

BAB III

LANDASAN TEORI

3.1 Sistem

Pengertian sistem dapat terbagi menjadi dua sesuai dengan pendekatannya dalam mendefinisikan sistem, yaitu pendekatan pada prosedur dan pada komponen atau elemen. Menurut pendekatan sistem pada prosedur, sistem didefinisikan merupakan suatu jaring kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau tujuan tertentu. Sedangkan untuk definisi sistem menurut pendekatan pada komponen/elemennya sistem adalah bagian-bagian elemen yang saling berinteraksi dan saling berhubungan untuk mencapai membentuk satu kesatuan (Astuti, 2013).

Sistem memiliki beberapa fungsi, namun fungsi sistem yang utama adalah menerima masukan, mengolah masukan, dan menghasilkan keluaran. Agar dapat menjalankan fungsi ini, sistem akan memiliki komponen-komponen input, proses, keluaran dan kontrol untuk menjamin bahwa semua fungsi dapat berjalan dengan baik.

a. Karakteristik Sistem

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen sistem terdiri dari sumber daya manusia, sumber daya perangkat keras, sumber daya perangkat lunak dan data.

2. Batasan Sistem

Batasan merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem, dipandang suatu kesatuan.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang lainnya atau dengan lingkungan luarnya. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu sistem ke sistem lainnya.

5. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal.

6. Keluaran Sistem

Keluaran sistem adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

b. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah akan mengubah masukan menjadi keluaran.

c. Sasaran Sistem

Suatu sistem mempunyai tujuan atau sasaran. Apabila suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai tujuannya. (Sulindawati & Fathoni, 2010)

3.2 Preprocessing Data

Preprocessing Data adalah salah satu faktor penting yang mempengaruhi keberhasilan pendekatan *Data Mining* dan *Machine Learning*. Umumnya, tahap *Data preprocessing* merupakan kualitas data. Data yang benar tetapi tidak lengkap menyajikannya, maka hasilnya semakin tidak dapat diandalkan. Selain itu, data *preprocessing* mencakup banyak tahap seperti pembersihan data, mengintegrasikan data, data transformasi. Namun, kinerja terbaik dari data algoritma pra-pengolahan didasarkan pada sifat masing-masing set data. Oleh karena itu, akan lebih baik jika memiliki metodologi yang baik untuk beradaptasi dan untuk memilih kinerja terbaik dari algoritma data *preprocessing* untuk setiap set data (Kularbphetong et al., 2010).

Preprocessing data diperlukan karena data *real* yang akan diolah adalah data kotor, tidak lengkap, dan mengandung *noise*. Kurang lengkapnya nilai pada atribut dan atribut hanya berisi agregat, data mengandung kesalahan atau *outlier* (keluar konteks) dan tidak konsisten dalam penamaan atau kode. Data kotor dikarenakan data yang didapat (data sumber) tidak

lengkap, tidak ada nilai yang tidak sesuai (valid) ketika data dikumpulkan dan adanya perbedaan pertimbangan (saran) pada saat analisis data. Data yang mengandung noise bisa berasal dari kesalahan manusia ketika memasukan data, sehingga data yang ada tidak sempurna. Kesalahan dalam transformasi karena adanya kerusakan data pada saat pengumpulan data. Data yang tidak konsisten, karena data dikumpulkan berasal dari data berbagai sumber data. Data yang redundan merupakan data duplikasi sehingga memerlukan pembersihan data.

Data *preprocessing* juga bertanggung jawab untuk memindahkan, mengompresi, dan normalisasi data mentah ke dalam format modul untuk meningkatkan kinerja selanjutnya (Osuna & Nagle, 1999). Dalam data *preprocessing* diperlukan komprehensif pemahaman untuk menentukan jenis hasil dari proses *preprocessing* tersebut (Mukhlash & Sitohang, 2007).

Khusus untuk data yang tidak konsisten dapat dilakukan dengan beberapa cara, yaitu :

1. Manual yaitu dengan mencari dan memperbaiki data yang tidak konsisten satu persatu.
2. Semi manual, yaitu dengan menggunakan bantuan sistem untuk mendeteksi data yang tidak konsisten, lalu memperbaikinya secara manual.
3. Otomatis, yaitu dengan menggunakan bantuan sistem sepenuhnya dalam mencari dan memperbaiki data yang tidak konsisten dengan menggunakan proses integrasi data (Han & Kamber, 2006)

3.3 Pengolahan Data

Pengolahan data (*Data Processing*) adalah suatu kegiatan manipulasi atau merubah simbol seperti huruf dan angka dengan tujuan meningkatkan nilai gunanya, sedangkan pemrosesan transaksi (*transaction processing*) mendeskripsikan pemrosesan data yang dilakukan terhadap data-data bisnis.

Pengolahan data mempunyai empat tugas utama :

a. Pengumpulan data

Saat perusahaan menyediakan produk dan jasa ke lingkungan, setiap kegiatan dijelaskan oleh sebuah record data. Sistem pengolahan data mengumpulkan data yang menjelaskan setiap tindakan internal perusahaan dan transaksi lingkungan perusahaan.

b. Manipulasi data

Operasi manipulasi data yang mengubah data menjadi informasi, meliputi :

1. Pengklasifikasian (*Classifying*), mengklasifikasikan data sesuai dengan kegunaan atau darimana data berasal
2. Penyortiran (*Sorting*), record data diatur dengan cara tertentu berdasarkan nilai kunci yang ditetapkan.
3. Penghitungan (*Calculating*), mengaplikasikan operasi logika dan matematika terhadap data untuk menghasilkan elemen tambahan dari data.
4. Perangkuman (*Summarizing*), melakukan penjumlahan total maupun subtotal dari suatu data item.

c. Penyimpanan data

Data disimpan pada media penyimpanan sekunder, dan dapat diambil sewaktu-waktu dibutuhkan. Data saling

terintegrasi secara logis hingga membentuk database (Ayuliana, 2011).

3.4 Levenshtein Distance (Edit Distance)

Levenshtein Distance dibuat oleh Vladimir Levenshtein pada tahun 1965. Perhitungan edit distance didapatkan dari matriks yang digunakan untuk menghitung jumlah perbedaan string antara dua string. Perhitungan jarak antara dua string ini ditentukan dari jumlah minimum operasi perubahan untuk membuat string A menjadi string B (Implementasi Algoritma Levenshtein Distance dan Metode Empiris untuk menampilkan saran perbaikan kesalahan pengetikan dokumen berbahasa Indonesia (Adriyani, et al., 2012).

Ada 3 macam operasi utama yang dapat dilakukan oleh algoritma ini yaitu :

1. Operasi Pengubahan Karakter

Operasi pengubahan karakter merupakan operasi menukar sebuah karakter dengan karakter lain contohnya penulis menuliskan string "yang" menjadi "yang". Dalam kasus ini karakter "m" diganti dengan huruf "n".

2. Operasi Penambahan Karakter

Operasi penambahan karakter berarti menambahkan karakter ke dalam suatu string. Contohnya string "kepad" menjadi string "kepada", dilakukan penambahan karakter "a" di akhir string. Penambahan karakter tidak hanya dilakukan di akhir kata, namun bias ditambahkan diawal maupun disisipkan di tengah string.

3. Operasi Penghapusan Karakter

Operasi penghapusan karakter dilakukan untuk menghilangkan karakter dari suatu string. Contohnya string "barur" karakter terakhir dihilangkan sehingga menjadi string "baru". Pada operasi ini dilakukan penghapusan karakter "r" (Adriyani, et al., 2012).

Algoritma ini berjalan mulai dari pojok kiri atas sebuah array dua dimensi yang telah diisi sejumlah karakter string awal dan string target dan diberikan nilai *cost*. Nilai *cost* pada ujung kanan bawah menjadi nilai *edit distance* yang menggambarkan jumlah perbedaan dua string. Misalnya terdapat dua string yaitu string asli dan string pembanding String asli (CS) : **aku** String pembanding (ST): **abu** Proses algoritma levenstein distance adalah sebagai berikut:

1. Jika panjang CS adalah 0 maka jarak CS ke ST adalah panjang ST dalam contoh diatas berarti bernilai 3.
2. jika panjang ST adalah 0 maka jarak CS ke ST adalah panjang CS dalam contoh diatas berarti bernilai 3.
3. Membuat matrik dengan ukuran $(CS+1) \times (ST+1)$

	a	b	u
a	1		
k	2		
u	3		

4. Melakukan pencocokan dengan melakukan pembandingan dari setiap karakter CS dengan karakter ST. Iterasi (1,1)

		a	b	u
	0	1	2	3
a	1	0		
k	2			
u	3			

5. Jika karakter 1 CS dengan karakter 1 ST sama maka nilai pada cell (1,1) diisi dengan 0 yang merupakan nilai matrik (x- 1 , y-1)

		a	b	u
0	1	2	3	
a	1	0		
k	2	1		
u	3			

Iterasi (1,2)

Jika karakter 1 CS dengan karakter 2 ST berbeda maka nilai cell didapatkan dari nilai terkecil dari:

- Nilai cell (x-1, y-1) + 1, pada contoh diatas nilainya adalah 2.
- Nilai cell (x-1, y) + 1, pada contoh diatas nilainya adalah 3.
- Nilai cell (x, y-1) + 1, pada contoh diatas nilainya adalah 1.

Maka jarak antara karakter 1 CS dengan karakter 2 ST bernilai 1. Setelah semua proses literasi selesai maka didapatkan jarak string asli (CS) dengan string pembanding (ST) pada cell (3,3) adalah sebesar 1 atau $Diff = 1$ (Junedy.S, 2014).

		a	b	u
0	1	2	3	
a	1	0	1	2
k	2	1	1	2
u	3	2	2	1

3.5 Basis Data

Basis data adalah dua atau lebih simpanan data dengan elemen-elemen data penghubung, yang dapat diakses lebih dari satu cara. Basis data dinyatakan dengan tehnik-tehnik formal dan manajemen basis data. Dari definisi diatas, maka dapat dikatakan bahwa basis

data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya (Iskandar & Rangkuti, 2008).

Basis data merepresentasi kumpulan fakta yang saling berhubungan dan disimpan secara bersama, untuk memenuhi berbagai kebutuhan. Sebuah database memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur database: ini dikenal sebagai database model atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah yaitu mewakili semua informasi dalam bentuk tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel.

3.6 Microsoft Visual Studio dan Bahasa Pemrograman C#

Microsoft Visual Studio adalah suatu *development tools* untuk membangun aplikasi-aplikasi dalam lingkungan *windows*. Keunggulan dari Visual Studio adalah pada interface desainnya yang menggunakan IDE (*Integrated Development Environment*). Jika diterjemahkan, maka artinya adalah lingkungan pengembangan yang terintegrasi.

IDE *Visual Studio* sudah terintegrasi dengan kompiler, .NET, fasilitas *debug*, fasilitas pendistribusian *file* program yang telah jadi, fasilitas *intelinsense* (*autocomplete modern*), dan lain

sebagainya. Para pengembang sangat dimanjakan oleh fasilitas yang ada pada IDE *Visual Studio*. *Visual Studio Express 2008* memiliki berbagai kelebihan dilihat dari hasil peningkatan performa dari seri *Visual Studio* sebelumnya (Tamsir et al., 2012).

Microsoft visual studio terdiri dari berbagai bahasa pemrograman antara lain C++ dan *C-Sharp* atau yang lebih dikenal dengan C#. Dalam sistem ini bahasa pemrograman yang digunakan adalah C#. Microsoft membuat C# seiring dengan pembuatan *Framework .NET*. *Chief Architect* dalam pembuatan C# adalah Anders Hejlsberg yang sebelumnya berperan dalam pembuatan *Borland Delphi* dan *Turbo Pascal* (Suryanto & Samiyono, 2011).

Bahasa Pemrograman C# merupakan sebuah bahasa pemrograman yang berorientasi objek. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek - aspek atau pun fitur bahasa yang terdapat pada bahasa - bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic dan lain - lain dengan beberapa penyederhanaan (Adelia & Setiawan, 2011). C# mendukung secara penuh konsep pemrograman berorientasi objek seperti *inheritance*, *polymorphism*, *encapsulation* dan *abstraction*. Selain pemrograman berorientasi objek, C# juga mendukung pemrograman berorientasi komponen, yang memungkinkan kita untuk menetapkan unit fungsional (komponen) yang berdiri sendiri dan terdokumentasi dengan menyediakan sebuah model dengan *properties*, *method*, *events* dan *metadata* tentang komponen yang dimaksud (Wicaksono, et al., 2014).

Bahasa Pemrograman C# menjanjikan produktifitas dan kemudahan yang ada di *Visual Basic* dengan kemampuan dan fleksibilitas yang ada di C/C++. Untuk mencapai produktifitas tinggi ini konsep-konsep sulit C++ disederhanakan dan fitur-fitur baru ditambahkan. Hal ini mungkin terasa mirip dengan Java, karena itulah C# bisa dianggap sebagai sepupu Java (Suryanto & Samiyono, 2011). Seperti pemrograman Java, C# juga tidak memperbolehkan *multiple inheritance* atau penggunaan *pointer*, tetapi C# menyediakan *garbage memory collection* pada saat *runtime* dan pada saat pengecekan akses memori. Meskipun bertentangan dengan pemrograman Java, C# tetap mempertahankan operasi unik yang terdapat pada bahasa pemrograman C++ seperti *overloading*, *enumerations*, *pre-processor directive*, *pointer* (pada *unmanaged/unsafe code*), dan fungsi *pointer*. Seperti halnya *Visual Basic*, bahasa pemrograman C# juga dilengkapi dengan *properties* (Mukadar, et al., 2014).

Bahasa pemrograman C# adalah salah satu dari banyak bahasa yang bisa dipakai untuk pemrograman .NET. Kelebihan utama bahasa ini adalah sintaksnya yang mirip C, namun lebih mudah dan lebih bersih (Suryanto & Samiyono, 2011). Karena Bahasa C# merupakan bahasa modifikasi dari bahasa C++ dan bahasa VB, maka bahasa C# mempunyai bahasa yang simpel dan efektif dibanding bahasa C++ dan VB, sehingga lebih mudah dalam menginputkan algoritma pemrograman. Selain itu bahasa C# tersebut dapat di compile menjadi suatu program installer yang dapat digunakan di komputer tanpa menggunakan *Visual Studio* (Jouhari A.M, et al., 2014).

3.7 Microsoft SQL Server

SQL (*Structured Query Language*) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara *de facto* merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua *server* basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya (Adelia & Setiawan, 2011).

SQL terdiri dari dua bahasa, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). Implementasi DDL dan DML berbeda untuk tiap sistem manajemen basis data (SMBD), namun secara umum implementasi setiap bahasa ini memiliki bentuk standar yang ditetapkan oleh ANSI.