JURNAL ILMIAH TEKNOLOGI DAN REKAYASA
Vol. 30, No. 03, December 2025, 215 – 228
p-ISSN: 1410-9093, e-ISSN: 2089-8088
https://ejournal.gunadarma.ac.id/index.php/tekno | tekrek@gunadarma.ac.id

# Optimasi Hyperparameter Berbasis Bayesian dengan Optuna untuk *Spectral Clustering - K-Means*: Studi Kasus pada Dataset Leukemia CuMiDa

**[1*]Rosalia Deviana Cahyaningrum, [2]Hendy Fergus Atheri Hura**

[1]Fakultas Teknologi dan Desain, Program Studi Informatika, Universitas Bunda Mulia, [2]Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi Matematika, Universitas Indonesia
[1]Jl. Jalur Sutera Barat Kav 7-9 Alam Sutera, Tangerang, Indonesia, [2]Kampus UI Depok, Pondok Cina, Depok, Indonesia
[1]rcahyaningrum@bundamulia.ac.id, [2]hendy.fergus@sci.ui.ac.id

**Abstract**

Leukemia is one of the cancers with the highest mortality rate worldwide; therefore, identifying its subtypes is crucial to support accurate diagnosis and effective treatment. The analysis of high-dimensional gene expression data, such as the CuMiDa dataset, still faces major challenges due to overlapping patterns and limited sample sizes. This study proposes the application of Bayesian Optimization using Optuna to perform hyperparameter tuning on the Spectral Clustering – K-Means method to improve the clustering performance of leukemia subtypes. Four key parameters (n_components, affinity method, n_neighbors, and gamma) were optimized through 1,000 iterations. The best configuration was obtained at n_components = 5 using the Nearest Neighbors method with n_neighbors = 6. The resulting Spectral Embedding matrix was then grouped using K-Means. The results showed that this approach achieved a clustering accuracy of 92,19%, outperforming both K-Means and Hierarchical Clustering when applied separately. Heatmap visualization demonstrated that the optimized method effectively grouped samples with similar gene expression patterns. This study demonstrates that the combination of Spectral Clustering–K-Means and Bayesian optimization using Optuna can improve the clustering quality of complex gene expression data and open up broader opportunities for application in other bioinformatics studies.

**Keywords**: *Bayesian Optimization*, gene expression, *K-Means*, Optuna, *Spectral Clustering*

**Abstrak**

Leukemia merupakan salah satu kanker dengan tingkat kematian tertinggi di dunia sehingga identifikasi subtipenya sangat penting untuk mendukung diagnosis dan pengobatan yang tepat. Analisis data ekspresi gen berdimensi tinggi, seperti *dataset* CuMiDa, masih menghadapi tantangan besar akibat pola yang saling tumpang tindih dan jumlah sampel yang terbatas. Penelitian ini mengusulkan penerapan optimasi *Bayesian* menggunakan Optuna untuk melakukan penyesuaian *hyperparameter* pada metode *Spectral Clustering – K-Means* guna meningkatkan performa klastering subtipe leukemia. Empat parameter kunci (*n_components*, *affinity method*, *n_neighbors*, dan *gamma*) dioptimasi melalui 1.000 iterasi. Konfigurasi terbaik diperoleh pada *n_components* = 5 dengan metode *Nearest Neighbors* dan *n_neighbors* = 6. Matriks *Spectral Embedding* yang dihasilkan kemudian dikelompokkan menggunakan *K-Means*. Hasil penelitian menunjukkan bahwa pendekatan ini mencapai akurasi klastering sebesar 92,19%, melampaui *K-Means* maupun *Hierarchical Clustering* secara terpisah. Visualisasi *heatmap* membuktikan bahwa metode yang dioptimasi ini mampu mengelompokkan sampel dengan pola ekspresi gen yang serupa secara efektif. Penelitian ini menunjukkan bahwa kombinasi *Spectral Clustering-K-Means* dan optimasi *Bayesian* menggunakan Optuna dapat meningkatkan kualitas klastering pada data ekspresi gen yang kompleks, serta membuka peluang penerapan lebih luas dalam studi bioinformatika lainnya.

* Corresponding Author

**Kata Kunci**: ekspresi gen, *K-Means*, Optimasi Bayesian, Optuna, *Spectral Clustering*

## 1. Pendahuluan

Leukemia (kanker darah) merupakan salah satu penyakit dengan tingkat kematian tertinggi di dunia. Leukemia adalah kanker darah yang menyerang sumsum tulang dan sel darah, ditandai dengan proliferasi sel abnormal yang tidak terkendali. Identifikasi subtipe leukemia sangat penting karena memengaruhi diagnosis, pengobatan, dan prognosis pasien [1].

Seiring meningkatnya kompleksitas data biomedis, penerapan teknik *machine learning* dalam analisis data kanker semakin berkembang. Integrasi teknologi *Artificial Intelligence* (AI) melalui metode *machine learning* dapat mendukung deteksi kanker secara dini dengan akurasi yang lebih tinggi, terutama pada *dataset* medis yang berukuran terbatas [2]. Perkembangan teknologi *microarray* memungkinkan peneliti menganalisis ekspresi ribuan gen secara bersamaan sehingga membuka peluang untuk mendeteksi pola molekuler pada berbagai jenis leukemia. Namun, data ekspresi gen memiliki tantangan khusus, yaitu dimensi yang tinggi, jumlah sampel terbatas, *noise* yang signifikan, serta pola subtipe yang tumpang tindih [3], [4].

Salah satu basis data yang mendukung pengembangan metode analisis ekspresi gen adalah *Curated Microarray Database* (CuMiDa). *Dataset* ini memuat data ekspresi gen kanker, termasuk leukemia, yang telah melalui tahap *background correction*, normalisasi, dan anotasi, sehingga banyak digunakan sebagai *benchmark* penelitian bioinformatika, baik untuk metode *supervised* maupun *unsupervised learning* [1], [3]. Dengan karakteristik data yang kompleks dan berdimensi tinggi, CuMiDa menyediakan landasan ideal bagi penerapan teknik klastering untuk mengungkap pola molekuler yang tersembunyi.

Klastering merupakan pendekatan *unsupervised* yang dapat digunakan untuk menganalisis *dataset* besar dengan banyak karakteristik dan membagi *dataset* besar tersebut ke dalam kelompok-kelompok kecil atau klaster [5]. Klastering menjadi salah satu pendekatan penting dalam bioinformatika untuk mengungkap pola-pola tersembunyi pada data ekspresi gen yang kompleks dan berdimensi tinggi. Klastering dikategorikan baik apabila setiap data tergabung ke dalam kelompok yang homogen tanpa adanya percampuran antar kelompok [6].

Dalam ranah klastering, *Spectral Clustering* dikenal sebagai pendekatan efektif untuk mendeteksi pola klaster non-linear. *Spectral Clustering* juga memiliki keunggulan utama menangani data berdimensi tinggi dengan pola non-konveks yang sulit dipecahkan oleh metode klasterisasi tradisional dengan memanfaatkan *Graph Structure Learning* (GSL) untuk membangun graf kemiripan yang optimal sehingga dapat meningkatkan kualitas *Spectral Embedding* dan hasil klasterisasi secara keseluruhan [7]. Sementara itu, pengembangan algoritma *Improved Automated Spectral Clustering* (IASC) berhasil mengatasi keterbatas *Spectral Clustering* konvensional dengan secara otomatis menentukan jumlah klaster melalui evaluasi kepadatan dan klasifikasi berbasis sudut kosinus, yang meningkatkan akurasi terutama pada data non-konveks [8]. Selain itu, konektivitas graf yang tinggi dengan pendekatan *multi-view Spectral Clustering* berbasis jalur (*path-based similarity)* terbukti tangguh terhadap *noise* dan *outlier* dalam aplikasi dunia nyata [9]. Namun, *Spectral Clustering* umumnya memerlukan algoritma lain sebagai tahap *post-processing* untuk mengelompokkan hasil *embedding* ke klaster akhir dan di bagian inilah *K-Means* sering digunakan karena kemampuannya menangkap struktur klaster sederhana pada ruang *embedding* yang telah direduksi [10]. *K-Means* mengungkapkan pola alami pada data tanpa perlu label atau kategori buatan manusia [11]. Tren terbaru dalam taksonomi algoritma klastering menunjukkan bahwa pendekatan *graph-based* seperti *Spectral Clustering* semakin diminati untuk menangani data berdimensi tinggi dan kompleks, terutama pada data kategorikal maupun numerik [12].

Performa *Spectral Clustering–K-Means* sangat bergantung pada pemilihan *hyperparameter* seperti jumlah klaster (k), skema normalisasi Laplacian, dan parameter kernel matriks similaritas. Pemilihan *hyperparameter* yang tidak optimal dapat menghasilkan klaster dengan kualitas rendah atau tidak stabil. Metode *tuning* tradisional seperti *grid search* atau *random search* seringkali memakan sumber daya komputasi besar dan tidak efisien untuk ruang parameter yang kompleks [4]. Hal ini diatasi denganadopsi optimasi *Bayesian* yang mampu membangun model probabilistik dari hasil evaluasi sebelumnya dan memilih kombinasi *hyperparameter* berikutnya yang memberikan hasil lebih baik. Secara umum, optimasi *Bayesian* sangat berguna untuk menyelesaikan masalah optimasi yang mahal, seperti *hyperparameter tuning*, desain eksperimen, sampai pemodelan sistem kompleks tanpa harus mencoba semua kemungkinan secara manual [13], [14].

Optuna adalah alat berbasis Python yang digunakan untuk mencari kombinasi parameter terbaik pada model *machine learning* secara otomatis dan efisien. Optuna, sebagai salah satu implementasi optimasi *Bayesian*, terbukti memiliki *trade-off runtime* dan skor performa yang baik pada kasus *Combined Algorithm Selection and Hyperparameter optimization (CASH)*, mengungguli HyperOpt, SMAC, dan Optunity di sebagian besar *dataset* riil. Fitur-fitur ini menjadikan Optuna efektif dan efisien untuk *hyperparameter tuning* secara otomatis pada berbagai model pembelajaran mesin [15]. Framework Optuna, dengan fitur *define-by-run* API, *pruning* otomatis, dan kemampuan *parallelization*, menjadi salah satu *open-source library* terkini yang mendukung pendekatan ini secara praktis, fleksibel, dan efisien [16].

Hingga saat ini penelitian pada data leukemia CuMiDa telah dilakukanmisalnya menggunakan pendekatan *Linear Programming* (LP) dan teknik *feature selection* [17]. Penelitian lain juga menggunakan metode *super-learner* yang menggabungkan beberapa model dasar dan memakai *Random Forest* sebagai *learner* akhir [18]. Namun, penelitian yang ada belum memanfaatkan optimasi *Bayesian* dengan Optuna untuk *hyperparameter tuning* kombinasi *Spectral Clustering–K-Means* pada data ekspresi gen leukemia dari CuMiDa. Oleh karena itu, penelitian ini bertujuan untuk menguji dan menganalisis lebih mendalam terhadap efektivitas kombinasi metode *Spectral Clustering–K-Means* yang dioptimasi menggunakan *Bayesian* melalui Optuna, untuk mengetahui sejauh mana pendekatan tersebut mampu meningkatkan pemisahan dan kualitas klaster pada data ekspresi gen leukemia dari CuMiDa.

## 2. Metode Penelitian

*Dataset* CuMiDa telah melalui tahap prapemrosesan data [3]. Selanjutnya data memasuki tahap algoritma *Spectral Clustering* yang intinya mereduksi dimensi *dataset* awal, disebut *Spectral Embedding*. Dalam pembentukan *Spectral Embedding*, penelitian ini menggunakan *Bayesian* untuk mengoptimalkan parameter-parameter dalam *Spectral Clustering*. Matriks hasil *Spectral Embedding* inilah yang kemudian diklaster menggunakan algoritma klastering *K-Means*.

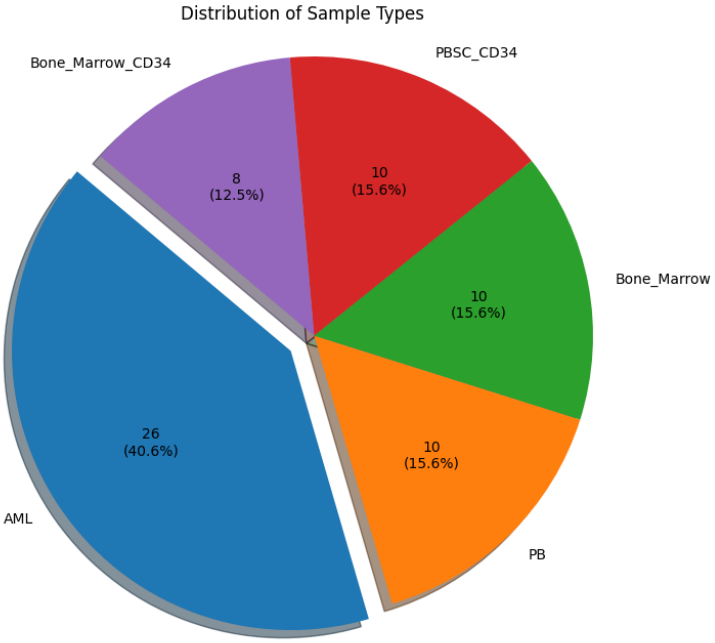
### 2.1 Pengumpulan Data

CuMiDa diperoleh dari *Structural Bioinformatics and Computational Biology Lab* (SBCB) yang terdiri atas 64 sampel data dan 22.285 ekspresi gen leukemia GSE9476. Data tersebut dapat diakses di https://sbcb.inf.ufrgs.br/cumida#datasets[19]. Sebagian *dataset* yang digunakan dalam penelitian ini ditunjukkan pada Gambar 1. *Dataset* telah diketahui terklaster menjadi 5 tipe kelas, yaitu AML, PB, Bone-Marrow, PBSC_CD34, dan Bone_Marrow_CD34. Distribusi sampel data berdasarkan kelasnya dapat dilihat pada Gambar 2. Telah dilakukan penelitian pada data tersebut menggunakan algoritma klastering *unsupervised learning: K-Means* dan *Hierarchical Clustering* (HC) [3] sehingga akan dibandingkan hasil klasteringnya menggunakan metode *Spectral Clustering–K-Means* yang dioptimasi menggunakan *Bayesian* melalui Optuna.

| | samples | type | 1007_s_at | 1053_at | 117_at | 121_at | 1255_g_at | 1294_at | 1316_at | 1320_at | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | Bone_Marrow_CD34 | 7.745245 | 7.811210 | 6.477916 | 8.841506 | 4.546941 | 7.957714 | 5.344999 | 4.673364 | ... |
| 1 | 12 | Bone_Marrow_CD34 | 8.087252 | 7.240673 | 8.584648 | 8.983571 | 4.548934 | 8.011652 | 5.579647 | 4.828184 | ... |
| 2 | 13 | Bone_Marrow_CD34 | 7.792056 | 7.549368 | 11.053504 | 8.909703 | 4.549328 | 8.237099 | 5.406489 | 4.615572 | ... |
| 3 | 14 | Bone_Marrow_CD34 | 7.767265 | 7.094460 | 11.816433 | 8.994654 | 4.697018 | 8.283412 | 5.582195 | 4.903684 | ... |
| 4 | 15 | Bone_Marrow_CD34 | 8.010117 | 7.405281 | 6.656049 | 9.050682 | 4.514986 | 8.377046 | 5.493713 | 4.860754 | ... |

5 rows × 22285 columns

Gambar 1. *Dataset* dari Ekspresi Microarray Leukemia




Gambar 2. Distribusi Sampel Gen berdasarkan Tipe Kelas

## 2.2 *Spectral Clustering*

Pada sebuah himpunan data yang terdiri dari $n$-titik, algoritma *Spectral Clustering* akan membentuk $n \times n$ matriks similaritas dan menghitung eigenvektor dari matriks tersebut. Misalkan diberikan sebuah himpunan $n$ titik data $x_1, x_2, \ldots, x_n$, dengan setiap $x_i \in \mathbb{R}_d$, graf similaritas $G = (V, E)$ didefinisikan sebagai sebuah graf tak berarah dimana simpul ke-$i$ berkorespondesi dengan titik data $x_i$. Untuk setiap busur $(i, j) \in E$, diberikan sebuah bobot $w_{ij} \geq 0$, yang mengartikan kemiripan/similaritas dari titik-titik data $x_i$ dan $x_j$. Matriks

$$W = (w_{ij})_{i,j=1}^{n} \tag{1}$$

disebut matriks similaritas (*similarity graph*). *Spectral Clustering* mempartisi data menjadi $k$ kelas yang saling asing (*disjoint*) sehingga setiap $x_i$ menjadi anggota di tepat salah satu kelas [20].

Penelitian ini menggunakan konstruksi graf terhubung penuh (*fully connected graph*). Semua titik akan dihubungkan dengan nilai kemiripan positif antar satu dengan lainnya, dan semua busur diberi bobot $s_{ij}$. Fungsi similaritas yang digunakan adalah fungsi similaritas Gauss, $s(x_i, x_j)$,

$$W = s(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \tag{2}$$

dengan parameter $\sigma$ menentukan "lebar" dari neighbourhood-nya [21].

Laplacian ternormalisasi digunakan dalam penelitian ini untuk meminimumkan similaritas data antar kelompok dan memaksimumkan similaritas antar data dalam satu kelompok. Laplacian ternormalisasi didefinisikan sebagai

$$L_{sym} \coloneqq D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = 1 - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \quad (3)$$

$D$ merupakan matriks diagonal dimana $d_i$ menandakan jumlah bobot semua busur yang bertetangga dengan simpul ke-$i$ dan $L = D - W$ [21].

Eigenvalue dari matriks Laplacian dihitung. *Spectral Clustering* menggunakan multiplisitas $n$ eigenvalue 0 atau mendekati 0 dari Graph Laplacian untuk mendapatkan informasi jumlah grup yang dapat dibentuk dari himpunan *vertex*. Matriks yang berisi *eigenvector* dari n *eigenvalue* tersebut menjadi representasi data yang selanjutnya akan diklastering.

### 2.3 *Hyperparameter Tuning* via Optimasi *Bayesian*

Optimasi *Bayesian* membangun model probabilistik untuk mempelajari pola dari hasil evaluasi sebelumnya. Kerangka kerja optimasi *Bayesian* memiliki dua komponen utama. Pertama, model perkiraan probabilistik yang terdiri dari distribusi awal (*prior*) yang menangkap perilaku fungsi tujuan yang tidak diketahui dan model observasi yang menjelaskan bagaimana data dihasilkan. Kedua, fungsi kerugian (*loss function*) yang menggambarkan seberapa optimal urutan percobaan yang dilakukan. Kerugian yang diharapkan akan diminimalkan untuk memilih urutan percobaan yang optimal [14]. Optimasi *Bayesian* memakai model perkiraan yang disebut *Gaussian Process* untuk memprediksi nilai rata-rata (mean) dan seberapa yakin (variansi) prediksi tersebut. Setelah fungsi dievaluasi di beberapa titik awal, model ini diperbaharui menggunakan aturan Bayes agar prediksi berikutnya makin akurat. Optimasi *Bayesian* memakai *acquisition function*, seperti *Expected Improvement* atau *Upper Confidence Bound* untuk memutuskan di mana harus mencoba lagi [22].

Optuna adalah sebuah *framework* modern untuk optimasi hyperparameter secara otomatis yang memanfaatkan pendekatan *Bayesian*. Optuna membantu menemukan kombinasi parameter terbaik menggunakan metode *define-by-run*, artinya membuat ruang pencarian parameter secara dinamis saat program berjalan tanpa perlu mendefinisikan semuanya di awal. Optuna juga memiliki fitur *pruning* otomatis yaitu memotong percobaan yang tidak menjanjikan sejak awal sehingga menghemat waktu dan sumber daya [16].

Dalam tahap ini terdapat empat parameter yang akan dioptimasi menggunakan metode *Bayesian*, yaitu:

a. *n_components*: menunjukkan jumlah dimensi output (*embedding*) yang merepresentasikan multiplisitas $n$ eigenvalue 0 atau mendekati 0. Artinya, data yang berdimensi $k$ (banyaknya kolom data) diproyeksikan ke ruang dimensi $n$ komponen, dengan $n < k$. Penelitian ini menggunakan $n\_components = 2,3, \dots ,10$.
b. *Affinity*: menunjukkan metode yang digunakan untuk membuat graf afinitas (*similiarity graph*). Ada dua metode yang akan dipilih, yaitu *Radial Basis Function* (RBF) dan *Nearest Neighbors*. *Nearest Neighbors* adalah metode menghubungkan simpul $v_i$ dengan simpul $v_j$ jika $v_j$ termasuk dalam $n$ tetangga terdekat dari $v_i$, kemudian busur-busur yang menghubungkan simpul-simpul bertetangga tersebut diberikan bobot berdasarkan similaritasnya [21]. Metode ini menghubungkan setiap titik data ke $n$ tetangga terdekatnya berdasarkan jarak Euclidean [7].
    - Jika *affinity = Nearest Neighbors* maka parameter *n_neighbours* yang dioptimasi, yaitu jumlah tetangga terdekat yang digunakan untuk membentuk matriks afinitas metode KNN. Penelitian ini menggunakan $n\_neighbors = 2,3, \dots ,10$.

RBF adalah metode yang memetakan data dari ruang berdimensi rendah ke ruang berdimensi lebih tinggi secara nonlinier, sehingga pola data yang awalnya sulit dipisahkan menjadi lebih mudah dipisahkan dengan garis lurus di ruang tersebut [23].

- Jika *affinity* = RBF maka parameter *gamma* yang dioptimasi, yaitu parameter dari kernel RBF yang mengatur sensitivitas kemiripan data. Penelitian ini menggunakan $gamma = 10^{-3}, 10^{-2}, \dots, 10$.

Proses optimasi *hyperparameter* pada penelitian ini dilakukan menggunakan bahasa pemrograman Python dengan bantuan *library* Optuna. Prosedur optimasi *hyperparameter Bayesian* diberikan oleh *pseudo-code* pada Gambar 3.

```
Algoritma: Optimasi Hyperparameter Berbasis Bayesian (Optuna)

Masukan:
    - Dataset X
    - Label sebenarnya y
    - Fungsi objektif f(·)
Keluaran:
    - Hyperparameter terbaik θ*

Langkah-langkah:

1. Inisialisasi studi dengan arah optimasi (misalnya: memaksimalkan skor purity)
2. Untuk setiap percobaan (trial) t dari 1 hingga N_trials lakukan:
      a. Bangkitkan kombinasi hyperparameter θ_t dari ruang pencarian
      b. Latih model klastering menggunakan parameter θ_t
      c. Hitung metrik performa f(θ_t) (misalnya purity score)
      d. Simpan hasil percobaan ke dalam studi
3. Setelah seluruh percobaan selesai, pilih parameter terbaik θ*
   yang menghasilkan nilai f(θ_t) tertinggi
4. Kembalikan θ* sebagai hyperparameter optimal
```

Gambar 3. *Pseudo-code* Optimasi *Hyperparameter*

Pada Gambar 3, Optuna akan memulai dengan membangkitkan sejumlah *trial* (kombinasi *hyperparameter* acak), kemudian mengevaluasi hasilnya berdasarkan metrik performa (pada penelitian ini digunakan metrik *purity score*). Selanjutnya, Optuna menggunakan pendekatan *Bayesian* untuk memperkirakan kombinasi parameter berikutnya yang berpotensi menghasilkan performa yang lebih baik. Proses ini diulang hingga mencapai jumlah *trial* maksimum atau tidak ditemukan peningkatan performa yang signifikan.

## 2.4 *K-Means* Clustering

*K-Means* digunakan sebagai metode untuk mengklaster hasil *Spectral Embedding*. Algoritma *K-Means* selalu memerlukan penentuan jumlah klaster ($k$) di awal. Jika nilai $k$ diubah, maka hasil klasterisasi juga bisa berbeda-beda [24]. Berikut langkah-langkah algoritma klastering *K-Means*:

- STEP 1: DEFINISIKAN BANYAKNYA KELAS KLASTER ($k$). PENELITIAN INI MENGGUNAKAN $k = 5$ KARENA *DATASET* DIKETAHUI BERASAL DARI 5 KELAS [19].
- STEP 2: PILIH *CENTROID* AWAL SECARA RANDOM UNTUK SETIAP KLASTER.
- STEP 3: HITUNG *SUM OF SQUARE ERROR* (SSE), YAITU KUADRAT DARI JARAK EUCLIDEAN, ANTARA SETIAP OBJEK DAN *CENTROID*:

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} dist^2(m_i, x) \tag{4}$$

dengan $x$ adalah titik data dalam klaster $C_i$ dan $m_i$ adalah *centroid* pada klaster $C$.

- Step 4: setiap objek di-*assigned* ke klaster terdekat.
- Step 5: *centroid* baru dihitung kembali untuk setiap klaster.
- Step 6: ulangi step 3 sampai 5 hingga posisi *centroid* tidak berubah [25].

2.5 Evaluasi Model

Hasil klastering dievaluasi menggunakan metode *Purity* (kemurnian). *Purity* adalah ukuran sejauh mana sebuah klaster hanya berisi satu kelas. Untuk setiap klaster, hitung jumlah titik data dari kelas yang paling banyak muncul di klaster tersebut, kemudian jumlahkan untuk semua kelas dan bagi dengan total jumlah titik data. Rumus dari *Purity* sebagai berikut:

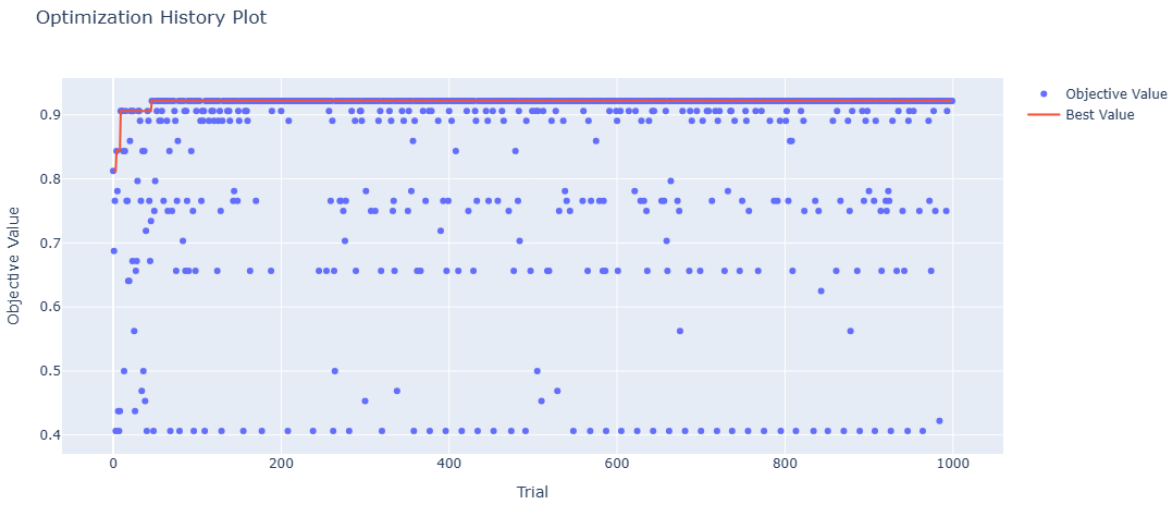
$$\frac{1}{N}\sum_{m\in M}\max_{d\in D}|m\cap d| \tag{5}$$

dengan $M$ klaster, $N$ titik data, dan $D$ kelas [26]. Selanjutnya, hasil evaluasi dengan metode *Purity* disebut sebagai akurasi dalam penelitian ini.

## 3. Hasil dan Pembahasan

3.1 Optimasi Parameter

Hasil optimasi *Bayesian* dalam penelitian ini terlihat pada Gambar 4. Sumbu mendatar menandakan jumlah percobaan optimasi (iterasi), yaitu 0 sampai 1.000 kali, sedangkan sumbu tegak menandakan nilai objektif yang ingin dimaksimalkan. Titik biru merepresentasikan nilai objektif pada setiap percobaan. Garis merah menandakan nilai objektif terbaik yang ditemukan hingga akhir percobaan. Terlihat bahwa nilai objektif semakin naik seiring bertambahnya percobaan optimasi. Pada iterasi awal (0 sampai 50), nilai objektif melonjak cepat dari sekitar 0,4 ke atas 0,9; artinya model menemukan kombinasi parameter yang bagus dengan cepat pada percobaan awal. Pada pertengahan iterasi (50 sampai 200), garis merah naik bertahap kemudian mulai stabil. Masih ada titik-titik yang nilai objektifnya lebih rendah, artinya terjadi proses eksplorasi yaitu mencoba mencari parameter lain untuk melihat apakah ada kombinasi yang lebih baik. Selanjutnya pada iterasi lanjut (200-1000), garis merah mendatar yang berarti tidak ada peningkatan signifikan dalam mencari nilai objektif terbaik. Banyak titik biru yang tersebar di bawah garis merah menandakan optimasi masih mengeksplorasi ruang parameter untuk memastikan tidak melewatkan potensi perbaikan.

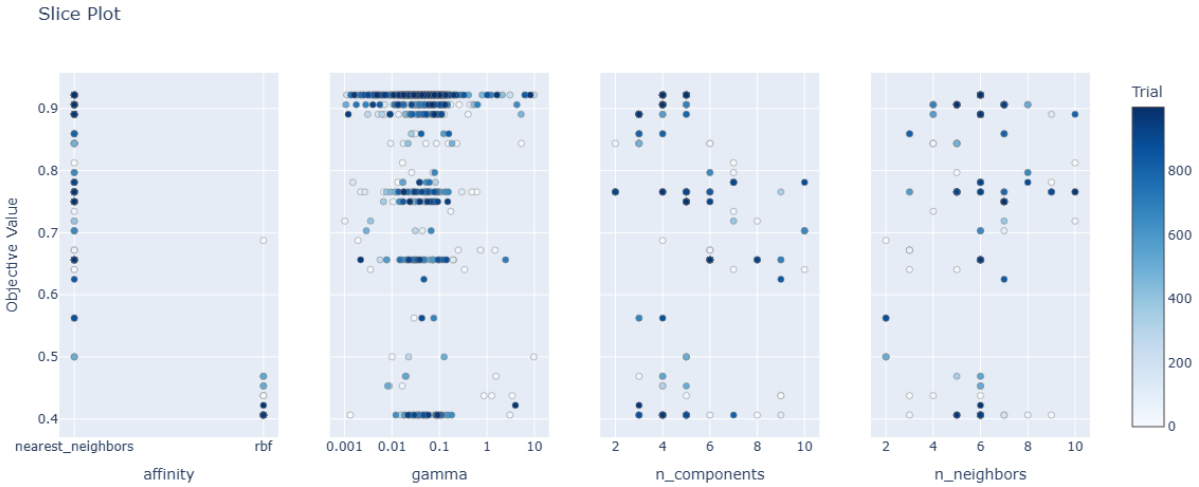



Gambar 4. Plot Hasil Optimasi *Bayesian*

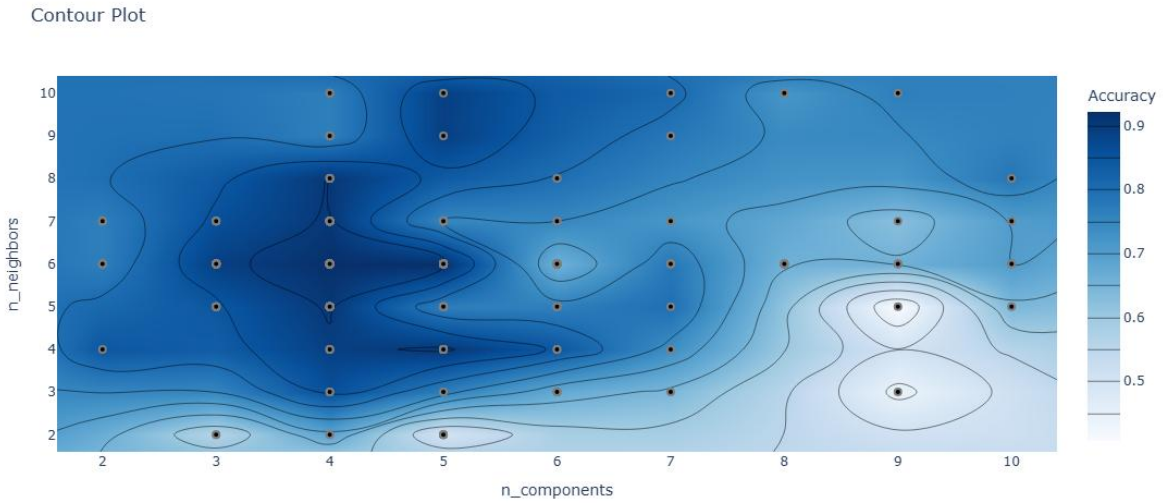
Detail hasil optimasi setiap parameter dapat dilihat pada Gambar 5. Sumbu mendatar menandakan nilai masing-masing parameter, sedangkan sumbu tegak menandakan nilai objektif yang ingin dimaksimalkan. Berikut penjelasan untuk masing-masing parameter:

- Pada parameter *affinity*, terdapat 2 parameter yang dipilih yaitu *Nearest Neighbors* dan RBF. Titik-titik untuk *Nearest Neighbors* umumnya menghasilkan nilai objektif lebih tinggi, terlihat dari banyak titik yang di atas 0,9; sedangkan titik-titik untuk RBF sebagian besar di bawah 0,5. Artinya untuk *dataset* ini, graf afinitas dengan *Nearest Neighbors* lebih konsisten menghasilkan kualitas klaster yang lebih baik dibandingkan RBF.
- Pada parameter gamma di RBF, terlihat titik-titik di gamma bervariasi mulai dari 0,001 sampai 10. Untuk sebagian besar nilai gamma, berapapun nilainya, nilai objektifnya relatif tetap rendah. Hal ini sesuai dengan hasil *affinity* bahwa secara umum variasi gamma tidak berpengaruh signifikan memperbaiki hasil.
- Pada parameter *n_components*, titik dengan nilai objektif tinggi, mendekati 0,9; lebih banyak pada *n_components* = 4 sampai 6. Artinya, data diproyeksikan ke ruang dimensi 4 sampai 6 komponen cenderung memberikan representasi klaster terbaik. Lebih detailnya mengenai pemilihan *n_components* terlihat pada Gambar 6. Sumbu mendatar menandakan jumlah dimensi output (jumlah kolom *Spectral Embedding*), sedangkan sumbu tegak menandakan *n_neighbors* yaitu jumlah tetangga terdekat. Hasil menunjukkan area biru tua, yaitu akurasi di atas 0,9; terkonsentrasi pada *n_components* sekitar 4 sampai 6 dan *n_neighbors* sekitar 5 sampai 8 yang berarti kombinasi proyeksi ke dimensi 4-6 dengan 5-8 tetangga terdekat memberikan kualitas klaster terbaik.
- Pada parameter *n_neighbors*, sebagian besar nilai objektif tinggi, yang lebih dari 0,9; terkonsentrasi di *n_neighbors* = 5 sampai 8. Artinya, matriks afinitas dengan membentuk 5 sampai 8 tetangga terdekat cenderung optimal.

Berdasarkan optimasi *Bayesian* dengan Optuna diperoleh hasil optimasi terbaik dari 1.000 percobaan, yaitu *n_components* = 5 dan *affinity* = *Nearest Neighbors* dengan *n_neighbors* = 6. Selanjutnya, hasil optimasi digunakan untuk membentuk matriks *Spectral Embedding*.




Gambar 5. Plot Hasil Optimasi setiap Parameter




Gambar 6. Kontur Pemilihan *n_components* dan *n_neighbors*

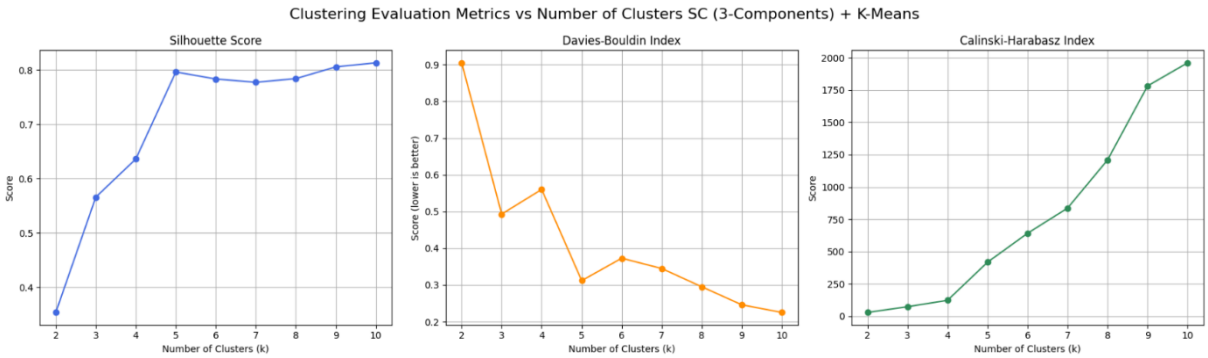
3.2 Klastering

Untuk mempermudah dalam melihat performa *Spectral Clustering,* maka penelitian ini menampilkan visualisasi dalam *n_components* = 3 (3D). Penelitian ini melakukan visualisasi 3D *dataset* menggunakan metode *Spectral Embedding* dengan *n_neighbors* = 4 yang dapat dilihat pada Gambar 7. Gambar 7 menunjukkan bahwa klaster terpisah secara spasial. Bone_Marrow (merah) tampak membentuk klaster yang kompak, Bone_Marrow_CD34 (biru) juga membentuk klaster yang relatif terpisah, sedangkan klaster AML (hijau) mulai terlihat terbagi menjadi subklaster yang lebih terdefinisi. Berdasarkan visualisasi, metode *Spectral Embedding* cukup baik dalam mengidentifikasi struktur klaster, dimana memberikan klaster antar tipe terpisah dan kompak untuk selanjutnya dianalisis menggunakan *K-Means*.


Gambar 7. Visualisasi *Dataset Spectral Embedding*

Hasil klastering kemudian dievaluasi akurasinya berdasarkan skor Silhouette, indeks Davies-Bouldin, dan indeks Calinski Harabasz. Gambar 8 menunjukkan nilai akurasi untuk *Spectral Embedding* 3 komponen. Pada Gambar 8, skor Silhoutte meningkat tajam di $k = 5$ mencapai sekitar 0,8 dan stabil tinggi hingga $k = 10$. Ini mengindikasikan struktur klaster yang jelas dan *well-separated* sejak $k = 5$. Kemudian indeks Davies-Bouldin menurun signifikan setelah $k = 5$, dimana semakin kecil indeks maka semakin bagus, dan terus

membaik mencapai nilai terndah di sekitar $k = 9 - 10$. Selanjutnya, indeks Calinski Harabasz naik drastis dari $k = 5$ ke atas, dimana semakin besar nilainya maka semakin jelas pemisahan antar klaster, hingga mencapai puncak sekitar $k = 10$. Berdasarkan ketiga nilai akurasi, *Spectral Embedding* memberi struktur klaster yang kuat dan *K-Means* menunjukkan performa sangat baik mulai $k = 5$ ke atas.




Gambar 8. Evaluasi *Spectral Embedding* 3 Komponen menggunakan *K-Means*

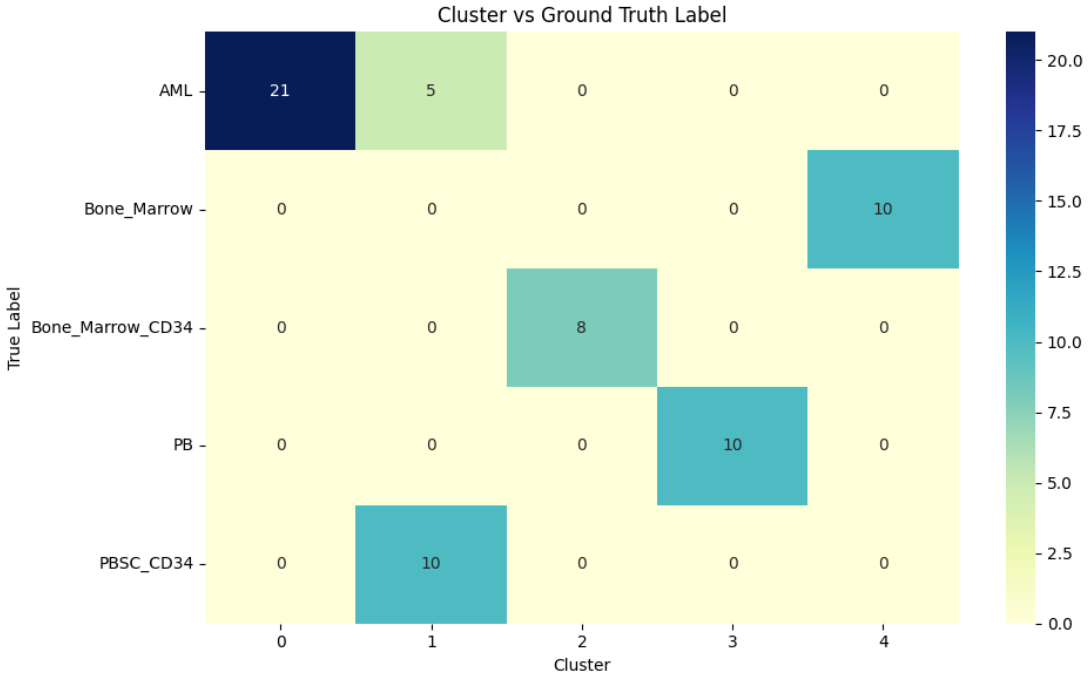
Mengacu pada hasil optimasi Bayesian, matriks hasil *Spectral Embedding* terdiri dari 5 kolom dan 64 baris. Sebagian matriks hasil *Spectral Embedding* ditunjukkan pada Gambar 9. Kolom matriks hasil *Spectral Embedding* ini berupa eigenvektor-eigenvektor. Matriks inilah yang kemudian diklaster berdasarkan barisnya menggunakan algoritma *K-Means*.

Hasil algoritma klastering *K-Means* dapat dilihat pada Gambar 10. Hasil menunjukkan ada 21 sampel anggota klaster ke-0 yang merupakan sampel kelas AML, ada 15 sampel anggota klaster ke-1 yang 10 diantaranya merupakan sampel kelas PBSC_CD34 dan 5 diantaranya salah pengelompokan, ada 8 sampel anggota klaster ke-2 yang merupakan sampel kelas Bone_Marrow_CD34, ada 10 sampel anggota klaster ke-3 yang merupakan sampel kelas PB, dan ada 10 sampel anggota klaster ke-4 yang merupakan sampel kelas Bone-Marrow.

| | x1 | x2 | x3 | x4 | x5 |
|---|---|---|---|---|---|
| **0** | -0.053277 | -0.002093 | 0.111102 | -0.038496 | 0.011491 |
| **1** | -0.053277 | -0.003184 | 0.132273 | -0.064502 | 0.022104 |
| **2** | -0.053277 | -0.003840 | 0.136104 | -0.069929 | 0.021193 |
| **3** | -0.053277 | -0.003583 | 0.134974 | -0.067981 | 0.021814 |
| **4** | -0.053277 | -0.005368 | 0.120019 | -0.047715 | -0.000713 |
| **...** | ... | ... | ... | ... | ... |
| **59** | -0.053277 | -0.074506 | -0.053735 | -0.043597 | 0.040244 |
| **60** | -0.053277 | -0.075205 | -0.054596 | -0.045656 | 0.041789 |
| **61** | -0.053277 | -0.067145 | -0.044776 | -0.024431 | 0.012094 |
| **62** | -0.053277 | -0.071923 | -0.050251 | -0.035779 | 0.025360 |
| **63** | -0.053277 | -0.070100 | -0.048008 | -0.031473 | 0.018678 |

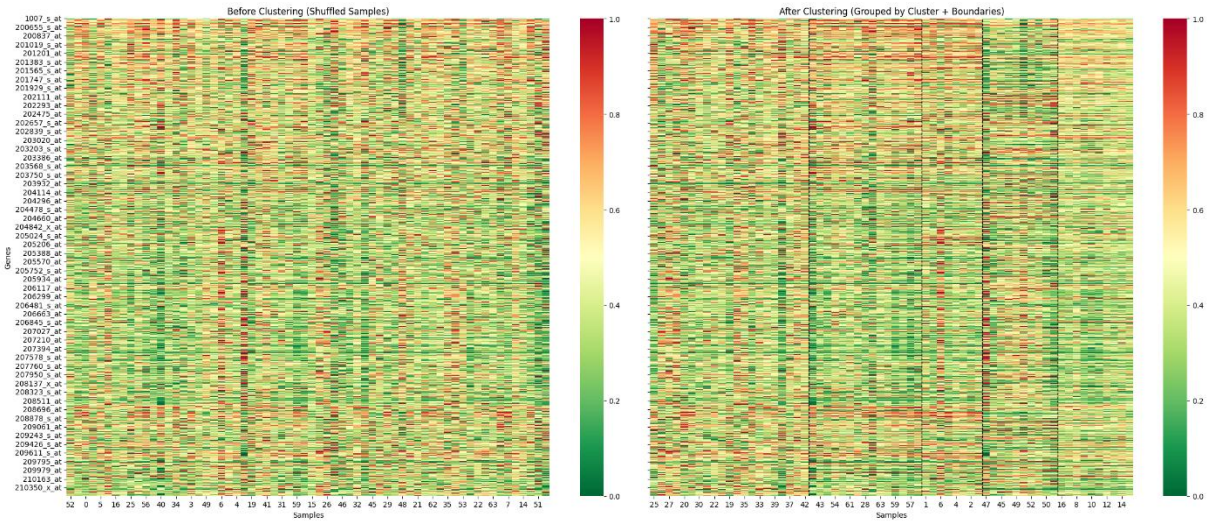
64 rows × 5 columns

Gambar 9. Matriks Hasil *Spectral Embedding*




Gambar 10. Pemetaan Hasil Klastering setiap Kelas

Perbandingan *heatmap* 10.000 gen *dataset* sebelum dan sesudah *Spectral Clustering – K-Means* (SC-KM) dapat dilihat pada Gambar 11. *Heatmap* sebelah kiri menampilkan data ekspresi gen awal yang masih acak sehingga pola kemiripan atau pola ekspresi gen sejenis belum tampak jelas. Warna hijau-kuning-merah masih tersebar merata menandakan variasi data namun belum terkelompok. *Heatmap* sebelah kanan menampilkan data yang sama namun sampel sudah dikelompokkan atau diurutkan berdasarkan hasil klastering sehingga sampel yang mirip berdekatan. Setiap blok vertikal terpisah oleh batas yang menunjukkan hasil pembagian sampel ke klaster. Dalam setiap blok, pola warna cenderung lebih seragam atau konsisten yang menandakan ekspresi gen dalam satu klaster memiliki kemiripan pola.




Gambar 11. Perbandingan *Heatmap* Gen

Perbandingan hasil klastering menggunakan metode *Hierarchical Clustering* (HC), *K-Means*, dan SC-KM dapat dilihat pada Tabel 1.

Tabel 1. Perbandingan Hasil Klastering

| Algoritma | Hasil Klastering | |
|---|---|---|
| | Pengelompokan tidak Benar | Akurasi |
| HC | 38 sampel | 40,63% [26] |
| *K-Means* | 21 sampel | 67,19% [26] |
| SC-KM | 5 sampel | 92,19% |

Hasil pada Tabel 1 menunjukkan bahwa metode SC-KM memberikan akurasi lebih tinggi dibandingkan kedua metode *unsupervised* lainnya. Hal ini berarti metode SC-KM lebih baik dalam melakukan pengelompokan. Hasil ini menunjukkan bahwa representasi *spectral embedding* pada SC-KM mampu memetakan pola hubungan antar sampel secara lebih efektif dibandingkan pendekatan konvensional. Ketika digabungkan dengan optimasi *hyperparameter* berbasis Bayesian, metode ini menghasilkan pemisahan klaster yang lebih akurat dan stabil pada data ekspresi gen leukemia.

## 4. Kesimpulan

Penelitian ini menunjukkan bahwa kombinasi optimasi *Bayesian* dengan Optuna dapat digunakan secara efektif untuk melakukan *hyperparameter tuning* pada metode *Spectral Clustering – K-Means* (SC-KM) untuk *dataset* ekspresi gen Leukemia (CuMiDa). Berdasarkan hasil optimasi 1.000 iterasi, diperoleh kombinasi parameter terbaik yaitu *n_components* = 5, *affinity* = *Nearest Neighbors*, dan *n_neighbors* = 6, yang mampu memproyeksikan data ke ruang *embedding* yang tepat dan membentuk matriks similaritas yang sesuai.

Berdasarkan evaluasi menggunakan tiga metrik klastering, yaitu skor Silhouette, indeks Davies-Bouldin, dan indeks Calinski Harabasz, diperoleh hasil yang konsisten bahwa reduksi dimensi menggunakan *Spectral Embedding* (3 eigen vektor) memberikan hasil yang optimal untuk *K-Mean*s. Hasil klastering juga menunjukkan bahwa metode SC-KM menggunakan optimasi *Bayesian* mampu memisahkan sampel ke dalam klaster dengan akurasi 92,19%, lebih tinggi dibandingkan *Hierarchical Clustering* (40,63%) maupun *K-Means* (67,19%). Hal ini menunjukkan bahwa pendekatan *Spectral Clustering* yang digabung dengan *K-Means* dan optimasi *hyperparameter* dengan *Bayesian* terbukti lebih efektif untuk mendeteksi pola subtipe leukemia pada data berdimensi tinggi. Visualisasi *heatmap* juga memperlihatkan pola ekspresi gen yang awalnya acak menjadi lebih terkelompok dan seragam di dalam masing-masing klaster. Secara keseluruhan, penelitian ini menunjukkan bahwa kombinasi *Spectral Clustering-K-Means* dan optimasi *Bayesian* menggunakan Optuna dapat meningkatkan kualitas klastering pada data ekspresi gen yang kompleks serta membuka peluang penerapan pendekatan ini atau menggunakan metode lain untuk mengklaster hasil *Spectral Embedding* seperti *Fuzzy C-Means*, *Partition Around Medoids* (PAM), atau pendekatan klastering alternatif lainnya pada masalah bioinformatika yang serupa.

## Daftar Referensi

[1] M. Ilyas, K. M. Aamir, S. Manzoor, and M. Deriche, “Linear programming based computational technique for leukemia classification using gene expression profile,” *PLoS One*, vol. 18, no. 10 October, Oct. 2023, doi: 10.1371/journal.pone.0292172.

[2] F. Joanda Kaunang, B. Hakim, F. Fraderic, S. Hartono, and A. Kristanto Mulyanto, “Breast Cancer Detection using Decision Tree and Random Forest,” 2025. doi: https://doi.org/10.30871/jaic.v9i2.9073.

[3] B. C. Feltes, E. B. Chandelier, B. I. Grisci, and M. Dorn, “CuMiDa: An Extensively Curated Microarray Database for Benchmarking and Testing of Machine Learning

Approaches in Cancer Research," *Journal of Computational Biology*, vol. 26, no. 4, pp. 376–386, Apr. 2019, doi: 10.1089/cmb.2018.0238.

[4] J. Shen *et al.*, "Deep learning approach for cancer subtype classification using high-dimensional gene expression data," *BMC Bioinformatics*, vol. 23, no. 1, Dec. 2022, doi: 10.1186/s12859-022-04980-9.

[5] A. Gupta, H. Sharma, and A. Akhtar, "A Comparative Analysis of K-Means and Hierachichal Clustering," *EPRA International Journal of Multidisciplinary Research (IJMR)-Peer Reviewed Journal*, no. 8, 2021, doi: 10.36713/epra2013.

[6] B. Hakim, F. Joanda Kaunang, C. Susanto, J. Salim, and R. Indradjaja, "Implementasi Machine Learning dalam Pengelompokkan Musik Menggunakan Algoritma K-Means Clustering," 2025. doi: https://doi.org/10.36080/idealis.v8i1.3357.

[7] K. Berahmand, F. Saberi-Movahed, R. Sheikhpour, Y. Li, and M. Jalili, "A Comprehensive Survey on Spectral Clustering with Graph Structure Learning," Jan. 2025, doi: https://doi.org/10.48550/arXiv.2501.13597.

[8] L. V. Xiaodan, "An Improved Automated Spectral Clustering Algorithm," *Journal of Information Processing Systems*, vol. 20, no. 2, pp. 185–199, Apr. 2024, doi: 10.3745/JIPS.04.0307.

[9] F. Sadjadi, V. Torra, and M. Jamshidi, "Preprocessed Spectral Clustering with Higher Connectivity for Robustness in Real-World Applications," *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, Dec. 2024, doi: 10.1007/s44196-024-00455-2.

[10] E. Al-sharoa and S. Aviyente, "A Unified Spectral Clustering Approach for Detecting Community Structure in Multilayer Networks," *Symmetry (Basel)*, vol. 15, no. 7, Jul. 2023, doi: 10.3390/sym15071368.

[11] D. Y. Bernanda, D. N. A. Jawawi, S. A. Halim, and F. Adikara, "Natural Language Processing For Requirement Elicitation In University Using Kmeans And Meanshift Algorithm," *Baghdad Science Journal*, vol. 21, no. 2, pp. 561–567, 2024, doi: 10.21123/bsj.2024.9675.

[12] M. Cendana and R. J. Kuo, "Categorical Data Clustering: A Bibliometric Analysis and Taxonomy," Jun. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/make6020047.

[13] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, "Recent Advances in Bayesian Optimization," *ACM Comput Surv*, vol. 55, no. 13s, 2023, doi: 10.1145/3582078.

[14] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," Jan. 01, 2016, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/JPROC.2015.2494218.

[15] S. Shekhar, A. Bansode, and A. Salim, "A Comparative study of Hyper-Parameter Optimization Tools," Jan. 2022, doi: https://doi.org/10.48550/arXiv.2201.06433.

[16] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," Jul. 2019, doi: https://doi.org/10.48550/arXiv.1907.10902.

[17] M. Ilyas, K. M. Aamir, S. Manzoor, and M. Deriche, "Linear programming based computational technique for leukemia classification using gene expression profile," *PLoS One*, vol. 18, no. 10 October, Oct. 2023, doi: 10.1371/journal.pone.0292172.

[18] S. Selvaraj *et al.*, "Super learner model for classifying leukemia through gene expression monitoring," *Discover Oncology*, vol. 15, no. 1, Dec. 2024, doi: 10.1007/s12672-024-01337-x.

[19] B. C. Feltes, E. B. Chandelier, B. I. Grisci, and M. Dorn, "CuMiDa: An Extensively Curated Microarray Database," SBCB Lab. Accessed: Oct. 26, 2025. [Online]. Available: https://sbcb.inf.ufrgs.br/cumida

[20] D. Yan, L. Huang, and M. I. Jordan, "Fast approximate spectral clustering," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 907–915. doi: 10.1145/1557019.1557118.

[21] U. von Luxburg, "A Tutorial on Spectral Clustering," Nov. 2007, [Online]. Available: http://arxiv.org/abs/0711.0189

[22] Z. Yi, Y. Wei, C. X. Cheng, K. He, and Y. Sui, "Improving sample efficiency of high dimensional Bayesian optimization with MCMC," Jan. 2024, doi: https://doi.org/10.48550/arXiv.2401.02650.

[23] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A Practical Guide to Support Vector Classification," 2003. Accessed: Oct. 23, 2025. [Online]. Available: https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[24] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," Aug. 01, 2020, *MDPI AG*. doi: 10.3390/electronics9081295.

[25] A. A. Mousa, M. A. El-Shorbagy, and M. A. Farag, "K-means-clustering based evolutionary algorithm for multi-objective resource allocation problems," *Applied Mathematics and Information Sciences*, vol. 11, no. 6, pp. 1681–1692, Nov. 2017, doi: 10.18576/amis/110615.

[26] S. Suresh Sikhakolli and A. Kiran Sikhakolli DrDY, "Effective Purity Method for Measuring the Clustering Accuracy and its Illustration," NY, USA, May 2023. doi: 10.5120/ijca2023922752.