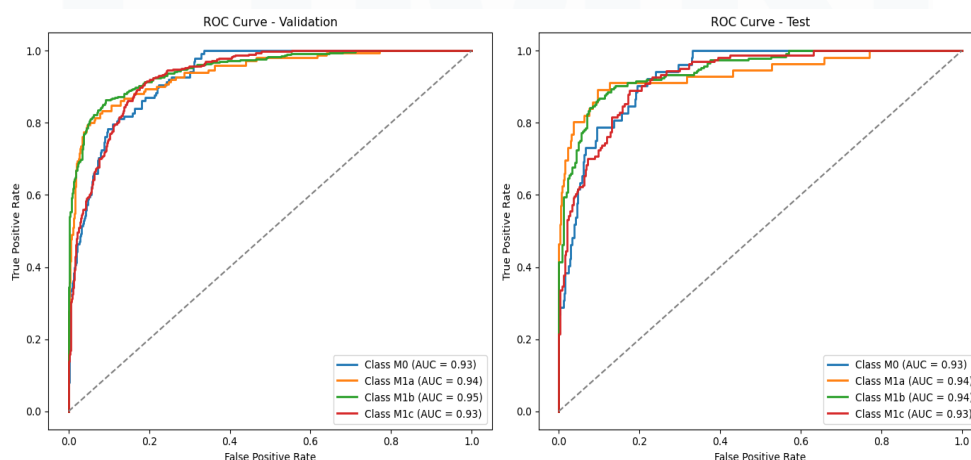
Gambar 4.1 memperlihatkan grafik akurasi dan grafik *loss* dari hasil iterasi pelatihan yang dilakukan dengan menggunakan *dataset* metastasis kanker prostat tanpa *image processing* pada model *Vision Transformer*. Dalam

pelatihan ini, teknik *early stopping* menghentikan iterasi pada *epoch* ke-62. Pada grafik akurasi, terlihat bahwa akurasi pelatihan meningkat secara konsisten dari sekitar 40% hingga mencapai 90% pada *epoch* ke-60, menandakan model mampu memahami pola data pelatihan dengan baik. Sementara itu, akurasi validasi cenderung fluktuatif, terutama di awal pelatihan, sebelum akhirnya stabil di kisaran 70-80%, yang masih lebih rendah dibandingkan akurasi pelatihan. Grafik *loss* menunjukkan penurunan *train loss* secara signifikan dari 1.3 menjadi sekitar 0.2 pada *epoch* ke-60. Namun, *validation loss* menunjukkan pola yang lebih fluktuatif dengan nilai yang lebih tinggi dibandingkan *train loss* dan cenderung stabil di kisaran 0.6. Perbedaan signifikan antara akurasi dan *loss* pada data pelatihan dan validasi ini mengindikasikan adanya *overfitting*. Secara keseluruhan, meskipun model belajar dengan baik pada data pelatihan, performa pada data validasi masih belum optimal.



Gambar 4.2 *Confusion Matrix* Model ViT Menggunakan *Dataset* Tanpa Pemrosesan Citra

Analisis *Confusion Matrix* dari Gambar 4.2 menunjukkan bahwa model memiliki kinerja yang baik dalam mengklasifikasikan kelas M1b, dengan jumlah *true positives* yang tinggi (538) dan jumlah *false positives* serta *false negatives* yang relatif rendah pada *confusion matrix* validasi. Namun, klasifikasi untuk kelas M0 dan M1a menunjukkan hasil yang kurang akurat, terlihat dari banyaknya kasus yang terklasifikasikan salah. Untuk kelas M0, model salah mengklasifikasikan 17 sebagai M1b dan 70 sebagai M1c. Sedangkan untuk kelas M1a, model salah mengklasifikasikan 3 sebagai M0, 32 sebagai M1b, dan 12 sebagai M1c. Untuk kelas M1b, model salah mengklasifikasikan 2 sebagai M0, 8 sebagai M1a, dan 69 sebagai M1c. Terakhir, untuk kelas M1c, model salah mengklasifikasikan 1 sebagai M0, 12 sebagai M1a, dan 41 sebagai M1b. Pada *confusion matrix* pengujian, tren yang sama juga terlihat, dengan kinerja yang baik dalam mengklasifikasikan M1b dan akurasi yang lebih rendah pada kelas M0 dan M1a. Kesalahan klasifikasi pada kelas M0 termasuk 8 yang terklasifikasikan sebagai M1b dan 30 sebagai M1c. Untuk kelas M1a, model salah mengklasifikasikan 11 sebagai M1b dan 8 sebagai M1c. Pada kelas M1b, kesalahan terjadi dengan 2 yang terklasifikasikan sebagai M1a dan 24 sebagai M1c, sedangkan pada kelas M1c, 3 salah terklasifikasikan sebagai M1a dan 18 sebagai M1b. Secara keseluruhan, model menunjukkan performa yang baik dalam mengklasifikasikan kelas M1b, namun masih kesulitan dengan kelas M0 dan M1a.



Gambar 4.3 ROC Curve Model ViT Menggunakan Dataset Tanpa Pemrosesan Citra

Gambar 4.3 menunjukkan dua kurva ROC yang menggambarkan performa model pada set validasi dan set uji, dengan empat kelas yang berbeda. Kelas M0 memiliki skor AUC 0,93 pada kedua set, menandakan performa yang baik. Kelas M1a memiliki skor AUC 0,94 pada kedua set, sedangkan kelas M1b memiliki skor AUC 0,95 pada set validasi dan 0,94 pada set uji. Kelas M1c memiliki skor AUC 0,93 pada kedua set. Secara keseluruhan, model menunjukkan performa yang sangat baik baik pada set validasi maupun set uji.

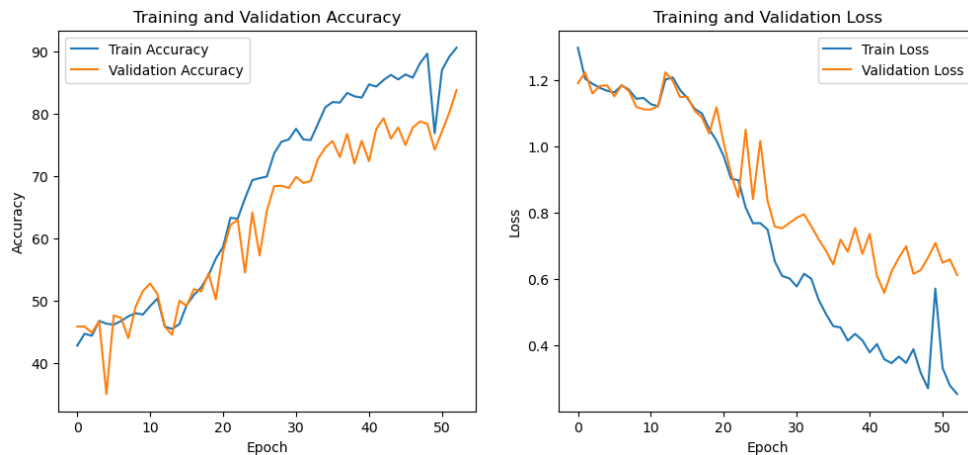
Tabel 4.1 Tabel Evaluasi Model ViT Menggunakan *Dataset* Tanpa Pemrosesan Citra

Class	<i>Precision</i>		<i>Recall</i>		<i>F-1 Score</i>	
	Val	Test	Val	Test	Val	Test
M0	0.88	1.00	0.33	0.25	0.48	0.40
M1a	0.81	0.86	0.69	0.66	0.74	0.75
M1b	0.86	0.85	0.87	0.89	0.86	0.87
M1c	0.72	0.70	0.88	0.87	0.79	0.78

Tabel 4.1 menunjukkan kinerja klasifikasi terhadap beberapa kelas (M0, M1a, M1b, M1c) berdasarkan metrik *Precision*, *Recall*, dan *F1 Score*. Kelas M0 memiliki *precision* yang tinggi (1.00 pada *test*), tetapi sangat rendah dalam *recall* (0.25), yang menunjukkan ketidakmampuannya untuk mendeteksi banyak kasus positif. Sebaliknya, kelas M1a menunjukkan keseimbangan yang lebih baik dengan *precision* 0.86 dan *recall* 0.66, menghasilkan *F1 Score* yang solid (0.75). Kelas M1b tampil sebagai yang paling seimbang dengan *precision* 0.85 dan *recall* 0.89, menghasilkan *F1 Score* tertinggi (0.87). Sementara itu, Kelas M1c memiliki *recall* yang sangat baik (0.87) tetapi dengan *precision* yang lebih rendah (0.70), menunjukkan banyak *false positives*.

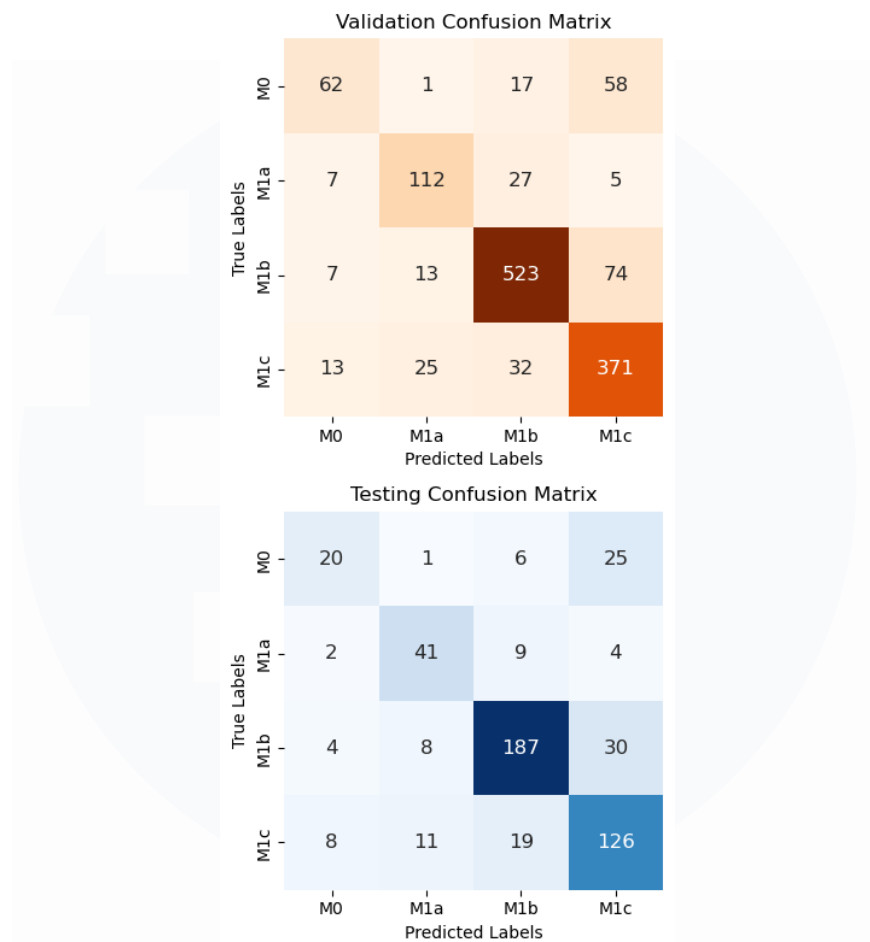
UNIVERSITAS
MULTIMEDIA
NUSANTARA

4.1.1.2 Pengujian Model ViT Menggunakan Data *Filter Gaussian*



Gambar 4.4 Grafik Akurasi dan *Loss* Model ViT Menggunakan *Dataset* dengan *Gaussian Filter*

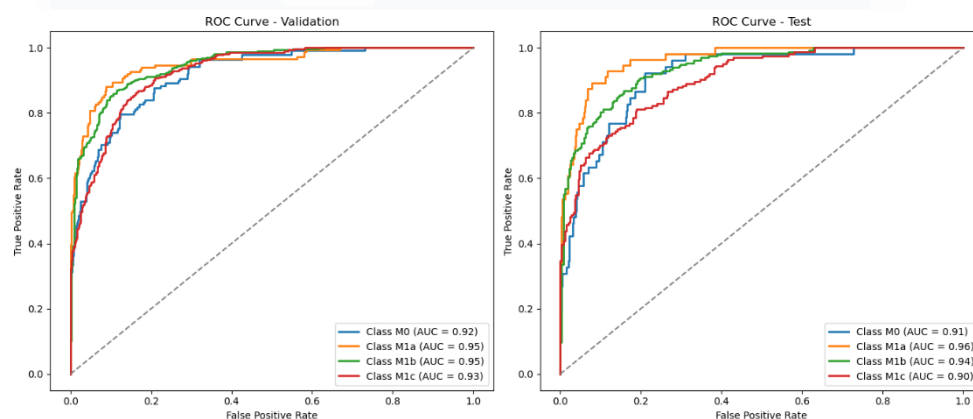
Gambar 4.2 menunjukkan kinerja model *Vision Transformer* (ViT) pada *dataset* yang telah diproses menggunakan *Gaussian Filter*. Pada grafik akurasi, terlihat bahwa akurasi pelatihan meningkat secara konsisten dari sekitar 40% hingga mencapai 90% pada *epoch* ke-50. Akurasi validasi juga menunjukkan peningkatan yang lebih stabil dibandingkan grafik sebelumnya, dengan fluktuasi yang lebih sedikit dan mencapai nilai mendekati 80-85% di akhir pelatihan. Sementara itu, pada grafik *loss*, *train loss* mengalami penurunan yang signifikan dari 1.3 di awal pelatihan menjadi sekitar 0.2 di akhir *epoch*, menunjukkan bahwa model berhasil meminimalkan kesalahan pada data pelatihan. *Validation loss* juga mengalami penurunan yang lebih stabil dan cenderung berhenti di sekitar 0.6, meskipun masih lebih tinggi dibandingkan *train loss*. Dibandingkan dengan *dataset* sebelumnya, penggunaan *Gaussian Filter* tampaknya berhasil mengurangi fluktuasi pada akurasi validasi dan membuat penurunan *loss* lebih stabil, sehingga model menunjukkan kinerja yang lebih baik dan lebih seimbang antara data pelatihan dan validasi.



Gambar 4.5 *Confusion Matrix* Model ViT Menggunakan *Dataset* dengan *Gaussian Filter*

Gambar 4.5 menunjukkan dua *confusion matrix* yang menggambarkan performa model klasifikasi pada data validasi dan data pengujian. Kelas M1c menunjukkan performa terbaik dengan jumlah prediksi benar terbanyak (371) dan jumlah prediksi salah paling sedikit. Kelas M1b juga memiliki jumlah prediksi benar yang signifikan (523), namun jumlah prediksi salahnya lebih tinggi dibandingkan M1c. Sementara itu, kelas M1a memiliki jumlah prediksi salah yang tinggi, menandakan kesulitan model dalam mengklasifikasikan kelas ini dengan benar. Kelas M0 menunjukkan jumlah prediksi salah yang lebih sedikit dibandingkan M1a, namun jumlah prediksi benar masih lebih rendah dibandingkan M1c dan M1b. Pada *confusion matrix* pengujian, kelas M1b memiliki jumlah prediksi benar terbanyak (187), menunjukkan performa terbaik pada data pengujian, sementara M1c memiliki jumlah prediksi benar

yang signifikan (126), namun juga menunjukkan prediksi salah yang tinggi. Kelas M1a memiliki jumlah prediksi benar yang cukup tinggi (41), namun juga memiliki prediksi salah yang relatif tinggi. Kelas M0 menunjukkan jumlah prediksi benar yang lebih rendah dibandingkan kelas lainnya, menandakan kesulitan model dalam mengklasifikasikan kelas ini dengan benar. Perbandingan antara kedua *confusion matrix* menunjukkan bahwa model menunjukkan performa yang lebih baik pada data validasi dibandingkan data pengujian, yang mengindikasikan kemungkinan terjadinya *overfitting*, di mana model terlalu baik dalam mempelajari data validasi namun kurang mampu menggeneralisasi pada data pengujian.



Gambar 4.6 ROC Curve Model ViT Menggunakan Dataset dengan Gaussian Filter

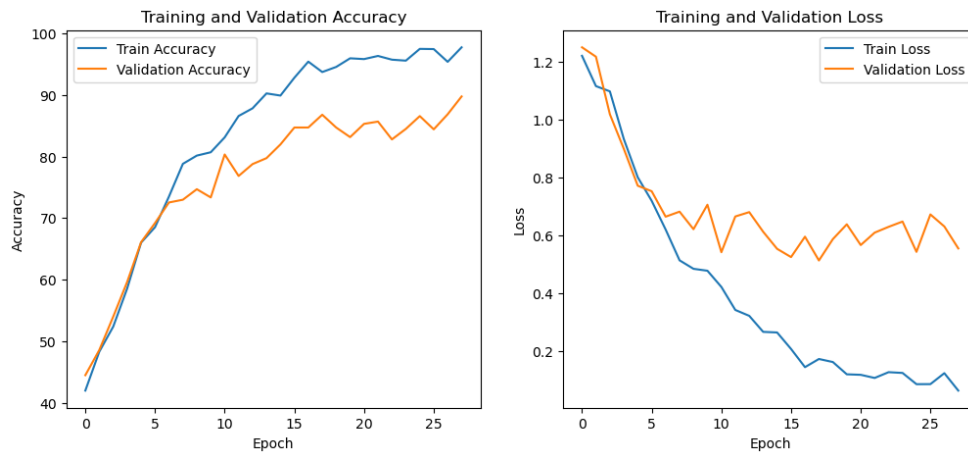
Gambar 4.6 menunjukkan dua kurva Receiver Operating Characteristic (ROC), satu untuk data validasi dan satu untuk data uji. Pada data validasi, model menunjukkan kinerja yang sangat baik dengan AUC yang tinggi untuk sebagian besar kelas: kelas M0 (0,92), M1a (0,95), M1b (0,95), dan M1c (0,93). Sedangkan pada data uji, meskipun kinerja model sedikit menurun, hasilnya tetap sangat baik dengan AUC masing-masing untuk kelas M0 (0,91), M1a (0,96), M1b (0,94), dan M1c (0,90). Secara keseluruhan, model ini mampu membedakan kelas-kelas dengan akurasi tinggi pada kedua *dataset*, meskipun ada sedikit perbedaan kinerja antara data validasi dan data uji, yang kemungkinan disebabkan oleh pelatihan yang menggunakan data validasi. Namun, model tetap menunjukkan kinerja yang sangat baik pada keduanya.

Tabel 4.2 Tabel Evaluasi Model ViT Menggunakan *Dataset* dengan *Gaussian Filter*

Class	<i>Precision</i>		<i>Recall</i>		<i>F-1 Score</i>	
	Val	Test	Val	Test	Val	Test
M0	0.70	0.59	0.45	0.38	0.55	0.47
M1a	0.74	0.67	0.74	0.73	0.74	0.70
M1b	0.87	0.85	0.85	0.82	0.86	0.83
M1c	0.73	0.68	0.84	0.77	0.78	0.72

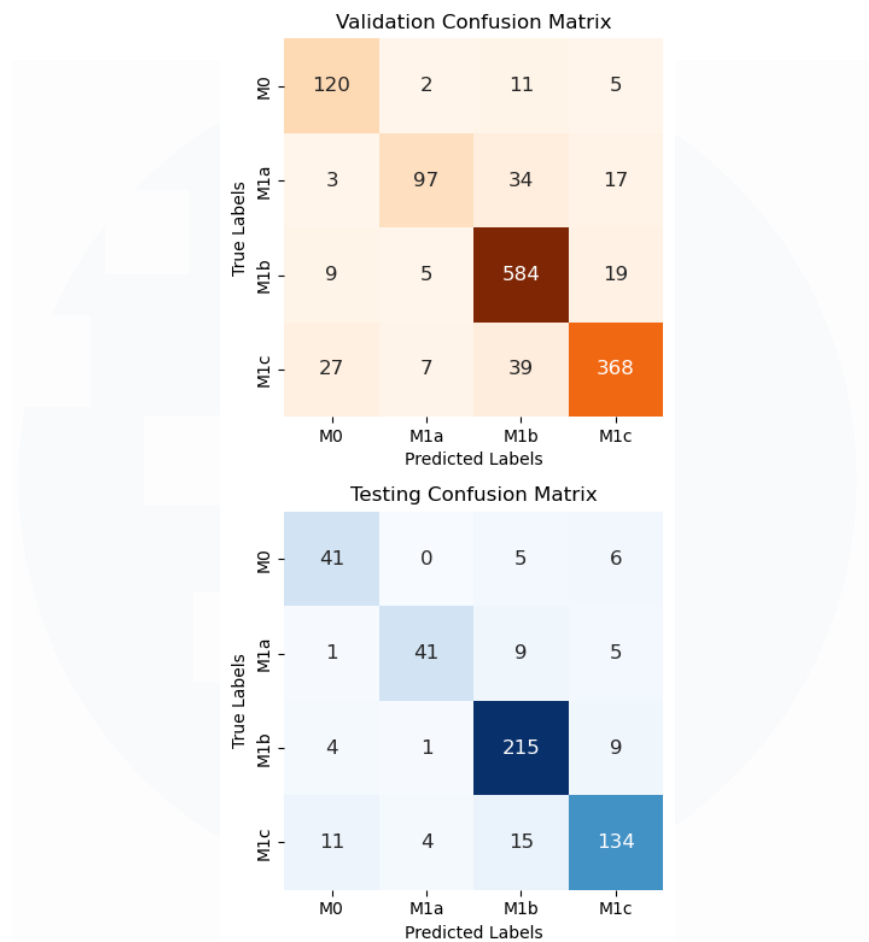
Tabel 4.2 menunjukkan bahwa Kelas M0 memiliki performa yang relatif rendah dibandingkan kelas lainnya, dengan nilai *Precision* dan *Recall* yang keduanya rendah, baik pada data validasi maupun pengujian. Hal ini menunjukkan bahwa model kesulitan dalam mengidentifikasi kelas ini, dan banyak prediksi yang salah meskipun mencoba mengidentifikasi kelas tersebut. *F1 Score* yang rendah (0.47) mengindikasikan ketidakseimbangan antara presisi dan *recall* pada kelas ini, yang memerlukan perhatian lebih. Di sisi lain, kelas M1a memiliki performa yang baik dengan nilai *Precision* dan *Recall* yang cukup tinggi dan seimbang, baik pada data validasi maupun pengujian. *F1 Score* yang tinggi (0.70) menunjukkan bahwa model dapat mengidentifikasi kelas ini dengan baik, dengan keseimbangan antara presisi dan *recall* yang baik. Kelas M1b menunjukkan performa terbaik di antara semua kelas, dengan *Precision* dan *Recall* yang sangat tinggi, serta *F1 Score* yang juga tinggi (0.83). Ini menunjukkan bahwa model sangat efektif dalam mengidentifikasi kelas ini, dengan keseimbangan yang sangat baik antara presisi dan *recall*. Kelas M1c, meskipun tidak sebaik M1b, tetap menunjukkan performa yang baik dengan *Precision* yang sedikit lebih rendah dibandingkan *Recall*. Meskipun ada banyak prediksi positif yang salah, *F1 Score* yang baik (0.72) menunjukkan bahwa model masih cukup efektif dalam mengidentifikasi kelas ini. Secara keseluruhan, kelas M0 membutuhkan perhatian lebih untuk meningkatkan performanya, sementara kelas M1b memberikan hasil yang sangat baik, diikuti oleh kelas M1a dan M1c. Untuk meningkatkan performa, analisis lebih lanjut terhadap data kelas M0 serta teknik peningkatan model atau pengumpulan data tambahan mungkin diperlukan.

4.1.1.3 Pengujian Model ViT Menggunakan Data *Filter Sobel*



Gambar 4.7 Grafik Akurasi dan Loss Model ViT Menggunakan *Dataset* dengan Sobel Operator

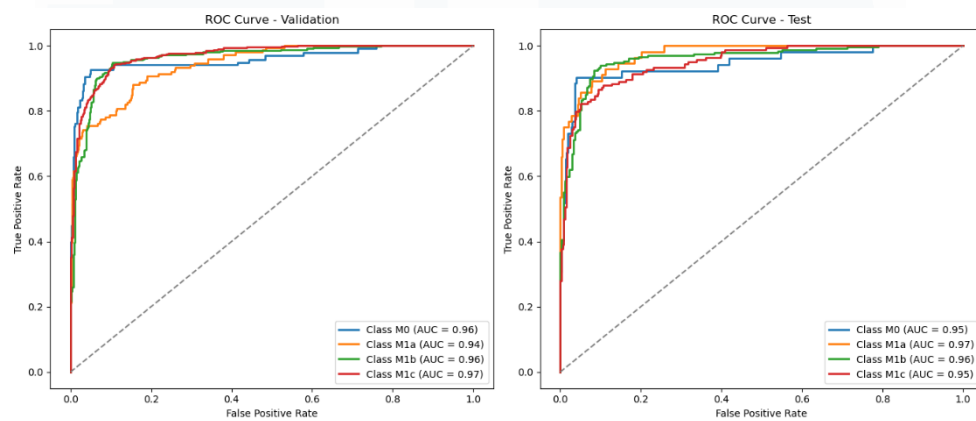
Grafik pada Gambar 4.7 menunjukkan kinerja model *Vision Transformer* (ViT) pada *dataset* yang telah diproses menggunakan *Sobel Operator*. Pada grafik akurasi, terlihat bahwa akurasi pelatihan meningkat dengan cepat dari sekitar 40% di awal pelatihan hingga mendekati 97% pada *epoch* ke-25, menunjukkan performa yang sangat baik pada data pelatihan. Akurasi validasi juga meningkat signifikan dan stabil di kisaran 85-90%, meskipun lebih rendah dibandingkan akurasi pelatihan, namun perbedaannya relatif kecil. Pada grafik *loss*, *train loss* turun drastis dari 1.2 menjadi mendekati 0.1 pada akhir *epoch*, yang menunjukkan bahwa model berhasil meminimalkan kesalahan pada data pelatihan. Sementara itu, *validation loss* juga menurun cukup stabil di awal tetapi cenderung stagnan di sekitar 0.5-0.6 setelah *epoch* ke-10. Dibandingkan dengan metode sebelumnya, penggunaan *Sobel Operator* menghasilkan peningkatan akurasi dan stabilitas *loss* yang lebih baik, akan tetapi jarak antara kurva *train* dan validasi pada grafik *loss* cukup besar, dengan nilai *loss* pada data validasi relatif stagnan dari *epoch* ke 5 sampai ke 25, yang menunjukkan adanya indikasi *overfitting* pada model ini.



Gambar 4.8 *Confusion Matrix* Model ViT Menggunakan *Dataset* dengan *Sobel Operator*

Gambar 4.8 menunjukkan dua *confusion matrix*, satu untuk set validasi dan satu untuk set pengujian, yang digunakan untuk mengevaluasi kinerja model klasifikasi. Pada set validasi, model menunjukkan kinerja yang baik dengan sebagian besar instance diklasifikasikan dengan benar pada sel diagonal. Namun, model kesulitan membedakan antara kelas M1a dan M1b, seperti yang terlihat dari jumlah instance yang salah diklasifikasikan antara kedua kelas ini. Kesalahan klasifikasi yang terjadi mencakup 2 data M1b yang diprediksi sebagai M1a, 11 data M0 yang diprediksi sebagai M1b, 5 data M0 yang diprediksi sebagai M1c, 17 data M1a yang diprediksi sebagai M1c, dan 19 data M1b yang diprediksi sebagai M1c. Pada set pengujian, kinerja model sedikit menurun dibandingkan dengan set validasi, menunjukkan adanya kemungkinan *overfitting*. Meskipun sebagian besar data masih diklasifikasikan

dengan benar, jumlah kesalahan klasifikasi meningkat. Kesalahan tersebut meliputi 5 data M0 yang diprediksi sebagai M1b, 6 data M0 yang diprediksi sebagai M1c, 9 data M1a yang diprediksi sebagai M1b, 5 data M1a yang diprediksi sebagai M1c, dan 9 data M1b yang diprediksi sebagai M1c. Secara keseluruhan, model kesulitan membedakan antara kelas M1a dan M1b, serta antara M1b dan M1c, yang mengindikasikan adanya tumpang tindih fitur atau kurangnya fitur yang memadai untuk membedakan kelas-kelas ini.



Gambar 4.9 ROC Curve Model ViT Menggunakan Dataset dengan Sobel Operator

Berdasarkan analisis kurva ROC dari Gambar 4.9, model menunjukkan kinerja yang sangat baik dalam klasifikasi multi-kelas, baik pada set validasi maupun set pengujian. Pada set validasi, Kelas M0 menunjukkan TPR yang tinggi dan FPR yang rendah di seluruh rentang ambang batas, dengan skor AUC 0,96, yang menandakan kemampuan diskriminasi yang sangat baik. Kelas M1a juga menunjukkan kinerja yang baik dengan AUC 0,94, sedikit lebih rendah dari Kelas M0. Kelas M1b memiliki AUC 0,96, yang juga menunjukkan kemampuan diskriminasi yang sangat baik, sementara Kelas M1c memiliki AUC tertinggi yaitu 0,97, yang menunjukkan kinerja terbaik di antara keempat kelas.

Pada set pengujian, Kelas M0 tetap menunjukkan kinerja yang sangat baik dengan AUC 0,95, menunjukkan kemampuan diskriminasi yang kuat. Kelas M1a menunjukkan AUC 0,97, sedikit lebih baik dibandingkan pada set validasi. Kelas M1b mengalami penurunan sedikit pada set pengujian, dengan

AUC 0,96, sementara Kelas M1c mempertahankan AUC 0,95, menunjukkan kinerja yang konsisten antara set validasi dan pengujian. Secara umum, model menunjukkan kemampuan diskriminasi yang kuat di semua kelas dengan AUC lebih dari 0,9, yang menandakan akurasi prediksi yang tinggi dan kemampuan model untuk digeneralisasi dengan baik pada data yang belum terlihat. Namun, ada sedikit penurunan kinerja pada Kelas M1b di set pengujian, yang mungkin disebabkan oleh *overfitting* pada data pelatihan. Hal ini menunjukkan perlunya optimasi lebih lanjut pada model, terutama untuk Kelas M1b, untuk meningkatkan kinerjanya secara keseluruhan.

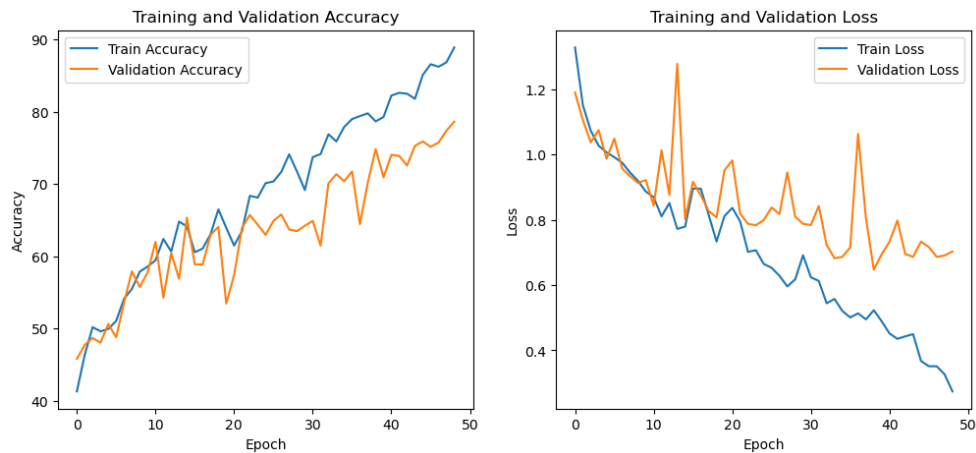
Tabel 4.3 Tabel Evaluasi Model ViT Menggunakan *Dataset* dengan *Sobel Operator*

Class	<i>Precision</i>		<i>Recall</i>		F-1 Score	
	Val	Test	Val	Test	Val	Test
M0	0.75	0.72	0.87	0.79	0.81	0.75
M1a	0.87	0.89	0.64	0.73	0.74	0.80
M1b	0.87	0.88	0.95	0.94	0.91	0.91
M1c	0.90	0.87	0.83	0.82	0.87	0.84

Tabel 4.3 menunjukkan metrik evaluasi untuk empat kelas (M0, M1a, M1b, M1c) berdasarkan nilai presisi, *recall*, dan *F1-score* untuk data validasi (Val) dan pengujian (Test). Kelas M0 memiliki presisi yang baik dengan nilai 0.75 untuk validasi dan 0.72 untuk pengujian, meskipun ada sedikit penurunan pada data pengujian. Kelas M1a menunjukkan peningkatan signifikan dalam presisi, dengan nilai 0.87 untuk validasi dan 0.89 untuk pengujian, menandakan bahwa model ini sangat efektif dalam mengidentifikasi positif. Kelas M1b menunjukkan presisi yang konsisten dengan nilai 0.87 untuk validasi dan 0.88 untuk pengujian, menunjukkan performa yang stabil. Kelas M1c memiliki presisi tertinggi di antara semua model, yaitu 0.90 untuk validasi dan 0.87 untuk pengujian, menunjukkan akurasi yang sangat baik dalam klasifikasi positif. Dalam hal *recall*, Kelas M0 menunjukkan nilai yang cukup baik, dengan 0.87 untuk validasi dan 0.79 untuk pengujian, meskipun ada penurunan pada pengujian. Kelas M1a menunjukkan penurunan *recall* pada data validasi (0.64) tetapi meningkat pada pengujian (0.73), mengindikasikan bahwa model

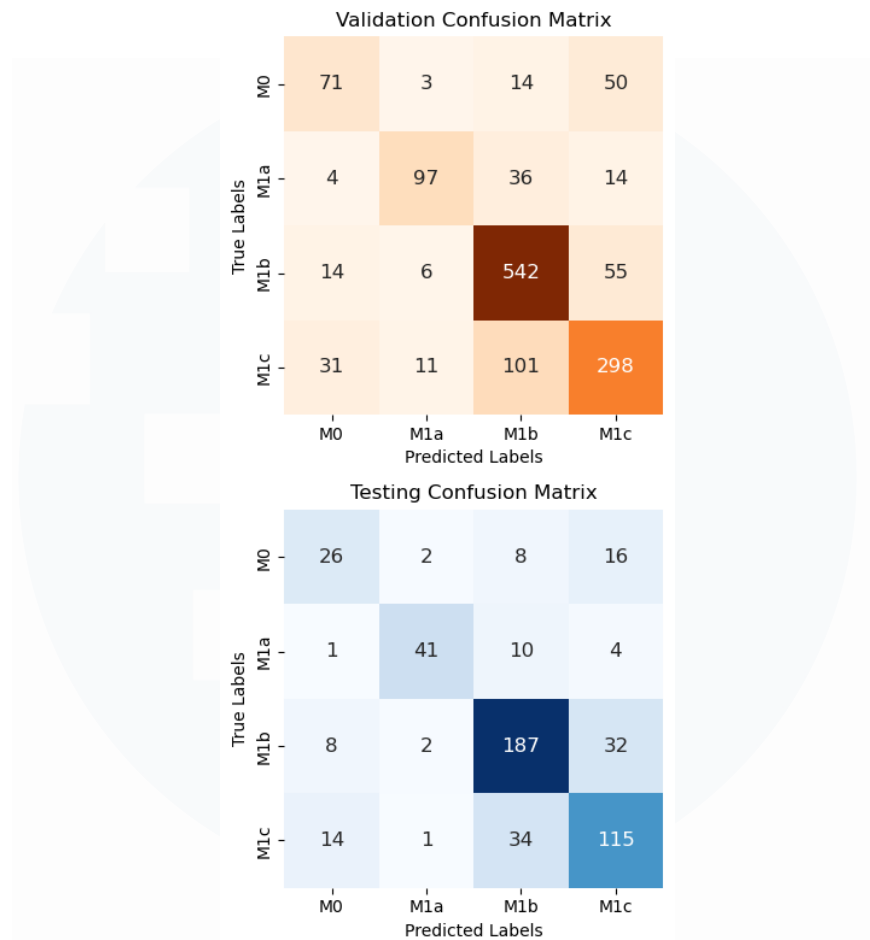
ini mungkin melewatkan beberapa kasus positif pada data validasi. Kelas M1b menunjukkan performa terbaik dengan *recall* 0.95 untuk validasi dan 0.94 untuk pengujian, menandakan kemampuan luar biasa dalam mendeteksi kasus positif. Kelas M1c memiliki *recall* yang baik dengan nilai 0.83 untuk validasi dan 0.82 untuk pengujian, menunjukkan performa yang solid dalam mendeteksi kasus positif. *F1-score* untuk M0 adalah 0.81 pada validasi dan 0.75 pada pengujian, menunjukkan keseimbangan yang baik antara presisi dan *recall* meskipun ada penurunan pada pengujian. *F1-score* kelas M1a adalah 0.74 pada validasi dan 0.80 pada pengujian, yang menunjukkan bahwa meskipun *recall* rendah, presisi yang tinggi membantu meningkatkan *F1-score* pada pengujian. Kelas M1b memiliki *F1-score* tertinggi di antara semua model, yaitu 0.91 untuk validasi dan 0.91 untuk pengujian, menunjukkan keseimbangan yang sangat baik antara presisi dan *recall*. Kelas M1c menunjukkan *F1-score* yang cukup baik dengan nilai 0.87 pada validasi dan 0.84 pada pengujian, yang mencerminkan keseimbangan yang baik antara presisi dan *recall*. Secara keseluruhan, kelas M1b memiliki performa terbaik dengan nilai *F1-score* yang tinggi, presisi yang baik, dan *recall* yang sangat baik. Kelas M1c juga menunjukkan performa yang baik, terutama dalam hal presisi. Sementara itu, Kelas M0 dan M1a memiliki beberapa kelemahan dalam hal *recall*, yang menunjukkan bahwa kemampuan model belum cukup baik dalam mendeteksi semua kasus positif.

4.1.1.4 Pengujian Model ViT Menggunakan Data *Filter Gabor*



Gambar 4.10 Grafik Akurasi dan Loss Model Vit Menggunakan Dataset dengan Gabor Filter

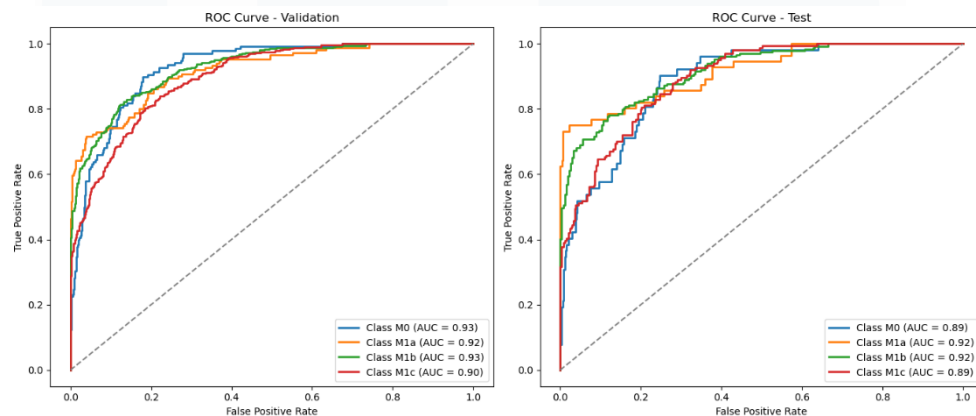
Berdasarkan Gambar 4.10, grafik akurasi dan *loss* model ViT yang dilatih dengan dataset yang telah difilter menggunakan Gabor Filter menunjukkan hasil yang baik, dengan akurasi *training* yang meningkat secara signifikan hingga mencapai sekitar 85%, meskipun terdapat sedikit kesenjangan dengan akurasi validasi yang lebih fluktuatif, mengindikasikan potensi *overfitting*. Di sisi lain, grafik *loss* menunjukkan penurunan yang stabil pada *loss training*, namun *loss validasi* mengalami fluktuasi yang lebih signifikan, menunjukkan kesulitan model dalam generalisasi pada data yang belum pernah dilihat. Secara keseluruhan, meskipun model menunjukkan performa yang baik, terdapat indikasi bahwa model belum sepenuhnya optimal dalam hal generalisasi.



Gambar 4.11 *Confusion Matrix* Model ViT Menggunakan *Dataset* dengan *Gabor Filter*

Pada Gambar 4.11, terdapat dua matrix yang membandingkan label sebenarnya dengan label prediksi untuk empat kelas: M0, M1a, M1b, dan M1c, yaitu untuk data validasi dan data pengujian. Untuk *confusion matrix* validasi, model berhasil mengklasifikasikan 71 data M0 dengan benar, namun ada 3 data yang salah diklasifikasikan sebagai M1a, 14 sebagai M1b, dan 50 sebagai M1c. Pada kelas M1a, model benar mengklasifikasikan 97 data, namun ada 4 data yang salah diklasifikasikan sebagai M0, 36 sebagai M1b, dan 14 sebagai M1c. Untuk kelas M1b, model berhasil mengklasifikasikan 542 data dengan benar, meskipun 14 data salah diklasifikasikan sebagai M0, 6 sebagai M1a, dan 55 sebagai M1c. Sedangkan pada kelas M1c, model berhasil mengklasifikasikan 298 data dengan benar, namun 31 data salah diklasifikasikan sebagai M0, 11 sebagai M1a, dan 101 sebagai M1b. Pada *confusion matrix* pengujian, model

berhasil mengklasifikasikan 26 data M0 dengan benar, tetapi 2 data salah diklasifikasikan sebagai M1a, 8 sebagai M1b, dan 16 sebagai M1c. Untuk kelas M1a, model benar mengklasifikasikan 41 data, tetapi ada 1 data yang salah diklasifikasikan sebagai M0, 10 sebagai M1b, dan 4 sebagai M1c. Pada kelas M1b, model berhasil mengklasifikasikan 187 data dengan benar, meskipun 8 data salah diklasifikasikan sebagai M0, 2 sebagai M1a, dan 32 sebagai M1c. Terakhir, pada kelas M1c, model benar mengklasifikasikan 115 data, tetapi 14 data salah diklasifikasikan sebagai M0, 1 sebagai M1a, dan 32 sebagai M1b.



Gambar 4.12 ROC Curve Model ViT Menggunakan Dataset dengan Gabor Filter

Gambar 4.12 menunjukkan dua kurva ROC, satu untuk data validasi dan satu lagi untuk data uji, yang menggambarkan kinerja model klasifikasi biner pada empat kelas berbeda (M0, M1a, M1b, dan M1c). Pada kurva ROC validasi, keempat kelas menunjukkan kinerja yang sangat baik dengan nilai AUC di atas 0,90, dengan kelas M0 memiliki AUC tertinggi (0,93) yang menunjukkan kemampuan model yang kuat dalam membedakan antara instance positif dan negatif. Kelas M1c memiliki AUC terendah (0,90), namun tetap menunjukkan kinerja klasifikasi yang baik. Untuk kurva ROC uji, hasilnya serupa, dengan nilai AUC di atas 0,89 untuk semua kelas, dan kelas M1a memiliki AUC tertinggi (0,92), sementara kelas M1c kembali menunjukkan AUC terendah (0,89). Secara keseluruhan, kurva ROC menunjukkan bahwa model memiliki kemampuan klasifikasi yang akurat pada kedua set data, dengan nilai AUC yang mendekati 1, yang mengindikasikan

prediksi yang kuat dan kesalahan yang rendah. Meskipun AUC untuk kelas M1c sedikit lebih rendah dibandingkan kelas lainnya, hasil tersebut masih tergolong baik. Dapat disimpulkan bahwa model ini menunjukkan kemampuan generalisasi yang baik, karena kinerjanya pada data validasi dan uji sangat mirip.

Tabel 4.4 Tabel Evaluasi Model ViT Menggunakan *Dataset* dengan *Gabor Filter*

Class	<i>Precision</i>		<i>Recall</i>		<i>F-1 Score</i>	
	Val	Test	Val	Test	Val	Test
M0	0.59	0.53	0.51	0.50	0.55	0.51
M1a	0.83	0.89	0.64	0.73	0.72	0.80
M1b	0.78	0.78	0.88	0.82	0.83	0.80
M1c	0.71	0.69	0.68	0.70	0.69	0.69

Berdasarkan Tabel 4.4, model ini menunjukkan kinerja yang bervariasi pada setiap kelas yang diuji. Untuk kelas M0, presisi dan *recall* masing-masing adalah 0.59 dan 0.51 pada *dataset* validasi, serta 0.53 dan 0.50 pada *dataset* uji, yang menunjukkan bahwa model kesulitan untuk mengidentifikasi kelas ini secara akurat. Skor *F1* untuk M0 juga rendah, yakni 0.55 pada *dataset* validasi dan 0.51 pada *dataset* uji, mengindikasikan bahwa perbaikan diperlukan untuk meningkatkan kinerja model pada kelas ini. Sebaliknya, kelas M1a menunjukkan hasil yang lebih baik, dengan presisi yang sangat tinggi (0.83 pada validasi dan 0.89 pada uji), yang berarti ketika model memprediksi kelas M1a, prediksinya sebagian besar benar. Namun, *recall*-nya lebih rendah (0.64 pada validasi dan 0.73 pada uji), yang menunjukkan bahwa beberapa data dari kelas M1a tidak terdeteksi. Meskipun demikian, skor *F1* yang solid (0.72 pada validasi dan 0.80 pada uji) menunjukkan adanya keseimbangan yang baik antara presisi dan *recall* untuk kelas ini. Kelas M1b memiliki *recall* yang sangat tinggi (0.88 pada validasi dan 0.82 pada uji), yang menunjukkan bahwa model berhasil menangkap sebagian besar data kelas ini. Presisinya juga cukup baik (0.78 pada validasi dan uji), sehingga skor *F1* yang diperoleh (0.83 pada validasi dan 0.80 pada uji) menunjukkan kinerja yang solid dan memadai. Sementara itu, kelas M1c memiliki presisi dan *recall* yang moderat (0.71 dan

0.68 pada validasi, 0.69 dan 0.70 pada uji), dengan skor $F1$ yang seimbang (0.69 pada kedua *dataset*), yang mencerminkan kinerja yang cukup baik, meskipun tidak sekuat kelas M1a atau M1b. Secara keseluruhan, model ini menunjukkan kinerja terbaik pada kelas M1b, diikuti oleh M1a dan M1c, sementara kelas M0 menunjukkan kinerja yang terlemah. Perbedaan antara metrik validasi dan uji menunjukkan bahwa model dapat lebih mengeneralisasi data yang belum terlihat untuk beberapa kelas, seperti M1a, namun tidak pada kelas M0. Kinerja model yang tidak merata di seluruh kelas ini bisa disebabkan oleh adanya ketidakseimbangan pada *dataset*, yang umum terjadi pada masalah klasifikasi multi-kelas.

4.1.1.5 Tabel Hasil Akurasi dan Loss Model ViT

Tabel 4.5 Tabel Hasil Akurasi dan Loss Model ViT

Mo del	<i>Image Processing</i>	Epoch	Accuracy (%)			Loss		
			Train	Val	Test	Train	Val	Test
ViT	-	62	91.14	79.81	79.04	0.2269	0.6198	0.0182
	<i>Gaussian</i>	53	94.33	83.82	74.65	0.1645	0.6114	0.0207
	<i>Sobel</i>	28	99.33	89.76	86.03	0.0255	0.5553	0.0178
	<i>Gabor</i>	49	90.54	78.62	73.65	0.2236	0.7029	0.0217

Tabel 4.5 menunjukkan hasil akurasi dan *loss* yang diperoleh pada setiap set data menggunakan model *Vision Transformer*. Hasil analisis akurasi dan *loss* dari model *Vision Transformer* yang diterapkan pada berbagai teknik pemrosesan gambar menunjukkan perbedaan performa yang signifikan pada *dataset* dengan teknik *image processing* yang berbeda. Metode *Sobel* menghasilkan hasil terbaik dengan akurasi pelatihan tertinggi mencapai 99.33%, diikuti oleh *Gaussian* dengan 94.33%, menunjukkan efektivitas *Sobel* dalam mendeteksi tepi yang tajam dalam gambar. Dalam hal akurasi validasi, *Sobel* juga unggul dengan 89.76%, sementara *Gaussian* dan *dataset* tanpa *image processing* memiliki akurasi validasi masing-masing sebesar 83.82% dan 79.81%. Meskipun *Gabor* memiliki akurasi pelatihan yang lebih baik dibandingkan *dataset* tanpa *image processing* dengan 90.54%, performa validasinya lebih rendah, hanya mencapai 78.62%. Dari segi *loss*, *Sobel*

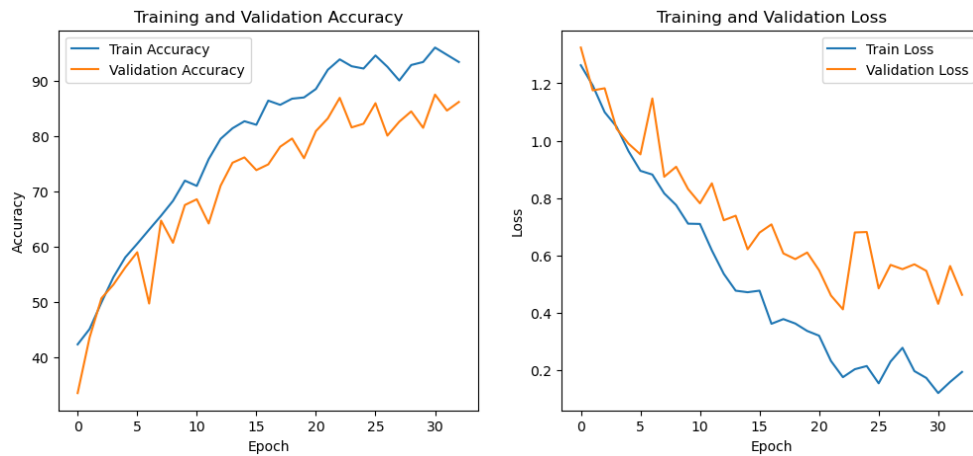
kembali menunjukkan hasil terbaik dengan *loss* pelatihan terendah yakni 0.0255, diikuti oleh *Gaussian* dengan 0.1645, mencerminkan kemampuan model *Sobel* dalam belajar dan menggeneralisasi data dengan baik. Sementara itu, *dataset* tanpa *image processing* dan *Gabor* memiliki *loss* pelatihan yang lebih tinggi, yaitu 0.2269 dan 0.2236, yang menunjukkan kesulitan dalam mempelajari pola data. Secara keseluruhan, hasil ini menunjukkan metode *Sobel* terbukti paling efektif untuk meningkatkan akurasi algoritma *Vision Transformer* berdasarkan metrik akurasi dan *loss*, sedangkan metode lain seperti *Gabor* dan *Original* mungkin perlu ditingkatkan atau dipadukan dengan teknik lain untuk mencapai performa yang lebih baik dalam klasifikasi gambar. Hasil ini menunjukkan bahwa teknik *image processing* dapat membantu algoritma *Vision Transformer* untuk mengklasifikasikan citra medis dengan lebih baik.

4.1.2 Data-efficient Image Transformer

Model untuk deteksi metastasis kanker prostat dibangun menggunakan arsitektur *Data-efficient Image Transformer* (DeiT) dengan parameter *default*. Proses pelatihan menggunakan metode *early stopping* dengan nilai *patience* 10 untuk menentukan jumlah *epoch* yang optimal. Optimizer Adam digunakan dengan *learning rate* 0.001, yang merupakan nilai *default*, dan *batch size* diatur sebesar 32. Masing-masing skenario pelatihan DeiT juga menggunakan empat jenis data, yaitu data asli atau *dataset* tanpa *image processing* (*original*), data dengan *filter Gaussian*, data dengan *operator Sobel*, dan data dengan *filter Gabor*. Hasil pelatihan dicatat dalam grafik akurasi dan *loss*, serta model dievaluasi dengan menggunakan *confusion matrix* dan kurva ROC untuk mengukur performa secara menyeluruh.

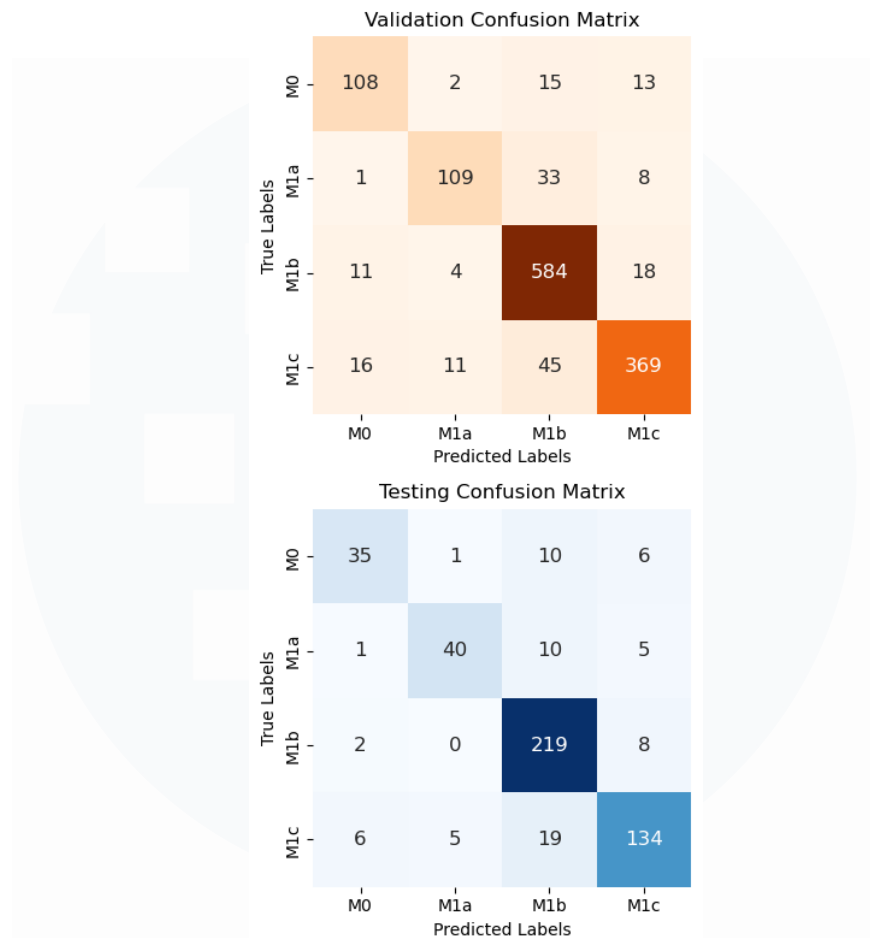
UNIVERSITAS
MULTIMEDIA
NUSANTARA

4.1.2.1 Pengujian Model DeiT dengan Data Tanpa *Image Processing*



Gambar 4.13 Grafik Akurasi dan *Loss* Model DeiT Menggunakan *Dataset* Tanpa Pemrosesan Citra

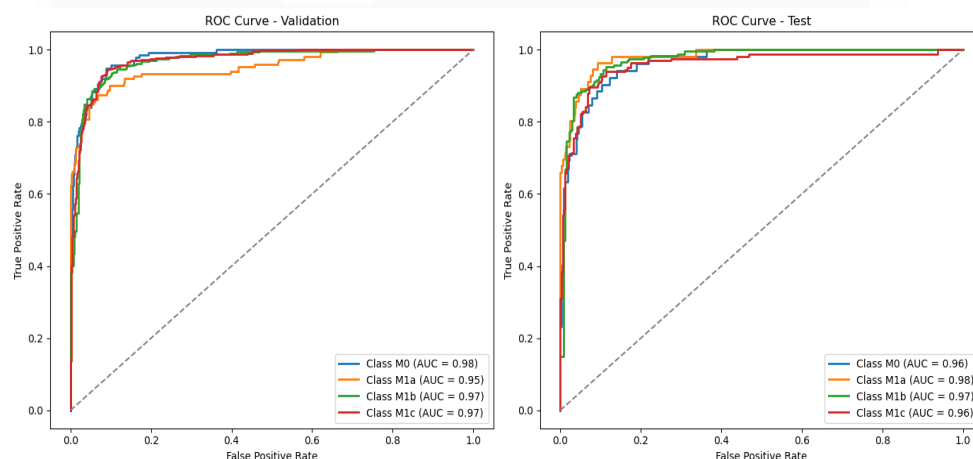
Gambar 4.13 memperlihatkan grafik akurasi dan grafik *loss* dari hasil iterasi pelatihan yang dilakukan dengan menggunakan *dataset* metastasis kanker prostat tanpa *image processing* pada model *Data-efficient Image Transformer*. Grafik akurasi menunjukkan peningkatan yang konsisten pada akurasi pelatihan dari sekitar 40% hingga mencapai lebih dari 90% pada *epoch* ke-32. Namun, akurasi validasi cenderung fluktuatif di awal pelatihan sebelum akhirnya stabil di kisaran 75-85%, yang masih lebih rendah dibandingkan dengan akurasi pelatihan. Pada grafik *loss*, terlihat bahwa *train loss* mengalami penurunan signifikan dari sekitar 1.2 menjadi mendekati 0.2, menandakan bahwa model mampu mempelajari data pelatihan dengan baik. Sebaliknya, *validation loss* menunjukkan pola yang lebih fluktuatif, terutama di pertengahan pelatihan, dengan nilai yang cenderung lebih tinggi dibandingkan *train loss*. Fluktuasi *validation loss* ini mengindikasikan adanya ketidaksesuaian antara kemampuan model dalam mempelajari data pelatihan dan kinerjanya pada data validasi.



Gambar 4.14 *Confusion Matrix* Model DeiT Menggunakan *Dataset* Tanpa Pemrosesan Citra

Berdasarkan Gambar 4.14, *confusion matrix* validasi menunjukkan hasil klasifikasi model pada *dataset* validasi, dimana nilai diagonal yang merupakan *True Positives* (TP) menunjukkan jumlah data yang diklasifikasikan dengan benar, seperti 108 data M0 diklasifikasikan dengan benar sebagai M0, 109 data M1a sebagai M1a, 584 data M1b sebagai M1b, dan 369 data M1c sebagai M1c. Sementara itu, nilai di luar diagonal yang merupakan *False Positives* (FP) menunjukkan data yang diklasifikasikan salah, dimana 15 data M0 diklasifikasikan salah sebagai M1b. Sedangkan, *confusion matrix* pengujian menunjukkan hasil klasifikasi model pada *dataset* pengujian. Pada *confusion matrix* ini, nilai diagonal juga menggambarkan *True Positives* (TP), seperti 35 data M0 yang diklasifikasikan dengan benar sebagai M0, 40 data M1a sebagai M1a, 219 data M1b sebagai M1b, dan 134 data M1c sebagai M1c. Di sisi lain,

False Positives (FP) juga terlihat di luar diagonal, seperti 10 data MO yang diklasifikasikan salah sebagai M1b. Secara keseluruhan, model menunjukkan kinerja yang lebih baik pada *dataset* validasi dibandingkan dengan *dataset* pengujian. Hal ini terlihat dari jumlah *True Positives* yang lebih tinggi dan *False Positives* yang lebih sedikit pada validasi. Ini mengindikasikan bahwa model kemungkinan mengalami *overfitting* pada *dataset* validasi, di mana model lebih "terbiasa" dengan data validasi yang lebih kecil atau lebih spesifik. Di sisi lain, performa model pada *dataset* pengujian yang lebih besar mungkin dipengaruhi oleh kesulitan model dalam menggeneralisasi data yang belum terlihat.



Gambar 4.15 ROC Curve Model DeiT Menggunakan *Dataset* Tanpa Pemrosesan Citra

Analisis kurva ROC dari Gambar 4.15 menunjukkan kinerja yang sangat baik dari model yang diuji, baik pada set validasi maupun set pengujian. Pada kurva ROC validasi, terdapat empat kelas yang dievaluasi, yaitu M0, M1a, M1b, dan M1c, dengan nilai AUC masing-masing 0,98, 0,95, 0,97, dan 0,97. Nilai AUC yang tinggi menunjukkan kemampuan model untuk membedakan antara kelas positif dan negatif dengan sangat baik, dengan semua kelas memiliki AUC di atas 0,9. Hal ini mengindikasikan performa yang sangat baik dalam membedakan antara berbagai kategori. Pada kurva ROC pengujian, hasilnya mirip dengan kurva validasi. Nilai AUC untuk masing-masing kelas adalah 0,96 untuk M0, 0,98 untuk M1a, 0,97 untuk M1b, dan 0,96 untuk M1c,

yang menunjukkan kinerja model yang konsisten. Semua kelas pada set pengujian juga menunjukkan nilai AUC di atas 0,9, mengonfirmasi daya diskriminasi yang kuat dari model ini. Secara keseluruhan, baik pada data validasi maupun data pengujian, model menunjukkan akurasi yang tinggi dan kemampuan yang baik dalam melakukan prediksi, bahkan pada data yang belum pernah dilihat sebelumnya.

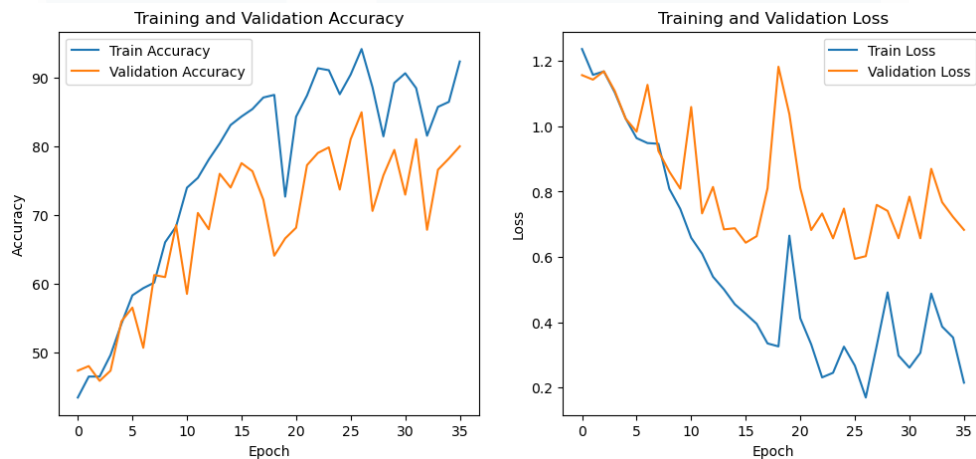
Tabel 4.6 Tabel Evaluasi Model DeiT Menggunakan Dataset Tanpa Pemrosesan Citra

Class	<i>Precision</i>		<i>Recall</i>		<i>F-1 Score</i>	
	Val	Test	Val	Test	Val	Test
M0	0.79	0.80	0.78	0.67	0.79	0.73
M1a	0.87	0.87	0.72	0.71	0.79	0.78
M1b	0.86	0.85	0.95	0.96	0.90	0.90
M1c	0.90	0.88	0.84	0.82	0.87	0.85

Berdasarkan Tabel 4.6, model klasifikasi ini menunjukkan kinerja yang bervariasi di antara empat kelas (M0, M1a, M1b, dan M1c). Pada kelas M0, meskipun *precision* cukup tinggi (0,79 untuk validasi dan 0,80 untuk uji), *recall* lebih rendah pada set uji (0,67), yang mengindikasikan bahwa model gagal mengidentifikasi sebagian besar instansi M0 yang sebenarnya, sehingga *F1 score* menurun dari 0,79 pada validasi menjadi 0,73 pada uji. Hal ini menunjukkan bahwa model mungkin tidak dapat menggeneralisasi dengan baik untuk kelas M0. Kelas M1a memiliki *precision* yang tinggi (0,87 untuk validasi dan uji), tetapi *recall* yang lebih rendah (0,72 untuk validasi dan 0,71 untuk uji), mengindikasikan bahwa meskipun model dapat diandalkan dalam prediksi positif, ia tidak berhasil menangkap banyak instansi M1a yang relevan. *F1 score* yang konsisten menunjukkan bahwa kinerja model stabil di kedua *dataset*. Sebaliknya, kelas M1b menunjukkan kinerja yang sangat baik dengan *precision* dan *recall* yang sangat tinggi (0,86 dan 0,95 untuk validasi, 0,85 dan 0,96 untuk uji), menghasilkan *F1 score* yang juga tinggi (0,90 untuk validasi dan uji), yang menandakan bahwa model sangat efektif dalam mengidentifikasi kelas ini. Kelas M1c juga menunjukkan kinerja yang kuat dengan *precision* dan *recall* yang tinggi (0,90 dan 0,84 untuk validasi, 0,88 dan

0,82 untuk uji), meskipun sedikit lebih rendah dibandingkan M1b. *F1 score* yang baik dan konsisten menunjukkan keseimbangan yang baik antara *precision* dan *recall*, serta kemampuan model untuk menggeneralisasi dengan baik. Secara keseluruhan, model memiliki kinerja terbaik pada kelas M1b dan M1c, tetapi untuk kelas M0 dan M1a, meskipun *precision* cukup tinggi, *recall* yang rendah menunjukkan bahwa model kesulitan dalam mengidentifikasi semua instansi relevan, yang dapat menjadi masalah jika tujuan utama adalah meminimalkan *false negatives*.

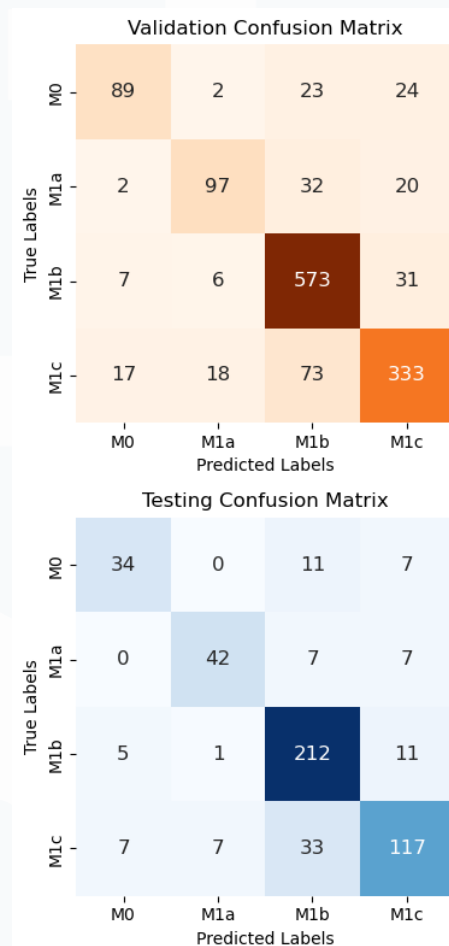
4.1.2.2 Pengujian Model DeiT Menggunakan Data *Filter Gaussian*



Gambar 4.16 Grafik Akurasi dan Loss Model DeiT Menggunakan *Dataset* dengan *Gaussian Filter*

Gambar 4.16 memperlihatkan grafik akurasi dan grafik *loss* dari hasil iterasi pelatihan yang dilakukan dengan menggunakan model DeiT dengan *dataset* yang telah diaplikasikan *Gaussian Filter*. Pada grafik akurasi, terlihat bahwa akurasi pelatihan meningkat secara konsisten dari sekitar 40% hingga mencapai 90% pada *epoch* ke-35, menandakan model mampu memahami pola data pelatihan dengan baik. Sementara itu, akurasi validasi cenderung fluktuatif di awal pelatihan, dan stabil di kisaran 70-80%, yang masih lebih rendah dibandingkan akurasi pelatihan. Grafik *loss* menunjukkan penurunan *train loss* secara signifikan dari sekitar 2 menjadi sekitar 0.2 pada *epoch* ke-35. Namun, *validation loss* menunjukkan pola yang lebih fluktuatif dengan nilai yang lebih tinggi dibandingkan *train loss* dan cenderung stabil di kisaran 0.6-0.8.

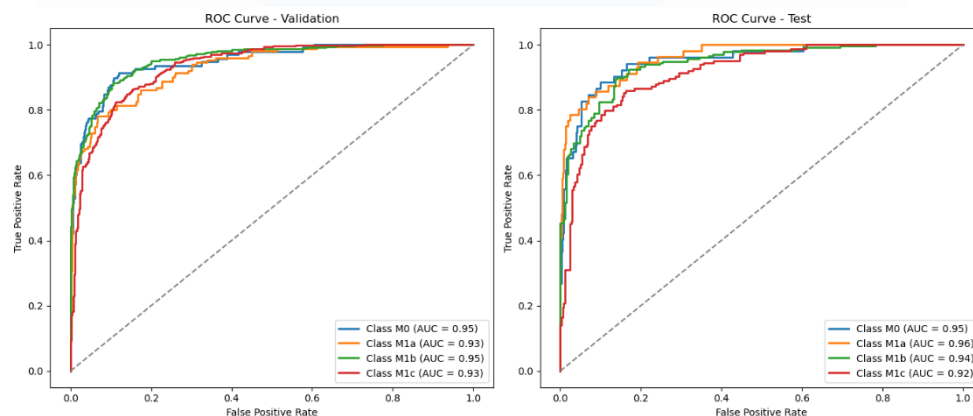
Perbedaan signifikan antara akurasi dan *loss* pada data pelatihan dan validasi ini mengindikasikan adanya *overfitting*. Secara keseluruhan, meskipun model belajar dengan baik pada data pelatihan, performa pada data validasi masih belum optimal.



Gambar 4.17 *Confusion Matrix* Model DeiT Menggunakan *Dataset* dengan *Gaussian Filter*

Analisis *confusion matrix* dari Gambar 4.17 menunjukkan bahwa model memiliki kinerja yang baik dalam mengklasifikasikan kelas M1b dan M1c, dengan jumlah *true positives* yang tinggi pada kedua kelas ini (573 untuk M1b dan 333 untuk M1c) pada data validasi. Namun, kesulitan muncul pada kelas M0 dan M1a, di mana model mengalami sejumlah kesalahan klasifikasi yang signifikan. Untuk kelas M0, model salah mengklasifikasikan 23 sebagai M1b dan 24 sebagai M1c, sementara untuk kelas M1a, model salah mengklasifikasikan 32 sebagai M0, 13 sebagai M1b, dan 3 sebagai M1c. Pada

kelas M1b, model salah mengklasifikasikan 31 sebagai M1c, sementara pada kelas M1c, 33 kesalahan klasifikasi terjadi dengan M1b. Pada data pengujian, tren serupa terlihat, meskipun kinerja model sedikit menurun, terutama pada kelas M0 dan M1a. Kesalahan klasifikasi untuk kelas M0 melibatkan 8 yang terklasifikasikan sebagai M1b dan 30 sebagai M1c. Untuk kelas M1a, model salah mengklasifikasikan 11 sebagai M1b dan 8 sebagai M1c. Pada kelas M1b, kesalahan terjadi dengan 2 yang terklasifikasikan sebagai M1a dan 24 sebagai M1c, sedangkan pada kelas M1c, 3 salah terklasifikasikan sebagai M1a dan 18 sebagai M1b. Secara keseluruhan, meskipun model telah menunjukkan kinerja yang cukup baik, terutama dalam mengenali M1b dan M1c, masih ada potensi untuk meningkatkan kemampuan model dalam mengurangi kesalahan klasifikasi, khususnya antara M0 dan kelas lainnya, melalui perbaikan pada representasi fitur atau penggunaan teknik regularisasi dan augmentasi data untuk mengatasi masalah *overfitting* dan meningkatkan kinerja pada data pengujian.



Gambar 4.18 ROC Curve Model DeiT Menggunakan Dataset dengan Gaussian Filter

Analisis Kurva ROC dari Gambar 4.18 menunjukkan kinerja model klasifikasi multi-kelas yang baik pada kedua set data, yaitu set validasi dan set pengujian. Pada set validasi, kurva ROC untuk keempat kelas—M0, M1a, M1b, dan M1c—relatif dekat dengan sudut kiri atas plot, yang menunjukkan kinerja model yang baik dalam membedakan antar kelas. Nilai AUC untuk kelas M0, M1a, M1b, dan M1c masing-masing adalah 0,95, 0,93, 0,95, dan

0,93, yang mengindikasikan bahwa model memiliki kemampuan yang sangat baik dalam membedakan antara kelas-kelas tersebut. Di sisi lain, pada set pengujian, model menunjukkan kinerja yang serupa dengan set validasi. Nilai AUC untuk masing-masing kelas pada set pengujian adalah 0,95 untuk MO, 0,96 untuk M1a, 0,94 untuk M1b, dan 0,92 untuk M1c. Hasil ini menunjukkan bahwa model dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya, tanpa menunjukkan tanda-tanda *overfitting*. Secara keseluruhan, hasil analisis ini menunjukkan bahwa model memiliki daya diskriminasi yang baik dan kinerja yang konsisten antara data pelatihan dan data pengujian, sehingga model ini dapat dipertimbangkan sebagai kandidat yang baik untuk diterapkan pada lingkungan nyata.

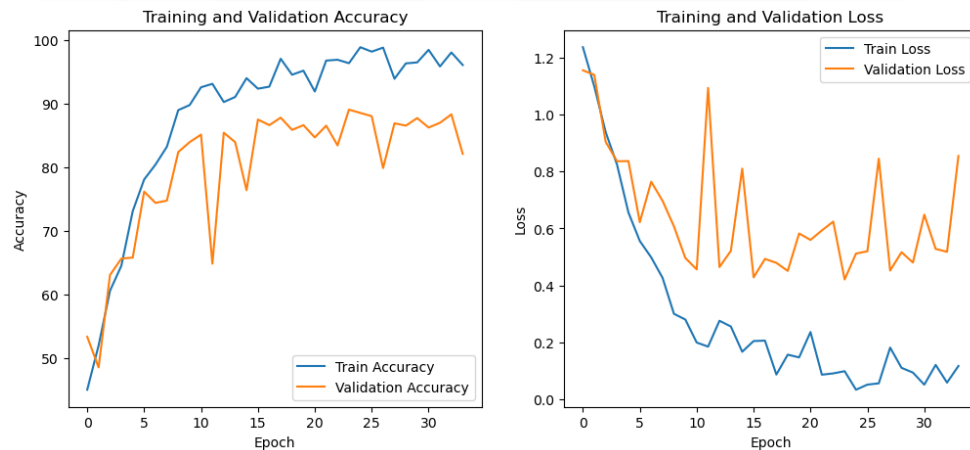
Tabel 4.7 Tabel Evaluasi Model DeiT menggunakan *Dataset* dengan *Gaussian Filter*

Class	<i>Precision</i>		<i>Recall</i>		<i>F-1 Score</i>	
	Val	Test	Val	Test	Val	Test
M0	0.77	0.74	0.64	0.65	0.70	0.69
M1a	0.79	0.84	0.64	0.75	0.71	0.79
M1b	0.82	0.81	0.93	0.93	0.87	0.86
M1c	0.82	0.82	0.76	0.71	0.78	0.76

Tabel 4.7 menunjukkan kinerja model klasifikasi untuk empat kelas (M0, M1a, M1b, M1c) berdasarkan metrik *Precision*, *Recall*, dan *F1 Score* mengungkapkan variasi dalam efektivitas masing-masing kelas. Kelas M0 memiliki *precision* yang cukup tinggi (0,77 pada validasi dan 0,74 pada *test*), tetapi *recall*-nya relatif rendah (0,64 pada validasi dan 0,65 pada *test*), yang menunjukkan bahwa meskipun model ini mampu menghasilkan prediksi positif yang akurat, ia kurang efektif dalam menemukan semua kasus relevan. Kelas M1a menunjukkan peningkatan kinerja pada *dataset* uji dengan *precision* 0,84 dan *recall* 0,75, yang menghasilkan *F1 Score* yang lebih tinggi (0,79 pada *test*), menandakan model ini bekerja lebih baik pada data yang belum terlihat sebelumnya. Kelas M1b tampil sebagai kelas dengan kinerja terbaik, dengan *precision* 0,82 dan *recall* 0,93 di kedua *dataset*, menghasilkan *F1 Score* yang sangat solid (0,87 pada validasi dan 0,86 pada *test*),

menjadikannya pilihan paling efektif dalam mengidentifikasi *true positives*. Sementara itu, kelas M1c memiliki *precision* yang tinggi (0,82 pada validasi dan *test*), tetapi *recall*-nya lebih rendah (0,76 pada validasi dan 0,71 pada *test*).

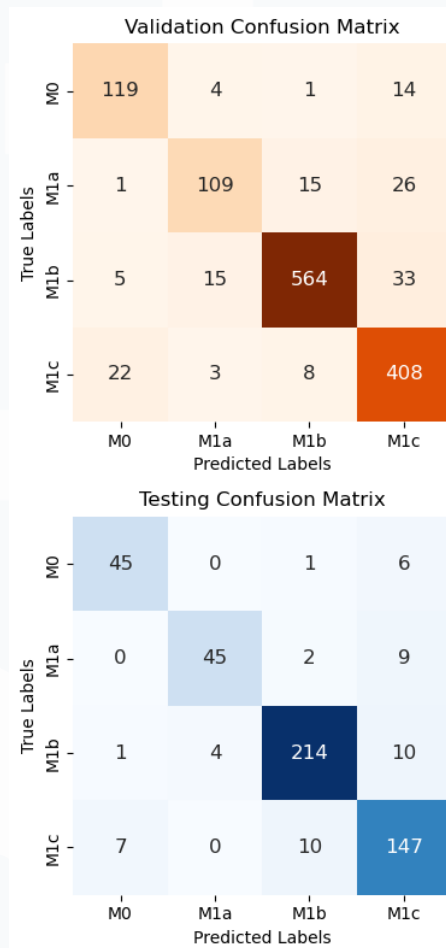
4.1.2.3 Pengujian Model DeiT Menggunakan Data *Filter Sobel*



Gambar 4.19 Grafik Akurasi dan *Loss* Model DeiT Menggunakan *Dataset* dengan *Sobel Operator*

Gambar 4.19 menunjukkan grafik akurasi dan *loss* yang diperoleh dari pelatihan model DeiT dengan *dataset* yang telah diberi *Sobel Filter* menunjukkan beberapa temuan penting. Pada grafik akurasi, terlihat bahwa akurasi pelatihan meningkat secara konsisten, dimulai dari sekitar 45% dan mencapai 95% pada *epoch* ke-30, yang menunjukkan bahwa model berhasil mempelajari pola data pelatihan dengan baik. Sebaliknya, akurasi validasi cenderung fluktuatif, terutama di awal pelatihan, dan akhirnya stabil di kisaran 80-90%, yang lebih rendah dibandingkan dengan akurasi pelatihan. Pada grafik *loss*, *loss* pelatihan menunjukkan penurunan yang signifikan dari 1.2 menjadi sekitar 0.1 pada *epoch* ke-30, yang menunjukkan bahwa model berhasil meminimalkan kesalahan prediksi pada data pelatihan. Namun, *loss* validasi memiliki pola yang lebih fluktuatif dan lebih tinggi dibandingkan dengan *loss* pelatihan, meskipun akhirnya stabil di sekitar 0.5. Secara keseluruhan, kedua grafik ini menunjukkan bahwa meskipun model DeiT berhasil mempelajari data pelatihan dengan baik, terdapat tanda-tanda *overfitting*, yang terlihat dari perbedaan signifikan antara akurasi dan *loss* pada data pelatihan dan validasi.

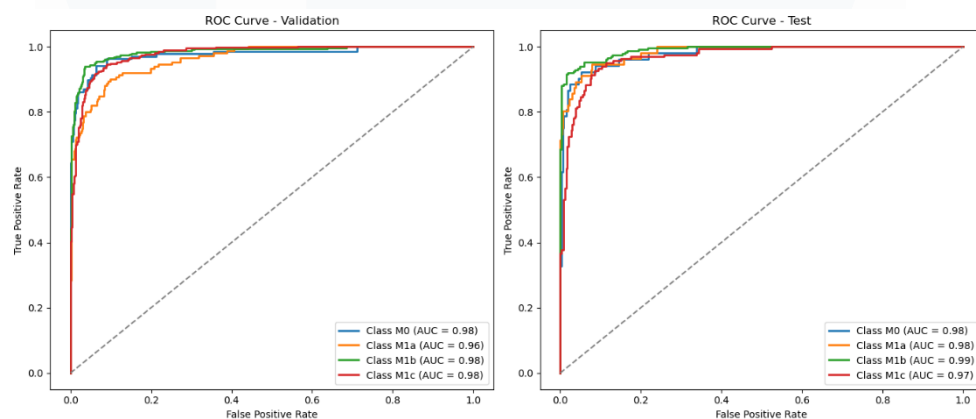
Hal ini mengindikasikan bahwa model perlu dimodifikasi untuk menghindari *overfitting* dan meningkatkan kemampuannya dalam menggeneralisasi pada data yang belum pernah dilihat sebelumnya.



Gambar 4.20 *Confusion Matrix* Model DeiT Menggunakan *Dataset* dengan *Sobel Operator*

Analisis *confusion matrix* dari Gambar 4.20 menunjukkan bahwa model berfungsi dengan baik di semua kelas pada *dataset* validasi dan pengujian, dengan akurasi yang tinggi secara keseluruhan. Pada set validasi, model berhasil mengklasifikasikan 119 data M0, 109 data M1a, 564 data M1b, dan 408 data M1c dengan benar, meskipun ada beberapa kesalahan klasifikasi, terutama M1c yang salah diidentifikasi sebagai M0 (22 data). Pada set pengujian, model sekali lagi menunjukkan akurasi tinggi, berhasil mengklasifikasikan 45 data M0, 45 data M1a, 214 data M1b, dan 147 data M1c dengan benar, dengan kesalahan klasifikasi yang minimal. Model sangat baik

dalam mengklasifikasikan M1b, dengan hanya 1 kesalahan klasifikasi di validasi dan 6 di pengujian, dan M1c juga diklasifikasikan dengan akurat, meskipun ada beberapa data M1c yang salah diklasifikasikan sebagai M1b (8 di validasi dan 4 di pengujian) atau M0 (22 di validasi dan 7 di pengujian). Meskipun kesalahan klasifikasi untuk M0 dan M1a minimal, area utama untuk perbaikan adalah kesalahan klasifikasi M1c sebagai M0. Hal ini menunjukkan bahwa model bisa mendapatkan keuntungan dari penyempurnaan lebih lanjut atau eksplorasi arsitektur yang berbeda untuk membedakan kelas-kelas ini dengan lebih baik. Secara keseluruhan, model menunjukkan hasil yang menjanjikan dengan akurasi tinggi, tetapi dapat ditingkatkan dengan mengatasi kesalahan klasifikasi spesifik ini.



Gambar 4.21 ROC Curve Model DeiT Menggunakan Dataset dengan Sobel Operator

Analisis kurva ROC untuk data validasi dan data uji dari Gambar 4.21 menunjukkan bahwa model bekerja sangat baik di semua empat kelas (M0, M1a, M1b, dan M1c). Untuk data validasi, skor AUC adalah sebagai berikut: M0 = 0,98, M1a = 0,96, M1b = 0,98, dan M1c = 0,98. Pada data uji, skor AUC tetap tinggi: M0 = 0,98, M1a = 0,98, M1b = 0,99, dan M1c = 0,97. Hasil ini menunjukkan bahwa model menunjukkan kinerja yang sangat baik dan kemampuan generalisasi yang kuat, karena kinerjanya pada set uji mirip dengan set validasi. Nilai AUC yang tinggi di semua kelas menunjukkan kemampuan model untuk membedakan antara kelas-kelas tersebut dengan efektif.

Tabel 4.8 Tabel Evaluasi Model Deit Menggunakan *Dataset* dengan *Sobel Operator*

Class	<i>Precision</i>		<i>Recall</i>		<i>F-1 Score</i>	
	Val	Test	Val	Test	Val	Test
M0	0.81	0.85	0.86	0.87	0.84	0.86
M1a	0.83	0.92	0.72	0.80	0.77	0.86
M1b	0.96	0.94	0.91	0.93	0.94	0.94
M1c	0.85	0.85	0.93	0.90	0.89	0.88

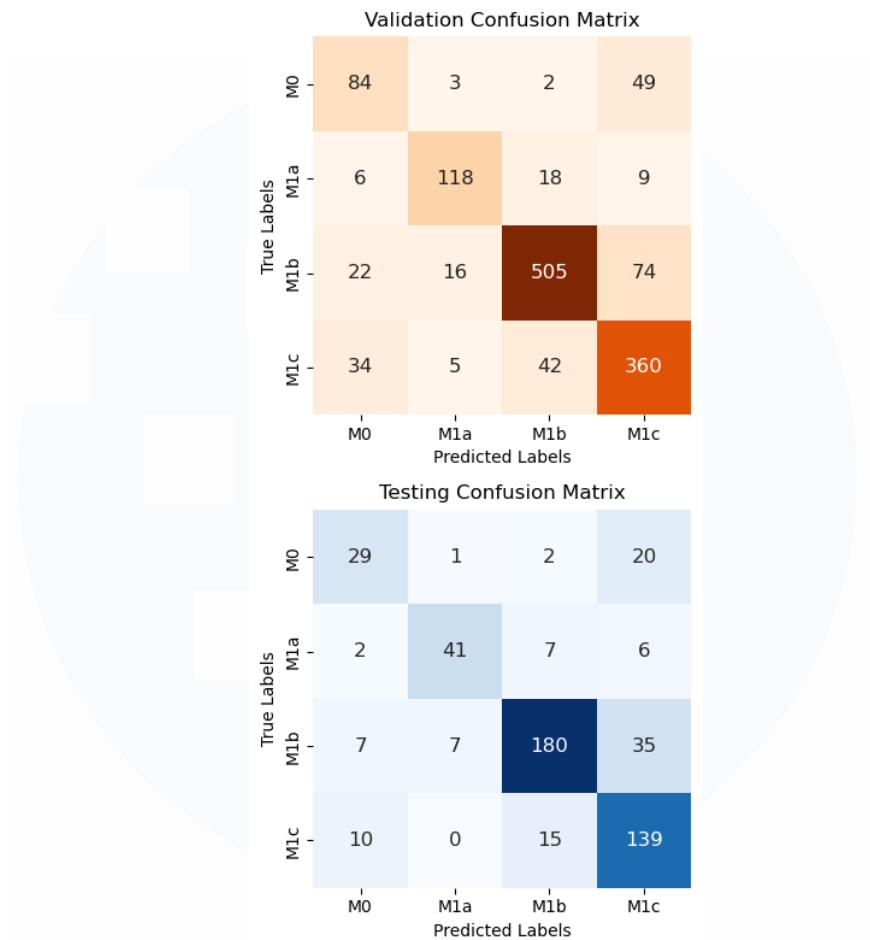
Berdasarkan Tabel 4.8 yang berisikan analisis metrik *Precision*, *Recall*, dan *F1 Score* untuk dua set data (*Validation* dan *Test*) pada masing-masing kelas (M0, M1a, M1b, M1c), dapat disimpulkan bahwa M1b menunjukkan performa terbaik secara keseluruhan, dengan nilai *precision*, *recall*, dan *F1 Score* yang tinggi dan konsisten antara *validation* dan *test* (*Precision Val*: 0.96, *Recall Val*: 0.91, *F1 Score Val*: 0.94; *Precision Test*: 0.94, *Recall Test*: 0.93, *F1 Score Test*: 0.94). M1a, meskipun memiliki *precision* yang tinggi (*Precision Val*: 0.83, *Precision Test*: 0.92), menunjukkan *recall* yang lebih rendah (*Recall Val*: 0.72, *Recall Test*: 0.80), yang mengindikasikan bahwa model ini lebih selektif dalam memprediksi kelas positif, sehingga *F1 Score*-nya juga terpengaruh (*F1 Score Val*: 0.77, *F1 Score Test*: 0.86). Kelas M0 dan M1c juga menunjukkan performa yang baik, dengan nilai *precision* dan *recall* yang relatif konsisten antara *validation* dan *test*, meskipun tidak setinggi M1b. Secara keseluruhan, meskipun model-model ini memiliki performa yang baik, ada ruang untuk perbaikan, terutama pada M1a dalam hal *recall* untuk meningkatkan keseimbangan antara *precision* dan *recall*.

4.1.2.4 Pengujian Model DeiT Menggunakan Data *Filter Gabor*



Gambar 4.22 Grafik Akurasi dan *Loss* Model DeiT Menggunakan *Dataset* dengan *Gabor Filter*

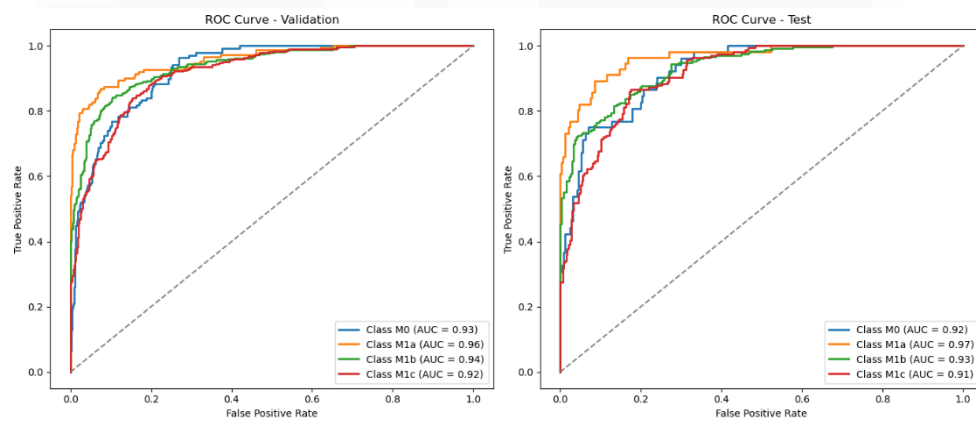
Gambar 4.22 menunjukkan grafik akurasi dan *loss* dari hasil pelatihan model DeiT menggunakan *dataset* yang telah diolah dengan *filter Gabor* menunjukkan performa yang baik. Akurasi pelatihan meningkat secara konsisten dari sekitar 45% hingga mencapai sekitar 85% pada *epoch* ke-30, menandakan bahwa model belajar dengan baik pada data pelatihan. Akurasi validasi awalnya fluktuatif, namun kemudian meningkat dan mendekati akurasi pelatihan pada *epoch* ke-30, yang menunjukkan bahwa model tidak mengalami *overfitting* yang signifikan. Pada grafik *loss*, terlihat penurunan signifikan pada *train loss* dari 1.1 menjadi sekitar 0.4, sementara *validation loss* juga menurun meskipun lebih fluktuatif. Perbedaan kecil antara *train loss* dan *validation loss* menunjukkan bahwa model tidak terlalu overfit pada data pelatihan. Secara keseluruhan, hasil ini mengindikasikan bahwa penggunaan *filter Gabor* pada *dataset* dapat membantu meningkatkan performa model DeiT tanpa menimbulkan masalah *overfitting* yang berarti.



Gambar 4.23 *Confusion Matrix* Model DeiT Menggunakan *Dataset* dengan *Gabor Filter*

Berdasarkan analisis *confusion matrix* dari Gambar 4.23, model ini menunjukkan kinerja yang kuat dalam memprediksi kelas M1b dan M1c, seperti yang dibuktikan dengan tingginya nilai *true positive* untuk kelas-kelas ini di kedua set validasi dan pengujian. Pada set validasi, *true positive* untuk M1b dan M1c masing-masing adalah 505 dan 360, sedangkan pada set pengujian, nilai-nilai ini adalah 180 dan 139. Model ini juga cukup baik dalam memprediksi M1a, dengan *true positive* sebanyak 84 pada set validasi dan 41 pada set pengujian, meskipun terdapat tingkat *false positive* yang lebih tinggi dibandingkan dengan kelas lainnya—yaitu 18 *false positive* untuk M1a pada set validasi dan 7 pada set pengujian. Namun, model ini kesulitan dalam memprediksi kelas M0 dengan akurat. Pada set validasi, nilai *true positive* untuk M0 hanya 49, dan pada set pengujian bahkan lebih rendah, yaitu 20.

Selain itu, tingkat *false negative* untuk M0 cukup tinggi, dengan 34 *false negative* pada set validasi dan 10 pada set pengujian, yang menunjukkan bahwa model sering mengklasifikasikan M0 sebagai kelas lain, terutama M1c (dengan 22 *false negative* pada set validasi dan 2 pada set pengujian). Secara keseluruhan, meskipun model ini menunjukkan potensi dalam memprediksi M1b dan M1c, perbaikan diperlukan dalam mengklasifikasikan M0 dan M1a untuk meningkatkan kinerja keseluruhannya.



Gambar 4.24 ROC Curve Model DeiT Menggunakan Dataset dengan Gabor Filter

Berdasarkan Gambar 4.24, model yang diuji menunjukkan kinerja yang baik pada kedua set data, yaitu set validasi dan set uji, dengan nilai AUC yang tinggi untuk semua kelas, yang semuanya berada di atas 0,9. Pada set validasi, nilai AUC untuk kelas MO adalah 0,93, kelas M1a 0,95, kelas M1b 0,94, dan kelas M1c 0,92. Sementara itu, pada set uji, nilai AUC untuk kelas MO sedikit menurun menjadi 0,92, kelas M1a meningkat menjadi 0,97, kelas M1b menurun sedikit menjadi 0,93, dan kelas M1c turun menjadi 0,91. Secara umum, model ini menunjukkan kinerja yang konsisten dalam membedakan kelas-kelas tersebut, dengan perbedaan nilai AUC yang relatif kecil antara set validasi dan set uji. Namun, ada sedikit penurunan kinerja pada set uji dibandingkan dengan set validasi, yang menunjukkan adanya potensi *overfitting*, terutama pada kelas M1a yang mengalami peningkatan AUC pada set uji. Meskipun demikian, model ini tetap menjanjikan dan menunjukkan kemampuan yang baik untuk aplikasi dunia nyata. Untuk memastikan

kemampuannya dalam menggeneralisasi, perlu dilakukan penyelidikan lebih lanjut terkait potensi *overfitting* dan eksplorasi teknik untuk meningkatkan kemampuan generalisasi model.

Tabel 4.9 Tabel Evaluasi Model Deit Menggunakan *Dataset* dengan *Gabor Filter*

Class	<i>Precision</i>		<i>Recall</i>		<i>F-1 Score</i>	
	Val	Test	Val	Test	Val	Test
M0	0.58	0.60	0.61	0.56	0.59	0.58
M1a	0.83	0.84	0.78	0.73	0.81	0.78
M1b	0.89	0.88	0.82	0.79	0.85	0.83
M1c	0.73	0.69	0.82	0.85	0.77	0.76

Berdasarkan analisis dari Tabel 4.9 yang menunjukkan metrik *Precision*, *Recall*, dan *F1 Score* untuk masing-masing kelas (M0, M1a, M1b, dan M1c) pada data validasi dan pengujian, dapat disimpulkan bahwa performa model bervariasi antar kelas. Kelas M0 menunjukkan performa yang relatif rendah, dengan nilai *Precision* dan *Recall* yang di bawah 0.61, mengindikasikan akurasi klasifikasi yang kurang baik dan kemampuan model yang terbatas dalam mendeteksi kelas ini. *F1 Score* yang mendekati 0.58 juga menunjukkan keseimbangan yang kurang optimal antara *Precision* dan *Recall*. Sebaliknya, kelas M1a memiliki performa yang lebih baik dengan *Precision* yang lebih tinggi (di atas 0.80), meskipun *Recall* yang sedikit lebih rendah (sekitar 73-78), menunjukkan ada sebagian contoh yang tidak terdeteksi. *F1 Score* untuk kelas M1a yang berkisar antara 0.78-0.81 menandakan keseimbangan yang lebih baik. Kelas M1b menunjukkan performa terbaik dengan *Precision* yang sangat tinggi (0.88-0.89) dan *Recall* yang juga cukup tinggi (0.79-0.82), menghasilkan *F1 Score* yang sangat baik (0.83-0.85), menandakan bahwa model ini sangat akurat dan efisien dalam mengklasifikasikan kelas ini. Sementara itu, kelas M1c memiliki *Precision* yang lebih rendah (0.69-0.73), namun diimbangi dengan *Recall* yang tinggi (0.82-0.85), yang menunjukkan bahwa meskipun ada beberapa kesalahan klasifikasi, model ini mampu mendeteksi sebagian besar contoh dari kelas ini. *F1 Score* untuk kelas M1c berkisar antara 0.76-0.77, menunjukkan keseimbangan yang cukup baik. Secara keseluruhan, M1b

adalah kelas yang paling dapat diandalkan, sementara M0 menunjukkan performa yang paling lemah. Kelas M1a dan M1c keduanya menunjukkan hasil yang baik, namun M1a lebih unggul dalam *Precision*, sementara M1c lebih unggul dalam *Recall*.

4.1.2.5 Tabel Hasil Akurasi dan Loss Model DeiT

Tabel 4.10 Tabel Hasil Akurasi dan Loss Model DeiT

Model	Image Processing	Epoch	Accuracy (%)			Loss		
			Train	Val	Test	Train	Val	Test
DeiT	-	33	93.85	86.12	85.43	0.1363	0.4629	0.0141
	<i>Gaussian</i>	36	94.23	80.03	80.84	0.1717	0.6830	0.0198
	<i>Sobel</i>	34	92.80	82.11	90.02	0.2025	0.8546	0.0122
	<i>Gabor</i>	32	89.77	76.84	77.64	0.2917	0.6293	0.0178

Berdasarkan analisis Tabel 4.10 yang menunjukkan performa model DeiT pada berbagai jenis pemrosesan citra, dapat disimpulkan bahwa pemrosesan tanpa *filter* memberikan hasil terbaik di semua metrik. Pada data latih, pemrosesan tanpa *filter* mencapai akurasi tertinggi (93.85%) dan memiliki *loss* terendah (0.1363), yang menunjukkan bahwa model dapat belajar dengan baik tanpa adanya pemrosesan tambahan. Sementara itu, pemrosesan *Gaussian*, meskipun memiliki akurasi latih yang tinggi (94.23%), menunjukkan *overfitting* dengan akurasi validasi dan pengujian yang lebih rendah. Pemrosesan *Sobel* memberikan hasil yang baik pada akurasi pengujian (90.02%), meskipun akurasi validasinya lebih rendah dibandingkan tanpa *filter*, menunjukkan kemampuannya dalam generalisasi data yang tidak terlihat. Sebaliknya, pemrosesan *Gabor* menunjukkan performa terendah di hampir semua metrik, dengan akurasi dan *loss* yang lebih buruk dibandingkan dengan pemrosesan lainnya, yang menunjukkan bahwa pemrosesan ini mungkin tidak cocok untuk *dataset* ini. Secara keseluruhan, pemrosesan tanpa *filter* tampaknya menjadi pilihan terbaik untuk model DeiT pada *dataset* ini, sedangkan pemrosesan *Gabor* perlu ditinjau atau dioptimalkan lebih lanjut. Hasil ini menunjukkan bahwa algoritma *Data-efficient Image Transformer*

tidak terlalu membutuhkan teknik *image processing* untuk membantu kinerja algoritma dalam mengklasifikasikan citra medis.

4.2 Pembahasan Hasil Perancangan

Tabel 4.11 Tabel Gabungan Hasil Akurasi dan Loss Model ViT dan DeiT

Model	Image Processing	Epoch	Accuracy (%)			Loss		
			Train	Val	Test	Train	Val	Test
ViT	-	62	91.14	79.81	79.04	0.2269	0.6198	0.0182
	Gaussian	53	94.33	83.82	74.65	0.1645	0.6114	0.0207
	Sobel	28	99.33	89.76	86.03	0.0255	0.5553	0.0178
	Gabor	49	90.54	78.62	73.65	0.2236	0.7029	0.0217
DeiT	-	33	93.85	86.12	85.43	0.1363	0.4629	0.0141
	Gaussian	36	94.23	80.03	80.84	0.1717	0.6830	0.0198
	Sobel	34	92.80	82.11	90.02	0.2025	0.8546	0.0122
	Gabor	32	89.77	76.84	77.64	0.2917	0.6293	0.0178

Berdasarkan analisis Tabel 4.11, model terbaik dapat ditentukan dengan mempertimbangkan akurasi pada data *test* serta nilai *loss* sebagai indikator stabilitas model. Untuk model ViT, penggunaan *image processing* dengan *sobel operator* menunjukkan hasil terbaik, dengan nilai akurasi paling tinggi dan *loss* paling rendah pada semua set data dibandingkan dengan teknik *image processing* lainnya. Sedangkan untuk model DeiT, DeiT tanpa *image processing* yang menunjukkan stabilitas dan nilai *loss* validasi yang lebih baik, dengan akurasi *test* 85.43% dan *loss test* terendah, yaitu 0.0141, serta *loss* validasi terbaik sebesar 0.4629.

Model *Vision Transformer* (ViT) dan *Data-efficient Image Transformer* (DeiT) menunjukkan hasil yang bervariasi tergantung pada jenis pemrosesan citra yang diterapkan. Secara umum, ViT menunjukkan kemampuan yang baik dalam mendeteksi metastasis kanker prostat, dengan akurasi pelatihan yang tinggi. Namun, model ini sering mengalami *overfitting*, terutama pada *dataset* tanpa pemrosesan citra atau dengan *filter Gabor*. Penggunaan *Sobel operator* pada ViT menghasilkan akurasi terbaik, yakni akurasi pelatihan sebesar 99.33% dan validasi 89.76%. *Filter Sobel* efektif dalam menyoroti tepi dan

fitur penting dalam citra, membantu model mengenali pola dengan lebih baik. Sebaliknya, tanpa pemrosesan citra, akurasi validasi menurun menjadi 79.81%, menunjukkan bahwa pemrosesan citra memiliki peran penting dalam meningkatkan performa model. Meski begitu, selisih besar antara akurasi pelatihan dan validasi mengindikasikan model terlalu terlatih pada data pelatihan dan kesulitan dalam generalisasi.

Dari analisis *confusion matrix*, ViT menunjukkan performa terbaik pada kelas M1b, namun kesulitan membedakan kelas M0 dan M1a. Hal ini kemungkinan disebabkan oleh ketidakseimbangan data atau kurangnya fitur yang relevan untuk membedakan kelas tersebut. Evaluasi dengan metrik *precision*, *recall*, dan *F1-score* juga menunjukkan bahwa kelas M1b memiliki performa terbaik, sementara kelas M0 menunjukkan kelemahan yang signifikan. Untuk meningkatkan kinerja model pada kelas M0, diperlukan augmentasi data atau penyesuaian bobot kelas agar model lebih sensitif terhadap pola yang kurang terwakili.

Di sisi lain, model DeiT menunjukkan kinerja yang lebih stabil dan konsisten dibandingkan ViT. Tanpa pemrosesan citra, model DeiT mencapai akurasi pelatihan 93.85%, dengan akurasi validasi 86.12%, menunjukkan kemampuan yang baik dalam mengidentifikasi pola pada data asli. Meskipun penggunaan *filter Gaussian* memberikan akurasi pelatihan tinggi (94.23%), performa validasi dan pengujian menunjukkan fluktuasi, mengindikasikan potensi *overfitting*. *Sobel operator* juga memberikan hasil baik pada DeiT, dengan akurasi pengujian mencapai 90.02%, meskipun akurasi validasinya lebih rendah. Seperti ViT, DeiT juga menunjukkan kelemahan dalam mengidentifikasi kelas M0, meskipun memiliki kemampuan generalisasi yang lebih baik untuk kelas M1b dan M1c.

Dari perbandingan antara kedua model, DeiT secara keseluruhan menunjukkan kinerja yang lebih stabil dan konsisten, terutama dalam hal kemampuan generalisasi pada data yang belum terlihat. Arsitektur DeiT terbukti lebih efisien dalam mempelajari representasi fitur yang relevan,

sementara ViT lebih rentan terhadap *overfitting*, khususnya ketika tidak menggunakan pemrosesan citra. Kedua model menunjukkan bahwa penggunaan teknik pemrosesan citra, seperti *Sobel operator*, dapat meningkatkan kinerja secara signifikan, sementara *filter Gabor* menunjukkan performa terendah pada keduanya.

Hasil evaluasi dari setiap model dalam penelitian ini menunjukkan bahwa model *Vision Transformer* dengan *image processing Sobel Filter* menghasilkan akurasi tertinggi dibandingkan dengan model lainnya. Model yang diperoleh dari percobaan ini akan dibandingkan dengan penelitian sebelumnya yang disajikan dalam Tabel 4.12 berikut.

Tabel 4.12 Perbandingan Hasil Penelitian Terdahulu

Model	Dataset	Image Processing	Test Accuracy
<i>Vision Transformers</i> (ViT) [13]	10,616 gambar	-	80.4%
<i>Vision Transformers</i> (ViT) [18]	Dataset gambar patologis prostat	Teknik pra-pemrosesan (crop, normalisasi) dan augmentasi data	76.7%
<i>Convolutional Neural Networks</i> (CNN) [16]	207 gambar	<i>Sobel Operator</i>	82.9%
Model yang diusulkan (<i>Vision Transformer + Sobel Operator</i>)	4986 gambar	<i>Sobel Operator</i>	86.03%

Tabel penelitian terdahulu menunjukkan perbandingan beberapa model *deep learning*, termasuk *Vision Transformers* (ViT) dan *Convolutional Neural Networks* (CNN), berdasarkan ukuran *dataset*, teknik pemrosesan gambar, dan akurasi pengujian. Dalam model *Vision Transformers* [13], digunakan *dataset* yang cukup besar yaitu 10,616 gambar, tanpa teknik pemrosesan gambar tambahan, yang menghasilkan akurasi sebesar 80.4%. Studi lain menggunakan *Vision Transformers* [18] dengan *dataset* gambar patologis prostat yang lebih

kecil serta menerapkan teknik pra-pemrosesan seperti cropping, normalisasi, dan augmentasi data, menghasilkan akurasi sebesar 76.7%. Di sisi lain, model CNN [16] menggunakan *dataset* kecil berisi 207 gambar dengan teknik *Sobel Operator* untuk deteksi tepi, mencapai akurasi yang lebih tinggi, yaitu 82.9%. Penelitian ini mengusulkan model yang menggabungkan *Vision Transformers* dengan *Sobel Operator* menggunakan *dataset* berukuran kecil, sebanyak 4986 gambar. Kombinasi ini menunjukkan hasil yang unggul dengan akurasi tertinggi, yaitu 86.03%. Hal ini menyoroti keunggulan mengintegrasikan teknik deteksi tepi seperti *Sobel Operator*, yang dapat meningkatkan kualitas fitur pada gambar, dengan kemampuan *Vision Transformers* dalam mempelajari representasi fitur kompleks.