BAB II

LANDASAN TEORI

2.1 *Clustering*

Clastering merupakan upaya untuk mengelompokkan *record,* observasi, atau mengelompokkan ke dalam kelas yang memiliki kesamaan objek (Kusuma V. M., 2017). Pengklasteran berbeda dengan klasifikasi yang tidak adanya variable target dalam pengklasteran. Pengklasteran tidak digunakan untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari target. Pengklasteran digunakan untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok – kelompok yang memiliki kemiripan .

2.2 Algoritma K-Means

Metode K-means merupakan algoritma klasterisasi yang paling tua dan banyak digunakan dalam berbagai aplikasi kecil hingga menengah karena kumudahan implementasinya (Suyatno, 2017). Ide dasar algoritma K-Means sangatlah sederhana, yaitu meminimalkan *Sym of Squared Error* (SSE) antara objek – objek data dengan empat langkah. Empat langkah tersebut antara lain :

1. Dari himpunan data yang akan diklasterisasi, dipilih sejumlah *k* objek secara acak sebagai centroid awal.
2. Setiap objek yang bukan centroid dimasukkan ke klaster terdekat berdasarkan ukuran jarak tertentu.

3. Setiap centroid diperbaharui berdasarkan rata – rata dari objek yang ada di dalam setiap kluster.
4. Langkah kedua dan ketiga tersebut diulang – ulang (diiterasi) sampai semua centroid stabil atau konvergen, dalam arti semua *centroid* yang dihasilkan pada itersi sebelumnya.

K-means juga memiliki kelemahan dan kelebihan dalam penggunaannya (Sari & D, 2014), seperti berikut :

- Kelebihan K-means

1. Selalu konvergen atau mampu melakukan klasterisasi.
2. Tidak membutuhkan operasi matematis yang rumit / sederhana
3. Beban komputasi relatif lebih ringan, sehingga klasterisasi bisa dilakukan dengan cepat walaupun relatif tergantung pada banyak jumlah data dan jumlah *cluster* yang ingin dicapai.

- Kekurangan K-means

1. K-means hanya dapat digunakan untuk data yang atributnya bernilai numerik. Jumlah *cluster* sebanyak k, harus ditentukan sebelum dilakukan perhitungan.
2. Nilai centroid yang diberikan diawal dapat mempengaruhi hasil klasterisasi apabila nilainya berbeda.
3. Solusi *cluster* yang dihasilkan hanya bersifat *local optima*, sehingga tidak diketahui apakah itu sudah merupakan konfigurasi optima atau belum.
4. Tergantung pada mean (rata – rata)

2.3 *Hamming Distance*

Hamming Distance digunakan untuk menghitung jumlah perbedaan dari dua deret bilangan biner yang mempunyai panjang sama sesuai dengan posisi dari setiap digit biner. *Hamming Distance* memiliki rumus sebagai berikut :

$$d_{ij} = q + r$$

Dengan q merupakan jumlah variabel dengan nilai 1 pada objek ke-i tapi bernilai 0 pada objek ke-j. Sedangkan r merupakan jumlah variabel yang dengan nilai 0 pada objek le-i tapi bernilai 1 pada objek ke-j. Cara kerja algoritma *Hamming Distance* yaitu dengan mengukur jarak antara dua string yang ukurannya sama dengan membandingkan karakter yang terdapat pada kedua *string* pada posisi yang sama (Rochmawati & Kusumaningrum, 2016)

Contoh :

Buah A (bentuk bulat, manis, berbiji, berair) dan buah B (bentuk tidak bulat, manis, tidak berbiji, dan tidak berair), maka dapat dipresentasikan sebagai deret biner sebagai berikut : A (1,1,1,1) dan B (0,1,0,0). *Hamming distance* antara A dan B dapat dicari dengan cara :

q= 3 (bernilai satu di A tapi bernilai 0 di B)

r = 0 (bernilai satu di B tapi bernilai 0 di A)

$$d_{BA} = \frac{(1111)\ XOR\ (0100)}{4} = \frac{3}{4} = 0.75$$

Selain *Hamming distance* ada banyak cara yang digunakan untuk menghitung jarak antar suatu objek dengan objek lainnya, antara lain adalah *Euclidean Distance, City Block (Manhattan) Distance, Chebyhev Distance, Minkowski Distance,* dan *Canberra Distance.*

2.4 PHP

PHP merupakan singkatan dari *Hypertext Prepocessor* yaitu bahas pemrograman *web server-side* yang bersifat *open source.* PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server server side HTML embedded scripting)*.. PHP juga memungkinkan kita untuk membuat sebuah halaman *website* secara dinamis, hal ini tidak mungkin dilakukan hanya dengan menggunakan HTML (Hitayatullah & Kawistara, 2014). Ada beberapa kelebihan dari PHP seperti berikut :

1. Dapat mengerjakan semua yang dapat dilakukan oleh program CGI (*Common Gateway Interface)* seperti menghasilkan isi halaman *web* yang dinamik dan menerima *cookies.*
2. Dukungan kepada banyak *database*, membuat halaman *web* yang menggunakan *database* sangan mudah dilakukan.
3. Dapat menggunakan berbagai database seperti MySQL, Microsoft Access, InterBase, mSQL, SyBbase, Dbase, Informix, SQL Server , dan lain-lain.
4. Dapat di unduh gratis.
5. Fitur PHP dapat dimodifikasi sesuai dengan kebutuhan.

Beberapa kelebihan PHP dari bahasa pemrograman *web*, antara lain :

a. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
b. *Web Server* yang mendukung PHP dapat ditemukan di mana-mana, mulai dari Apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
c. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
d. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
e. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.5 MySQL

MySQL adalah salah satu aplikasi DBMS yang sudah banyak digunakan oleh para pemrogram aplikasi *website. Database Management System* (DBMS) adalah aplikasi yang digunakan untuk mengolah basis data (Hitayatullah & Kawistara, 2014). MySQL memiliki beberapa kelebihan dibandingkan *database* lain, antara lain :

1. MySQL sebagai *Database Management System* (DBMS)
2. Sebagai *Relation Database Managament System* (RDMS)
3. Merupakan *database* yang *open source*

4. Dapat dijadikan sebagai *Database Server*
5. Dapat dijadiakan sebagai *Database Client*
6. Mampu menerima *query* bertumpuk dalam satu permintaan atau yang disebut *Multi-Threding*
7. Dapat menyimpan data dalam ukuran besar hingga berukuran *Gigabyte.*

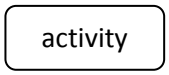
2.6 Unified Modelling Language (UML)

Unified Modelling Language (UML) merupakan sebuah bahasa yang telah menjadi standar dalam industri visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Sugiarti, 2013) UML mendefinisikan beberapa diagram sebagai berikut :

1. *Activity diagram*, atau diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem proses bisnis yang ada dalam perangkat lunak. Dalam *activity diagram* memperlihatkan aliran kerja atau proses sederhana dari sistem yang bekerja. Notasi-notasi yang digunakan dalam *activity diagram* dapat dilihat pada tabel 2.1

Tabel 2.1 Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	*Initial node*	*Initial node* digunakan untuk memulai suatu aktivitas.

Simbol	Nama	Keterangan
	*Activity*	*Activity* merupakan notasi yang menggambarkan pelaksanaan dari beberapa proses dalam aliran pekerjaan.
	*Transition*	*Transition* merupakan notasi yang digunakan untuk memperlihatkan aliran kontrol dari aktivitas yang satu
		ke aktivitas yang lain.
	*Final-activity node*	*Final-activity node* digunakan untuk mengakhiri suatu aktivitas.

2. *Use case diagram*, merupakan pemodelan untuk menggambarkan kelakuan *(behavior)* sistem yang akan dibuat. Diagram ini mendiskripsikan sebuah interaksi antara satu atau lebih aktor dalam sistem yang dibuat. Diagram ini menunjukkan fungsi-fungsi dalam sistem dan menunjukkan siapa saja yang berhak melakukan fungsi – fungsi tersebut. Notasi-notasi yang digunakan dalam *use case diagram* dapat dilihat pada tabel 2.2.

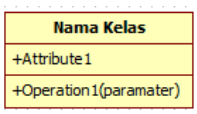
Tabel 2.2 Simbol *Use Case Diagram*

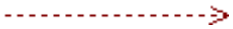
Simbol	Nama	Keterangan
	*Actor*	*Actor* adalah pengguna sistem. *Actor* tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan *input* atau memberikan *output*, maka aplikasi tersebut juga bisa dianggap sebagai *actor*.
	*Use Case*	*Use case* digambarkan sebagai lingkaran *elip*s dengan nama *use case* dituliskan didalam *elips* tersebut.
	Association	*Association* digunakan untuk menghubungkan *actor* dengan *use case*. *Association* digambarkan dengan sebuah garis yang menghubungkan antara *Actor* dengan *Use Case*.
	*Extend*	*Extend* digunakan untuk menghubungkan *use case* yang satu dengan yang lainnya jika *use case*

Simbol	Nama	Keterangan
		tersebut merupakan bagian dari *use case* yang lain.
<<include>>	*Include*	*Include* digunakan untuk menghubungkan *use case* yang satu dengan *use case* yang lain, dimana *use case* tersebut harus dilakukan sebelum *use case* yang lain dilakukan

3. *Class diagram*, menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variable – variable yang dimiliki oleh suatu kelas. Atribut mendiskripsikan property dengan sebaris teks didalam kotak kelas tersebut. Operasi atau metode adlah fungsi – fungsi yang dimiliki oleh suatu kelas. Kelas diagram memiliki tiga area pokok yaitu nama, atribut dan operasi.

Tabel 2.3 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur system

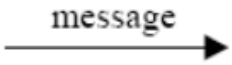
Simbol	Deskripsi
Paket/*package* 	Paket/*package* merupakan sebuah bungkusan dari satu atau lebih kelas (kumpulan kelas)
Assosiasi	Asosiasi merupakan hubungan antar kelas dengan makna umum, asosiasi biasanya jua disertai dengan *multiplicity*
Generalisasi 	Generalisasi merupakan hubungan generalsasi dan spesialisasi (umum khusus) antara dua kelas dimana fungsi yang satu adalah fungsi yang lebih umum dari fungsi yang lainnya
Dependency 	Dependency merupakan hubungan antarkelas yang saling bergantung, membutuhkan satu sama lain.
Agregasi 	Agregasi merupakan hubungan antar kelas dimana satu kelas merupakan semua bagian dari kelas-kelas yang lain.

4. *Sequence diagram*, menggambarkan kelakuan objek pada *Use* Case dengan mendiskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk

menggambarkan diagram ini harus mengetahui objek – objek yang terlibat dalam sebuah *Use Case* beserta metode –metode yang dimiliki kelas yang diintansiasi menjadi onjek itu. Notasi-notasi yang digunakan dalam *sequence diagram* dapat dilihat pada tabel 2.4

Tabel 2.4 Simbol *Sequence Diagram*

Simbol	**Deskripsi**	**Simbol**
: Object1	*Object*	*Object* merupakan *instance* dari sebuah
		class dan dituliskan tersusun secara *horizontal*. Digambarkan sebagai sebuah *class* (kotak) dengan nama *objek* didalamnya yang diawali dengan sebuah titik koma
	*Actor*	*Actor* juga dapat berkomunikasi dengan *object*, maka actor juga dapat diurutkan sebagai kolom. Simbol *Actor* sama dengan simbol pada *Actor Use Case Diagram.*
	Lifeline	*Lifeline* mengindikasikan keberadaan sebuah *object* dalam basis waktu. Notasi untuk *Lifeline* adalah garis putus-putus *vertical* yang ditarik dari sebuah *objek.*

Simbol	Deskripsi	Simbol
	*Activation*	*Activation* dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah *lifeline*. *Activation* mengindikasikan sebuah *objek* yang akan melakukan sebuah aksi.
	*Message*	*Message*, digambarkan dengan anak panah *horizontal* antara *Activation*.
		Message mengindikasikan komunikasi antara *object-objec*t.

2.7 Entity Relationship Diagram (ERD)

ERD adalah pemodelan basis data yang paling banyak digunakan. ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data rasional, sehingga jika penyimpanan basis data menggunakan OODBMS maka peranccangan basis data tidak perlu menggunakan ERD (A.S & Shalahuddin, 2016).

Sesuai dengan namanya ada 2 komponen utama pembentuk model keterhubungan entitas yaitu *entity* (*entitas*) dan relasi (*relation*). Entitas menyatakan suatu object yang memprasentasikan suatu himpunan atau sesuatu didunia nyata yang mempunyai peranan dalam sistem yang sedang dibangun, sedangkan relasi merupakan sebuah kumpulan dari beberapa entitas atau relasi yang memiliki tipe sama. Pada model *entity relationship* diagram hubungan antar *file* direlasikan dengan kunci relasi (*relation key*),

yang merupakan kunci utama dari masing-masing file. (Fathansyah, 2004). Untuk membantu gambaran relasi secara lengkap terdapat juga tiga macam relasi dalam hubungan atribut dalam satu *file*, yaitu :

a. *One to one relationship*

Hubungan antara *file* pertama dan *file* kedua adalah satu berbanding satu. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan table dan relasi antar keduanya digambarkan dengan panah tunggal.

b. *One to Many relationship*

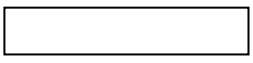
Hubungan antara *file* pertama dan *file* kedua adalah satu berbanding banyak atau dapat pula dibalik banyak berbanding satu. Hubungan tersebut dapat digambarkan dengan panah banyak untuk menunjukan hubungan banyak tersebut.

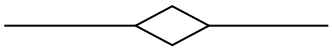
c. *Many to many relationship*

Hubungan antara *file* pertama dan *file* kedua adalah banyak berbanding banyak. Hubungan tersebut dapat digambarkan dengan panah ganda untuk menunjukkan hubungan banyak tersebut.

Berikut simbol-simbol yang digunakan dalam ERD:

Tabel 2.5 Simbol ERD

Gambar	Keterangan
	Entitas
	Hubungan
	Penghubung (*link*)

Gambar	Keterangan
	*One to One*
	*One to Many*
	*Many to Many*

2.8 Notepad++

Notepad++ adalah sebuah *text editor* yang sangat berguna bagi setiap orang dan khususnya bagi para *developer* dalam membuat program. Notepad++ menggunakan komponen *Scintilla* untuk dapat menampilkan dan menyuntingan teks dan berkas kode sumber berbagai bahasa pemrograman yang berjalan diatas sistem operasi Microsoft Windows.

Selain manfaat dan kemampuannya menangani banyak bahasa pemrograman, Notepad ++ juga dilisensikan sebagai perangkat *free*. Jadi, setiap orang yang menggunakannya tidak perlu mengeluarkan biaya untuk membeli aplikasi ini karena *sourceforge.net* sebagai layanan yang memfasilitasi Notepad ++ membebaskannya untuk digunakan.

Beberbapa daftar bahasa program yang didukung oleh Notepad++ adalah C, C++, Java, C#, XML, HTML, PHP, Javascript, Perl Pascal, dan lain-lain yang dapat bekerja pada sistem operasi windows.

2.9 Penelitian Terkait

1. **Analisis *Clustering* Menggunakan Metode K-Means Dalam Pengelompokkan Penjualan Produk Pada Swalayan Fadhila (Metisen & Sari, 2015)**

   Penelitian tersebut menggunakan metode algoritma K-means dengan pemrosesan menggunakan software yang dibuat sehingga dapat dengan mudah menentukan dan mengkasifikasikan produk yang laku dan kurang laku. Penelitian tersebut menghasilkan data antara penghitungan manual eqiuvalen dengan data nonmanual.

2. **Studi Perbandingan Algoritma Pencarian String dalam Metode *Approximate String Matching* untuk Identifikasi Kesalahan Pengetikan Teks (Rochmawati & Kusumaningrum, 2016)**

   Pada penelitian ini membahas mengenai proses identifikasi pengetikan teks dimana kata baku berubah menjadi kata tidak baku karena ejaan yang digunakan tidak sesuai. Pengidentifikasian yang dilakukan menggunakan beberapa metode pencarian string yaitu *Levenshtein Distance, Hamming Distance, Damerau Levenshtein Distance* dan *Jaro Winkler Distance*. Keempat metode tersebut dibandingkan hasilnya untuk melihan metode pencarian mana yang terbaik.

3. **Pemanfaatan Metode K-Means *Clustering* Dalam Penentuan Penerima Beasiswa (Hastuti, 2013)**

Dalam penelitian ini menggunakan beberapa kriteria yang digunakan dalam proses *Clustering* yaitu, Indeks Prestasi Kumulatif (IPK), jumlah tanggungan keluarga, dan penghasilkan total orang tua. Kretiria tersebut digunakan untuk proses *Clustering* yang digunakan untuk mengelompokkan mahasiswa yang direkomendasikan menerima beasiswa, dipertimbangkan menerima beasiswa, dan tidak menerima beasiswa.