

BAB 2

TINJAUAN PUSTAKA

2.1 Profil SMAN 1 Serang Baru

Penelitian ini dilaksanakan di sekolah SMAN 1 Serang Baru. Berikut merupakan uraian profil dari sekolah SMAN 1 Serang Baru.

2.1.1 Sejarah SMAN 1 Serang Baru

SMAN 1 Serang Baru (SMANSASERA) merupakan salah satu Lembaga Pendidikan Menengah Atas di kabupaten Bekasi, Provinsi Jawa Barat yang menyelenggarakan Program Pendidikan, yang pembangunan fisiknya dimulai sejak tahun 2007, di atas tanah seluas 8.000 m², dan telah menerima siswa sejak tahun 2006 dengan nama SMA Negeri 1 Serang Baru, yang diresmikan pada tanggal 6 Januari 2007.

Perkembangan SMAN 1 Serang Baru dari sejak berdiri pada tahun 2007 oleh Kabupaten Bekasi berpindah pengelolaan ke Provinsi Jawa Barat pada 2017, sedangkan nama SMAN 1 Serang Baru – berlaku sama sejak tahun pelajaran 2007 dengan berdasarkan SK Pendirian 421/kep.15-Disdik/2007.

Secara geografis, SMAN 1 Serang Baru terletak di Perumahan Kota Serang Baru (KSB), sebelah selatan Kabupaten Bekasi, yang masih dapat dijangkau dari berbagai sudut di Kabupaten Bekasi, Kota Bekasi, Kabupaten Bogor, dan Jakarta dengan mudah, serta dapat dengan mudah diakses oleh industri-industri yang berada di kawasan industri Deltamas, Jababeka, dan Karawang. Sehingga akses yang dapat dilakukan untuk kepentingan akademik baik pembelajaran di sekolah maupun di luar sekolah (dunia usaha / dunia industri) dapat dilakukan dengan baik.

2.1.2 Visi, Misi, dan Tujuan SMAN 1 Serang Baru

2.1.2.1 Visi

Sekolah Unggul, Berakhlakul Karimah, Disiplin, Berdaya Saing, Nyaman, Berprestasi.

2.1.2.2 Misi

1. Memperkuat Imtaq dan budaya silaturahmi sekolah.
2. Meningkatkan budaya disiplin dengan konsisten dan konsekuen melaksanakan SOP setiap kegiatan di sekolah.
3. Memberdayakan semua *stakeholder* dan menjadi tauladan.
4. Menjaga nama baik dan melestarikan lingkungan sekolah.
5. Meningkatkan kerjasama dengan semua pihak yang saling menguntungkan (simbiosis mutualisme).

2.1.2.3 Tujuan

Mewujudkan insan yang berakhlak mulia, cerdas dalam berkarya, profesional dalam bekerja sesuai bidang dan kompetensi masing-masing.

2.1.3 Logo SMAN 1 Serang Baru

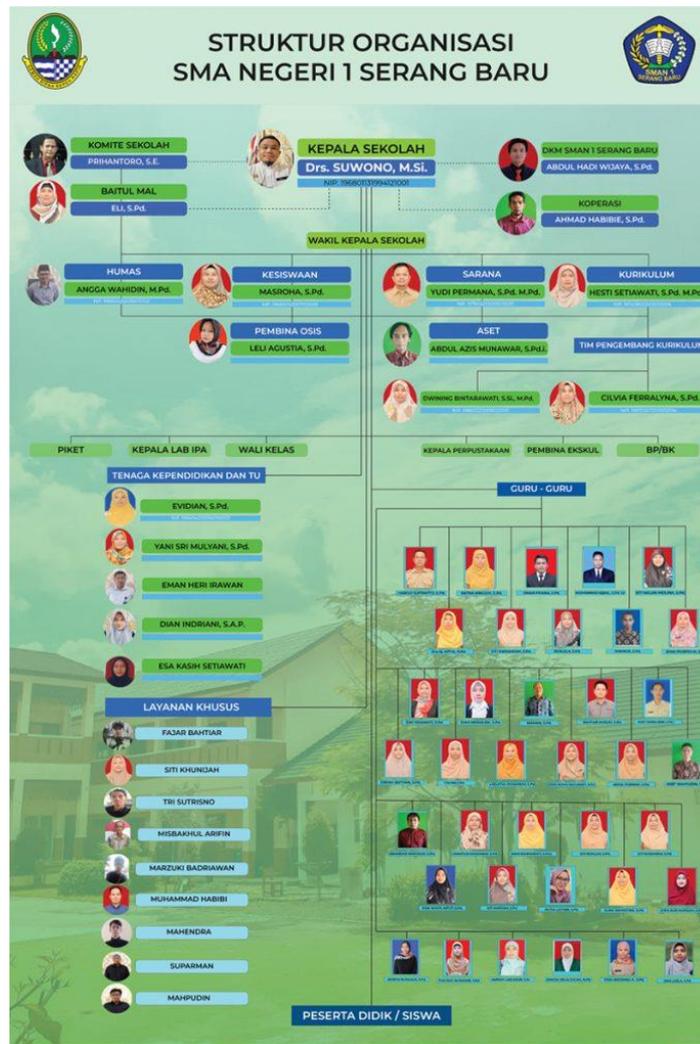
Logo dari sekolah SMAN 1 Serang Baru dapat dilihat pada Gambar 2.1.



Gambar 2.1 Logo SMAN 1 Serang Baru

2.1.4 Struktur Organisasi

Struktur organisasi dari Sekolah SMAN 1 Serang Baru dapat dilihat pada Gambar 2.2.



Gambar 2.2 Struktur Organisasi SMAN 1 Serang Baru

2.1.4.1 Jobdesk

Berikut merupakan jobdesk dari Guru Bimbingan Konseling (BK) selaku pihak yang akan dibantu dalam penelitian ini:

Guru BK bertanggung jawab untuk memantau serta memastikan setiap siswa berperilaku baik sesuai dengan visi dan misi pendidikan yang dicanangkan oleh pihak sekolah. Selain itu, Guru BK juga memiliki tanggung jawab untuk menggali dan mengarahkan potensi dan kondisi positif yang dimiliki siswa agar dapat dipelihara dan dikembangkan dengan lebih baik lagi sebagai bekal untuk kesuksesan bagi siswa di masa depan. Guru BK juga bertanggung jawab atas segala

urusan yang berkaitan dengan beasiswa, seperti menginformasikan adanya beasiswa yang ditawarkan kepada seluruh siswa, memilih siswa untuk direkomendasikan sebagai penerima beasiswa, serta memaksimalkan adanya beasiswa yang masuk ke sekolah SMAN 1 Serang Baru.

2.2 Landasan Teori

Landasan teori ini berisikan teori-teori pendukung yang digunakan dalam proses analisis dan implementasi pada permasalahan yang diangkat dalam penelitian ini.

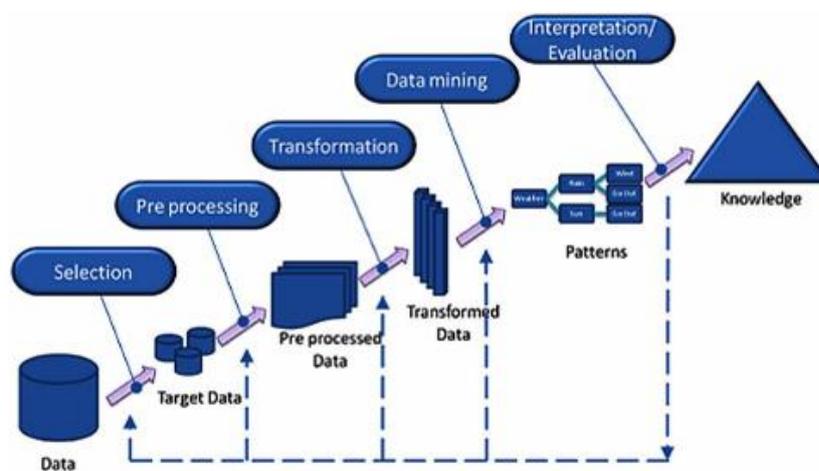
2.2.1 Data

Data adalah sekumpulan keterangan atau fakta yang dibuat dengan simbol, angka, kata-kata maupun kalimat [7]. Data didapatkan melalui sebuah proses pencarian serta pengamatan berdasarkan sumber-sumber tertentu. Data merupakan fakta mentah yang perlu dilakukan pengolahan agar data tersebut dapat menghasilkan informasi yang dapat dipahami dengan mudah.

Data dapat dikelompokkan berdasarkan data menurut sifat, sumber, cara memperoleh, dan waktu pengumpulan [8]. Data menurut sifatnya dibedakan menjadi data kualitatif dan data kuantitatif. Data kualitatif adalah data yang tidak berbentuk angka, sedangkan data kuantitatif adalah data yang dinyatakan dalam bentuk angka. Data menurut sumbernya mengacu kepada sumber perolehan data, yakni data eksternal dan internal. Data internal adalah data yang bersumber dari suatu organisasi atau kelompok, sedangkan data eksternal adalah data yang bersumber dari luar organisasi atau kelompok tertentu. Data menurut cara memperolehnya dibedakan menjadi data primer dan sekunder. Data primer adalah data yang dikumpulkan dan diolah langsung dari objeknya sedangkan data sekunder adalah data yang diperoleh dalam bentuk jadi yang sebelumnya telah diolah oleh pihak lain. Data menurut waktu pengumpulannya dibedakan menjadi data *cross section* dan data berkala (*time series*). Data *cross section* adalah data yang dikumpulkan pada periode tertentu, sedangkan data berkala adalah data yang dikumpulkan dari waktu ke waktu.

2.2.2 Knowledge Discovery in Databases

Knowledge Discovery in Databases (KDD) adalah proses dalam metode data mining untuk mengekstraksi apa yang dianggap sebagai suatu pengetahuan sesuai dengan spesifikasi ukuran dan batasan menggunakan *database* dengan melakukan proses *preprocessing*, *sub sampling*, dan transformasi *database* yang diperlukan [6]. Adapun tahapan dari proses *KDD* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Proses *Knowledge Discovery in Databases*

Berikut adalah penjelasan dari masing-masing tahapan :

1. *Selection*

Tahap ini terdiri dari pembuatan atau pengumpulan data set yang akan diolah untuk dilakukan penemuan pengetahuan (*knowledge discovery*).

2. *Pre processing*

Tahap ini terdiri dari proses pembersihan data mencakup membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data.

3. *Transformation*

Pada tahap ini, terjadi proses transformasi pada data yang telah dipilih, sehingga data tersebut dapat dipastikan telah sesuai untuk proses *data*

mining. Proses ini merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam *database*.

4. *Data Mining*

Tahap ini merupakan proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam data mining sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses *KDD* secara keseluruhan.

5. *Interpretation / Evaluation*

Tahap ini merupakan bagian dari proses *KDD* yang mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya. Kemudian, pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan.

Proses *KDD* diawali dengan pengembangan pemahaman tentang domain aplikasi, pengetahuan sebelumnya yang relevan, serta tujuan dari pengguna akhir. Proses ini harus dilanjutkan dengan konsolidasi pengetahuan, dengan cara menggabungkan pengetahuan yang didapatkan ke dalam sistem. Proses *KDD* bersifat interaktif dan berulang, melibatkan banyak langkah dengan banyak keputusan yang dibuat oleh pengguna [6].

2.2.3 *Data Mining*

Data mining adalah serangkaian proses menggali atau menambang suatu kumpulan data untuk menghasilkan pengetahuan yang selama ini tidak diketahui secara manual [9]. *Data mining* dapat digunakan untuk memenuhi tujuan tertentu. Salah satunya adalah *data mining* dapat digunakan sebagai sarana untuk mencari pola baru yang sebelumnya tidak terdeteksi. Secara sederhana, *data mining* akan mengubah data mentah menjadi data siap pakai yang dapat digunakan untuk pengambilan keputusan.

Secara umum ada dua jenis metode pada *data mining*, yakni metode *predictive* dan metode *descriptive*. Metode *predictive* adalah proses untuk menemukan pola dari data yang menggunakan beberapa variabel untuk memprediksi variabel lain yang tidak diketahui jenis atau nilainya. Teknik yang termasuk dalam *predictive mining* yakni klasifikasi, regresi dan deviasi. Sedangkan, metode *descriptive* adalah proses untuk menemukan suatu karakteristik penting dari kumpulan data. Teknik *data mining* yang termasuk dalam *descriptive mining* adalah *clustering*, *association* dan *sequential mining* [10].

2.2.4 Clustering

Clustering adalah proses mempartisi sekumpulan objek data atau hasil observasi ke dalam himpunan bagian. *Clustering* juga dikenal sebagai segmentasi data dalam beberapa aplikasi, karena *clustering* akan mengelompokkan data yang besar ke dalam kelompok berdasarkan kesamaannya. Dalam konteks pembelajaran mesin, *clustering* merupakan salah satu contoh dari unsupervised learning [11]. Setiap bagian memiliki karakteristik sedemikian rupa sehingga objek-objek dalam sebuah *cluster* memiliki kemiripan satu sama lain yang berbeda dengan objek di *cluster* lainnya [12]. Hasil *clustering* dapat dikatakan baik jika data yang ada di dalam suatu *cluster* memiliki kemiripan yang tinggi serta memiliki ketidakmiripan yang tinggi dengan data yang ada di *cluster* lain. Kualitas hasil *cluster* yang terbentuk bergantung pada karakteristik data dan algoritma yang digunakan, sehingga kesamaan karakteristik antar data dengan algoritma sangat dibutuhkan agar dapat menghasilkan hasil *clustering* yang baik.

Clustering digunakan secara luas tidak hanya untuk mengatur dan mengkategorikan data, tetapi juga berguna untuk kompresi data dan konstruksi model [13]. Kegunaan lain dari *clustering* yaitu sebagai segmentasi data yang berguna untuk memprediksi dan menganalisa masalah bisnis, serta mengidentifikasi obyek dalam berbagai bidang. Dengan kepadatan dan jarak daerah dari sebuah kelompok, akan memudahkan penemuan dan identifikasi pola distribusi dan korelasi antar atribut.

2.2.5 Metode Clustering

Secara umum, terdapat dua jenis metode *clustering* yang sering digunakan, yakni *Hierarchical Clustering* dan *Non-Hierarchical Clustering*. *Hierarchical Clustering* merupakan proses *clustering* dengan cara membentuk partisi rekursif dari kumpulan data menjadi kelompok yang lebih kecil secara berurutan. Metode ini digambarkan sebagai sebuah pohon berakar di mana setiap daun mewakili titik data dan setiap simpul internal mewakili *cluster* yang berisi daun keturunannya [14]. Metode ini biasa digunakan pada data yang jumlahnya tidak terlalu banyak dan jumlah *cluster* yang akan dibentuk belum diketahui. Sedangkan *Non-Hierarchical Clustering* merupakan proses *clustering* di mana jumlah *cluster* dapat ditentukan sebelumnya [15].

Algoritma yang digunakan dalam metode *Hierarchical Clustering* adalah *Agglomerative Hierarchical Clustering* dan *Divisive Hierarchical Clustering*. Algoritma *agglomerative* (pemusatan) merupakan metode *Hierarchical Clustering* dengan pendekatan bawah-atas (*bottom-up*) [16]. Sedangkan *divisive* (pembagian) adalah metode *Hierarchical Clustering* yang dimulai dari satu *cluster* tunggal kemudian dipisah sampai setiap objek berada dalam *cluster* yang terpisah. Kemudian, algoritma yang sering digunakan dalam metode *Non-Hierarchical Clustering* adalah K-Means. K-Means merupakan salah satu algoritma *Non-Hierarchical Clustering* yang berusaha mempartisi data yang ada ke dalam bentuk dua atau lebih *cluster* menggunakan “*mean*” yaitu rata-rata sebagai pusat *cluster*-nya [17].

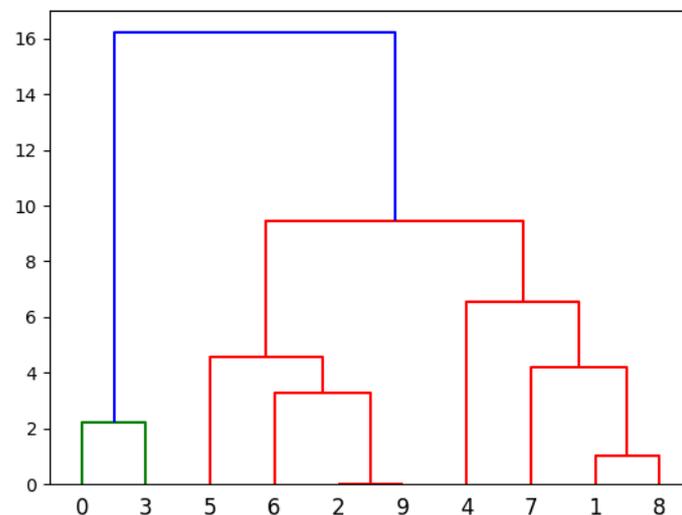
2.2.5.1 Agglomerative Hierarchical Clustering

Agglomerative merupakan suatu strategi pengelompokan hirarki yang dimulai dengan setiap objek dalam satu *cluster* yang terpisah kemudian bergabung dengan membentuk *cluster* yang semakin membesar [18]. *Agglomerative Hierarchical Clustering* adalah metode *hierarchical clustering* yang akan membentuk satu *cluster* tunggal dengan menyatukan N buah *cluster*. Metode ini diawali dengan meletakkan setiap data sebagai satu *cluster* tersendiri (*atomic*

cluster) kemudian menyatukan beberapa *cluster* tersebut menjadi sebuah *cluster* tunggal [19].

Berikut merupakan langkah-langkah dari algoritma *Agglomerative Hierarchical Clustering* :

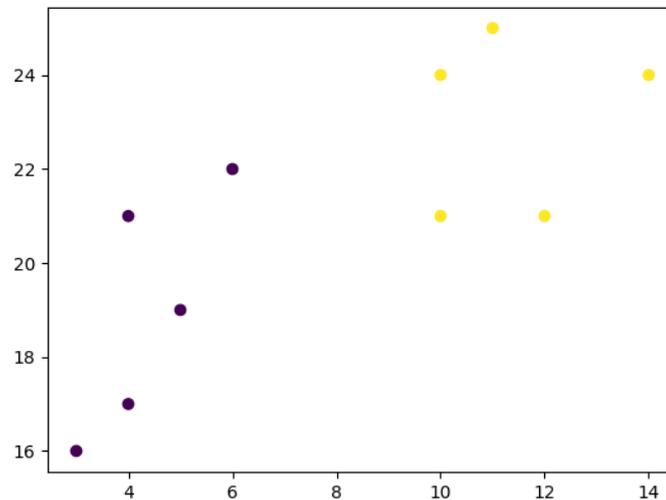
1. Hitung matrik jarak antar data menggunakan *Euclidean Distance*.
 2. Gabungkan kelompok terdekat menjadi satu berdasarkan penghitungan *ward*.
 3. Hitung kembali jarak antara *cluster* yang baru terbentuk dengan titik data lainnya.
 4. Ulangi langkah 2 sampai dengan 3 sampai terbentuk 1 *cluster* untuk seluruh data. Langkah ini akan diulang sampai terbentuk 1 *cluster* sesuai dengan ide dasar *Agglomerative Hierarchical Clustering*.
 5. Evaluasi *cluster* terbaik menggunakan perhitungan *DBI (Davies Bouldin Index)*.
 6. Bentuk dendogram untuk memvisualisasikan *cluster* yang terbentuk.
- Contoh bentuk dendogram dapat dilihat pada Gambar 2.4.



Gambar 2.4 Dendogram

Visualisasi dari proses *clustering* tidak selalu menggunakan dendogram. Banyak jenis-jenis visualisasi yang dapat menampilkan hasil dari proses penemuan

pengetahuan menggunakan *data mining*, salah satunya adalah *Scatter Plot*. Contoh bentuk *scatter plot* dapat dilihat pada Gambar 2.5.



Gambar 2.5 *Scatter Plot*

2.2.5.2 Ward

Analisis *cluster* dengan metode *Ward* merupakan metode yang berusaha untuk meminimalkan *variance* antar objek dalam satu *cluster* [20]. Metode ini akan menggabungkan dua *cluster* yang memiliki nilai *cost* terkecil ketika dua *cluster* tersebut digabungkan [21]. Metode *Ward* paling tepat digunakan untuk memproses variabel kuantitatif, dan bukan variabel biner. Hasil perhitungan analisis *cluster* dengan metode *Ward* dapat diperoleh dengan rumus :

$$Ward(S_i, S_j) = \frac{N_{S_i} N_{S_j}}{N_{S_i} + N_{S_j}} d(c_{S_i}, c_{S_j})$$

Nilai N_{S_i} dan c_{S_i} masing-masing mewakili kardinalitas dan *centroid* dari *cluster* S_i , sementara N_{S_j} dan c_{S_j} mewakili kardinalitas dan *centroid* dari *cluster* S_j , dan $d(c_{S_i}, c_{S_j})$ adalah fungsi yang mengembalikan jarak antara *centroid* masing-masing *cluster* S_i dan S_j .

2.2.6 Perhitungan Jarak

Dalam analisis data, jarak digunakan untuk mengukur seberapa dekat atau seberapa jauh antara dua objek atau titik data. Ada beberapa perhitungan jarak yang umum digunakan, seperti *Euclidean Distance*, *Manhattan Distance*, *Minkowski Distance*, dan lain sebagainya. Perhitungan jarak untuk mengukur kedekatan biasanya digunakan dalam pengelompokan untuk membentuk *cluster-cluster* berdasarkan kedekatan jarak antar masing-masing objek sehingga keberagaman antar objek dalam *cluster* lebih kecil daripada keberagaman antar *cluster* [22]. Ketika sebuah algoritma menggunakan jarak minimum untuk mengukur jarak antar *cluster*, algoritma tersebut akan disebut sebagai *nearest-neighbor clustering algorithm*. Ketika sebuah algoritma menggunakan jarak maksimum untuk mengukur jarak antar *cluster*, algoritma tersebut akan disebut sebagai *farthest-neighbor clustering algorithm*. Pengukuran minimum dan maksimum sebelumnya mewakili dua ekstrim dalam pengukuran jarak antar *cluster* [12].

2.2.6.1 Euclidean Distance

Euclidean Distance adalah perhitungan jarak dari dua buah titik dalam *euclidean space* untuk mempelajari hubungan antara sudut dan jarak [23]. *Euclidean space* diperkenalkan oleh Euclid, salah seorang matematikawan Yunani sekitar tahun 300 B.C.E.

Berikut adalah rumus perhitungan *Euclidean Distance* :

$$dist(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Euclidean Distance memiliki kelebihan dan kekurangannya tersendiri. Kelebihan dari perhitungan *Euclidean Distance* adalah [24] :

1. Perhitungan jarak menggunakan *Euclidean Distance* lebih umum digunakan pada metode K-Means.
2. *Euclidean Distance* mempunyai hasil yang lebih optimal dibandingkan dengan perhitungan lain.

3. Referensi dari *Euclidean Distance* lebih banyak karena secara umum perhitungan *Euclidean Distance* digunakan dalam perhitungan jarak pada metode K-Means.

Selain memiliki kelebihan, *Euclidean Distance* juga memiliki kekurangan. Kekurangan pada perhitungan *Euclidean Distance* adalah [24] :

1. Perhitungan manual menggunakan *Euclidean Distance* cenderung lebih lama karena jumlah iterasi *Euclidean Distance* lebih banyak dibandingkan dengan *Manhattan Distance*.

2.2.7 *Davies-Bouldin Index*

Diperlukan sebuah alat ukur untuk menilai keakuratan dari beberapa *cluster* yang telah terbentuk dari hasil proses *clustering*. Salah satu alat ukur atau metode yang dapat digunakan untuk mengevaluasi *cluster* yang terbentuk adalah dengan menggunakan *Davies-Bouldin Index (DBI)*. *Davies-Bouldin Index* merupakan metode yang diperkenalkan oleh David L. Davis dan Donald W. Bouldin sebagai alat ukur untuk mengevaluasi *cluster*. *Davies-Bouldin Index* akan mengevaluasi *cluster* berdasarkan kuantitas dan kedekatan antar anggota *cluster*. Perhitungan nilai dari metode *Davies-Bouldin Index* yakni berdasarkan perbandingan rasio *cluster* ke-i dan *cluster* ke-j. Semakin kecil nilai yang didapatkan dari perhitungan *Davies-Bouldin Index*, maka akan semakin baik *cluster* yang dihasilkan [29].

Dalam *DBI*, *Sum of Square within Cluster (SSW)* merupakan persamaan yang digunakan untuk mencari nilai dari matriks kohesi di dalam sebuah *cluster* ke-i. Nilai *SSW* dapat dihitung dengan rumus [30] :

$$SSW_i = \frac{1}{m_i} \sum_{j=i}^{m_i} d(X_j, C_i)$$

Selain itu, ada persamaan *Sum of Square Between Cluster (SSB)* yang digunakan untuk mengetahui nilai separasi antar *cluster*. Nilai *SSB* dapat dihitung dengan rumus [30] :

$$SSB_{i,j} = d(C_i, C_j)$$

Setelah memperoleh nilai kohesi dan separasi menggunakan persamaan sebelumnya, selanjutnya dilakukan pengukuran rasio untuk mengetahui nilai perbandingan antara *cluster* ke-*i* dan *cluster* ke-*j*. *Cluster* akan dikatakan baik jika memiliki nilai kohesi sekecil mungkin dan nilai separasi yang sebesar mungkin. Nilai rasio dihitung dengan rumus [30] :

$$R_{i,j} = \frac{SSW_i + SSW_j}{SSB_{i,j}}$$

Nilai rasio yang telah diperoleh akan digunakan untuk mencari nilai *Davies-Bouldin Index*. Nilai *Davies-Bouldin Index* dapat dihitung dengan rumus [30] :

$$DBI = \frac{1}{k} \sum_{i=1}^k \max_{i \neq j} (R_{i,j})$$

Nilai yang didapat dari rumus tersebut merupakan hasil yang menjadi ukuran validitas dari *cluster* yang diuji. Semakin kecil nilai yang didapatkan (non-negatif ≥ 0), maka semakin baik *cluster* yang diperoleh [30].

2.2.8 Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan standar untuk pengembangan perangkat lunak dan sistem [31]. *UML* menyediakan notasi dan metode yang konsisten untuk menggambarkan struktur, fungsi, dan interaksi antar komponen dalam suatu sistem. Dengan *UML*, tim pengembang dapat dengan mudah berkomunikasi dan memahami sistem yang sedang dikembangkan melalui model visual. *UML* mendukung berbagai jenis diagram, seperti *class diagram*, *sequence diagram*, dan *activity diagram* yang memungkinkan para pengembang untuk merancang, menganalisis, dan mendeskripsikan sistem secara efektif. Dengan mempelajari dan menerapkan *UML*, pengembang dapat meningkatkan produktivitas dan kualitas dalam proses pengembangan perangkat lunak.

2.2.8.1 Use Case

Use case adalah suatu kasus atau situasi untuk memenuhi satu atau lebih dari kebutuhan pengguna pada suatu sistem, sebuah *use case* menangkap suatu fungsi

yang disediakan oleh sistem [31]. *Use case* adalah cara untuk menggambarkan secara jelas kebutuhan sistem dari sudut pandang pengguna. *Use case* menjelaskan nilai yang sistem berikan kepada pengguna dengan jelas dan terperinci.

Use case diagram dan *use case description* memiliki peran penting dalam pemodelan sistem menggunakan UML. *Use case diagram* memberikan gambaran visual tentang bagaimana aktor (entitas eksternal) berinteraksi dengan *use case* (fungsionalitas sistem) dalam sistem tersebut. Diagram ini membantu kita untuk memahami hubungan antara aktor dan fungsionalitas yang diperlukan oleh sistem. Di sisi lain, *use case description* memberikan penjelasan lebih rinci mengenai fungsionalitas yang ada. *Use case description* akan menjelaskan bagaimana pengguna atau aktor berinteraksi dengan sistem dan apa yang diharapkan dari sistem saat menjalankan setiap *use case* [31].

2.2.8.2 Activity Diagram

Activity diagram digunakan untuk menggambarkan tindakan-tindakan tingkat tinggi yang saling terhubung untuk mewakili proses yang terjadi dalam sistem [31]. Diagram ini berguna untuk memodelkan proses bisnis dan sering digunakan dalam manajemen proses bisnis (BPM). *Activity diagram* menggunakan simbol-simbol yang mirip dengan notasi flowchart yang sudah dikenal luas, sehingga mudah dipahami oleh berbagai kalangan. Diagram ini juga memiliki hubungan dengan flowchart, state diagram UML, data flow diagram, dan Petri Nets.

2.2.8.3 Class Diagram

Class diagram adalah sebuah diagram yang digunakan untuk menggambarkan struktur sistem berorientasi objek. Diagram ini menampilkan kelas-kelas yang ada dalam sistem beserta hubungan-hubungannya [31]. *Class diagram* membantu dalam memahami jenis-jenis objek yang ada dalam sistem dan bagaimana objek-objek tersebut saling berhubungan. *Class diagram* juga membantu dalam memodelkan kebutuhan sistem berdasarkan *use case* yang telah ditentukan.

2.2.8.4 *Sequence Diagram*

Sequence diagram adalah jenis diagram yang digunakan untuk menggambarkan urutan interaksi antara bagian-bagian sistem [31]. *Sequence diagram* dapat menggambarkan interaksi apa yang akan dipicu saat sebuah *use case* tertentu dieksekusi dan dalam urutan interaksi apa peristiwa-peristiwa tersebut akan terjadi. *Sequence diagram* menunjukkan berbagai informasi lainnya tentang sebuah interaksi, tetapi keunggulan utamanya adalah cara sederhana dan efektif dalam mengkomunikasikan urutan peristiwa dalam sebuah interaksi.

2.2.9 Python

Python merupakan bahasa pemrograman interpretatif, interaktif dan berbasis *object oriented (OO)* yang diciptakan oleh Guido van Rossum pada tahun 1991. Bahasa ini meliputi modul, *exceptions*, *dynamic typing*, *very high level dynamic data types*, dan *classes*. Python memiliki *syntax* yang sangat bersih dan memiliki antarmuka ke banyak *system calls* dan *libraries*. Selain itu, python juga dapat digunakan sebagai *extension language* untuk perangkat lunak yang membutuhkan antarmuka yang dapat diprogram [32].

Python muncul selama beberapa dekade terakhir untuk membantu tugas komputasi ilmiah, termasuk untuk menganalisis dan memvisualisasikan dataset dengan jumlah yang sangat besar. Kegunaan bahasa pemrograman python untuk kebutuhan *data science* didukung dengan ekosistem *third-party packages* atau *libraries* yang berjumlah besar [33]. Selain memudahkan untuk kebutuhan *data science*, bahasa pemrograman python juga dapat mempermudah pekerjaan lainnya karena bahasa ini bersifat multifungsi. Tidak hanya digunakan untuk *data processing*, namun bahasa ini juga bisa digunakan untuk tugas lain seperti membuat *website* dan tampilan *GUI (Graphical User Interface)*.

2.2.10 Website

Website merupakan halaman informasi yang disediakan di internet dan bisa diakses oleh seluruh orang di seluruh dunia. Teknologi *website* mengolah data menjadi sebuah informasi dengan cara mengidentifikasi, mengumpulkan, mengelola dan menyediakan informasi untuk dapat diakses secara bersama-sama

[34]. Penggunaan *website* untuk pembangunan suatu sistem lebih efisien bagi pengguna daripada aplikasi *mobile* ataupun *desktop*. Karena pada dasarnya, *website* dapat diakses melalui *browser* tanpa harus melakukan instalasi tambahan.

Website merupakan fasilitas internet yang menghubungkan banyak dokumen dalam lingkup lokal maupun jarak yang sangat jauh. Dokumen pada *website* ditampilkan dalam bentuk halaman-halaman yang disebut dengan *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu halaman ke halaman yang lain (*hyper text*), baik diantara halaman yang disimpan dalam *server* yang sama maupun *server* lain yang tersebar di seluruh dunia. Halaman diakses dan dibaca melalui *browser* seperti Netscape Navigator atau Internet Explorer atau berbagai aplikasi *browser* lainnya [35].

2.2.11 Black Box Testing

Black Box testing adalah metode pengujian aplikasi yang mencakup sisi luar perangkat lunak, dimulai dari pengujian tampilan hingga aksi masukan atau input action [36]. Tujuan utama pada black box testing adalah untuk benar-benar tidak memperhatikan perilaku dan struktur internal dari program. Sebaliknya, fokuslah untuk menemukan situasi di mana program tidak berperilaku sesuai dengan spesifikasinya [37]. Akan tetapi, jenis verifikasi perangkat lunak pada black box testing ini tidak hanya digunakan ketika tidak ada akses ke arsitektur dan kode program, tetapi juga bisa digunakan ketika struktur komponen internal atau *source code* sistem diketahui [38].

2.2.12 User Acceptance Test

User Acceptance Test merupakan salah satu teknik yang dapat dilakukan untuk mengukur seberapa mudah aplikasi dapat digunakan oleh pengguna atau *user* [39]. *User Acceptance Test* juga digunakan untuk mengidentifikasi cacat atau kesalahan dalam aplikasi atau perangkat lunak yang sedang dikembangkan [40]. *User Acceptance Test* memiliki tujuan untuk memastikan bahwa perangkat lunak yang sedang dikembangkan telah memenuhi persyaratan dan kebutuhan pengguna atau tujuan bisnis dengan baik. Kriteria-kriteria pengujian yang digunakan berasal

dari pengguna atau *user*. Jika perangkat lunak telah memenuhi semua kriteria yang ditetapkan oleh penguji, maka *User Acceptance Test* dianggap telah berhasil.