BAB IV
IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi Menggunakan Aplikasi *RapidMiner*

Implementasi metode *K-Means Clustering* dalam proses pengelompokan minat dan bakat siswa dapat dilakukan secara efektif dengan menggunakan aplikasi *RapidMiner*, yang merupakan salah satu perangkat lunak analitik data berbasis visual dan tanpa kode (*no-code*). *RapidMiner* menyediakan antarmuka berbasis *drag-and-drop* yang memungkinkan pengguna dari kalangan pendidikan untuk melakukan analisis data secara praktis dan efisien tanpa harus memiliki kemampuan pemrograman. Dalam konteks ini, *RapidMiner* dapat digunakan untuk memproses data siswa dan mengelompokkannya ke dalam beberapa klaster berdasarkan kesamaan karakteristik minat dan bakat yang dimiliki oleh setiap individu.

Tahapan pertama dari proses implementasi ini adalah menyiapkan data siswa yang memuat informasi mengenai minat dan bakat mereka. Data tersebut dapat diperoleh dari hasil kuesioner minat bakat, asesmen psikologis, maupun hasil observasi guru terhadap aktivitas siswa baik di dalam maupun di luar lingkungan sekolah. Setiap siswa biasanya dievaluasi dalam berbagai aspek, seperti minat dalam bidang akademik (seperti matematika, sains, dan bahasa), minat dalam bidang non-akademik (seperti seni dan olahraga), serta potensi bakat kognitif dan motorik.

Setelah data terkumpul, langkah berikutnya adalah melakukan *preprocessing* data di dalam *RapidMiner*. Tahapan ini melibatkan proses pembersihan data dari nilai kosong (missing values), normalisasi data agar berada dalam skala yang seragam (menggunakan teknik seperti *Z-Score* atau *Min-Max Normalization*), serta transformasi atribut non-numerik menjadi numerik (menggunakan teknik seperti *one-hot encoding*). Semua proses ini penting agar algoritma *K-Means* dapat bekerja secara optimal dalam mengelompokkan data.

Setelah data siap, pengguna dapat memulai proses klasterisasi dengan menyeret operator *K-Means* ke dalam *process panel* di *RapidMiner*. Dalam penerapan metode ini, pengguna perlu menentukan jumlah klaster (*k*) yang diinginkan berdasarkan tujuan analisis. Misalnya, apabila tujuan klasterisasi adalah untuk mengelompokkan siswa ke dalam tiga kelompok besar berdasarkan minat dominan mereka (akademik, seni, dan olahraga), maka nilai *k* yang digunakan adalah tiga. Proses ini dilakukan dengan menyeret operator seperti *Read Excel* untuk memuat data, *Replace Missing Values* untuk membersihkan data, *Normalize* untuk menyesuaikan skala data, dan *Select Attributes* untuk memilih atribut yang relevan. Setelah itu, operator *K-Means* digunakan untuk melakukan klasterisasi dan dilanjutkan dengan operator *Cluster Model* serta *Scatter Plot* untuk mengevaluasi dan memvisualisasikan hasil klasterisasi.

Hasil klasterisasi yang ditampilkan dalam bentuk *scatter plot* akan menunjukkan bagaimana setiap siswa dikelompokkan berdasarkan kesamaan karakteristik. Setiap titik dalam visualisasi tersebut merepresentasikan satu siswa, dan setiap warna atau simbol menunjukkan klaster tempat siswa tersebut

tergabung. Visualisasi ini sangat membantu dalam mengidentifikasi pola distribusi minat dan bakat di antara siswa, sehingga pengguna dapat dengan cepat mengenali kelompok-kelompok yang memiliki potensi atau kecenderungan serupa.

4.2 Implementasi Algoritma *K-Means* Menggunakan *RapidMiner*

Algoritma *K-Means* adalah salah satu metode pembelajaran tanpa supervisi yang paling sering diterapkan dalam pengolahan data untuk mengelompokkan data menjadi beberapa kategori berdasarkan kesamaan di antara data tersebut. Cara kerja algoritma ini adalah dengan membagi sekumpulan data ke dalam sejumlah k kluster berdasarkan jarak terdekat ke titik pusat kluster yang dikenal sebagai *centroid*. Dalam penerapannya menggunakan perangkat lunak *RapidMiner*, proses pengelompokan dengan *K-Means* dapat dilakukan dengan cara yang efisien dan mudah dipahami melalui antarmuka visual yang berbasis pada alur kerja, yang memfasilitasi eksplorasi dan percobaan terhadap data.

Pelaksanaan algoritma *K-Means* di *RapidMiner* dimulai dengan menyiapkan data yang akan dianalisis. Data ini dapat berasal dari berbagai sumber seperti *file* Excel, CSV, basis data SQL, atau platform penyimpanan awan. Setelah data dimasukkan ke dalam *RapidMiner* menggunakan operator seperti *Read* CSV atau *Retrieve*, langkah selanjutnya adalah data *preprocessing*. Proses ini meliputi pemilihan atribut yang relevan, normalisasi nilai numerik menggunakan operator seperti *Normalize*, serta transformasi tipe data jika diperlukan. Proses normalisasi

sangat krusial karena algoritma *K-Means* peka terhadap skala yang berbeda antar atribut.

Setelah data disiapkan, pengguna dapat menambahkan operator *K-Means* ke dalam panel proses. Parameter penting yang perlu ditentukan dalam operator ini adalah jumlah kluster (k) yang diinginkan. Nilai k dapat ditentukan berdasarkan pemahaman awal tentang data atau melalui pendekatan yang lebih sistematis seperti metode *elbow*, *silhouette*, atau gap *statistic*, yang semuanya dapat didukung dengan operator analisis tambahan di *RapidMiner*. Selanjutnya, pengguna dapat mengonfigurasi parameter seperti cara pengukuran jarak (contohnya *Euclidean Distance* atau *Cosine Similarity*) dan jumlah maksimal iterasi untuk memastikan proses konvergensi tetap optimal. Proses klasterisasi dilakukan dengan membentuk *centroid* awal secara acak, lalu menghitung jarak setiap data terhadap *centroid* tersebut untuk mengelompokkan data ke kluster terdekat. Proses ini berlanjut hingga tidak ada perubahan signifikan pada *centroid* atau sampai batas iterasi maksimum tercapai. Hasil dari operator *K-Means* adalah model kluster yang bisa divisualisasikan melalui operator *Cluster Model Viewer* atau *Scatter Plot*, yang menunjukkan sebaran data berdasarkan atribut yang dipilih beserta warna yang membedakan setiap kluster.

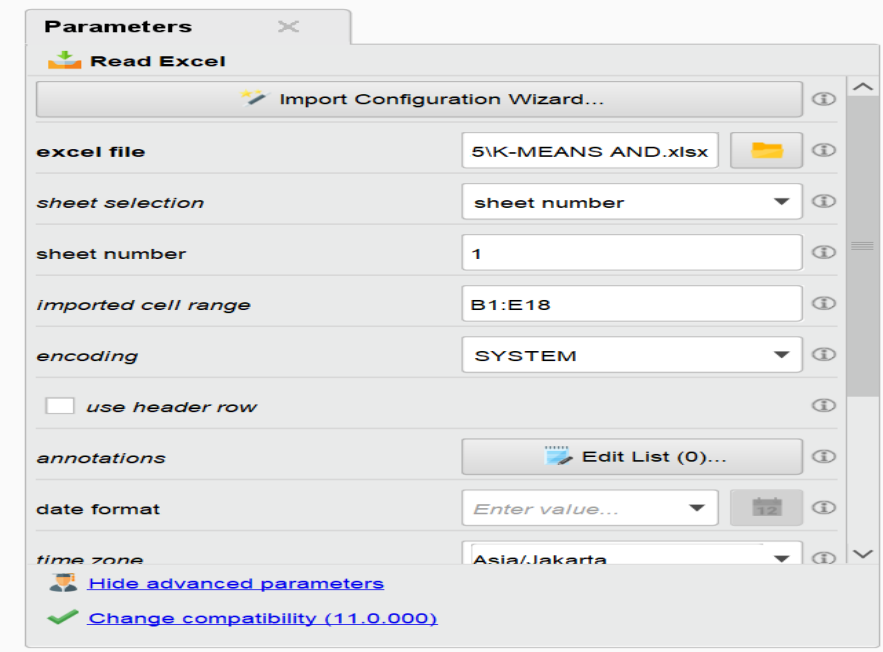
RapidMiner juga menawarkan operator evaluasi seperti *Cluster Performance*, yang dapat dipakai untuk menghitung indeks validitas kluster seperti total kuadrat dalam kluster, *Davies-Bouldin Index*, dan *Silhouette Coefficient*. Evaluasi ini sangat membantu dalam menilai seberapa baik pemisahan kluster yang terbentuk, sekaligus memberikan umpan balik dalam

menentukan nilai k yang paling optimal. Beberapa penelitian menyarankan untuk mengintegrasikan algoritma *K-Means* dengan metode visualisasi berbasis PCA (*Principal Component Analysis*) agar dapat memahami struktur kluster dengan lebih baik, yang juga didukung oleh berbagai operator di *RapidMiner*. Dengan fitur yang ada, penerapan algoritma *K-Means* menggunakan *RapidMiner* dapat meningkatkan keefektifan analisis data tanpa memerlukan keterampilan pemrograman yang tinggi. Hal ini membuat *RapidMiner* menjadi platform yang sangat cocok untuk keperluan klasterisasi di bidang pendidikan, bisnis, dan penelitian ilmiah.

Dalam pengaplikasian metode *K-Means* menggunakan aplikasi *RapidMiner* operator yang digunakan diantaranya :

1. *Read* Excel

Fungsi *Read* Excel pada *RapidMiner* digunakan untuk mengimpor data dari file Excel ke dalam proses analisis. Operator ini memungkinkan pengguna memilih sheet, menentukan rentang sel, dan mengatur format data. Dengan *Read* Excel, pengguna dapat mempersiapkan data awal sebelum dilakukan preprocessing atau pemodelan data lebih lanjut. Gambar dibawah menunjukkan bahwa pengguna sedang dalam tahap awal proses data mining, yaitu mengimpor data dari Excel ke *RapidMiner* untuk selanjutnya dianalisis. Rentang data yang diimpor hanya mencakup sebagian sheet (B1:E18), tidak menggunakan *header*, dan disiapkan untuk tahap selanjutnya seperti normalisasi atau klasterisasi menggunakan algoritma *K-Means*.



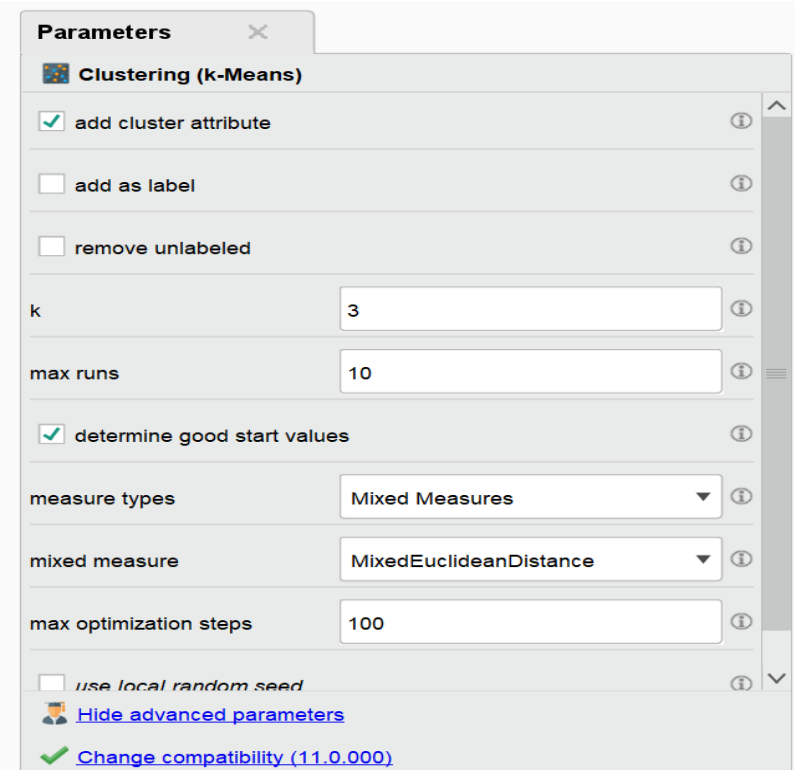
Gambar 4.1 *Read* Excel

Gambar 4.1 menunjukkan pengaturan parameter pada node *Read Excel* di KNIME *Analytics Platform. File* yang digunakan bernama *"5\K-MEANS AND.xlsx"* dengan metode pemilihan sheet berdasarkan nomor, yaitu sheet nomor 1. Rentang sel yang diimpor adalah dari B1 hingga E18. *Encoding* diset pada opsi "*SYSTEM*" dan opsi "*use header row*" tidak dicentang, yang berarti baris pertama tidak dianggap sebagai nama kolom. Zona waktu diatur ke Asia/Jakarta. Pengguna juga memiliki opsi untuk mengatur format tanggal dan menambahkan anotasi, namun keduanya belum diisi.

2. Operator *K-Means*

Operator *K-Means* pada *RapidMiner* berfungsi untuk mengelompokkan data ke dalam beberapa klaster berdasarkan kemiripan nilai atribut. Operator ini menggunakan algoritma *centroid* untuk membagi data menjadi *k* kelompok, di mana setiap kelompok memiliki pusat (*centroid*) yang merepresentasikan rata-rata

dari anggota klaster tersebut. Dalam prosesnya, *K-Means* akan secara iteratif memperbarui posisi *centroid* dan mengelompokkan ulang data hingga mencapai konvergensi. Operator ini sangat efektif digunakan untuk analisis segmentasi, termasuk dalam kasus klasterisasi minat dan bakat siswa, karena mampu mengidentifikasi pola tersembunyi dan membentuk kelompok berdasarkan karakteristik yang serupa.



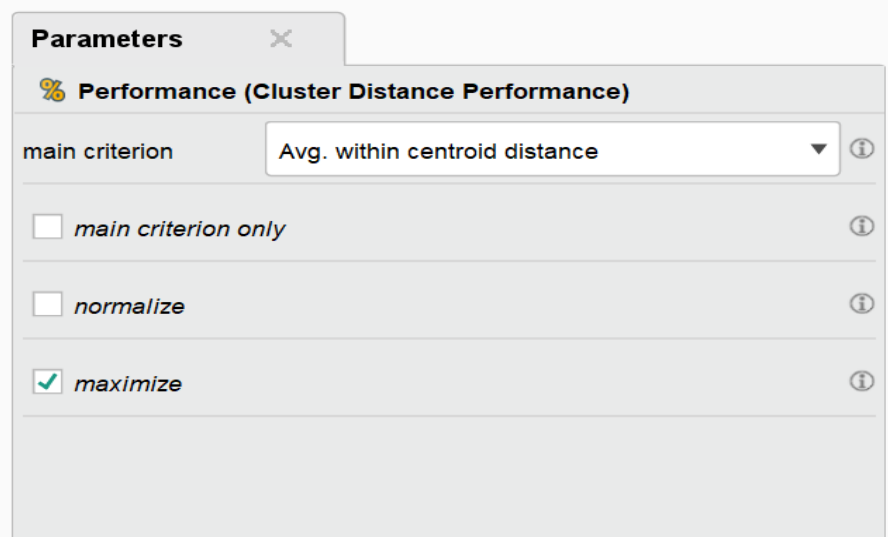
Gambar 4.2 Operator *K-Means*

Konfigurasi pada gambar menunjukkan bahwa pengguna ingin membagi data ke dalam 3 kelompok menggunakan algoritma *K-Means* dengan mempertimbangkan atribut campuran (numerik dan kategorikal), menggunakan pendekatan *Mixed Euclidean Distance*, dan dengan pengaturan iterasi yang cukup untuk menghasilkan hasil klasterisasi yang optimal dan stabil. Hasilnya akan

ditampilkan sebagai atribut tambahan yang dapat digunakan untuk analisis lanjutan terhadap data yang telah dikelompokkan.

3. Performance (Cluster Distance Performance)

Fungsi *Performance* (*Cluster Distance Performance*) pada *RapidMiner* digunakan untuk mengevaluasi kualitas hasil klasterisasi dengan menghitung seberapa baik objek dalam satu klaster memiliki kesamaan, dan seberapa besar perbedaannya dengan klaster lain. Operator ini menghitung metrik seperti *Within-Cluster Distance* (jarak dalam klaster) dan *Between-Cluster Distance* (jarak antar klaster), yang berguna untuk menilai kepadatan dan pemisahan klaster. Semakin kecil jarak dalam klaster dan semakin besar jarak antar klaster, semakin baik hasil klasterisasi. Evaluasi ini membantu menentukan apakah jumlah klaster yang dipilih sudah optimal.

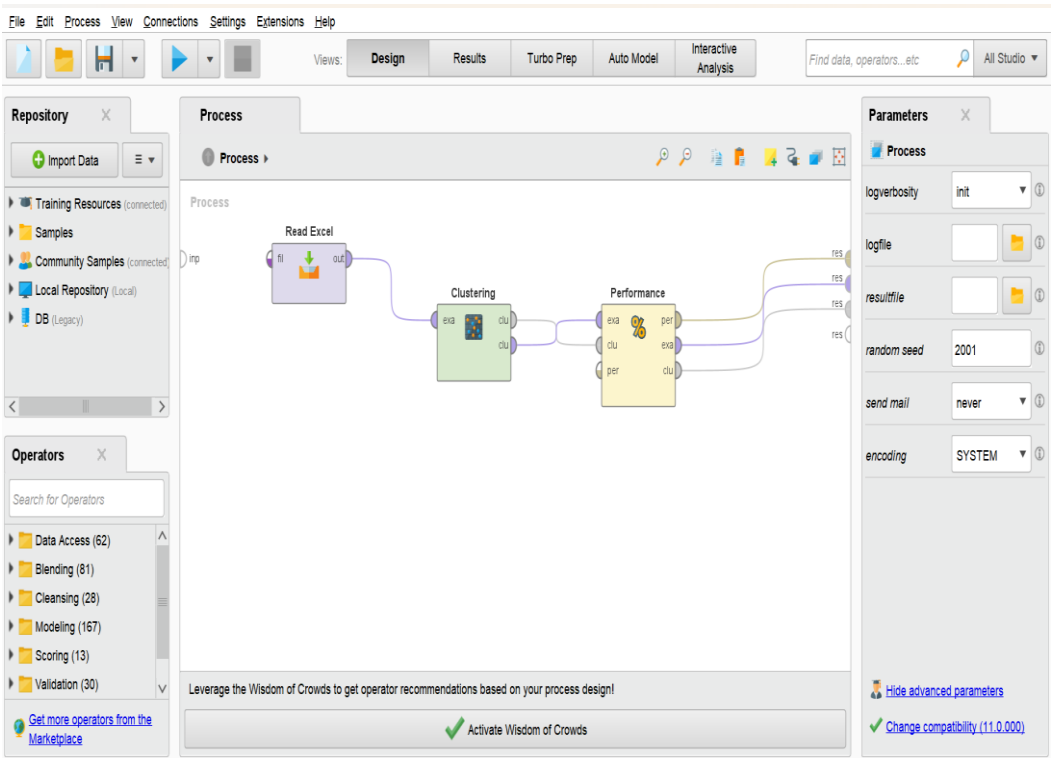


Gambar 4.3 *Performance (Cluster Distance Performance)*

Gambar 4.3 menunjukkan konfigurasi parameter pada operator *Performance (Cluster Distance Performance)* di *RapidMiner*. Pada bagian *main*

criterion, pengguna memilih metrik *Avg. within centroid distance*, yang berfungsi untuk menghitung rata-rata jarak data terhadap pusat klasternya. Opsi *main criterion only* dan *normalize* tidak dicentang, sehingga metrik lain masih dapat dihitung dan data tidak dinormalisasi otomatis. Sementara itu, opsi *maximize* dicentang, yang menandakan bahwa sistem akan mencari nilai kinerja tertinggi berdasarkan metrik yang dipilih. Pengaturan ini bertujuan mengevaluasi kepadatan klaster dan menentukan kualitas hasil klasterisasi secara kuantitatif.

4. Desain *K-Means* pada *RapidMiner*

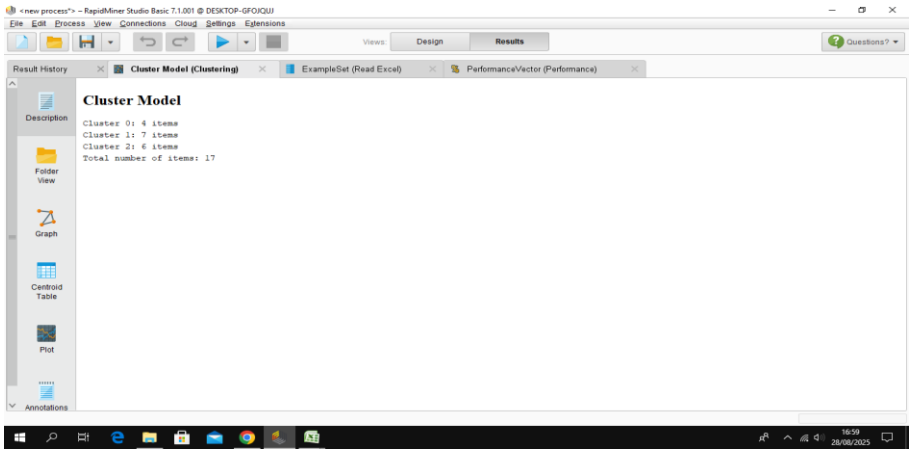


Gambar 4.4 Desain *K-Means*

4.3 Hasil dan Pembahasan

Hasil dan analisis dari penggunaan *K-Means* dalam aplikasi *RapidMiner* menunjukkan bahwa data berhasil dikelompokkan ke dalam beberapa klaster sesuai dengan jumlah klaster yang telah ditetapkan sebelumnya. Proses ini

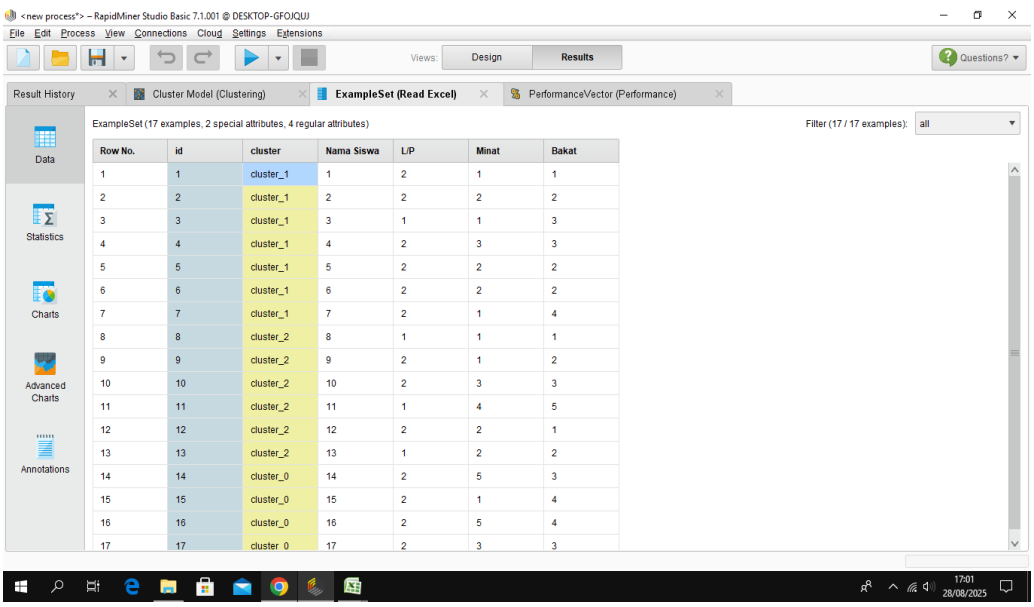
dimulai dengan langkah pra-pemrosesan data, kemudian dilanjutkan dengan penerapan algoritma *K-Means* menggunakan operator "*K-Means Clustering*" di *RapidMiner*, di mana sistem secara otomatis menghitung pusat klaster (*centroid*) dan mengelompokkan data berdasarkan jarak terdekat dari setiap *centroid* dengan metode Jarak *Euclidean*. Berdasarkan hasil yang diperoleh, *RapidMiner* menyajikan distribusi data dalam setiap klaster, rata-rata atribut untuk masing-masing klaster, serta visualisasi dua dimensi yang memudahkan pemahaman pola. Evaluasi menggunakan "*Performance (Cluster Distance Performance*)" menunjukkan nilai jarak dalam klaster (*intra-cluster distance*) yang rendah dan jarak antara klaster (*inter-cluster distance)* yang relatif tinggi, yang menandakan bahwa hasil klasterisasi cukup efektif. Hal ini membuktikan bahwa algoritma *K-Means* pada *RapidMiner* dapat dimanfaatkan secara efisien untuk mengidentifikasi pola tersembunyi dalam data dengan struktur klaster yang jelas dan terpisah.



Gambar 4.5 *Cluster Model*

Gambar 4.5 menampilkan hasil dari proses klasterisasi menggunakan algoritma *K-Means* di *RapidMiner*, yang ditampilkan pada

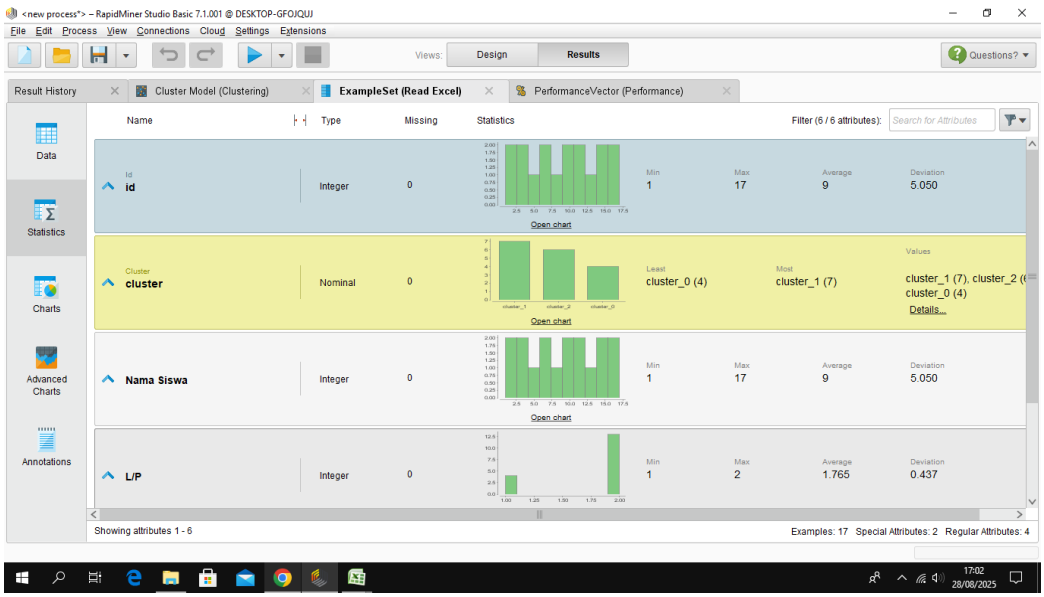
jendela *Cluster* Model (*Clustering*). Hasil tersebut menunjukkan bahwa data berhasil dikelompokkan menjaditigaklaster,yaitu *Cluster* 0 dengan 4 item,*Cluster* 1dengan 7 item, dan *Cluster*2 dengan 6 item. Secara keseluruhan terdapat 17 item yang dikelompokkan. Distribusi ini menunjukkan bahwa mayoritas data termasuk dalam *Cluster* 0, sedangkan dua klaster lainnya memiliki jumlah item lebih sedikit. Informasi ini berguna untuk memahami penyebaran data dalam masing-masing kelompok dan karakteristik dominan pada tiap klaster.



Gambar 4.6 *ExampleSet*

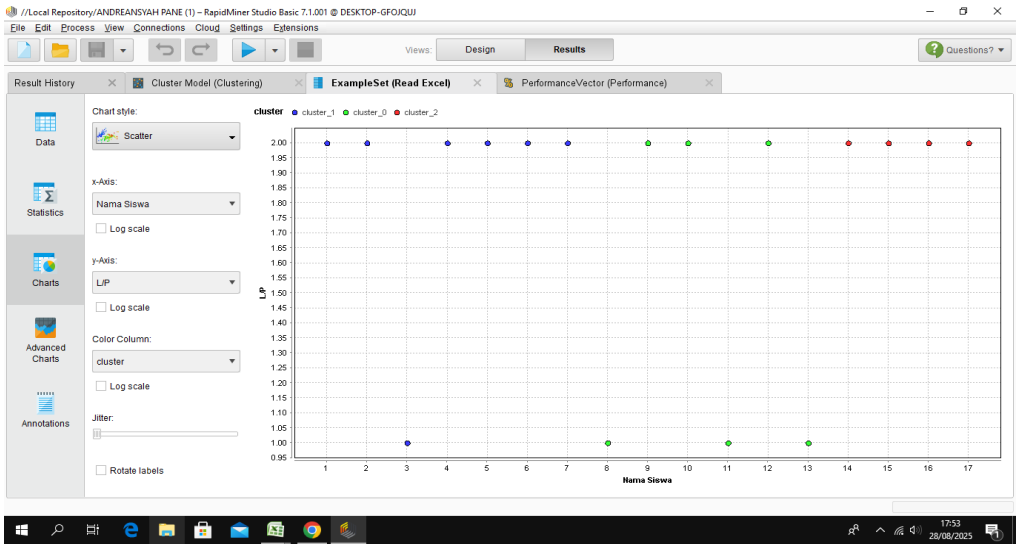
Gambar 4.6 menunjukkan tampilan hasil klasterisasi pada tab *ExampleSet* (*Clustering*) di *RapidMiner*. Data terdiri dari 18 baris yang masing-masing memiliki atribut id, *cluster*, dan tiga atribut numerik dan Kolom *cluster* menampilkan hasil pengelompokan data, yaitu ke dalam *cluster*_1, *cluster*_2, dan *cluster*_0. Mayoritas data masuk ke *cluster*_0,

sesuai dengan hasil sebelumnya. Tampilan ini memberikan gambaran jelas tentang bagaimana tiap data dikelompokkan berdasarkan kemiripan nilai pada atribut sehingga mempermudah analisis karakteristik tiap klaster secara visual dan numerik.



Gambar 4.7 *Statistic*

Gambar 4.7 menunjukkan tampilan tab Statistics di *RapidMiner* yang menampilkan analisis statistik deskriptif terhadap atribut dari hasil klasterisasi. atribut bertipe data integer tanpa nilai yang hilang. Atribut id memiliki nilai minimum 1, maksimum 17, rata-rata 9 dan deviasi standar 5.050. Atribut nama siswa memiliki nilai minimum 1, maksimum 17, rata-rata 9 dan deviasi 5.050. Sedangkan atribut L/P memiliki rentang nilai antara1 sampai 2, rata-rata1.765, dan deviasi 0,437. Informasi ini berguna untuk melihat sebaran nilai data sebelum dilakukan interpretasi lebih lanjut terhadap hasil klasterisasi



Gambar 4.8 *Scatter Plot*

Gambar yang dilampirkan menunjukkan hasil visualisasi klasterisasi menggunakan grafik scatter plot di *RapidMiner*. Sumbu X merepresentasikan atribut D, sedangkan sumbu Y menunjukkan nilai atribut E. Warna titik menunjukkan pembagian klaster, yaitu *cluster*_0 (biru), *cluster*_1 (hijau), dan *cluster*_2 (orange). Setiap titik mewakili satu item data, dan distribusinya menggambarkan pola keterkelompokan berdasarkan atribut D dan E. Titik-titik biru terlihat mendominasi area tengah dan bawah, sementara titik oranye menyebar ke area atas. Visualisasi ini membantu memahami sebaran data dan kekompakan masing-masing klaster secara intuitif dalam dua dimensi.