

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 *Customer Service***

Menurut Edward (2009), customer service adalah hasil penggabungan dari apa yang organisasi lakukan guna memenuhi kebutuhan pelanggan dan menghasilkan kepuasan pelanggan. Dengan kata lain, customer service merupakan bagian dari perusahaan yang menghubungkan antara perusahaan dan pelanggannya guna memenuhi dan memahami ekspektasi pelanggan.

Dalam kenyataannya, customer service dibagi ke dalam beberapa tipe, antara lain:

- Customer service before sale, merupakan customer service yang menyambut datangnya pelanggan, contohnya adalah pegawai yang menyambut datangnya pelanggan.
- Customer service during sale, merupakan customer service yang melayani pelanggannya ketika hendak melakukan pembelian, contohnya adalah pegawai yang membantu pelanggan dalam menentukan produk yang akan dibeli.
- Customer service after sale, merupakan customer service yang melayani pelanggannya setelah proses penjualan terjadi, contohnya adalah pegawai yang melayani layanan purna jual sebuah produk.

#### **2.2 *Automasi / Otomasi***

Menurut Random House Webster College Dictionary, *automation is the technique, method or system of operation or controlling a process by highly automatic means, as by electronic device, reducing human intervention to a*

*minimum*. Dalam pengertian tersebut memiliki arti bahwa otomasi merupakan sebuah teknik, metode, atau sistem operasi yang mengendalikan setiap proses atau kegiatan secara otomatis, melalui bantuan alat elektronik, sehingga dapat meminimlasir campur tangan dari manusia.

Berdasarkan pengertian diatas, penulis dapat menyimpulkan bahwa otomasi merupakan sebuah metode untuk meringankan kegiatan manusia dengan bantuan teknologi yang ada. Dengan adanya otomasi, kegiatan rutin yang dilakukan oleh manusia dapat diringankan bebannya.

### **2.3 Knowledge-Based System**

Menurut Teft (1989) *Knowledge-based system* adalah sebuah program komputer yang didesain untuk menstimulasi penyelesaian masalah berdasarkan kebiasaan atau pengetahuan manusia. *Knowledge-based system* merupakan sebuah program yang menyimpan informasi sangat besar, hampir keseluruhannya merupakan fakta dan peraturan, yang kemudian informasi tersebut akan diolah untuk menghasilkan sebuah solusi dalam menyelesaikan permasalahan yang biasanya menggunakan keahlian manusia. Selain itu juga *knowledge-based system* juga dapat berkomunikasi dengan pengguna guna memperoleh informasi yang lebih spesifik dalam menyelesaikan permasalahan yang dimaksud.

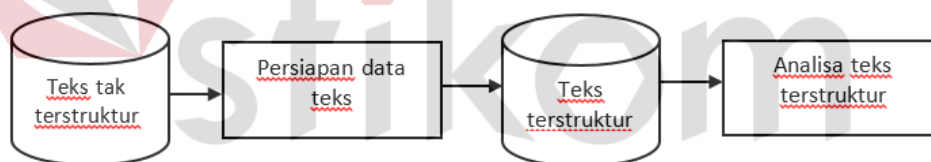
*Knowledge-based system* berbeda dengan program konvensional yang ada, dimana *knowledge-based system* dapat menyajikan lebih dari satu solusi dari permasalahan yang ada. Dan apabila dibandingkan dengan basis data, *knowledge-based* juga memiliki perbedaan yaitu dimana *knowledge-based system* lebih bisa digunakan untuk membantu dalam pengambilan keputusan. Akan tetapi,

*knowledge-based system* masih memerlukan bantuan dari basis data guna menyimpan data dan informasi yang dimiliki.

## 2.4 Text Mining

Menurut (Miller, 2005) pada buku *Data and Text Mining A Business Applications Approach*, *text mining* ialah pemrosesan secara otomatis atau semi-otomatis yang melibatkan struktur mengesankan pada teks dan penggalian informasi yang relevan dari teks. *Text Mining* berkaitan dengan kata. Kata-kata tersebut ditranskrip dan disimpan didalam berkas elektronik, mewakili data mentah untuk analisa.

Sama seperti halnya *data mining*, jumlah data yang diperlukan untuk *text mining* tidaklah bergantung pada besarnya ukuran basis data atau jumlah koleksi dokumen. Hal ini dikarenakan *text mining* merupakan sebuah proses yang menggunakan pendekatan kata dalam melakukan penelitiannya.



Gambar 2.1 Proses Text Mining

## 2.5 Natural Language Processing

Menurut Suhartono (2013), Natural Language Processing atau biasa disebut dengan NLP, merupakan sebuah cabang ilmu *Artificial Intelligent* yang berfokus pada pengolahan bahasa natural. Bahasa natural merupakan bahasa umum yang digunakan untuk berkomunikasi antar manusia. Sedangkan untuk bahasa

komputer sendiri, diperlukan sebuah pemrosesan tersendiri agar apa yang dimaksud oleh pengguna dapat dipahami oleh komputer.

Pustejovsky dan Stubbs (2012) menjelaskan bahwa ada beberapa area utama pada NLP, diantaranya,

- a. *Question Answering System (QAS)*, merupakan kemampuan komputer untuk menjawab pertanyaan yang diajukan oleh penggunanya. Dalam hal ini, pengguna tidak lagi memasukkan kata kunci berkenaan dengan jawaban yang diinginkan melainkan pengguna memasukkan pertanyaan secara langsung ke dalam komputer.
- b. *Summarization*, merupakan pembuatan ringkasan dari kumpulan dokumen atau email sehingga pengguna dibantu untuk mengkonversikan dokumen teks yang besar ke dalam sebuah slide presentasi.
- c. *Machine Translation*, salah satu hasil aplikasi yang telah menarapkan cabang ilmu ini adalah Google Translate, dimana output dari aplikasi yang dibuat dapat memahami bahasa manusia sekaligus menerjemahkannya ke dalam bahasa lain seperti dari bahasa Indonesia ke dalam bahasa Inggris, atau sebaliknya.
- d. *Speech Recognition*, merupakan cabang ilmu NLP yang cukup sulit. Hal ini dikarenakan komputer harus memahami bahasa manusia yang diucapkan. Dan bentuk kalimat yang sering digunakan dalam cabang ilmu ini adalah kalimat tanya dan perintah.
- e. *Document Classification*, adalah salah satu cabang ilmu NLP yang paling sukses. Pekerjaan yang dilakukan melalui cabang ilmu ini yaitu menentukan dimana tempat terbaik dokumen yang baru diinputkan ke

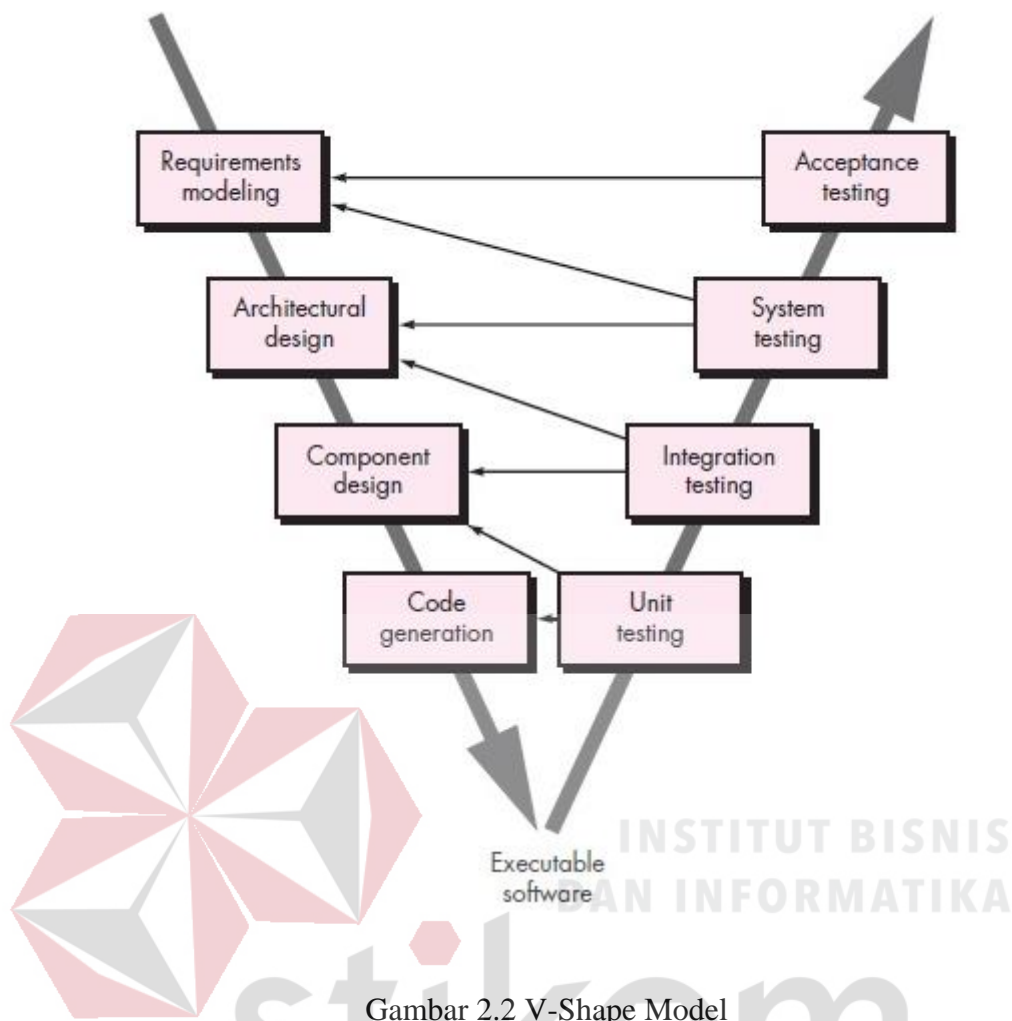
dalam sistem. Hal ini sangat berguna pada aplikasi spam filtering, news article classification, dan movie review.

## 2.6 *System Development Cycle (SDLC)*

Menurut (Efraim, Rainer, & Potter, 2003), *system development life-cycle* (SDLC) adalah metode pengembangan sistem tradisional yang digunakan sebagian besar organisasi saat ini. SDLC juga merupakan sebuah kerangka kerja (*framework*) yang terstruktur dan berisi proses-proses sekuensial dimana sistem informasi dikembangkan.

## 2.7 *V-Shaped Model*

Dalam buku *Software Engineering a Practitioner's Approach Seventh Edition*, *V-Shaped Model* merupakan bentuk model pengembangan software yang bergerak ke sisi kiri "V" secara berlawanan. Persyaratan dasar dalam masalah, disempurnakan menjadi semakin lebih rinci dan teknis secara representasi dari masalah dan solusinya. Sehingga setelah kode telah dihasilkan, pengembang software akan bergerak naik sisi kanan V yang pada dasarnya akan melakukan serangkaian tes (*Assurance*) atau tindakan kualitas yang memvalidasi masing-masing model dibuat sebagai pengembang software. Pada kenyataannya, tidak ada perbedaan mendasar antara siklus hidup klasik dan V-model. Dimana *V-Shaped Model* menyediakan cara tentang bagaimana memvisualisasikan sebuah tindakan terverifikasi dan tervalidasi yang diterapkan untuk pekerjaan rekayasa sebelumnya.



Gambar 2.2 V-Shape Model

Model ini disebut dengan *V-Shaped* karena tahap demi tahap yang dilalui akan terverifikasi dan tervalidasi secara urut. Model ini merupakan model variasi dalam representasi dari model *waterfall* dalam *Software Engineering*. Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem hingga ke tahap selanjutnya sesuai uraian berikut :

- a. *Requirement Modeling*, terdiri dari skenario, informasi dan analisa yang merupakan tindakan persyaratan pemodelan yang menghasilkan model berbasis scenario dari sudut pandang berbagai sistem “aktor” yang menggambarkan perilaku.

- b. *Architectural Design*, merupakan desain arsitektur perangkat lunak yang menganggap dua tingkat desain piramida tentang desain data dan desain arsitektur. Desain data memungkinkan pengembangan untuk mewakili komponen data yang arsitektur dalam sistem konvensional dan definisi kelas (meliputi atribut dan operasi) dalam sistem berorientasi objek. Desain arsitektur berfokus pada representasi dari struktur komponen perangkat lunak, sifat mereka, dan interaksi.
- c. *Component Design*, merupakan sebuah langkah lanjutan setelah langkah pertama dari desain arsitektur selesai. Pada tahap ini, secara keseluruhan data dan struktur program perangkat lunak telah ditetapkan. Tujuannya adalah untuk menerjemahkan model desain ke dalam perangkat lunak operasional. Namun tingkat abstraksi dari model desain yang ada relative tinggi, dan tingkat abstraksi dari program operasional rendah.
- d. *Code Generation*, merupakan proses pembuatan sistem menggunakan bahasa pemrograman yang telah ditentukan. Dan pada tahapan ini dibutuhkan kesesuaian antara semantik kode dan kebenaran sintaksis.
- e. *Unit Testing*, merupakan sebuah langkah yang berfokus dalam upaya verifikasi pada unit terkecil dari desain perangkat lunak hingga komponen perangkat lunak atau modul. Hal ini menggunakan deskripsi desain tingkat komponen sebagai panduan, jalur kontrol penting yang diuji untuk mengungkap kesalahan dalam batas modul. Uji unit berfokus pada logika proses internal dan struktur data dalam batas-batas komponen.
- f. *Integration Testing*, adalah teknik yang sistematis untuk membangun arsitektur perangkat lunak sementara pada saat yang sama melakukan tes

untuk mengungkap kesalahan yang terkait dengan interfacing. Tujuannya adalah untuk mengambil komponen unit diuji dan membangun struktur program yang telah ditentukan oleh desain.

g. *System Testing*, merupakan serangkaian tes yang berbeda yang tujuan utamanya adalah untuk sepenuhnya melaksanakan sistem berbasis computer. Meskipun masing-masing tes memiliki tujuan yang berbeda, namun semua pekerjaan untuk memverifikasi bahwa unsur sistem telah terintegrasi dengan baik dan melakukan fungsi *allocated*(pengalokasian).

h. *Acceptance Testing*, merupakan tahap pengujian penerimaan yang dilakukan oleh pelanggan dalam upaya untuk melaksanakan semua fitur dan mengikuti seperangkat prinsip-prinsip serta konsep-konsep yang berlaku untuk *coding* dan *testing*.

## 2.8 Confusion-Matrix

*Confusion matrix* merupakan perhitungan yang terdiri dari informasi tentang data actual dan prediksi yang telah diklasifikasi oleh sistem. Performa dari sistem biasanya dievaluasi data yang ada menggunakan matrix (Hamilton, 2012). Bentuk sederhana dari *confusion matrix* adalah sebagai berikut,

		Prediksi	
		Negatif	Positif
Aktual	Negatif	A	B
	Positif	C	D

Gambar 2.3 Bentuk dari *Confusion Matrix* (Hamilton, 2012)

Keterangan:

- A merupakan jumlah prediksi yang benar bahwa hal tersebut bernilai negatif



- B merupakan jumlah prediksi yang salah bahwa hal tersebut bernilai negatif
- C merupakan jumlah prediksi yang salah bahwa hal tersebut bernilai positif
- D merupakan jumlah prediksi yang benar bahwa hal tersebut bernilai positif

Dari bentuk matrix diatas didapatkan beberapa formula standar yang diantaranya,

$$Accuracy = \frac{A + D}{A + B + C + D}$$

$$Error Rate = 1 - Accuracy$$

*Accuracy* merupakan nilai dari proporsi prediksi yang bernilai benar, sedangkan *error rate* digunakan untuk mengukur nilai kesalahan yang dimiliki.

