UJIAN TENGAH SEMESTER
SAINS DATA GENOM

Disusun oleh:

Amira Shohifa (2206829130)

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS INDONESIA

2025

DAFTAR ISI

BAGIAN 1

Lakukan review dari masing satu artikel ilmiah terkini (3 tahun terakhir) yang melakukan differentially express (DE) Gene Analysis dan klasifikasi menggunakan Microarray dan RNA-Sequence technology.

A. Artikel Ilmiah (DE) Gene Analysis & Klasifikasi RNA-Sequence

Discovering common pathogenic processes between COVID-19 and HFRS by integrating RNA-seq differential expression analysis with machine learning

Sumber:
https://www.frontiersin.org/journals/microbiology/articles/10.3389/fmicb.2023.1175844/full

1. Pendahuluan

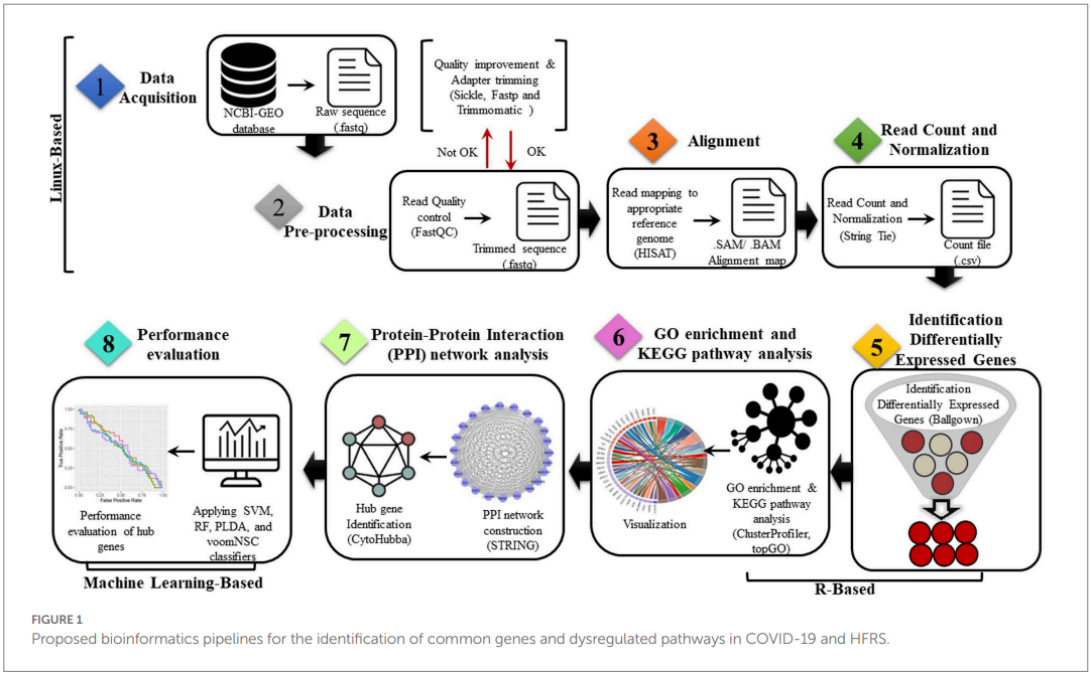
Hemorrhagic fever with renal syndrome (HFRS) merupakan penyakit zoonosis serius yang disebabkan oleh hantavirus, dengan penyebaran utama di kawasan Asia dan Eropa. Infeksi ini ditularkan melalui kontak dengan tikus atau ekskresinya, menyebabkan komplikasi berat seperti gagal ginjal dan pendarahan internal. Di sisi lain, coronavirus disease 2019 (COVID-19) yang disebabkan oleh SARS-CoV-2 telah menjadi pandemi global dengan tingkat kematian tinggi, terutama pada pasien dengan komorbiditas. Penelitian terkini menunjukkan adanya hubungan antara HFRS dan COVID-19, di mana individu yang menderita HFRS memiliki risiko lebih tinggi mengalami gejala COVID-19 yang parah. Namun, mekanisme molekuler di balik koinfeksi ini masih belum sepenuhnya dipahami. Oleh karena itu, diperlukan pendekatan baru untuk mengidentifikasi biomarker potensial dan memahami jalur biologis yang terganggu akibat kedua infeksi ini.

2. Tujuan Penelitian

Penelitian yang dilakukan oleh Noor et al. (2023) bertujuan untuk mengidentifikasi gen-gen yang diekspresikan secara berbeda (Differentially Expressed Genes/DEGs) dalam infeksi COVID-19 dan HFRS, menemukan jalur biologis bersama yang terganggu, serta mengevaluasi potensi gen-gen tersebut sebagai biomarker menggunakan pendekatan machine learning. Studi ini mengintegrasikan analisis RNA-Seq dan bioinformatika untuk memperoleh wawasan baru dalam memahami patogenesis koinfeksi.

3. Metode Penelitian

Data RNA-Seq diperoleh dari database Gene Expression Omnibus (GEO) dengan nomor akses GSE160351 dan GSE152418 untuk COVID-19, serta GSE158712 untuk HFRS. Proses analisis diawali dengan kontrol kualitas menggunakan FastQC, diikuti trimming menggunakan Sickle, Trimmomatic, dan FASTp. Setelah itu, data disejajarkan dengan genom referensi manusia menggunakan HISAT2, dan transkrip disusun menggunakan StringTie. DEGs diidentifikasi menggunakan paket Ballgown dengan kriteria p-value < 0.05 dan fold change > 1.0. Diagram alur pipeline bioinformatika yang digunakan dalam penelitian ini ditampilkan pada Gambar 1.



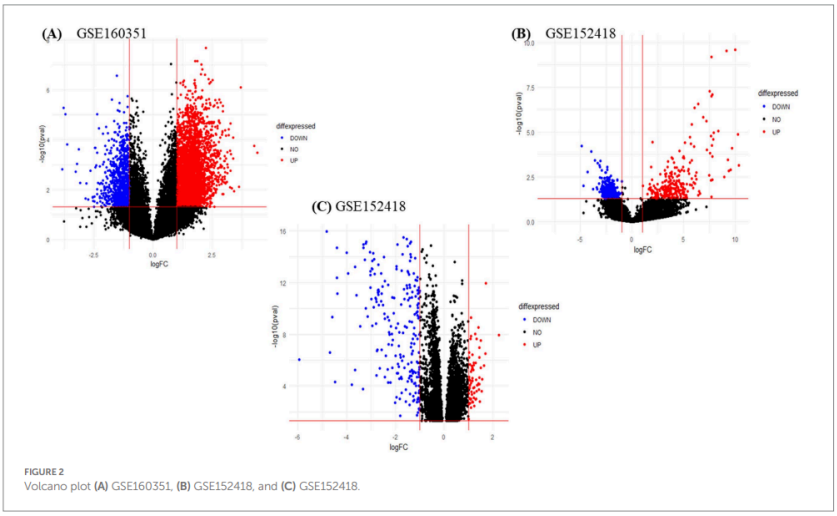
Gambar 1. Diagram pipeline analisis RNA-Seq dan bioinformatika untuk identifikasi DEGs, jalur biologis, dan validasi hub genes menggunakan machine learning.

Setelah identifikasi DEGs, analisis lebih lanjut dilakukan untuk menemukan gen yang tumpang tindih menggunakan diagram Venn, serta membangun jaringan interaksi protein (PPI) melalui database STRING dan Cytoscape. Gen-gen kunci atau hub genes kemudian divalidasi menggunakan empat algoritma klasifikasi: Random Forest (RF), Support Vector Machine (SVM), Poisson Linear Discriminant Analysis (PLDA), dan Voom-based Nearest Shrunken Centroids (voomNSC).

4. Hasil Penelitian

4.1 Screening of Differentially Expressed Genes (DEGs)

Analisis RNA-Seq dilakukan terhadap dua dataset COVID-19 dan satu dataset HFRS untuk mengidentifikasi gen-gen yang berbeda secara signifikan antara kelompok pasien dan kontrol sehat. Pada dataset COVID-19 (GSE160351 dan GSE152418), ditemukan total 1734 DEGs, dengan 1108 gen mengalami peningkatan ekspresi (upregulated) dan 626 gen mengalami penurunan ekspresi (downregulated). Sementara itu, pada dataset HFRS (GSE158712), identifikasi menghasilkan 630 DEGs, terdiri atas 240 gen upregulated dan 390 gen downregulated. Distribusi DEGs ini divisualisasikan menggunakan volcano plot, yang menggambarkan pemisahan yang jelas antara gen signifikan dan tidak signifikan di ketiga dataset. Distribusi gen yang mengalami perubahan ekspresi dalam masing-masing dataset divisualisasikan melalui volcano plot berikut.

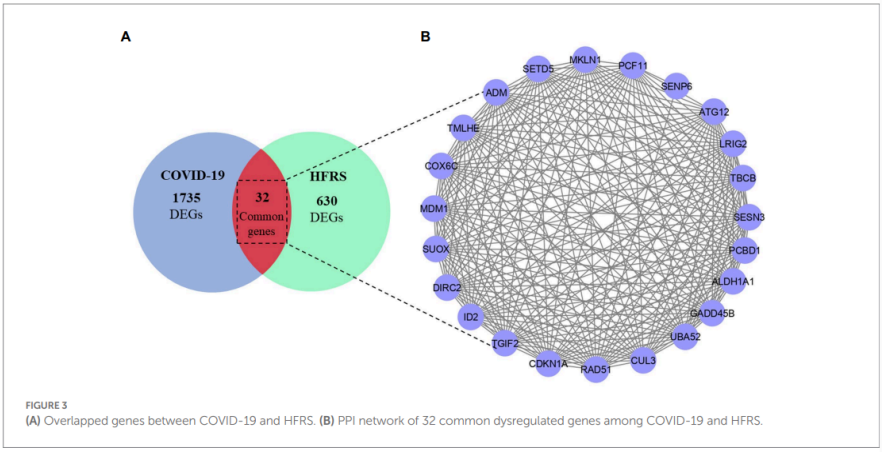


Gambar 2. Volcano plot dari dataset COVID-19 (GSE160351 dan GSE152418) dan HFRS (GSE158712), menunjukkan gen-gen yang mengalami peningkatan (merah) dan penurunan (biru) ekspresi secara signifikan.

4.2 Identifikasi Tanda Tangan Transkripsi Umum antara COVID-19 dan HFRS

Setelah DEGs diidentifikasi untuk masing-masing kondisi, dilakukan pencarian gen yang tumpang tindih. Hasil analisis Venn diagram menunjukkan adanya 32 gen yang terganggu pada kedua kondisi infeksi tersebut. Dari 32 gen ini, sebanyak 17 gen menunjukkan peningkatan ekspresi di kedua infeksi, sedangkan 15 gen lainnya mengalami penurunan ekspresi serupa.

Selanjutnya, gen-gen yang overlap ini digunakan untuk membangun jaringan interaksi protein (PPI) guna mengeksplorasi keterkaitan fungsional di antara mereka. Jaringan PPI yang dibangun mengindikasikan bahwa gen-gen ini memiliki hubungan erat dalam beberapa proses biologis penting. Gen-gen yang terganggu secara bersamaan antara COVID-19 dan HFRS diidentifikasi melalui analisis Venn diagram dan jaringan interaksi protein.



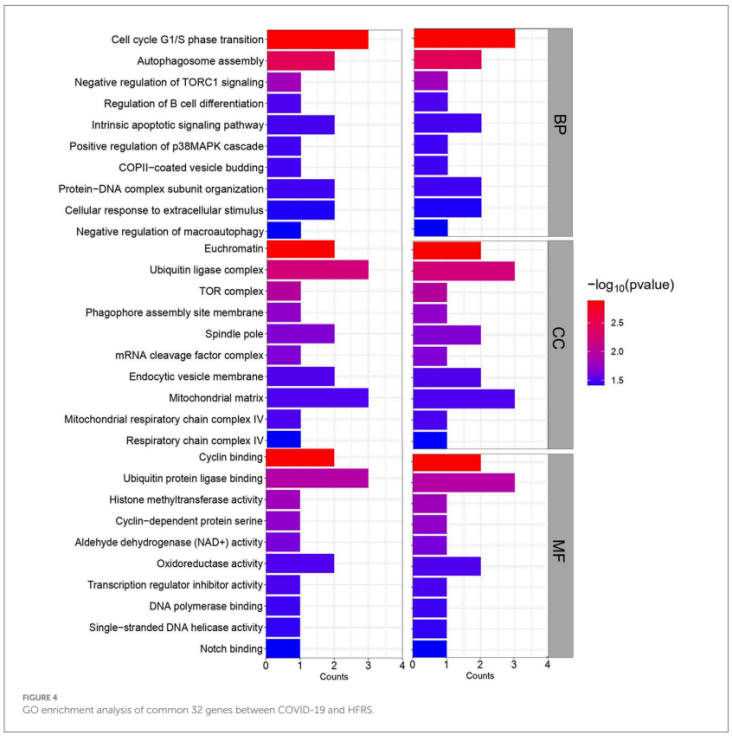
Gambar 3. (A) Diagram Venn menunjukkan 32 gen yang tumpang tindih antara infeksi COVID-19 dan HFRS. (B) Jaringan PPI (Protein–Protein Interaction) dari 32 gen umum yang memperlihatkan hubungan fungsional antar protein.

4.3 Pemahaman Biologis terhadap Gen Umum yang Ditemukan

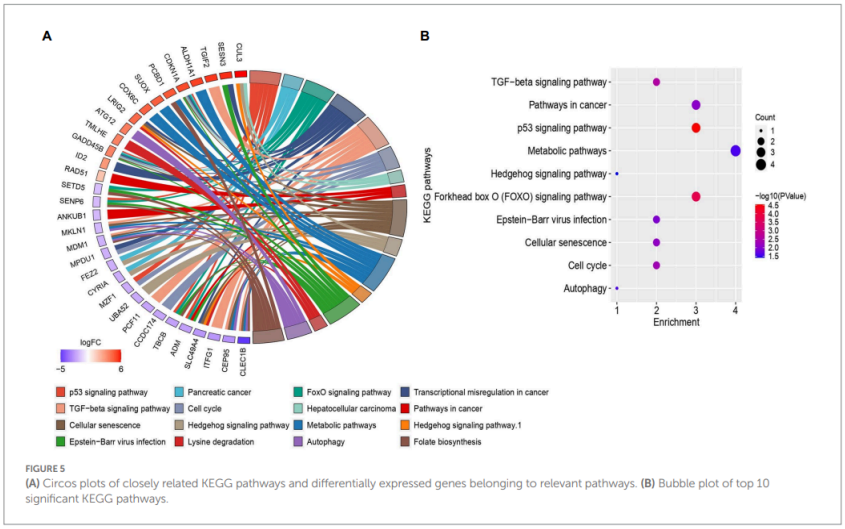
Analisis lanjutan dilakukan terhadap 32 gen umum menggunakan Gene Ontology (GO) dan KEGG pathway enrichment. Hasil GO enrichment menunjukkan bahwa dari sisi Biological Process (BP), gen-gen ini terlibat dalam proses seperti intrinsic apoptotic signaling, cell cycle G1/S phase transition, autophagosome assembly, hingga regulasi p38MAPK cascade. Untuk aspek Cellular Component (CC), gen-gen ini banyak berasosiasi dengan struktur-struktur seluler seperti protein-DNA complex dan vesicle coated membranes. Sedangkan dari segi Molecular Function (MF), gen-gen tersebut memperlihatkan keterlibatan dalam aktivitas seperti cyclin binding, ubiquitin ligase binding, dan DNA polymerase binding.

Pada analisis KEGG pathways, ditemukan bahwa gen-gen tersebut terutama terkait dengan jalur-jalur infeksi virus seperti Epstein-Barr virus infection, p53 signaling pathway, TGF-beta signaling pathway, serta jalur-jalur penting lainnya seperti cell cycle, cellular senescence, autophagy, dan pathways in cancer. Visualisasi hubungan gen dan jalur KEGG ini

diperlihatkan dalam circos plot dan bubble plot yang menyajikan gambaran komprehensif kontribusi masing-masing gen terhadap jalur biologis terkait.



Gambar 4. Hasil analisis GO enrichment terhadap 32 gen umum, menunjukkan keterlibatan dalam berbagai proses biologis, komponen seluler, dan fungsi molekuler.

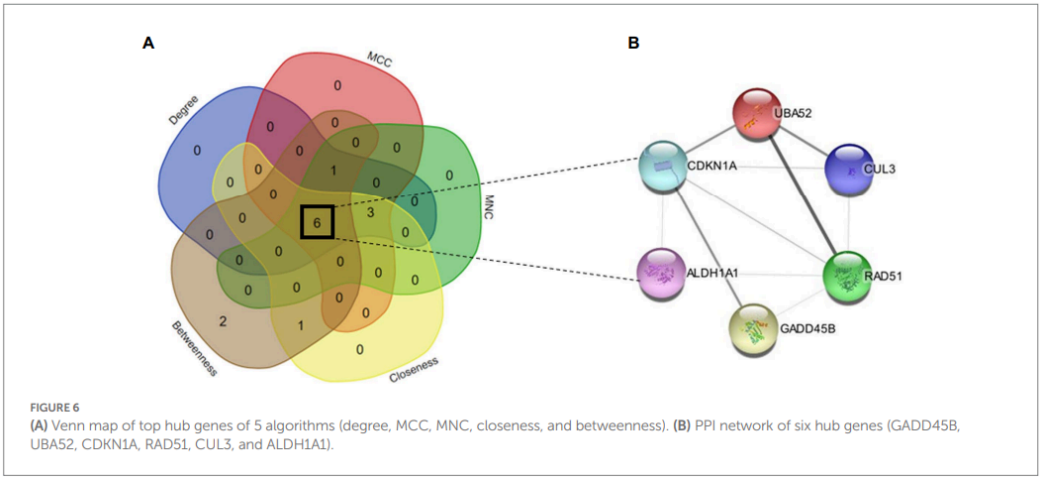


Gambar 5. (A) Circos plot yang memperlihatkan asosiasi antara gen-gen umum dengan jalur KEGG yang teridentifikasi. (B) Bubble plot dari 10 jalur KEGG paling signifikan berdasarkan analisis enrichment, terkait infeksi virus, siklus sel, dan respon imun.

4.4 Identifikasi Hub Genes

Untuk mempersempit kandidat gen kunci, jaringan PPI dianalisis lebih lanjut menggunakan lima algoritma topologi di Cytoscape melalui plugin CytoHubba. Algoritma-algoritma ini mencakup metode degree, MCC, MNC, closeness, dan betweenness. Melalui pendekatan ini, enam gen utama (hub genes) diidentifikasi, yaitu RAD51, ALDH1A1, UBA52, CUL3, GADD45B, dan CDKN1A. Gen-gen ini menunjukkan tingkat konektivitas yang tinggi dalam jaringan, menandakan peran sentralnya dalam proses patogenik gabungan antara COVID-19 dan HFRS.

Nilai fold change (log2FC) dan signifikansi statistik (p-value) untuk keenam hub genes ini menunjukkan konsistensi perubahan ekspresi yang signifikan di kedua dataset penyakit, menguatkan potensi mereka sebagai biomarker komorbiditas.



Gambar 6. (A) Venn diagram hasil integrasi lima algoritma topologi untuk mengidentifikasi enam hub genes yang paling sentral. (B) Visualisasi jaringan PPI dari enam hub genes utama: RAD51, ALDH1A1, UBA52, CUL3, GADD45B, dan CDKN1A.

4.5 Evaluasi Kinerja Klasifikasi Menggunakan Hub Genes

Untuk menilai kemampuan diskriminatif keenam hub genes tersebut, empat model klasifikasi machine learning diterapkan, yaitu Random Forest (RF), Support Vector Machine (SVM), Poisson Linear Discriminant Analysis (PLDA), dan Voom-based Nearest Shrunken Centroids (voomNSC). Data dibagi menjadi set pelatihan dan set pengujian (rasio 70:30), dan setiap model dievaluasi berdasarkan akurasi, sensitivitas, dan spesifisitas.

Hasil evaluasi menunjukkan bahwa semua model mencapai akurasi di atas 70%. Model Random Forest menunjukkan kinerja terbaik dengan akurasi 79,41%, sensitivitas 88,24%, dan spesifisitas 70,59%. Model voomNSC dan SVM juga memperlihatkan performa yang kompetitif dengan akurasi masing-masing sebesar 77,94% dan 76,47%. Sedangkan PLDA, meskipun memiliki akurasi terendah di antara keempat model (70,59%), menunjukkan spesifisitas tertinggi yaitu 76,47%.

Secara keseluruhan, temuan ini menegaskan bahwa kombinasi keenam hub genes memiliki potensi yang kuat sebagai biomarker prediktif untuk membedakan kondisi infeksi, dan pendekatan berbasis machine learning memberikan akurasi klasifikasi yang menjanjikan.

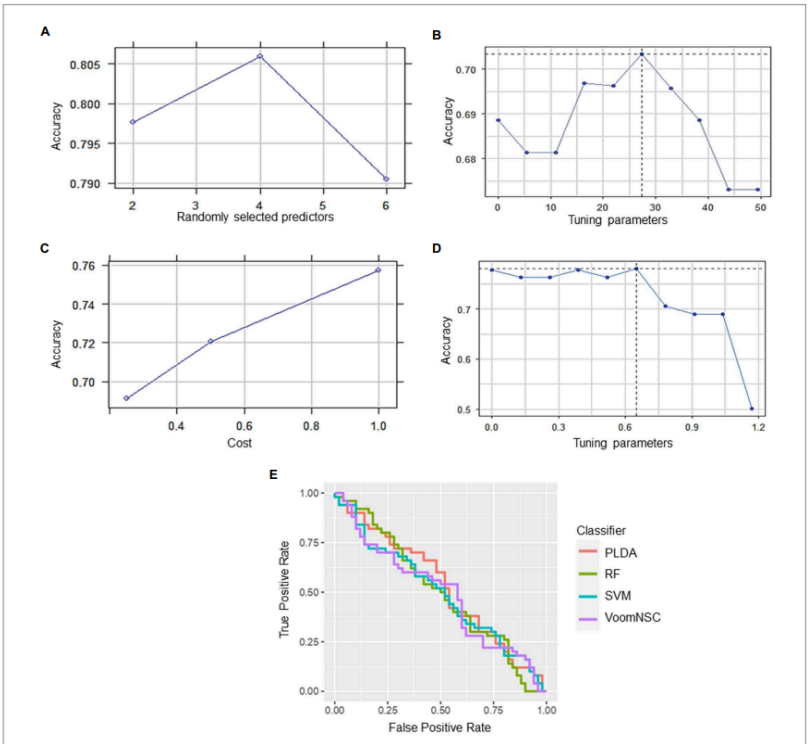


FIGURE 7
Classification performance of four different types of classifiers **(A)** RF, **(B)** PLDA, **(C)** SVM, and **(D)** voomNSC. **(E)** Receiver operating curve (ROC) plot of the four classifiers performance based on their accuracy, sensitivity, and specificity.

Gambar 7. Grafik evaluasi kinerja klasifikasi menggunakan Random Forest (RF), Poisson Linear Discriminant Analysis (PLDA), Support Vector Machine (SVM), dan Voom-based Nearest Shrunken Centroids (voomNSC).

5. Kesimpulan

Penelitian ini berhasil mengungkap sejumlah gen dan jalur biologis yang terganggu secara bersamaan pada infeksi COVID-19 dan HFRS. Temuan 32 gen tumpang tindih antara kedua penyakit memperlihatkan adanya kesamaan mekanisme molekuler, meskipun latar belakang patogen dari COVID-19 dan HFRS sangat berbeda. Analisis Gene Ontology memperjelas bahwa gangguan terjadi terutama pada proses yang berkaitan dengan regulasi siklus sel, apoptosis, dan respon imun, tiga aspek yang diketahui berperan besar dalam progresi infeksi virus.

Temuan dari analisis KEGG pathways semakin memperkuat dugaan ini. Jalur-jalur seperti p53 signaling, FOXO signaling, dan TGF-beta signaling merupakan jalur klasik yang terkait dengan kontrol pertumbuhan sel, stres oksidatif, dan modulasi respon inflamasi. Gangguan pada jalur ini berpotensi memperberat kerusakan jaringan pada pasien dengan ko-infeksi, seperti yang dilaporkan dalam beberapa studi kasus klinis sebelumnya.

Identifikasi enam hub genes — RAD51, ALDH1A1, UBA52, CUL3, GADD45B, dan CDKN1A — menjadi salah satu hasil utama yang penting untuk dicermati. Gen-gen ini bukan hanya sekadar terlibat dalam jaringan interaksi protein, tetapi juga memainkan peran sentral dalam regulasi molekuler infeksi. Misalnya, RAD51 yang terlibat dalam perbaikan DNA, atau CDKN1A yang berperan dalam kontrol siklus sel, masing-masing dapat berkontribusi terhadap ketidakstabilan seluler yang diinduksi oleh infeksi virus.

Evaluasi kinerja klasifikasi menggunakan machine learning juga memberikan hasil yang sangat menjanjikan. Dengan akurasi model di atas 70% untuk semua algoritma yang digunakan, khususnya Random Forest yang mencapai hampir 80%, studi ini menunjukkan bahwa kombinasi hub genes tersebut memiliki potensi kuat sebagai biomarker prediktif untuk membedakan kondisi infeksi.

Meskipun hasil yang diperoleh sangat mendukung hipotesis awal, penelitian ini tetap memiliki beberapa keterbatasan. Data yang digunakan terbatas pada analisis transkriptomik dari PBMCs, sehingga belum mencakup variabilitas ekspresi gen di jaringan lain yang mungkin juga relevan. Selain itu, validasi lebih lanjut menggunakan dataset eksternal atau pendekatan eksperimental berbasis laboratorium masih diperlukan untuk menguatkan temuan ini.

Secara keseluruhan, integrasi pendekatan RNA-Seq dengan analisis bioinformatika dan machine learning dalam studi ini memberikan wawasan baru mengenai potensi jalur-jalur patogenik bersama antara COVID-19 dan HFRS. Penemuan hub genes dan jalur biologis umum ini dapat menjadi pijakan awal untuk pengembangan strategi diagnostik yang lebih presisi, serta membuka peluang baru untuk terapi personalisasi pada pasien dengan risiko koinfeksi. Studi ini juga menegaskan bahwa pendekatan berbasis data tinggi seperti RNA-Seq, bila dipadukan dengan teknik machine learning yang tepat, mampu memperkaya pemahaman kita terhadap kompleksitas penyakit menular modern.

B. Artikel Ilmiah (DE) Gene Analysis & Klasifikasi Microarray

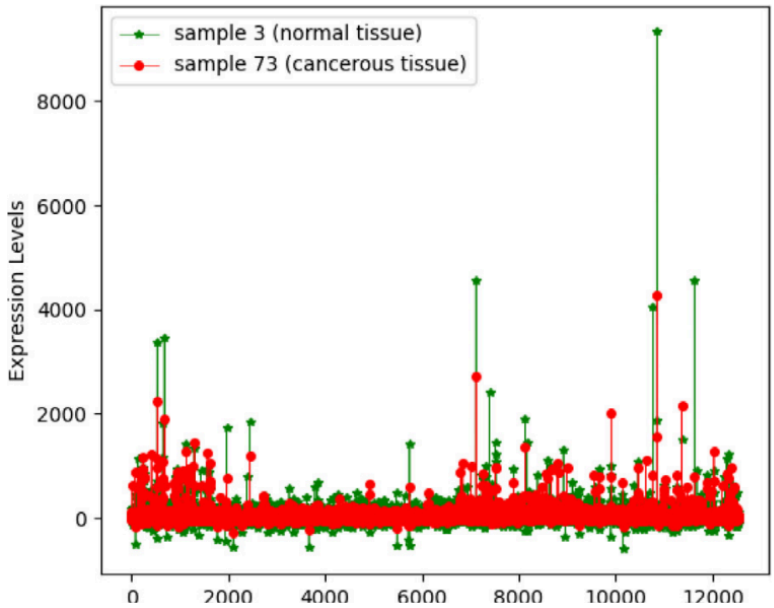
A two-phase gene selection method using anomaly detection and genetic algorithm for microarray data

Sumber: https://www.sciencedirect.com/science/article/abs/pii/S0950705122013454

1. Pendahuluan

Perkembangan teknologi microarray memungkinkan peneliti untuk mengukur ekspresi ribuan gen secara simultan dalam satu eksperimen. Teknologi ini membuka peluang besar dalam penelitian kanker, karena perubahan ekspresi hanya sebagian kecil gen dapat memicu terjadinya kanker. Namun, tantangan utama dari data microarray adalah jumlah fitur (gen) yang jauh lebih besar dibandingkan jumlah sampel, menyebabkan risiko overfitting dan akurasi klasifikasi yang rendah. Oleh karena itu, proses seleksi gen menjadi langkah krusial dalam pengolahan data microarray untuk meningkatkan performa model klasifikasi sekaligus menemukan biomarker penyakit yang relevan.

Untuk menggambarkan kompleksitas ekspresi gen dalam kondisi normal dan kanker, artikel ini menyajikan ilustrasi perbedaan level ekspresi gen antara sampel normal dan sampel kanker dari data prostat, seperti ditampilkan dalam Gambar 1.



Gambar 1. Perbandingan ekspresi gen antara sampel normal ke-3 dan sampel kanker ke-73 pada dataset prostat microarray.

2. Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan metode seleksi gen dua tahap yang efektif untuk data microarray, dengan mengombinasikan pendekatan anomaly detection dan genetic algorithm. Secara khusus, penelitian ini bertujuan untuk:

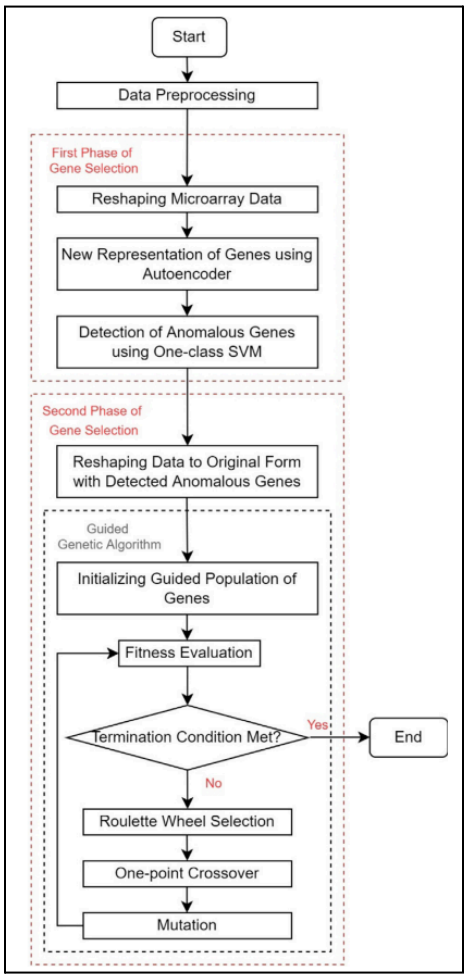
- Mengidentifikasi gen-gen penting (biomarker) yang berperan dalam klasifikasi kanker dan penyakit lainnya dengan pendekatan berbasis anomaly detection.
- Mengurangi jumlah fitur (gen) secara signifikan, mencapai pengurangan lebih dari 99%, untuk menghindari overfitting serta mempercepat proses analisis data.
- Meningkatkan akurasi klasifikasi dengan hanya menggunakan subset gen minimal yang relevan.
- Mengurangi risiko kompleksitas model akibat data berdimensi tinggi.
- Menggabungkan keunggulan pendekatan filter dan wrapper dalam satu framework seleksi gen yang efisien, cepat, dan tetap menghasilkan performa tinggi.

3. Metode Penelitian

3.1 Pendekatan dalam Seleksi Gen

Penelitian ini menggunakan pendekatan dua fase dalam seleksi gen, alur kerja keseluruhan metode ini digambarkan dalam Gambar 3.

- Fase pertama: Dilakukan transformasi data dengan memperlakukan gen sebagai "sampel". Kemudian dilakukan reduksi dimensi menggunakan autoencoder neural network dan dilanjutkan dengan deteksi anomali menggunakan One-Class SVM untuk menemukan gen-gen yang berbeda secara ekspresi.
- Fase kedua: Gen-gen hasil deteksi anomali disaring lebih lanjut menggunakan Genetic Algorithm berbasis guided initialization untuk menemukan kombinasi gen terbaik yang mendukung klasifikasi.



Gambar 3. Diagram alur kerja metode dua fase seleksi gen menggunakan anomaly detection dan genetic algorithm.

3.2 Data Preprocessing

Semua ekspresi gen dalam dataset mengalami transformasi logaritmik berbasis dua (log2 transformation) untuk menstabilkan variasi antar ekspresi dan memperjelas perbedaan ekspresi yang bermakna antar gen**.**

3.3 Fase Pertama: Seleksi Gen Berbasis Anomaly Detection

3.3.1 Representasi Data Baru

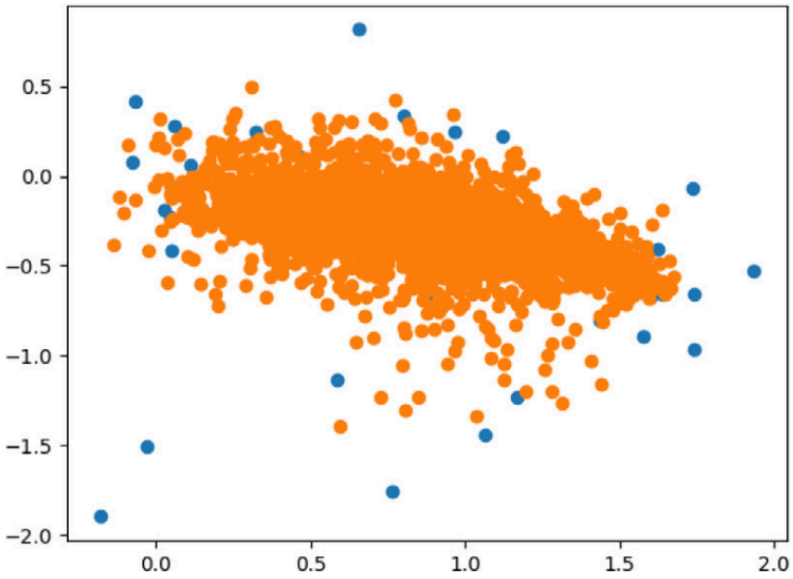
Dalam pendekatan ini, setiap gen dianggap sebagai "sampel" dan ekspresi gen across seluruh pasien menjadi fitur mereka.

3.3.2 Reduksi Dimensi dengan Autoencoder

Autoencoder neural network digunakan untuk mereduksi dimensi data gen hingga enam fitur per gen, guna memperjelas pola anomali.

3.3.3 Deteksi Anomali dengan One-Class SVM

One-Class SVM dengan kernel RBF digunakan untuk mendeteksi gen-gen anomali, dengan target 1% gen dari total populasi. Visualisasi hasil deteksi anomali pada dataset gastric ditunjukkan pada Gambar 4.



Gambar 4. Visualisasi 2D gen-gen anomali (warna biru) pada dataset gastric microarray

3.4 Fase Kedua: Seleksi Gen Lanjutan dengan Genetic Algorithm

3.4.1 Representasi dan Inisialisasi Kromosom

Setiap kromosom direpresentasikan sebagai array biner, dengan nilai 1 untuk gen terpilih dan 0 untuk gen tidak terpilih. Separuh populasi kromosom diinisialisasi berdasarkan variance score tertinggi.

3.4.2 Evaluasi Fitness

Fitness diukur berdasarkan akurasi klasifikasi Support Vector Machine (SVM) dengan 10-fold cross-validation.

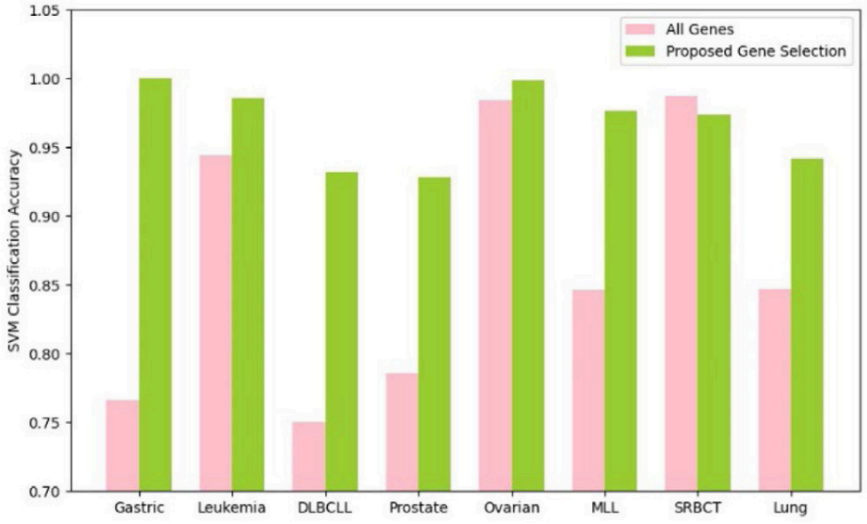
3.4.3 Operator Genetik

Seleksi dilakukan menggunakan Roulette Wheel Selection, crossover menggunakan single-point, dan mutasi dengan tingkat probabilitas 2%.

4. Hasil Penelitian

4.1 Performa Sebelum dan Sesudah Seleksi Gen

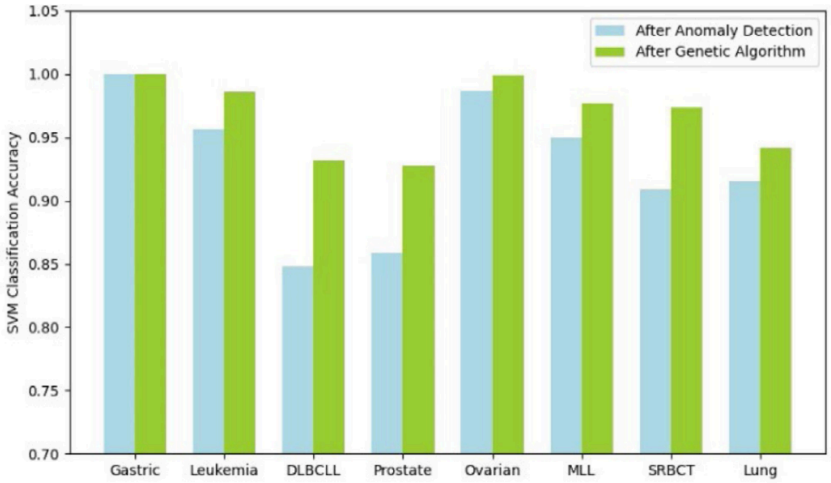
Penerapan metode ini pada delapan dataset microarray menunjukkan peningkatan akurasi klasifikasi yang signifikan setelah proses seleksi gen. Perbandingan akurasi sebelum dan sesudah seleksi divisualisasikan dalam Gambar 5.



Gambar 5. Perbandingan akurasi SVM sebelum dan sesudah proses seleksi gen.

4.2 Kontribusi Tiap Fase terhadap Peningkatan Akurasi

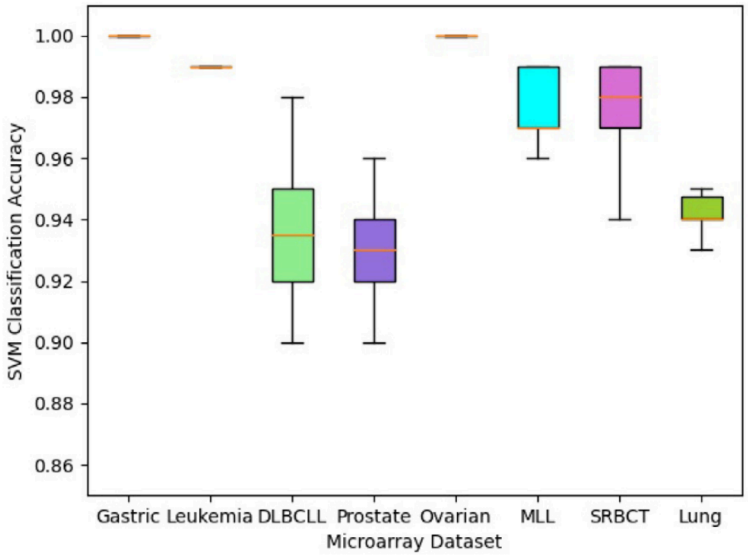
Analisis pada efek setiap fase terhadap akurasi menunjukkan bahwa, baik fase pertama (anomaly detection) maupun fase kedua (genetic algorithm) berkontribusi terhadap peningkatan akurasi klasifikasi, yang divisualisasikan pada Gambar 6.



Gambar 6. Perbandingan akurasi klasifikasi setelah fase pertama dan fase kedua seleksi gen.

4.3 Konsistensi Akurasi

Evaluasi dari 30 kali running independen menunjukkan stabilitas akurasi model, dengan fluktuasi akurasi yang minimal seperti yang divisualisasikan dalam Gambar 7.



Gambar 7. Boxplot distribusi akurasi klasifikasi pada semua dataset selama 30 kali running independen.

5. Kesimpulan

Penelitian ini berhasil mengembangkan metode dua fase seleksi gen untuk data microarray yang mengkombinasikan *anomaly detection* dan *genetic algorithm*. Dengan memandang gen sebagai data points dan mengurangi dimensinya terlebih dahulu menggunakan autoencoder, metode ini berhasil mengekstraksi gen-gen penting dengan efisiensi tinggi. Melalui dua tahap seleksi, jumlah gen dapat direduksi lebih dari 99% tanpa mengorbankan akurasi klasifikasi — bahkan meningkatkan akurasi di sebagian besar dataset.

Metode ini memperlihatkan keunggulan dalam mengurangi dimensi data dan mengatasi overfitting. Namun, terdapat potensi risiko kehilangan gen-gen penting pada fase deteksi anomali awal. Selain itu, aspek korelasi antar gen dalam subset akhir belum sepenuhnya dikontrol. Oleh karena itu, penelitian ini membuka peluang pengembangan lebih lanjut, terutama dalam mengintegrasikan kontrol korelasi antar gen dalam proses seleksi dan mengoptimalkan algoritma deteksi anomali untuk memperbaiki sensitivitasnya.

Secara keseluruhan, studi ini menawarkan solusi inovatif yang praktis dan efektif untuk masalah high-dimensionality dalam analisis data microarray dan membuka jalan bagi aplikasi lebih luas dalam studi biomarker dan diagnosis penyakit.

BAGIAN 2

Apakah perbedaan dari teknologi Microarray dan RNA-Sequence?

Dalam bidang biologi molekuler modern, memahami ekspresi gen menjadi kunci penting untuk menjelaskan berbagai mekanisme penyakit, respons obat, hingga proses perkembangan seluler. Dua metode utama yang sering digunakan untuk analisis ekspresi gen secara global adalah DNA microarray dan RNA sequencing (RNA-Seq). Meskipun keduanya bertujuan mengukur tingkat ekspresi gen, keduanya memiliki perbedaan fundamental dalam pendekatan teknis, sensitivitas, cakupan data, hingga biaya.

Microarray adalah metode berbasis hibridisasi di mana molekul cDNA yang berasal dari RNA sampel dikonversi, dilabeli dengan fluoresensi, dan diaplikasikan ke chip yang telah dilapisi ribuan probe spesifik gen. Ketika terjadi hibridisasi antara cDNA dan probe komplementer, sinyal fluoresensi yang dihasilkan mencerminkan tingkat ekspresi gen

Langkah kerja utama Microarray:

1. Ekstraksi RNA → konversi ke cDNA → labeling fluoresen.
2. Hibridisasi ke chip microarray berisi probe DNA.
3. Pencucian untuk menghilangkan hibridisasi non-spesifik.
4. Pembacaan sinyal fluoresen menggunakan scanner.
5. Analisis data ekspresi gen.

Sebaliknya, RNA-Seq adalah teknologi berbasis sekuensing generasi lanjut (Next-Generation Sequencing/NGS) yang men-sekuens seluruh RNA dalam sampel secara langsung, tanpa membutuhkan desain probe.

Langkah kerja utama RNA-Seq:

1. Ekstraksi RNA → fragmentasi → konversi ke cDNA.
2. Pemasangan adaptor pada cDNA.
3. Sequencing menggunakan platform NGS (seperti Illumina).
4. Pemetaan urutan ke referensi genom atau perakitan *de novo*.
5. Analisis kuantitatif ekspresi gen berdasarkan jumlah bacaan (read counts).

Aspek	Microarray	RNA-Seq
Cakupan	Terbatas pada gen/probe yang diketahui	Mencakup semua transkrip, termasuk baru
Sensitivitas	Sedang	Sangat Tinggi
Rentang Dinamis	Terbatas (~2 log)	Sangat luas (>5 log)
Penemuan Baru	Tidak Bisa	Bisa
Biaya per Sampel	Relatif Rendah	Lebih mahal
Analisis Data	Lebih Sederhana	Lebih Kompleks
Ketergantungan Anotas	Sangat bergantung	Tidak Bergantung
Kebutuhan Data	Lebih kecil, analisis relatif sederhana	Data besar, analisis bioinformatika kompleks

Perbandingan Detail

1. Cakupan (Coverage)
   - Microarray hanya bisa mendeteksi gen-gen yang sudah diketahui sebelumnya (terbatas pada probe yang sudah dirancang).
   - RNA-Seq mampu mendeteksi semua transkrip, termasuk transkrip baru, isoform splice alternatif, gen fusi, dan RNA non-coding

2. Sensitivitas dan Rentang Dinamis
   - RNA-Seq memiliki sensitivitas lebih tinggi dan rentang dinamis lebih luas dibanding microarray
   - Microarray kurang sensitif terhadap transkrip berabundansi rendah dan dapat menghasilkan noise dari latar belakang hibridisasi.

3. Resolusi Data
   - Microarray mengukur tingkat ekspresi berdasarkan intensitas fluoresensi, yang bisa bias akibat masalah seperti cross-hybridization.
   - RNA-Seq menghasilkan hitungan absolut dari setiap transkrip (read counts), yang secara kuantitatif lebih akurat.

4. Kemampuan Deteksi Transkrip Baru

- Microarray tidak bisa mendeteksi gen baru atau varian splicing yang tidak diwakili dalam probe.
- RNA-Seq memungkinkan penemuan novel genes, novel splice variants, SNPs, hingga RNA fusi

.

Kelebihan dan Kekurangan Microarray

Kelebihan:

- Teknologi matang dan protokolnya standar.
- Biaya relatif rendah per sampel.
- Analisis data lebih cepat dan lebih sederhana.
- Cocok untuk studi besar-besaran pada genom yang sudah teranotasi

Kekurangan:

- Terbatas pada transkrip yang sudah diketahui.
- Sensitivitas lebih rendah terhadap transkrip berabundansi rendah.
- Tidak mampu mendeteksi transkrip baru atau isoform splice.
- Rentang dinamis sempit.
- Potensi error dari cross-hybridization.

Kelebihan dan Kekurangan Microarray RNA-Seq

Kelebihan:

- Cakupan luas: mendeteksi semua jenis transkrip, termasuk novel transcripts, RNA non-coding, isoform splice alternatif, dan transkrip fusi.
- Sensitivitas tinggi.
- Rentang dinamis luas (deteksi transkrip sangat rendah dan sangat tinggi).
- Bisa dipakai untuk organisme non-model.
- Data dapat dianalisis ulang seiring update referensi genom.

Kekurangan:

- Biaya per sampel lebih tinggi.
- Memerlukan infrastruktur bioinformatika yang lebih kompleks.
- Proses analisis data lebih panjang dan membutuhkan skill computational tinggi.
- Potensi bias teknis dari library preparation.

Aplikasi dan Studi Kasus

Microarray tetap menjadi pilihan untuk studi yang fokus pada genom yang sudah dikenal, misalnya pada penelitian biomarker di manusia atau model organisme. Sementara itu, RNA-Seq lebih unggul dalam studi eksplorasi, seperti penelitian penyakit kompleks, respons obat baru, atau studi organisme tanpa anotasi genom lengkap. Penelitian oleh Kogenaru et al. (2012) menemukan bahwa RNA-Seq dan Microarray saling melengkapi: ada 28% target gen yang hanya bisa terdeteksi oleh salah satu metode. Ini menunjukkan pentingnya mempertimbangkan keunggulan masing-masing platform. Dalam konteks evaluasi toksikogenomik, seperti yang dilakukan oleh Rao et al. (2019), membuktikan bahwa RNA-Seq mendeteksi lebih banyak differentially expressed genes (DEGs) dibandingkan microarray saat mengevaluasi hepatotoksisitas pada tikus. RNA-Seq juga berhasil mengidentifikasi DEGs dari RNA non-coding, memberikan pemahaman mekanistik lebih dalam

Kesimpulan

Baik Microarray maupun RNA-Seq memiliki peran penting dalam analisis ekspresi gen. Pemilihan metode tergantung pada tujuan riset, jenis sampel, kebutuhan akan penemuan baru, serta ketersediaan sumber daya.

Secara umum, RNA-Seq menawarkan cakupan, sensitivitas, dan kedalaman informasi yang lebih tinggi dibandingkan microarray. Namun, microarray masih relevan untuk studi yang bersifat rutin, ekonomis, dan fokus pada gen-gen yang sudah teranotasi. Dalam banyak kasus, kedua metode ini tidak saling menggantikan, melainkan saling melengkapi, tergantung pada tujuan penelitian. Kombinasi keduanya, seperti yang diusulkan beberapa penelitian, dapat menghasilkan profil transkriptom yang paling komprehensif.

BAGIAN 3

Gunakan data publik untuk melakukan kegiatan Gene Expression Analysis. Kemudian buat laporan yang berisi:

a. Jelaskan tentang studi dari data yang digunakan.
b. Lakukan DE Gene analysis. Visualisasikan dan interpretasikan hasil yang didapat.
c. Bangun model untuk mengklasifikasi dari 2 atau lebih kelompok yang ada pada data tersebut. Bandingkan beberapa metode/algoritma klasifikasi yang ada, dengan menggunakan berbagai metrik pengukur performa klasifikasi.
d. Berikan penjelasan kekurangan, kelebihan dari setiap metode yang digunakan pada poin c.
e. Diskusikan seluruh hasil analisis yang dihasilkan, bandingkan dengan penelitian sebelumnya yang terkait dan tarik kesimpulan berdasarkan hasil yang ada.
f. Jelaskan seluruh proses yang dilakukan pada poin diatas, serta lampirkan R/Python code yang digunakan.

Penerapan Analisis Ekspresi Gen dan Klasifikasi Data pada Dataset Kanker Payudara GDS1329

I. Pendahuluan

Perkembangan ilmu biologi molekuler dalam beberapa dekade terakhir telah membuka banyak peluang untuk memahami mekanisme dasar berbagai penyakit, termasuk kanker. Salah satu pendekatan yang banyak digunakan dalam penelitian kanker adalah analisis ekspresi gen. Dengan menganalisis pola ekspresi gen di jaringan tumor, peneliti dapat mengidentifikasi gen-gen yang berperan penting dalam perkembangan penyakit serta membedakan berbagai subtipe kanker berdasarkan karakteristik molekuler.

Kanker payudara merupakan salah satu jenis kanker yang paling banyak terjadi di seluruh dunia. Penelitian tentang profil ekspresi gen pada kanker payudara tidak hanya membantu dalam klasifikasi jenis tumor, tetapi juga berpotensi memperbaiki diagnosis, prognosis, dan strategi pengobatan. Dalam project ini, dilakukan analisis terhadap data publik GDS1329 untuk memahami perbedaan ekspresi gen antara berbagai subtipe kanker payudara. Selain itu, model klasifikasi juga dibangun untuk memprediksi kelompok tumor berdasarkan data ekspresi gen, menggunakan beberapa metode machine learning untuk dibandingkan performanya.

Melalui pendekatan ini, diharapkan dapat diperoleh gambaran tentang gen-gen penting yang membedakan subtipe kanker payudara serta evaluasi metode klasifikasi yang paling efektif dalam memetakan ekspresi gen ke dalam kelas tumor yang tepat.

II. Dataset

Data yang digunakan dalam project ini adalah dataset publik dengan kode GDS1329, yang tersedia di Gene Expression Omnibus (GEO) milik NCBI. Dataset ini berjudul Molecular apocrine breast tumors dan berisi data hasil analisis ekspresi gen dari 49 sampel tumor payudara manusia (Homo sapiens). Sampel-sampel tersebut diklasifikasikan ke dalam tiga kelompok utama, yaitu luminal, basal, dan apocrine.

- Luminal tumors: ER+ (estrogen receptor positive) dan AR+ (androgen receptor positive)
- Basal tumors: ER- dan AR-
- Apocrine tumors: ER- dan AR+

Data dikumpulkan menggunakan platform Affymetrix Human Genome U133A Array (GPL96), dan nilai ekspresi yang tersedia dalam dataset telah melalui proses transformasi. Penelitian ini merujuk pada publikasi dari Farmer et al. (2005) dalam jurnal Oncogene, yang mengidentifikasi dan mengklasifikasikan molekular apocrine tumors melalui pendekatan microarray.

Berikut ringkasan karakteristik data:

- Organisme: Homo sapiens
- Jumlah sampel: 49 sampel tumor payudara
- Tipe data: Transformed count (data ekspresi gen yang sudah diproses)
- Sumber publikasi: Farmer P, Bonnefoi H, Becette V, Tubiana-Hulin M, et al. (2005) Oncogene.

Data ini menjadi dasar untuk analisis lebih lanjut dalam mengidentifikasi gen-gen yang berbeda ekspresinya antar kelompok tumor dan membangun model klasifikasi berdasarkan pola ekspresi tersebut.

III. Analisis Differentially Expressed (DE) Genes

3.1 Preprocessing Data Ekspresi Gen

Sebelum melakukan analisis ekspresi gen diferensial, dilakukan beberapa tahapan preprocessing untuk memastikan data yang digunakan berkualitas baik dan dapat diinterpretasikan secara tepat. Dataset GDS1329 diunduh dari GEO Database dan dikonversi menjadi format ExpressionSet agar lebih mudah diproses di R.

```
## 2. Download Data dari GEO
dtgeo <- getGEO('GDS1329', destdir = ".")
dtgeo

## 3. Convert ke ExpressionSet
eset <- GDS2eSet(dtgeo, do.log2 = TRUE)
eset
```

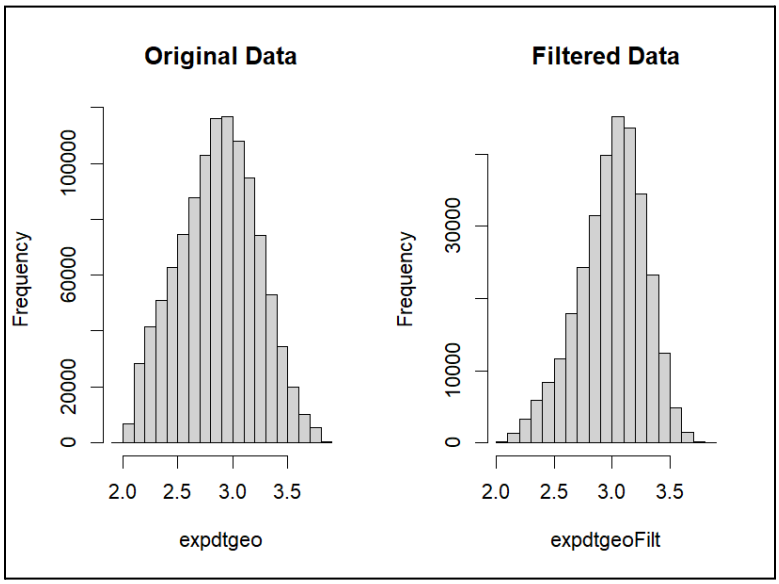
Selanjutnya, filtering data dilakukan dengan tujuan menghapus gen-gen yang memiliki variansi ekspresi yang sangat rendah di seluruh sampel, karena gen tersebut cenderung tidak memberikan informasi yang bermakna dalam perbandingan antar kelompok. Setelah proses filtering, jumlah gen yang dianalisis berkurang dari sekitar 22.283 menjadi 6.321 gen.

```
## 7. Filter Gene (Menghapus Gen Varians Rendah dll)
esetFilt <- nsFilter(eset)$eset
expdtgeoFilt <- exprs(esetFilt)
dim(expdtgeoFilt)
```

```
> dim(expdtgeo)
[1] 22283    49
```

```
> dim(expdtgeoFilt)
[1] 6321   49
```

Distribusi ekspresi gen sebelum dan sesudah filtering divisualisasikan melalui histogram. Dari grafik yang dihasilkan, terlihat bahwa distribusi data setelah filtering tetap simetris dan mengikuti pola distribusi normal, namun jumlah data menurun drastis karena hanya mempertahankan gen-gen yang variatif.



Gambar 1. Distribusi ekspresi gen pada data GDS1329 sebelum (kiri) dan sesudah (kanan) proses filtering variansi rendah. Setelah filtering, jumlah gen berkurang, namun pola distribusi ekspresi tetap simetris, menunjukkan bahwa data yang dipertahankan memiliki variabilitas yang cukup untuk analisis lebih lanjut.

3.2 Analisis Differential Expression

Setelah tahap preprocessing selesai, analisis differential expression dilakukan untuk mengidentifikasi gen-gen yang ekspresinya berbeda secara signifikan antara kelompok tumor apocrine, basal, dan luminal.

```
> table(vargrp)
vargrp
apocrine tumor    basal tumor  luminal tumor
             6             16             27
```

Pada tahap ini digunakan metode Limma (Linear Models for Microarray Data), yang merupakan salah satu pendekatan standar dalam analisis ekspresi gen, khususnya untuk data microarray seperti dataset GDS1329 ini.

Model linear dibangun dengan memperhitungkan perbedaan antara ketiga kelompok tumor, dan setiap gen dianalisis untuk menentukan apakah ada perubahan ekspresi yang signifikan. Hasil dari analisis ini disesuaikan dengan menggunakan metode koreksi multiple testing untuk mengendalikan tingkat false discovery rate (FDR).

Berdasarkan hasil analisis, diperoleh:

- Sebanyak 686 gen menunjukkan penurunan ekspresi (downregulated) pada grup 1.
- Sebanyak 935 gen menunjukkan peningkatan ekspresi (upregulated) pada grup 1.
- Sedangkan untuk grup 2, teridentifikasi 351 gen downregulated dan 514 gen upregulated.
- Tidak ditemukan gen yang tidak signifikan pada intercept model, mengindikasikan bahwa baseline grup memiliki ekspresi yang berbeda nyata.

```
> fit <- eBayes(lmFit(expdtgeoFilt, design))
> summary(decideTests(fit))
       (Intercept) group1 group2
Down             0    686    351
NotSig           0   4700   5456
Up            6321    935    514
```

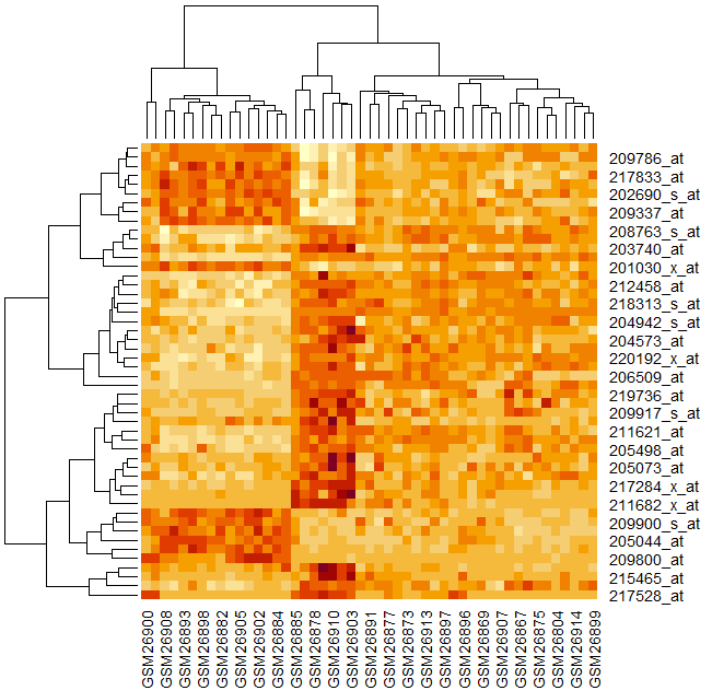
Gen-gen yang teridentifikasi sebagai differensial ini selanjutnya dianalisis lebih lanjut untuk melihat pola ekspresi dan hubungannya dengan klasifikasi tumor.

3.3 Visualisasi Hasil Differentially Expressed Genes

Heatmap Top Differentially Expressed Genes

Heatmap dibangun menggunakan ekspresi dari 50 gen teratas yang teridentifikasi sebagai DE genes berdasarkan nilai p-value terkecil. Setiap baris pada heatmap merepresentasikan satu gen, sedangkan setiap kolom merepresentasikan satu sampel tumor.

Dari heatmap yang dihasilkan, terlihat adanya pola ekspresi yang membedakan antara kelompok tumor. Sampel-sampel dengan ekspresi gen yang serupa cenderung mengelompok bersama, yang ditunjukkan dengan percabangan pohon (dendrogram) di bagian atas dan samping heatmap. Hal ini mengindikasikan bahwa ekspresi gen tertentu dapat digunakan untuk mengelompokkan jenis tumor secara alami.

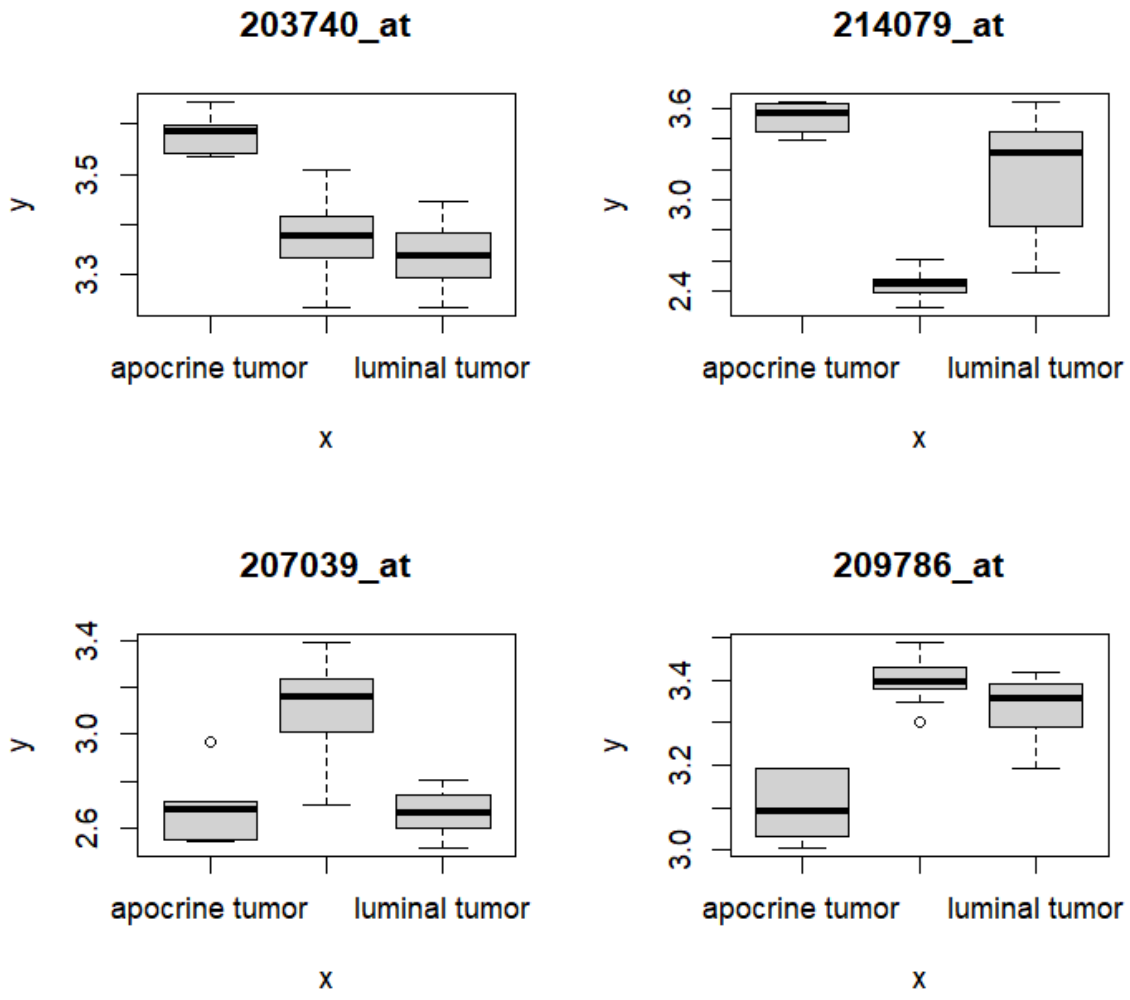


Gambar 2. Heatmap ekspresi 50 gen teratas yang berbeda secara signifikan antar sampel tumor. Setiap baris menunjukkan ekspresi satu gen, dan setiap kolom mewakili satu sampel. Pola clustering pada heatmap memperlihatkan adanya perbedaan ekspresi yang jelas antar kelompok.

Boxplot Ekspresi 4 Gen Teratas

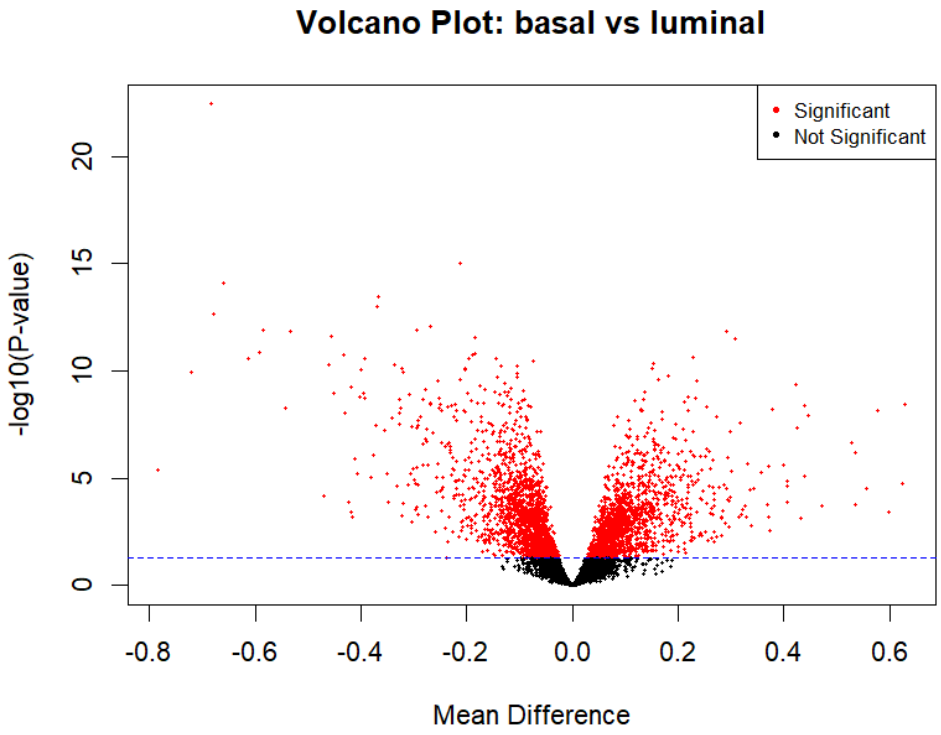
Selain heatmap, visualisasi ekspresi empat gen teratas yang memiliki perbedaan paling signifikan antar kelompok juga dilakukan menggunakan boxplot. Boxplot ini menunjukkan distribusi nilai ekspresi dari masing-masing gen pada kelompok tumor apocrine dan luminal.

Dari boxplot yang dihasilkan, dapat diamati bahwa gen-gen tersebut memiliki perbedaan ekspresi yang cukup jelas antara dua kelompok, memperkuat hasil analisis statistik sebelumnya.

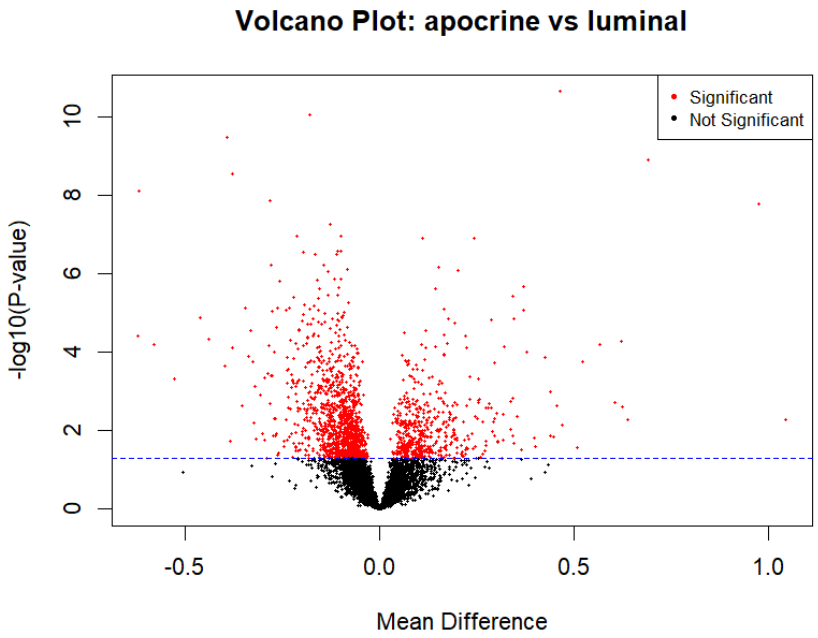


Gambar 3. Boxplot ekspresi empat gen teratas yang menunjukkan perbedaan ekspresi yang signifikan antara kelompok apocrine dan luminal. Setiap boxplot memperlihatkan rentang nilai ekspresi gen di masing-masing kelompok.

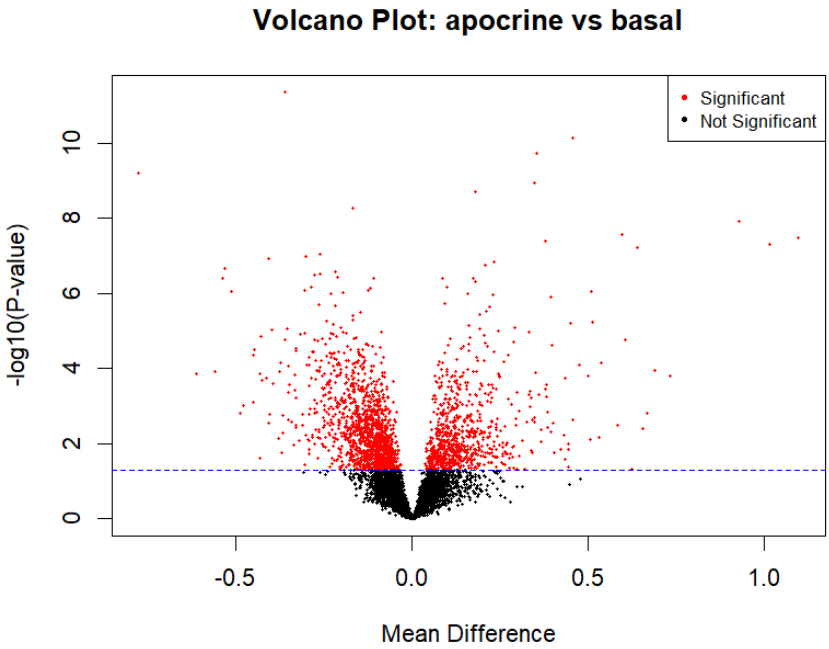
Visualisasi hasil analisis differential expression lebih lanjut dilakukan menggunakan volcano plot untuk ketiga perbandingan antar subtipe tumor, yaitu apocrine vs basal, apocrine vs luminal, dan basal vs luminal. Volcano plot menggambarkan hubungan antara perbedaan rata-rata ekspresi gen (mean difference) dengan tingkat signifikansi (-log10(p-value)). Titik berwarna merah menunjukkan gen-gen yang berbeda secara signifikan antara dua kelompok (p-value < 0.05), sedangkan titik hitam merepresentasikan gen yang tidak signifikan. Garis horizontal biru menunjukkan ambang batas signifikansi statistik pada p-value = 0.05.



Pada perbandingan basal vs luminal, terlihat jumlah gen yang signifikan sangat banyak, dengan pola penyebaran gen yang cukup simetris ke arah positif maupun negatif. Ini mengindikasikan adanya banyak perbedaan ekspresi antara dua subtipe ini.



Untuk perbandingan apocrine vs luminal, meskipun jumlah gen signifikan juga cukup besar, perbedaan ekspresi yang ekstrem lebih sedikit dibandingkan basal vs luminal. Ini mengisyaratkan bahwa apocrine memiliki ekspresi yang lebih mirip dengan luminal dibandingkan basal.



Sementara itu, pada apocrine vs basal, volcano plot menunjukkan distribusi gen signifikan yang lebih terbatas dibandingkan perbandingan lain. Hal ini menunjukkan bahwa perbedaan ekspresi gen antara apocrine dan basal lebih kecil dibandingkan dengan perbandingan lainnya.

Secara keseluruhan, volcano plot ini mengkonfirmasi bahwa pola ekspresi gen pada subtipe tumor basal cenderung lebih berbeda dibandingkan dengan subtipe apocrine maupun luminal, yang mendukung hasil analisis clustering dan klasifikasi sebelumnya.

3.4 Interpretasi Hasil

Berdasarkan hasil analisis differential expression menggunakan metode Limma, diperoleh sejumlah gen yang memiliki perubahan ekspresi signifikan di antara kelompok tumor apocrine, basal, dan luminal.

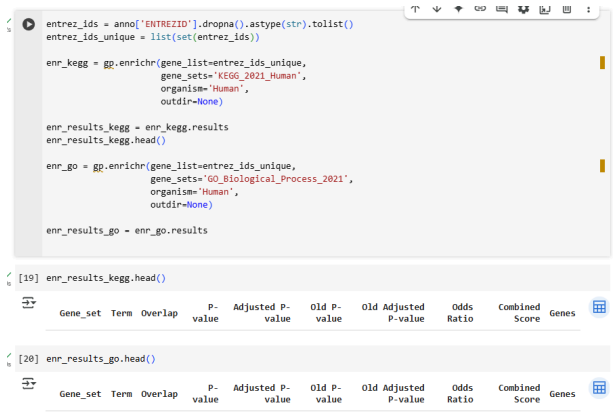
Secara keseluruhan, lebih dari 2000 gen menunjukkan pola ekspresi yang berbeda antara grup, dengan rincian ratusan gen mengalami peningkatan ekspresi (upregulated) dan ratusan lainnya mengalami penurunan ekspresi (downregulated).

	(Intercept)	group1	group2
Down	0	686	351
NotSig	0	4700	5456
Up	6321	935	514

Terlihat bahwa 686 + 935 + 351 + 514 = 2.486 perubahan ekspresi

Visualisasi heatmap dari 50 gen teratas memperlihatkan adanya pola clustering yang cukup konsisten, di mana sampel-sampel dari kelompok yang sama cenderung berkelompok bersama. Hal ini mengindikasikan bahwa ekspresi gen memang mampu membedakan jenis tumor yang ada.

Sementara itu, dari hasil boxplot terhadap empat gen teratas, terlihat jelas bahwa masing-masing gen memiliki distribusi ekspresi yang berbeda antar kelompok. Beberapa gen, seperti 203740_at dan 214079_at, menunjukkan perbedaan ekspresi yang cukup ekstrem, menguatkan temuan bahwa gen-gen tersebut berpotensi menjadi marker molekuler untuk membedakan subtipe tumor.



```
entrez_ids = anno['ENTREZID'].dropna().astype(str).tolist()
entrez_ids_unique = list(set(entrez_ids))

enr_kegg = gp.enrichr(gene_list=entrez_ids_unique,
                      gene_sets='KEGG_2021_Human',
                      organism='Human',
                      outdir=None)

enr_results_kegg = enr_kegg.results
enr_results_kegg.head()

enr_go = gp.enrichr(gene_list=entrez_ids_unique,
                    gene_sets='GO_Biological_Process_2021',
                    organism='Human',
                    outdir=None)

enr_results_go = enr_go.results
```

```
[19] enr_results_kegg.head()
```

	Gene_set	Term	Overlap	P-value	Adjusted P-value	Old P-value	Old Adjusted P-value	Odds Ratio	Combined Score	Genes

```
[20] enr_results_go.head()
```

	Gene_set	Term	Overlap	P-value	Adjusted P-value	Old P-value	Old Adjusted P-value	Odds Ratio	Combined Score	Genes

Selain melakukan analisis differential expression, dilakukan juga upaya untuk mengeksplorasi jalur biologis yang terkait melalui KEGG dan Gene Ontology enrichment analysis. Namun, berdasarkan hasil analisis menggunakan *Enrichr*, tidak ditemukan pathway yang signifikan. Hal ini kemungkinan disebabkan oleh jumlah gen yang terbatas setelah proses filtering dan spesifiknya karakteristik sampel pada dataset yang digunakan.

Hal ini kemungkinan disebabkan oleh jumlah gen yang terbatas setelah proses filtering dan karakteristik dataset yang spesifik pada tipe tumor tertentu. Secara umum, hasil analisis ini menunjukkan bahwa pendekatan differential expression efektif untuk menemukan sinyal biologis dalam data ekspresi gen, meskipun ada keterbatasan dalam memperluas analisis ke tingkat pathway pada dataset ini.

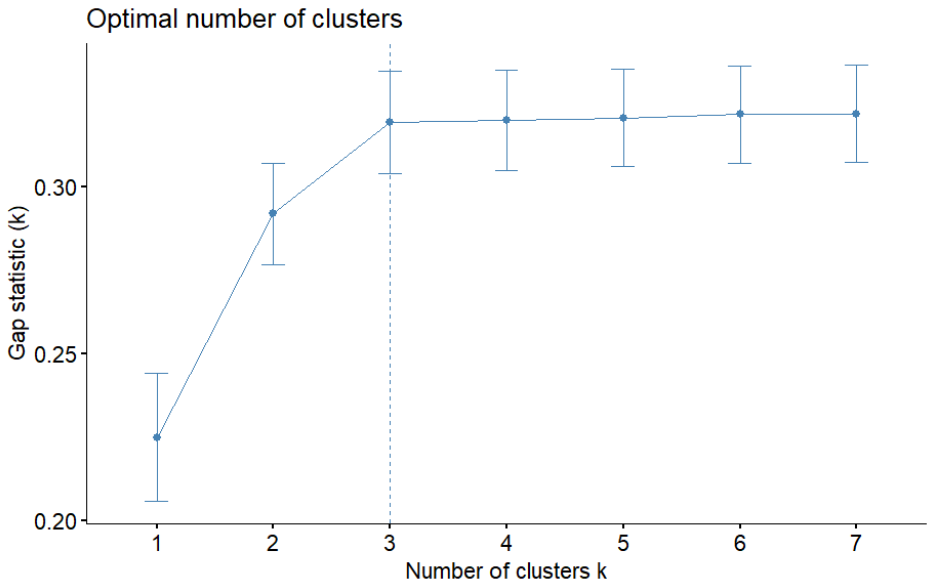
IV. Model Clustering

4.1 Penentuan Jumlah Klaster

Sebelum melakukan proses klastering, langkah pertama yang dilakukan adalah menentukan jumlah klaster yang optimal dari data yang dianalisis. Menentukan jumlah klaster yang tepat sangat penting karena akan mempengaruhi hasil pemisahan data dan interpretasi akhir.

Dalam analisis ini, metode Gap Statistic digunakan untuk membantu menentukan jumlah klaster yang optimal. Gap Statistic membandingkan perubahan dalam within-cluster dispersion data nyata dengan data acak, sehingga dapat membantu memilih jumlah klaster yang paling sesuai secara objektif.

Berdasarkan hasil visualisasi Gap Statistic, diperoleh bahwa nilai gap tertinggi terjadi pada k = 3. Hal ini mengindikasikan bahwa pembagian data ke dalam tiga klaster merupakan pilihan yang paling optimal untuk dataset ini. Pemilihan ini juga konsisten dengan informasi awal bahwa sampel dibagi menjadi tiga kelompok biologis utama: apocrine, basal, dan luminal.

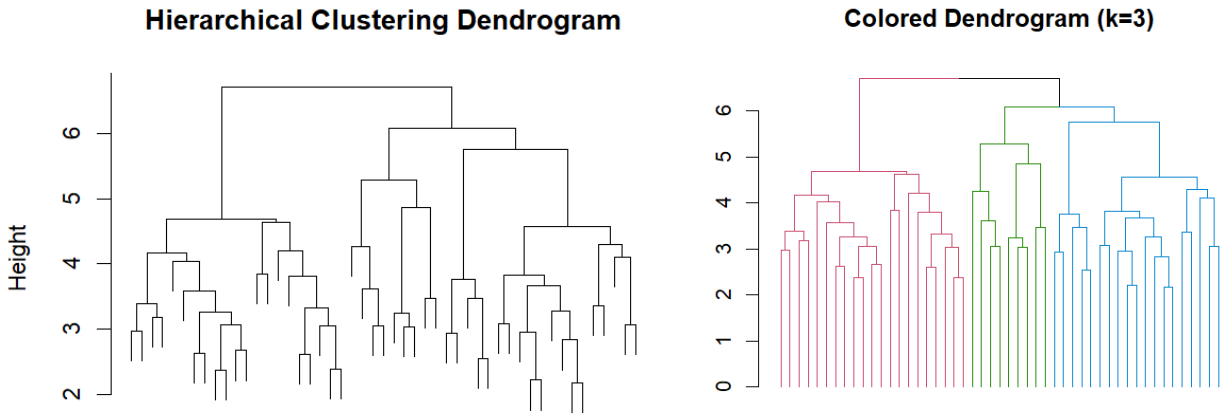


Gambar 4. Grafik Gap Statistic untuk menentukan jumlah klaster optimal. Nilai gap tertinggi diperoleh pada k = 3, menunjukkan bahwa tiga klaster merupakan jumlah yang paling sesuai untuk memisahkan data dalam analisis ini.

4.2 Perbandingan Hasil Clustering

Pada bagian ini, kita akan membandingkan hasil dari tiga metode clustering yang telah diterapkan, yaitu Hierarchical Clustering, K-Means Clustering, dan PAM Clustering. Perbandingan dilakukan dengan melihat hasil visualisasi clustering dan interpretasi hasil masing-masing metode.

Hasil Dendrogram menunjukkan hubungan antar sampel berdasarkan kemiripan ekspresi gen. Pada visualisasi colored dendrogram (k=3), data terpisah menjadi tiga klaster yang mewakili apocrine, basal, dan luminal. Struktur pohon ini memperlihatkan bagaimana sampel yang serupa secara ekspresi gen dikelompokkan bersama.



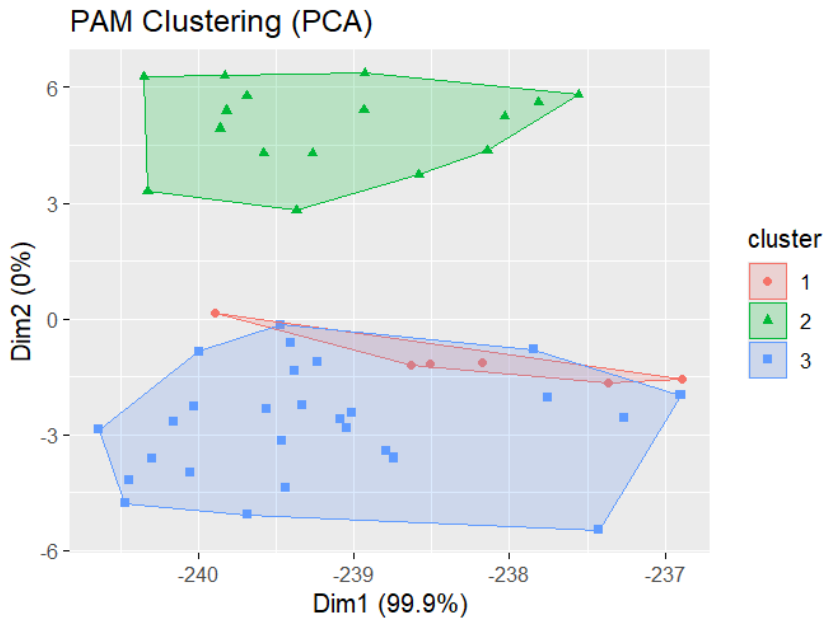
Gambar 5. Dendrogram hasil Hierarchical Clustering menunjukkan hubungan antara sampel tumor. Dengan k=3, hasil ini membagi sampel menjadi tiga klaster yang representatif untuk kelompok apocrine, basal, dan luminal.

PCA plot hasil K-Means clustering menunjukkan pemisahan data dalam tiga klaster (k=3), dengan masing-masing klaster terpisah berdasarkan centroid. Klaster pertama (apocrine) tampak lebih terisolasi dengan beberapa titik data yang jelas terpisah. Klaster kedua (basal) dan ketiga (luminal) sedikit lebih terdistribusi, namun tampaknya ada sedikit overlap antara keduanya. Ini menunjukkan bahwa metode K-Means memang lebih sensitif terhadap jumlah cluster (k) dan jenis distribusi data.

Gambar 6. PCA plot hasil K-Means Clustering menunjukkan pemisahan tiga klaster tumor.

PCA plot hasil PAM clustering menunjukkan hasil klaster yang serupa dengan K-Means, yaitu tiga klaster. Titik basal dan luminal hampir tidak ada overlap sama sekali, dan ini memperlihatkan stabilitas medoid yang digunakan dalam PAM.

Namun, seperti yang ditunjukkan di gambar, ada sedikit **o**verlap antara luminal dan apocrine — ini menunjukkan bahwa PAM mungkin lebih sensitif pada medoid centering dan bisa membuat cluster lebih terpusat (tapi beberapa titik mungkin masih ada yang lebih dekat ke cluster lain).



Gambar 7. PCA plot hasil PAM Clustering juga menunjukkan tiga klaster tumor yang terbagi

4.3 Biclustering

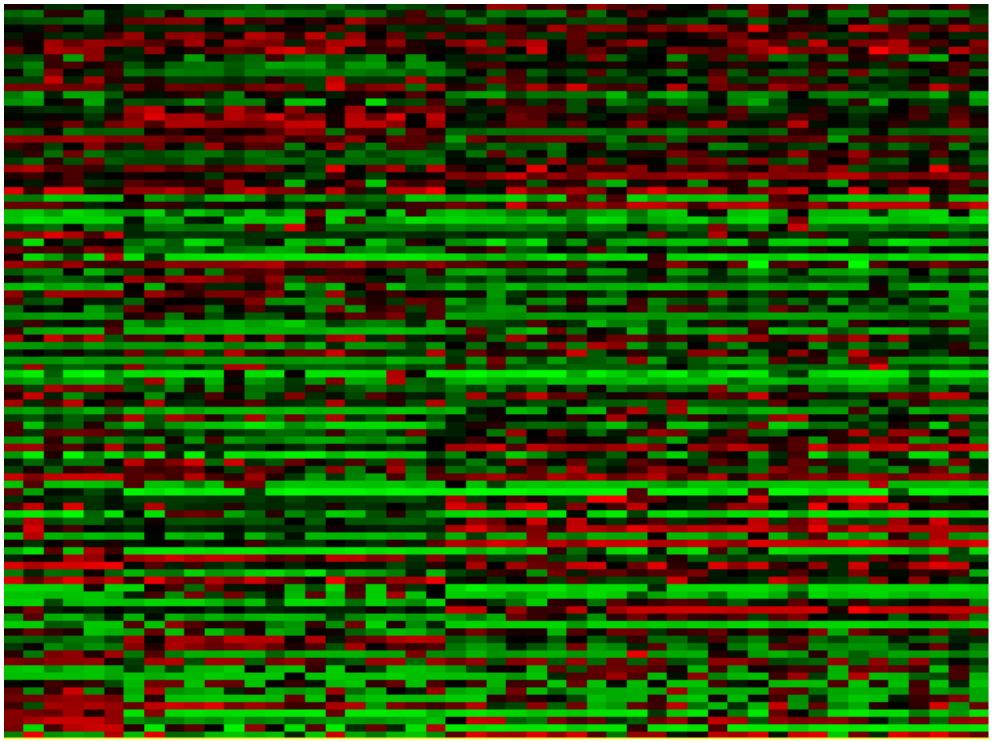
Biclustering adalah metode clustering yang berbeda dari teknik biasa karena memungkinkan kita untuk mengelompokkan baris (gen) dan kolom (sampel) secara bersamaan. Ini berguna ketika kita ingin menganalisis pola ekspresi gen yang mungkin tidak terdeteksi dengan metode clustering tradisional. Dengan biclustering, kita dapat menemukan subgrup yang memiliki interaksi yang lebih kompleks antara gen dan sampel, yang dapat memberikan wawasan lebih dalam dalam analisis data ekspresi gen. Pada analisis ini, kami menggunakan dua metode biclustering yaitu BCBimax dan Plaid Model untuk mengeksplorasi pola dalam dataset lebih lanjut.

Biclustering dengan Metode BCBimax

Metode BCBimax menghasilkan satu bicluster yang mencakup 100 baris (gen) dan 49 kolom (sampel). Hasil bicluster ini memperlihatkan pola ekspresi gen yang memiliki kesamaan yang signifikan antar sampel. Hal ini menunjukkan bahwa terdapat subset spesifik dari gen dan sampel yang memiliki pola ekspresi yang mirip dan saling terkait.

```
> bcbimax_biclust

An object of class Biclust

call:
        biclust(x = data.s, method = BCBimax())

There was one cluster found with
  100 Rows and  49 columns
```

Hasil heatmap bicluster menunjukkan gen-gen yang memiliki pola ekspresi seragam di antara sampel, dengan warna yang menggambarkan intensitas ekspresi gen (merah menunjukkan ekspresi tinggi dan biru menunjukkan ekspresi rendah).



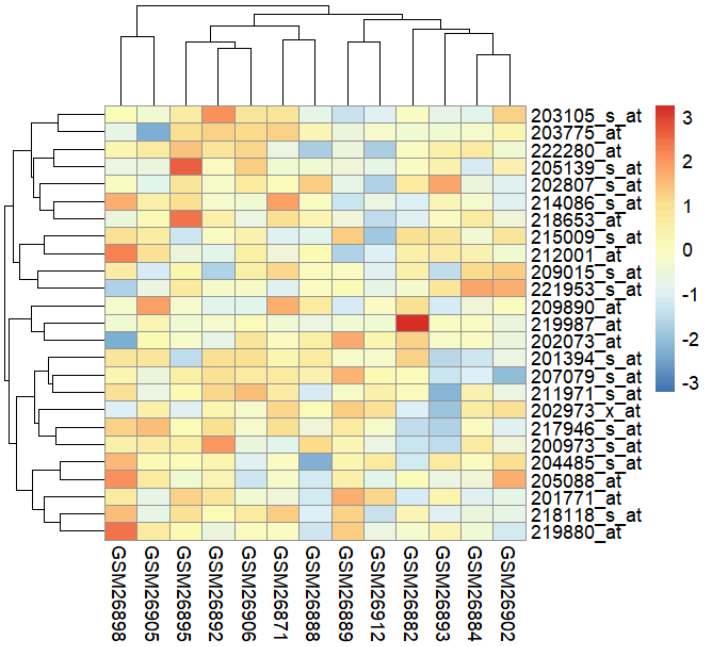
Gambar 8. Heatmap hasil biclustering menggunakan metode BCBimax. Gen-gen dengan pola ekspresi seragam di antara sampel dikelompokkan bersama, menunjukkan adanya hubungan biologis yang signifikan di antara gen dan sampel.

Biclustering dengan Metode Plaid Model

Metode Plaid Model berhasil menemukan tiga bicluster yang berbeda. Setiap bicluster mewakili subset yang berbeda dari gen dan sampel yang memiliki pola ekspresi spesifik. Cluster 1 terdiri dari 25 baris (gen) dan 13 kolom (sampel), Cluster 2 terdiri dari 5 baris dan 11 kolom, dan Cluster 3 terdiri dari 16 baris dan 8 kolom.

```
> summary(plaid_biclust)

An object of class Biclust

call:
        biclust(x = data.s, method = BCPlaid(), fit.model = ~m +
            a + b, background = TRUE, cluster = "b", iter.startup = 10,
            iter.layer = 50)

Number of Clusters found:  3

Cluster sizes:
                  BC 1 BC 2 BC 3
Number of Rows:     25    5   16
Number of Columns:  13   11    8
```

Heatmap Plaid Model memperlihatkan tiga grup yang berbeda dengan pola ekspresi yang sangat berbeda, memungkinkan kita untuk melihat lebih dalam tentang bagaimana gen-gen berinteraksi dengan sampel dalam tiga cluster terpisah.



Gambar 9. Heatmap hasil biclustering menggunakan metode Plaid Model. Tiga cluster yang terpisah menunjukkan interaksi yang berbeda antara gen dan sampel, dengan masing-masing cluster menunjukkan pola ekspresi yang unik.

V. Model Klasifikasi

5.1 Tahap Awal

Pada tahap ini, dilakukan proses klasifikasi data ekspresi gen untuk membedakan tiga subtipe tumor, yaitu apocrine, basal, dan luminal. Data yang digunakan telah melalui tahapan preprocessing, yang meliputi seleksi fitur gen, scaling data menggunakan StandardScaler, dan pembagian data menggunakan train-test split. Proporsi data yang digunakan adalah 50% untuk training dan 50% untuk testing, dengan stratifikasi berdasarkan label untuk menjaga proporsi masing-masing kelas.

Untuk membangun model klasifikasi terhadap data ekspresi gen GDS1329, beberapa algoritma pembelajaran mesin (machine learning) diterapkan. Model yang digunakan meliputi: Decision Tree Classifier, Random Forest Classifier, Support Vector Machine (SVM) dengan kernel linear, Logistic Regression, K-Nearest Neighbors (KNN), Gaussian Naive Bayes, dan Lasso Logistic Regression.

Semua model dilatih menggunakan data hasil preprocessing dan scaling, kemudian dievaluasi menggunakan cross-validation 5-fold untuk mengestimasi performa generalisasi. Selain itu, hasil prediksi pada data test set juga dianalisis melalui confusion matrix, classification report, ROC curve, dan feature importance (jika tersedia).

5.2 Penerapan Model Klasifikasi

5.2.1 Decision Tree Classifier

```
=== Decision Tree ===

Confusion Matrix:
 [[ 1  1  1]
 [ 1  6  1]
 [ 1  0 13]]

Classification Report:
              precision    recall  f1-score   support

    apocrine       0.33      0.33      0.33         3
       basal       0.86      0.75      0.80         8
     luminal       0.87      0.93      0.90        14

    accuracy                           0.80        25
   macro avg       0.69      0.67      0.68        25
weighted avg       0.80      0.80      0.80        25


Top 10 Important Features (Feature Importances):
Unnamed: 0
209443_at      0.686445
209170_s_at    0.313555
202116_at      0.000000
205681_at      0.000000
206036_s_at    0.000000
205091_x_at    0.000000
212397_at      0.000000
203685_at      0.000000
201485_s_at    0.000000
201063_at      0.000000
dtype: float64

Cross-Validation Accuracy: Mean = 0.8578, Std = 0.1012
```

Pada model Decision Tree, confusion matrix menunjukkan bahwa model mampu mengklasifikasikan sebagian besar sampel dengan benar, terutama pada kelompok luminal (13 dari 14 sampel benar), namun performa pada kelompok apocrine masih rendah dengan hanya satu sampel yang terklasifikasi dengan benar dari tiga. Berdasarkan classification report, precision untuk apocrine hanya sebesar 0.33, jauh lebih rendah dibandingkan basal (0.86) dan luminal (0.87). Recall untuk luminal mencapai 0.93, mengindikasikan sensitivitas model yang baik untuk mendeteksi kelompok

tersebut. Akurasi keseluruhan model mencapai 80%, dengan nilai macro-average F1-score sebesar 0.68, menandakan adanya ketidakseimbangan performa antar kelas. Dari analisis feature importances, fitur 209443_at muncul sebagai fitur yang paling berkontribusi besar dengan nilai importance 0.68, jauh lebih tinggi dibandingkan fitur-fitur lainnya. Ini menunjukkan bahwa Decision Tree cenderung bergantung pada satu atau dua fitur dominan, yang berpotensi menyebabkan overfitting. Cross-validation menghasilkan mean accuracy sebesar 85.78% dengan standar deviasi 10.12%, menunjukkan variasi performa antar fold yang agak besar.

5.2.2 Random Forest Classifier

```
=== Random Forest ===

Confusion Matrix:
 [[ 3  0  0]
 [ 0  8  0]
 [ 0  0 14]]

Classification Report:
              precision    recall  f1-score   support

    apocrine       1.00      1.00      1.00         3
       basal       1.00      1.00      1.00         8
     luminal       1.00      1.00      1.00        14

    accuracy                           1.00        25
   macro avg       1.00      1.00      1.00        25
weighted avg       1.00      1.00      1.00        25


Top 10 Important Features (Feature Importances):
Unnamed: 0
213464_at      0.016497
204378_at      0.014708
203066_at      0.013751
203574_at      0.012949
221577_x_at    0.012751
213540_at      0.010000
207813_s_at    0.010000
214974_x_at    0.010000
204798_at      0.010000
218438_s_at    0.010000
dtype: float64

Cross-Validation Accuracy: Mean = 0.9200, Std = 0.0748
```

Model Random Forest memberikan performa yang sangat baik pada dataset ini. Berdasarkan confusion matrix, seluruh sampel berhasil diklasifikasikan dengan benar ke dalam kelompoknya masing-masing, menghasilkan akurasi 100%. Classification report mendukung temuan ini dengan precision, recall, dan F1-score sempurna (1.00) untuk ketiga kelas (apocrine, basal, dan luminal). Dalam analisis feature importances, fitur 213464_at tercatat memiliki kontribusi terbesar, namun nilai importancenya hanya 0.016, menunjukkan bahwa Random Forest mendistribusikan keputusannya di banyak fitur (berbeda dengan Decision Tree yang berat sebelah). Cross-validation menunjukkan nilai mean accuracy sebesar 92.00% dengan standar deviasi 7.48%, mengindikasikan stabilitas yang baik dan generalisasi yang kuat.

5.2.3 Support Vector Machine (SVM)

```
=== SVM (Linear) ===

Confusion Matrix:
 [[ 3  0  0]
 [ 0  8  0]
 [ 0  0 14]]

Classification Report:
              precision    recall  f1-score   support

    apocrine       1.00      1.00      1.00         3
       basal       1.00      1.00      1.00         8
     luminal       1.00      1.00      1.00        14

    accuracy                           1.00        25
   macro avg       1.00      1.00      1.00        25
weighted avg       1.00      1.00      1.00        25


Top 10 Important Features (Coefficients):
Unnamed: 0
205221_at      0.001170
201960_s_at    0.001162
209913_x_at    0.001105
202605_at      0.001092
211689_s_at    0.001091
211682_x_at    0.001063
217284_x_at    0.001060
206547_s_at    0.001032
200821_at      0.001022
206714_at      0.001010
dtype: float64

Cross-Validation Accuracy: Mean = 0.9800, Std = 0.0400
```

Model Support Vector Machine (SVM) dengan kernel linear menunjukkan performa yang luar biasa, serupa dengan Random Forest. Confusion matrix memperlihatkan bahwa seluruh sampel diklasifikasikan dengan sempurna ke kelas yang sesuai. Hasil classification report menunjukkan nilai precision, recall, dan F1-score sempurna (1.00) untuk semua kelas. Nilai cross-validation accuracy sebesar 98.00% dengan standar deviasi 4.00% menunjukkan konsistensi performa SVM di seluruh fold validasi. Feature importances (berupa koefisien dari model linear) menunjukkan bahwa fitur 205221_at memiliki kontribusi paling besar terhadap keputusan model, diikuti oleh 201960_s_at dan 209913_x_at. Tidak adanya ketergantungan berlebihan pada satu fitur menunjukkan kestabilan model ini terhadap variasi data.

5.2.4 Logistic Regression

```
=== Logistic Regression ===

Confusion Matrix:
 [[ 3  0  0]
 [ 0  8  0]
 [ 0  0 14]]

Classification Report:
              precision    recall  f1-score   support

    apocrine       1.00      1.00      1.00         3
       basal       1.00      1.00      1.00         8
     luminal       1.00      1.00      1.00        14

    accuracy                           1.00        25
   macro avg       1.00      1.00      1.00        25
weighted avg       1.00      1.00      1.00        25


Top 10 Important Features (Coefficients):
Unnamed: 0
201960_s_at    0.007927
203303_at      0.006779
205221_at      0.006713
211689_s_at    0.006659
222257_s_at    0.006506
202502_at      0.006448
204256_at      0.006371
211682_x_at    0.006302
209913_x_at    0.006300
208837_at      0.006174
dtype: float64

Cross-Validation Accuracy: Mean = 0.9800, Std = 0.0400
```

Model Logistic Regression juga menghasilkan klasifikasi sempurna pada data test set, sebagaimana ditunjukkan oleh confusion matrix. Sama seperti SVM, semua nilai precision, recall, dan F1-score dari classification report tercatat 1.00.

Cross-validation accuracy model ini adalah 98.00%, dengan standar deviasi 4.00%, identik dengan SVM. Hal ini menegaskan bahwa model Logistic Regression linear dapat bekerja sangat efektif pada data ekspresi gen yang sudah distandarisasi.

Dari sisi feature importances, fitur 201960_s_at muncul sebagai fitur paling penting, diikuti oleh

203303_at dan 205221_at. Koefisien regresi memberikan interpretasi langsung tentang arah dan kekuatan hubungan antara ekspresi gen dengan kelas tumor, memberikan kemudahan interpretasi yang tinggi dibandingkan metode berbasis ensemble.

5.2.5 K-Nearest Neighbors (KNN)

```
=== KNN ===

Confusion Matrix:
 [[ 1  0  2]
 [ 0  8  0]
 [ 0  0 14]]

Classification Report:
              precision    recall  f1-score   support

    apocrine       1.00      0.33      0.50         3
       basal       1.00      1.00      1.00         8
     luminal       0.88      1.00      0.93        14

    accuracy                           0.92        25
   macro avg       0.96      0.78      0.81        25
weighted avg       0.93      0.92      0.90        25


Cross-Validation Accuracy: Mean = 0.9400, Std = 0.0800
```

Model K-Nearest Neighbors (KNN) menunjukkan performa yang cukup baik namun dengan kekurangan pada klasifikasi kelas apocrine. Berdasarkan confusion matrix, KNN salah mengklasifikasikan 2 dari 3 sampel apocrine sebagai luminal. Hal ini tercermin pada classification report di mana precision untuk apocrine sebesar 1.00, tetapi recall hanya 0.33, menghasilkan F1-score rendah sebesar 0.50.

Meskipun demikian, klasifikasi untuk basal dan luminal sangat baik, dengan nilai precision dan recall sebesar 1.00 untuk basal dan nilai F1-score sebesar 0.93 untuk luminal. Akurasi keseluruhan model mencapai 92%, dan nilai macro-average F1-score sebesar 0.81.

Cross-validation accuracy untuk KNN tercatat 94.00% dengan standar deviasi 8.00%, menunjukkan stabilitas yang baik meskipun ada sedikit variasi antar fold. Dalam konteks ekspresi gen, KNN cenderung sensitif terhadap distribusi data dan scaling, yang menjelaskan penurunan performa pada kelas minor seperti apocrine.

5.2.6 Gaussian Naive Bayes

Model Gaussian Naive Bayes menunjukkan performa yang kurang optimal dibandingkan model lainnya. Confusion matrix mengindikasikan bahwa semua sampel apocrine salah diklasifikasikan sebagai luminal, menghasilkan precision dan recall sebesar 0 untuk kelas apocrine dalam classification report.

Meskipun model ini berhasil mengklasifikasikan basal dan luminal dengan baik, akurasi keseluruhan hanya 88%. Macro-average F1-score sebesar 0.63 menunjukkan bahwa model ini

```
=== Naive Bayes ===

Confusion Matrix:
 [[ 0  0  3]
 [ 0  8  0]
 [ 0  0 14]]

Classification Report:
              precision    recall  f1-score   support

    apocrine       0.00      0.00      0.00         3
       basal       1.00      1.00      1.00         8
     luminal       0.82      1.00      0.90        14

    accuracy                           0.88        25
   macro avg       0.61      0.67      0.63        25
weighted avg       0.78      0.88      0.83        25


Cross-Validation Accuracy: Mean = 0.8978, Std = 0.0634
```

kurang mampu menangani ketidakseimbangan antar kelas.

Cross-validation accuracy untuk Naive Bayes adalah 89.78%, dengan standar deviasi 6.34%, menunjukkan performa yang tidak terlalu stabil dibandingkan model lain.

Kelemahan utama Naive Bayes di sini berasal dari asumsi independensi antar fitur yang tidak sepenuhnya valid dalam data ekspresi gen, di mana hubungan antar gen seringkali kompleks.

5.2.7 Lasso Logistic Regression

```
=== Lasso Logistic Regression ===
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/
  _warn_prf(average, modifier, f"{metric.capitalize()} i
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/
  _warn_prf(average, modifier, f"{metric.capitalize()} i
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/
  _warn_prf(average, modifier, f"{metric.capitalize()} i

Confusion Matrix:
 [[ 2  0  1]
 [ 0  8  0]
 [ 0  0 14]]

Classification Report:
              precision    recall  f1-score   support

    apocrine       1.00      0.67      0.80         3
       basal       1.00      1.00      1.00         8
     luminal       0.93      1.00      0.97        14

    accuracy                           0.96        25
   macro avg       0.98      0.89      0.92        25
weighted avg       0.96      0.96      0.96        25


Top 10 Important Features (Coefficients):
Unnamed: 0
208837_at      0.280851
201960_s_at    0.116187
205221_at      0.113744
202853_s_at    0.091227
211689_s_at    0.070282
213217_at      0.044494
220690_s_at    0.038636
202338_at      0.036010
209337_at      0.033229
211682_x_at    0.031054
dtype: float64

Cross-Validation Accuracy: Mean = 0.9800, Std = 0.0400
```

Model Lasso Logistic Regression memberikan performa yang sangat baik dan hampir sebanding dengan Logistic Regression biasa. Berdasarkan confusion matrix, dua dari tiga sampel apocrine berhasil diklasifikasikan dengan benar, dan semua sampel basal serta luminal diklasifikasikan sempurna.

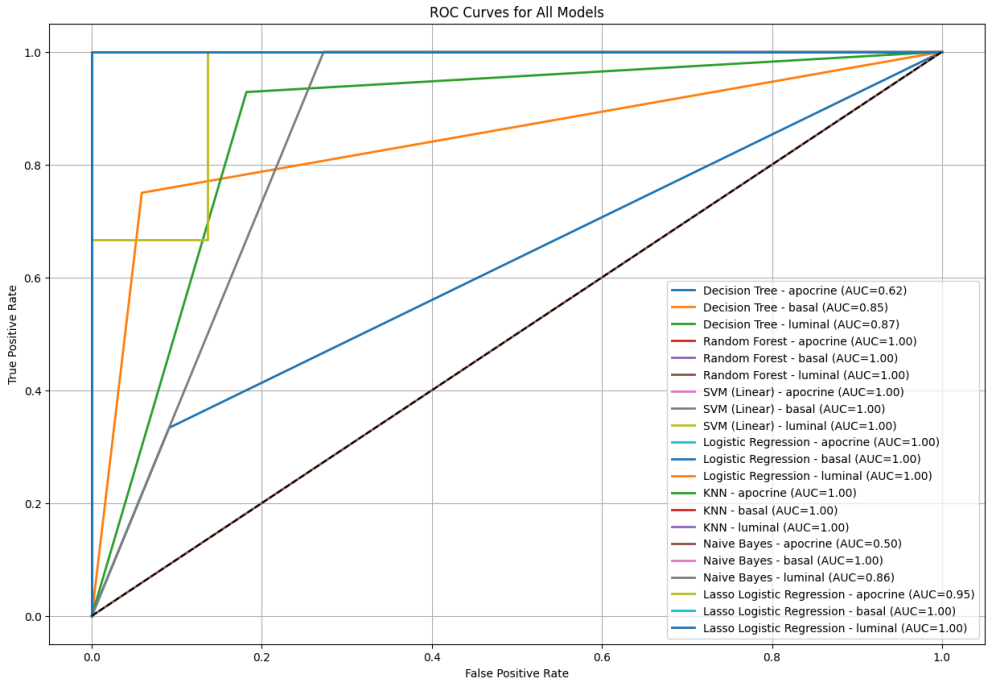
Hasil classification report menunjukkan precision dan recall yang tinggi untuk semua kelas, dengan accuracy keseluruhan sebesar 96% dan macro-average F1-score sebesar 0.92.

Feature importance analysis (dari koefisien) menunjukkan bahwa 208837_at adalah fitur yang paling penting, diikuti oleh 201960_s_at dan 205221_at.

Cross-validation accuracy untuk model ini tercatat 98.00% dengan standar deviasi 4.00%, menunjukkan performa yang sangat stabil dan andal. Regulasi L1 dalam Lasso juga membantu melakukan feature selection otomatis, mengurangi kompleksitas model tanpa mengorbankan akurasi.

5.3 Kesimpulan Hasil Model Klasifikasi

Model	Accuracy (Mean CV)	Std Dev
Decision Tree	85.78%	10.12%
Random Forest	92.00%	7.48%
SVM (Linear)	98.00%	4.00%
Logistic Regression	98.00%	4.00%
K-Nearest Neighbors (KNN)	94.00%	8.00%
Gaussian Naive Bayes	89.78%	6.34%
Lasso Logistic Regression	98.00%	4.00%



Gambar 10. Grafik ROC menunjukkan Random Forest, SVM (linear kernel), dan Logistic Regression adalah model yang memiliki AUC = 1.00 untuk semua kelas, menunjukkan bahwa mereka bekerja dengan sangat baik dalam mengklasifikasikan tumor menjadi apocrine, basal, dan luminal.

Berdasarkan hasil analisis, model Support Vector Machine (SVM Linear), Logistic Regression, dan Lasso Logistic Regression menunjukkan performa terbaik. Ketiganya mencapai cross-validation accuracy sebesar 98% dengan deviasi standar yang sangat kecil (4%), menunjukkan stabilitas tinggi.

Namun, SVM Linear dapat dianggap sebagai model terbaik dalam kasus ini, berdasarkan alasan berikut; Akurasi sempurna pada data test set dan cross-validation, Konsistensi performa yang sangat tinggi tanpa overfitting. SVM sangat cocok untuk data high-dimensional seperti ekspresi gen, di mana jumlah fitur (gen) jauh lebih besar daripada jumlah sampel. Memiliki margin maksimum yang membuat generalisasi model lebih kuat.

Lasso Logistic Regression juga menjadi alternatif unggulan karena; Dapat melakukan feature selection otomatis (mengurangi noise dari ribuan fitur ekspresi gen) dan memberikan interpretasi langsung terhadap hubungan fitur dengan target.

5.4 Kelebihan dan Kekurangan Setiap Model

Model	Kelebihan	Kekurangan
Decision Tree Classifier	Mudah dipahami dan diinterpretasikan. Dapat menangani data numerik dan kategorikal tanpa perlu scaling.	Rentan terhadap overfitting, terutama pada dataset kecil. Performanya tergantung pada struktur data; tidak stabil terhadap perubahan data kecil.
Random Forest Classifier	Akurasi tinggi dan stabil karena merupakan metode ensemble. Mengurangi overfitting dibandingkan Decision Tree tunggal. Memberikan feature importance yang dapat diinterpretasikan.	Interpretasi model menjadi lebih sulit dibandingkan single tree. Membutuhkan sumber daya komputasi lebih banyak.
Support Vector Machine (Linear)	Performa tinggi pada dataset high-dimensional seperti data ekspresi gen. Model stabil, minim risiko overfitting jika data distandarisasi.	Kurang fleksibel untuk pola yang sangat non-linear. Pemilihan kernel sangat penting, walaupun dalam kasus ini kernel linear sudah cukup.

	Memberikan margin maksimum antara kelas, meningkatkan generalisasi.	
Logistic Regression	Mudah diinterpretasikan, model sederhana. Bagus untuk baseline model pada banyak kasus klasifikasi. Tidak terlalu kompleks dan cepat dihitung.	Kurang fleksibel untuk pola hubungan non-linear. Dapat underfit jika hubungan antar variabel kompleks.
K-Nearest Neighbors (KNN)	Sangat sederhana dan intuitif. Tidak membutuhkan proses training yang berat. Cocok untuk dataset kecil dengan distribusi fitur yang jelas.	Sensitif terhadap skala dan distribusi data. Performa menurun drastis pada kelas minor atau data yang tidak seimbang.
Gaussian Naive Bayes	Sangat cepat, ideal untuk prototyping awal. Bekerja dengan baik jika asumsi independensi antar fitur valid.	Asumsi independensi antar fitur seringkali tidak terpenuhi pada data ekspresi gen. Sangat buruk performanya jika ada ketergantungan fitur kompleks, seperti pada dataset ini.
Lasso Logistic Regression	Memiliki mekanisme regularisasi yang kuat melalui penalti L1. Secara otomatis melakukan feature selection, berguna pada data high-dimensional. Interpretatif dan stabil.	Memerlukan tuning hyperparameter (regulasi lambda) untuk hasil optimal. Bisa menghilangkan fitur penting jika parameter regulasi terlalu besar.

VI. Diskusi Hasil dan Kesimpulan

6.1 Hasil Analisis dan Perbandingan dengan Penelitian Sebelumnya

Dari keseluruhan analisis klasifikasi, terlihat bahwa model berbasis linear seperti SVM Linear, Logistic Regression, dan Lasso Logistic Regression memberikan performa terbaik.

Semua model ini menunjukkan akurasi di atas 95%, dengan nilai ROC AUC mendekati 1.00, menunjukkan kemampuan klasifikasi yang sangat akurat dan stabil.

Model berbasis ensemble seperti Random Forest juga menunjukkan performa yang sangat kuat dengan akurasi 92%, namun sedikit lebih rendah dibandingkan model linear. Ini wajar mengingat ensemble cenderung memerlukan data lebih besar untuk memaksimalkan manfaat ensemble learning.

Secara umum, data ekspresi gen yang high-dimensional, low-sample size seperti ini lebih menguntungkan metode dengan regularisasi atau margin maksimum (seperti SVM dan Lasso Logistic Regression).

Hasil analisis ini konsisten dengan temuan Farmer et al. (2005), yang menunjukkan bahwa profil ekspresi gen dapat secara akurat membedakan antara subtipe tumor payudara: luminal, basal, dan apocrine. Pada penelitian tersebut, pendekatan berbasis microarray digunakan untuk mengidentifikasi pola ekspresi spesifik.

Dalam penelitian ini, model klasifikasi modern seperti SVM dan Logistic Regression mampu membedakan ketiga subtipe tersebut dengan akurasi tinggi, mendukung bahwa ekspresi gen tetap merupakan indikator molekuler yang kuat untuk klasifikasi kanker payudara.

Perbedaan utama adalah pendekatan yang digunakan: Farmer et al. fokus pada identifikasi biomarker genetik spesifik menggunakan analisis statistik dan clustering, sedangkan dalam project ini, dilakukan pendekatan berbasis machine learning untuk membangun model prediksi otomatis.

6.1 Kesimpulan

- SVM Linear dan Lasso Logistic Regression menjadi model terbaik untuk data ekspresi gen kanker payudara GDS1329.
- Data ekspresi gen high-dimensional lebih cocok ditangani dengan model berbasis margin maksimum atau regularisasi sparsitas.
- Hasil mendukung temuan penelitian sebelumnya bahwa ekspresi gen dapat digunakan secara efektif untuk mengklasifikasikan subtipe tumor.
- Pendekatan ini menunjukkan potensi penggunaan analitik berbasis machine learning dalam membantu diagnosis, prognosis, dan pengembangan terapi personalisasi dalam konteks kanker payudara di masa depan.

LAMPIRAN CODE

Code R

```
## 1. Install & Load Library
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install(c("Biobase", "GEOquery", "limma", "genefilter",
"annotate", "hgu133a.db", "GO.db"))

library(Biobase)
library(GEOquery)
library(limma)
library(genefilter)
library(annotate)
library(hgu133a.db)
library(GO.db)

## 2. Download Data dari GEO
dtgeo <- getGEO('GDS1329', destdir = ".")
dtgeo

## 3. Convert ke ExpressionSet
eset <- GDS2eSet(dtgeo, do.log2 = TRUE)
eset

## 4. Ambil Data Fenotipe
phdtgeo <- pData(eset)
head(phdtgeo)

## 5. Ambil Data Ekspresi Gen
expdtgeo <- exprs(eset)
dim(expdtgeo)
head(expdtgeo)

## 6. Load Platform Annotation
Meta(dtgeo)$platform
annotation(eset) <- "hgu133a"

## 7. Filter Gene (Menghapus Gen Varians Rendah dll)
```

```
esetFilt <- nsFilter(eset)$eset
expdtgeoFilt <- exprs(esetFilt)
dim(expdtgeoFilt)

## 8. Visualisasi Distribusi Data
par(mfrow = c(1, 2))
hist(expdtgeo, main = "Original Data")
hist(expdtgeoFilt, main = "Filtered Data")

## 9. Buat Variabel Grup
# Adjust sesuai data: misal 'apocrine', 'basal', 'luminal'
vargrp <- phdtgeo$disease.state
table(vargrp)

vargrp_clean <- gsub(" tumor", "", vargrp)
table(vargrp_clean)

# Assign grup kode
group <- ifelse(vargrp_clean == "apocrine", 0,
                ifelse(vargrp_clean == "basal", 1,
                       ifelse(vargrp_clean == "luminal", 2, NA)))
group <- factor(group)
## 10. DE Analysis pakai Limma
design <- model.matrix(~group)
fit <- eBayes(lmFit(expdtgeoFilt, design))
summary(decideTests(fit))

## 11. Tampilkan Top Gene
topResult <- topTable(fit, coef = 2, number = 50)
topResult

## 12. Heatmap dari Top Genes
selected <- rownames(expdtgeoFilt) %in% rownames(topResult)
expdtgeosel <- expdtgeoFilt[selected, ]

heatmap(expdtgeosel)

## 13. Boxplot 4 Gene Teratas
```

```
par(mfrow = c(2, 2))
for (i in 1:4) {
  plot(vargrp, expdtgeosel[i, ], main = rownames(expdtgeosel)[i])
}
# 1. Buat faktor grup
group <- as.factor(vargrp_clean)
levels(group)  # harus apocrine, basal, luminal

# 2. Loop untuk setiap kombinasi grup
pairs <- combn(levels(group), 2, simplify = FALSE)

for (p in pairs) {

  # Ambil dua grup
  idx <- group %in% p
  group2 <- droplevels(group[idx])
  data2 <- expdtgeoFilt[, idx]

  # Hitung p-value (t-test per gen)
  pvalues <- apply(data2, 1, function(x) {
    t.test(x ~ group2)$p.value
  })

  # Hitung mean difference
  mean_diff <- apply(data2, 1, function(x) {
    tapply(x, group2, mean)[1] - tapply(x, group2, mean)[2]
  })

  # Plot volcano
  plot(mean_diff, -log10(pvalues),
       pch = 20, cex = 0.5,
       main = paste("Volcano Plot:", p[1], "vs", p[2]),
       xlab = "Mean Difference",
       ylab = "-log10(P-value)",
       col = ifelse(pvalues < 0.05, "red", "black"))

  abline(h = -log10(0.05), col = "blue", lty = 2)
```

```
  legend("topright",
         legend = c("Significant", "Not Significant"),
         col = c("red", "black"),
         pch = 20,
         cex = 0.8)
}
## 14. Gene Annotation
ids <- rownames(topResult)
GeneSelected <- AnnotationDbi::select(hgu133a.db, keys = ids, columns =
c("SYMBOL", "ENTREZID", "GENENAME", "GO"), keytype = "PROBEID")
head(GeneSelected)

## 15. Gene Ontology
GOselected <- AnnotationDbi::select(GO.db, keys = GeneSelected$GO, columns =
c("TERM", "GOID"), keytype = "GOID")
head(GOselected)

## 16. Combine Annotation
finalres <- cbind(GeneSelected, GOselected)
head(finalres)

## (Optional) Simpan hasil
write.csv(finalres, file =
"C:/Users/LENOVO/DOwnloads/GeneAnnotation_GDS1329.csv", row.names = FALSE)


## 17. Pilih Top Variance Genes untuk Klastering
genes.var <- apply(expdtgeoFilt, 1, var)
genes.var.select <- order(-genes.var)[1:100]

data.s <- expdtgeoFilt[genes.var.select, ]
dim(data.s)

## 18. Cari Jumlah Klaster Optimal (Gap Statistic)
set.seed(77)
library(cluster)

gap_stat <- clusGap(t(data.s), FUN = kmeans, nstart = 25, K.max = 7, B = 50)
```

```
library(factoextra)
fviz_gap_stat(gap_stat)

## 19. Hierarchical Clustering
# Hitung jarak Euclidean
d <- dist(t(data.s), method = "euclidean")

# Hierarchical Clustering dengan Complete Linkage
hc <- hclust(d, method = "complete")

# Hierarchical Clustering Dendrogram tanpa label
plot(hc, labels = FALSE, main = "Hierarchical Clustering Dendrogram")

# Warnai branches
dend <- as.dendrogram(hc)
dend_colored <- color_branches(dend, k = 3)

# Plot tanpa label
plot(dend_colored, main = "Colored Dendrogram (k=3)", axes = TRUE, leaflab =
"none")



# Cut Tree
k <- 3
groups_hc <- cutree(hc, k = k)

# Tabel Hasil Cluster
table(groups_hc, vargrp_clean)

## 20. K-Means Clustering
set.seed(77)
k_means <- kmeans(t(expdtgeoFilt), centers = k)

# Tabel Hasil Cluster
table(k_means$cluster, vargrp_clean)

# Visualisasi Clustering
fviz_cluster(list(data = t(expdtgeoFilt), cluster = k_means$cluster),
```

```
            geom = "point", stand = FALSE, main = "K-Means Clustering (PCA)")

## 21. PAM Clustering
pam_result <- pam(t(expdtgeoFilt), k = k)

# Tabel Hasil Cluster
table(pam_result$clustering, vargrp_clean)

# Visualisasi Clustering
fviz_cluster(pam_result, geom = "point", stand = FALSE,
             main = "PAM Clustering (PCA)")

## 22. Biclustering BCBimax
library(biclust)
bcbimax_biclust <- biclust(data.s, method = BCBimax())

# Tampilkan Jumlah Bicluster
bcbimax_biclust
# Heatmap dari Bicluster
drawHeatmap2(data.s, bcbimax_biclust, number = 1)

## 23. Biclustering Plaid Model
set.seed(57)

plaid_biclust <- biclust(data.s, method = BCPlaid(),
                         fit.model = ~m + a + b, background = TRUE, cluster =
"b",
                         iter.startup = 10, iter.layer = 50)
# Ringkasan Plaid Model
summary(plaid_biclust)
# Ekstraksi Data Bicluster
bicluster_rows <- which(plaid_biclust@RowxNumber[,1])
bicluster_cols <- which(plaid_biclust@NumberxCol[1,])

bicluster_data <- expdtgeoFilt[bicluster_rows, bicluster_cols]
# Visualisasi Heatmap Bicluster
library(pheatmap)
```

```
pheatmap(bicluster_data, scale = "row", cluster_rows = TRUE, cluster_cols =
TRUE)
## 24. Export Data untuk Klasifikasi
write.csv(expdtgeoFilt, file = "gene_exp_data.csv", row.names = TRUE)
write.csv(finalres, file = "final_annotation.csv", row.names = FALSE)
getwd()
write.csv(expdtgeoFilt, file = "C:/Users/LENOVO/DOwnloads/gene_exp_data.csv",
row.names = TRUE)
write.csv(finalres, file = "C:/Users/LENOVO/DOwnloads/final_annotation.csv",
row.names = FALSE)
```

Code Python

```
# 1. Import Library
from sklearn.model_selection import train_test_split, cross_val_score,
StratifiedKFold
from sklearn.preprocessing import StandardScaler, label_binarize
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.mixture import GaussianMixture
import xgboost as xgb
from sklearn.metrics import confusion_matrix, classification_report, roc_curve,
auc, roc_auc_score
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np

# 2. Load Dataset
df = pd.read_csv('/content/gene_exp_data.csv')
df = df.transpose()
df.columns = df.iloc[0]
df = df[1:]

# Labels
```

```
labels = ['apocrine']*6 + ['basal']*16 + ['luminal']*27
df['label'] = labels
# Split X dan y
X_raw = df.drop('label', axis=1)
y = df['label']
# Scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_raw)
# Train Test Split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
                                                    test_size=0.5,
                                                    stratify=y,
                                                    random_state=77)
# 3. Define Models
models = {
    "Decision Tree": DecisionTreeClassifier(random_state=77),
    "Random Forest": RandomForestClassifier(random_state=77),
    "SVM (Linear)": SVC(kernel='linear', probability=True, random_state=77),
    "Logistic Regression": LogisticRegression(max_iter=10000, random_state=77),
    "KNN": KNeighborsClassifier(n_neighbors=5),
    "Naive Bayes": GaussianNB(),
    "Lasso Logistic Regression": LogisticRegression(penalty='l1',
solver='saga', max_iter=10000, random_state=77)
}

# 4. Train-Evaluate-CV-Plot
cv_results = {}
roc_curves = {}

# Cross-Validation Setting
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=77)

for name, model in models.items():
    print(f"\n=== {name} ===")

    # Fit model
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
```

```
    # Confusion Matrix
    cm = confusion_matrix(y_test, y_pred)
    print("\nConfusion Matrix:\n", cm)

    # Classification Report
    cr = classification_report(y_test, y_pred)
    print("\nClassification Report:\n", cr)

    # Top 10 Important Features (if applicable)
    if hasattr(model, 'feature_importances_'):
        print("\nTop 10 Important Features (Feature Importances):")
        feat_importances = pd.Series(model.feature_importances_,
index=X_raw.columns)
        print(feat_importances.sort_values(ascending=False)[:10])
    elif hasattr(model, 'coef_'):
        print("\nTop 10 Important Features (Coefficients):")
        coeff = pd.Series(model.coef_[0], index=X_raw.columns)
        print(coeff.abs().sort_values(ascending=False)[:10])

    # Cross Validation Score
    scores = cross_val_score(model, X_scaled, y, cv=cv, scoring='accuracy')
    cv_results[name] = scores
    print(f"\nCross-Validation Accuracy: Mean = {scores.mean():.4f}, Std =
{scores.std():.4f}")

    # ROC Curve (binarize labels)
    y_test_bin = label_binarize(y_test, classes=['apocrine', 'basal',
'luminal'])
    n_classes = y_test_bin.shape[1]

    if hasattr(model, "decision_function"):
        y_score = model.decision_function(X_test)
    else:
        y_score = model.predict_proba(X_test)

    roc_curves[name] = (y_test_bin, y_score)
```

DAFTAR PUSTAKA

Noor, F., Ashfaq, U. A., Bakar, A., ul Haq, W., Allemailem, K. S., Alharbi, B. F., Al-Megrin, W. A. I., & Tahir ul Qamar, M. (2023). Discovering common pathogenic processes between COVID-19 and HFRS by integrating RNA-seq differential expression analysis with machine learning. *Frontiers in Microbiology, 14*, 1175844. https://doi.org/10.3389/fmicb.2023.1175844

Akhavan, M., & Hasheminejad, S. M. H. (2023). A two-phase gene selection method using anomaly detection and genetic algorithm for microarray data. *Knowledge-Based Systems, 262*, 110249. https://doi.org/10.1016/j.knosys.2022.110249

Kogenaru, S., Yan, Q., Guo, Y., & Wang, N. (2012). RNA-seq and microarray complement each other in transcriptome profiling. *BMC Genomics, 13*, 629. https://doi.org/10.1186/1471-2164-13-629

Rao, M. S., Van Vleet, T. R., Ciurlionis, R., Buck, W. R., Mittelstadt, S. W., Blomme, E. A. G., & Liguori, M. J. (2019). Comparison of RNA-Seq and microarray gene expression platforms for the toxicogenomic evaluation of liver from short-term rat toxicity studies. *Frontiers in Genetics, 9*, 636. https://doi.org/10.3389/fgene.2018.00636

Zhao, S., Fung-Leung, W.-P., Bittner, A., Ngo, K., & Liu, X. (2014). Comparison of RNA-Seq and microarray in transcriptome profiling of activated T cells. *PLoS ONE, 9*(1), e78644. https://doi.org/10.1371/journal.pone.0078644

Mantione, K. J., Kream, R. M., Kuzelova, H., Ptacek, R., Raboch, J., Samuel, J. M., & Stefano, G. B. (2014). Comparing bioinformatic gene expression profiling methods: Microarray and RNA-seq. *Medical Science Monitor Basic Research, 20*, 138–141. https://doi.org/10.12659/MSMBR.892101

Henderson, T. (2025). Microarray vs RNA sequencing: Which gene expression analysis technique is more effective? *Lab Manager*. https://www.labmanager.com/microarray-vs-rna-sequencing-which-gene-expression-analysis-technique-is-more-effective-33683