

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1. Kata**

Kata menurut KBBI adalah unsur bahasa yang diucapkan atau dituliskan yang merupakan perwujudan kesatuan perasaan dan pikiran yang dapat digunakan dalam berbahasa. Kata merupakan satuan bahasa yang dapat berdiri sendiri atau merupakan satuan bebas yang paling kecil yang dapat berdiri sendiri. Selain itu, kata merupakan unsur bahasa yang diucapkan atau dituliskan dan merupakan perwujudan kesatuan perasaan dan pikiran yang dapat digunakan dalam berbahasa. Dalam penelitian ini kata memiliki peranan penting dalam menentukan kesalahan kata dan rekomendasi pemilihan perbaikan kata [9].

#### **2.2. Kesalahan Eja**

Kesalahan ejaan merupakan kesalahan dalam input ejaan kata. Kesalahan penulisan ejaan terjadi disebabkan beberapa hal, diantaranya ketidaktahuan penulis akan kata yang dicari, kesalahan ini karena kurang teliti dalam penulisan, penyebab ketiga adalah kesalahan yang bersifat teknis [13].

Berdasarkan bentuknya, Kesalahan ejaan terdapat dua kategori yaitu.

1. Kesalahan non kata Kesalahan ejaan non kata (non-word) adalah kesalahan yang berfokus pada kata yang terbentuk umumnya oleh kesalahan pengetikan. Kesalahan ejaan non kata ini menghasilkan kata-kata yang tidak masuk akal. Contoh kata **nasi** yang menjadi **nsi, nas, nai**.
2. Kesalahan kata yang sebenarnya Kesalahan kata yang sebenarnya (real-word) adalah kesalahan yang menekankan pada penanganan penempatan kata dalam kalimat. Kesalahan kata yang sebenarnya menghasilkan kata sah lainnya. Contoh **nasi** yang menjadi **basi, napi. basi** dan **napi** sendiri memiliki makna yang berbeda dengan **nasi**.

### 2.3. *Typo Checking*

Typo checking adalah proses pemeriksaan kata untuk mendeteksi kata yang salah eja dan memberikan kandidat kata yang benar. *typo checking* memiliki 2 cara dalam pendeteksian kesalahan yaitu..

1. Pendeteksian kesalahan Non-word yaitu pendeteksian kesalahan kata yang tidak memiliki arti. Contoh **nasi** menjadi **nsi**. Sistem hanya mendeteksi kesalahan kata yang tidak memiliki arti.
2. Pendeteksian kesalahan Real-word yaitu pendeteksian kesalahan kata yang memiliki makna/arti lain seperti **nasi** yang menjadi **basi**.

Pada penelitian ini hanya akan mendeteksi kesalahan non-word yaitu fokus pada pendeteksian kesalahan kata yang tidak memiliki arti seperti **nasi** menjadi **nsi**.

### 2.4. *Pencocokan String*

Pencocokan *string* (*string matching*) secara garis besar dapat dibedakan menjadi dua yaitu *exact string matching* dan *inexact string matching* [15].

#### 1. *Exact String Matching*

Merupakan pencocokan string secara tepat dengan susunan karakter dalam string yang dicocokkan memiliki jumlah maupun urutan karakter dalam string yang sama. Contoh : kata “**baju**” akan menunjukkan kecocokan hanya dengan kata “**baju**”.

#### 2. *In Exact String Matching*

Disebut juga *fuzzy string matching*, merupakan pencocokan string secara samar, maksudnya pencocokan string di mana string yang dicocokkan memiliki kemiripan di mana keduanya memiliki susunan karakter yang berbeda (mungkin jumlah atau urutannya) tetapi string-string tersebut memiliki kemiripan baik kemiripan tekstual/penulisan (*approximate string matching*) atau kemiripan ucapan (*phonetic string matching*).

### 2.5. *Penelitian Terdahulu*

Beberapa penelitian terdahulu yang berkaitan dengan penelitian *spelling checker* khususnya tentang teknik pencocokan string, seperti berikut.

Dalam penelitian yang ditulis oleh Muhamad Bahari Ilmi, Nitia Rahmi yang berjudul Penerapan Algoritma *Levenshtein Distance* untuk Mengoreksi Kesalahan Pengejaan pada Editor Teks memiliki kesimpulan bahwa itu algoritma Levenshtein Distance yang menerapkan program dinamis untuk menghitung kemiripan sebuah string. Setelah tingkat kemiripan sebuah string diperoleh, secara brute force dicari string pada kamus yang tingkat kemiripannya tinggi dengan string tersebut.

Dalam penelitian yang ditulis oleh Radiant Victor Imbar yang berjudul Implementasi *Cosine Similarity* dan Algoritma *Smith-Waterman* untuk Mendeteksi Kemiripan Teks dijelaskan bahwa mengatasi struktur teks yang tidak konsisten dan kompleks, diperlukan konsep *text mining*. Berfokuskan pada teks berbahasa Indonesia dan berkonsepkan *text mining*, penelitian ini bertujuan untuk mengembangkan sebuah aplikasi yang mengimplementasikan *cosine similarity* yang berguna untuk mengukur kesamaan teks berdasarkan kemunculan kata-kata dalam teks tersebut dan algoritma *Smith-Waterman* yang berfungsi untuk menghitung kemiripan teks berdasarkan urutan kata.

## **2.6. Kamus Online**

Kamus online adalah situs web yang menyediakan layanan menterjemahkan kata atau kalimat layaknya kamus yg ada di toko buku. di dalam situs ini kita dapat mengubah suatu kata dalam satu bahasa ke bahasa yang lain. beberapa kamus juga memberikan layanan pengertian atau makna/maksud dari suatu kata. Saat ini kamus online bahasa Indonesia terdiri dari beberapa jenis seperti; KBBI Online Kemdikbud, Kamus Besar Bahasa Indonesia – Universitas Indonesia [16], Kateglo, dan lain sebagainya. Pada penelitian ini kamus yang digunakan adalah KBBI karena KBBI mempunyai cukup banyak kata sehingga membantu untuk digunakan dalam sistem sehingga baik pemakai atau penganalisis sistem mempunyai dasar pengertian sama tentang masukan, keluaran dan proses.

## 2.7. Preprocessing

*Preprocessing* adalah proses pengolahan data asli sebelum diolah pada tahap selanjutnya. Masukan data awal berupa dokumen. Tahapan preprocessing pada penelitian ini meliputi, *case folding*, *filtering* dan tokenisasi kata.

### 2.7.1. Case Folding

*Case folding* merupakan tahapan yang mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf ‘a’ sampai dengan ‘z’ yang diterima.

Pada penelitian ini *case folding* digunakan untuk mengubah dokumen input menjadi kata kecil. Contoh:

**Tabel 2.1 Contoh Case Folding**

Sebelum Case Folding	Setelah Case Folding
Nasi	nasi
Bahasa Indonesia	bahasa indonesia

### 2.7.2. Filtering

Tahap *filtering* adalah tahap menghapus karakter-karakter dari data masukan. *Filtering* berguna untuk menghilangkan karakter-karakter yang dianggap tidak valid (karakter angka dan symbol).

Pada penelitian ini *filtering* digunakan untuk menghapus karakter simbol seperti “(,)-,’,!,@,#,\$,%,&\*,,,”, “=,+,” dan karakter angka “(0-9)” dikarenakan karakter simbol dan angka tidak dibutuhkan dalam penelitian ini. Sehingga keluaran hanya menampilkan karakter huruf “(a-z)”. Contoh:

**Tabel 2.2 Contoh Filtering**

Sebelum Filtering	Setelah Filtering
“Menanggapi kicauan @SBYudhoyono”	Menanggapi kicauan SBYudhoyono”

### 2.7.3. Tokenisasi Kata

Tokenisasi kata adalah pemotongan *string input* berdasarkan tiap kata yang menyusunnya. Pemecahan kalimat menjadi kata-kata tunggal dilakukan dengan *menscan* kalimat dengan pemisah (*delimiter*) spasi.

Pada penelitian ini tokenisasi kalimat berguna untuk mendapatkan list kata dari dokumen input. Contoh:

**Tabel 2.3 Contoh Tokenisasi Kata**

Sebelum Tokenisasi Kata	Sesudah Tokenisasi Kata	
Ayah budi adalah seorang seniman	Ayah	seorang
	budi	seniman
	adalah	
Dia juga berprofesi sebagai seorang wartawan	dia	sebagai
	juga	seorang
	berprofesi	wartawan

## 2.8. Algoritma *Levenshtein*

Algoritma ini lebih dikenal dengan Levenshtein Distance. Algoritma ini ditemukan oleh Vladimir Losifovich Levenshtein yang merupakan ilmuwan dari Rusia pada tahun 1965 [17]. Algoritma ini mengukur kesamaan antara 2 string untuk menentukan nilai terkecil atau disebut dengan nilai edit distance, 2 string yang akan dibandingkan nantinya akan lebih dikenal dengan string sumber (s) dengan string target (t), contoh.

1. Jika (s) = "coba", dan (t) = "coba" ==> maka  $Levenshtein(s,t) = 0$ , karena kedua string tersebut sama / tidak ada perbedaan.
2. Jika (s) = "coba", dan (t) = "cona" ==> maka  $Levenshtein(s,t) = 1$ , karena adanya perbedaan antara kedua string tersebut di huruf yang ketiga yakni 'b' dengan 'n'.

Contohnya jika ada 2 buah kata x = BARU dengan y = BATU, berapakah nilai edit distance perbedaan huruf dari kedua kata tersebut.

### A. Pembuatan tabel matriks

**Tabel 2.4 Pembuatan Tabel Matriks**

d[i,j]		B	A	R	U
	0	1	2	3	4
B	1				
A	2				
T	3				
U	4				

**B. Pengisian tabel matriks**

ketika  $i = 1$ , dan  $j = 1$

\* $x[i] = B$ ,  $y[i] = B$ ,  $cost = 0$

kemudian ambil nilai minimal dari

- $d[i-1,j] + 1$   $\implies d[1-1,1] + 1 = 1$
- $d[i,j-1] + 1$   $\implies d[1,1-1] + 1 = 2$
- $d[i-1,j-1] + cost$   $\implies d[1-1,1-1] + 0 = 0$

**Tabel 2.5 Pengisian Tabel Matriks**

d[i,j]		B	A	R	U
	0	1	2	3	4
B	1	<b>0</b>			
A	2				
T	3				
U	4				

C. Hasil pengisian matriks.

**Tabel 2.6 Hasil Edit Distance**

d[i,j]		B	A	R	U
	0	1	2	3	4
B	1	0	1	2	3
A	2	1	0	1	2
T	3	2	1	1	2
U	4	3	2	2	<b>1</b>

Dari sini dapat kita ketahui nilai edit distance kedua kata BARU dan BATU bernilai 1 (satu)

## 2.9. Algoritma *Smith-Waterman*

Algoritma *Smith-waterman* pertama ditemukan oleh Temple F. Smith dan Michael S. Waterman pada 1981. Algoritma ini melakukan penjajaran sekuens lokal, dengan memberikan daerah yang sama diantara 2 sekuens,

menyejajarkan 2 sekuens yang sebagian sama, dan juga dapat menjajarkan 2 subsekuens ke sekuens itu sendiri [18].

Algoritma ini menggunakan algoritma *Needleman-Wunsch* sebagai dasar, yang dimana algoritma tersebut menyejajarkan sekuens global. Kedua algoritma ini menggunakan teknik program dinamis. Perbedaan dari algoritma *Needleman Wunsch* adalah algoritma *Smith-Waterman* mencari best local alignment yaitu kecocokan substring pada 2 sekuens, sedangkan algoritma *Needleman-Wunsch* mencari *best global alignment* yaitu kecocokan dari panjang ujung ke ujung suatu sekuens yang terlibat.

Program dinamis yang digunakan untuk mencari alignment optimal pada 2 sekuens menggunakan nilai (scores) untuk setiap kecocokan dan ketidakcocokan pada matriks nilai (*scoring matrices*). Dengan mencari nilai tertinggi pada matriks, alignment dapat secara akurat ditemukan.

Langkah dasar untuk algoritma *Smith-Waterman* adalah.

1. Inisialisasi sebuah matriks.
2. Mengisi matriks dengan nilai yang sesuai.

Melacak kembali sekuens yang memiliki alignment yang sesuai.

## 2.10. Algoritma *Needleman-Wunsch*

Algoritma *Needleman-Wunsch* merupakan perluasan algoritma pencocokan string pada teks atau barisan sebagai salah satu implementasi program dinamis. Algoritma ini merupakan salah satu algoritma yang digunakan untuk menentukan tingkat kesamaan atau kecocokan dua buah teks [19].

Prosedur algoritma *Needleman-Wunsch* adalah sebagai berikut.

1. Tetapkan nilai untuk setiap kecocokan karakter, ketidakcocokan karakter, serta nilai penalti apabila salah satu karakter dari kedua teks yang dibandingkan digeser sehingga diganti dengan karakter celah kosong.
2. Nilai kecocokan karakter boleh sama untuk setiap karakter, boleh tidak (bergantung definisi yang diinginkan).

Nilai kecocokan (apabila dua karakter sama) haruslah ditetapkan sebagai suatu nilai positif.

### 2.11. Pengujian Akurasi

Pengujian ini bertujuan untuk menentukan akurasi yang dilakukan dalam penelitian ini. Dalam penelitian ini menggunakan pengujian pendeteksian kesalahan kata, pengujian akurasi perbaikan dan rekomendasi.

Tujuan dari pengujian pendeteksian kesalahan kata yaitu untuk melihat berapa akurasi pendeteksian kesalahan kata. pengukuran akurasi pendeteksian kesalahan kata yang biasa digunakan dalam penelitian pengecekan kesalahan pengetikan yaitu [17]:

$$\begin{aligned} & \textit{Akurasi Pendeteksian Kesalahan Kata} \\ & = \left( \frac{\textit{Jumlah Kata Salah Sesuai}}{\textit{Jumlah Kata Terdeteksi Salah}} \right) \times 100\% \end{aligned}$$

dimana,

Jumlah kata salah sesuai : yaitu jumlah kata yang dianggap salah

Jumlah kata terdeteksi salah : yaitu jumlah total kata yang terdeteksi salah.

Tujuan dari pengujian ini untuk melihat berapa banyak kata yang rekomendasinya tepat. Berdasarkan skenario yang dibuat Pengujian akurasi rekomendasi pada penelitian ini menggunakan pengujian akurasi. Pengukuran akurasi perbaikan yang biasa digunakan dalam penelitian pengecekan kesalahan pengetikan yaitu [17]:

$$\textit{Akurasi Rekomendasi} = \left( \frac{\textit{Jumlah Rekomendasi Sesuai}}{\textit{Jumlah Kata Terdeteksi Salah}} \right) \times 100 \%$$

dimana,

Jumlah rekomendasi sesuai : yaitu jumlah total kata yang memiliki rekomendasi yang dianggap sesuai.

Jumlah kata terdeteksi salah : yaitu jumlah total kata yang terdeteksi salah.

Tujuan dari pengujian ini untuk melihat berapa banyak kata yang memiliki perbaikan tepat pengukuran akurasi perbaikan yang biasa digunakan dalam penelitian pengecekan kesalahan pengetikan yaitu [17]:

$$\text{Akurasi Perbaikan} = \left( \frac{\text{Jumlah Perbaikan Sesuai}}{\text{Jumlah Kata Terdeteksi Salah}} \right) \times 100\%$$

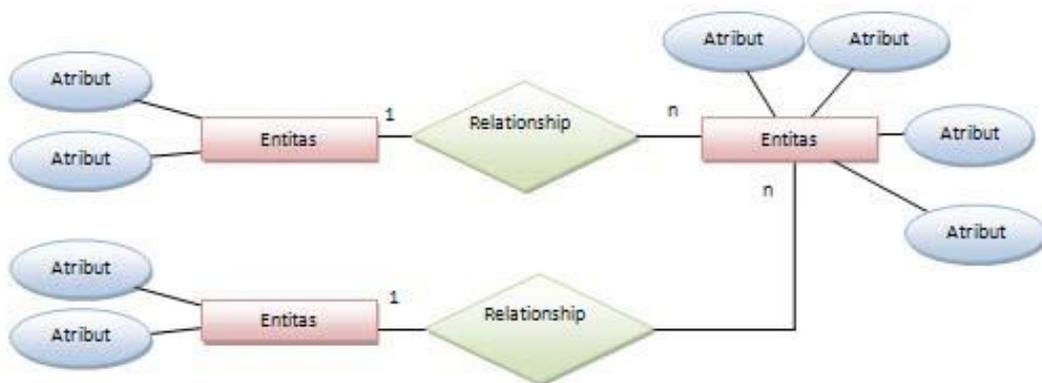
dimana,

Jumlah perbaikan benar : yaitu jumlah total kata yang memiliki perbaikan yang dianggap sesuai.

Jumlah kata terdeteksi salah : yaitu jumlah total kata yang terdeteksi salah.

### 2.12. ERD (*Entity Relationship Diagram*)

Diagram E-R adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas, atribut dan hubungan antar entitas. Huruf E sendiri menyatakan entitas dan R menyatakan hubungan (dari kata Relationship). Model ini dinyatakan dalam bentuk diagram. Perlu diketahui bahwa model seperti ini tidak mencerminkan bentuk fisik yang nantinya akan disimpan dalam database, melainkan hanya bersifat konseptual [20]. Contoh tampilan ERD yaitu seperti pada gambar 2.1.



**Gambar 2.1 Tampilan ERD**

## **2.13. Model Perangkat Lunak**

### **2.13.1. Pemrograman Berorientasi Objek**

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut [17].

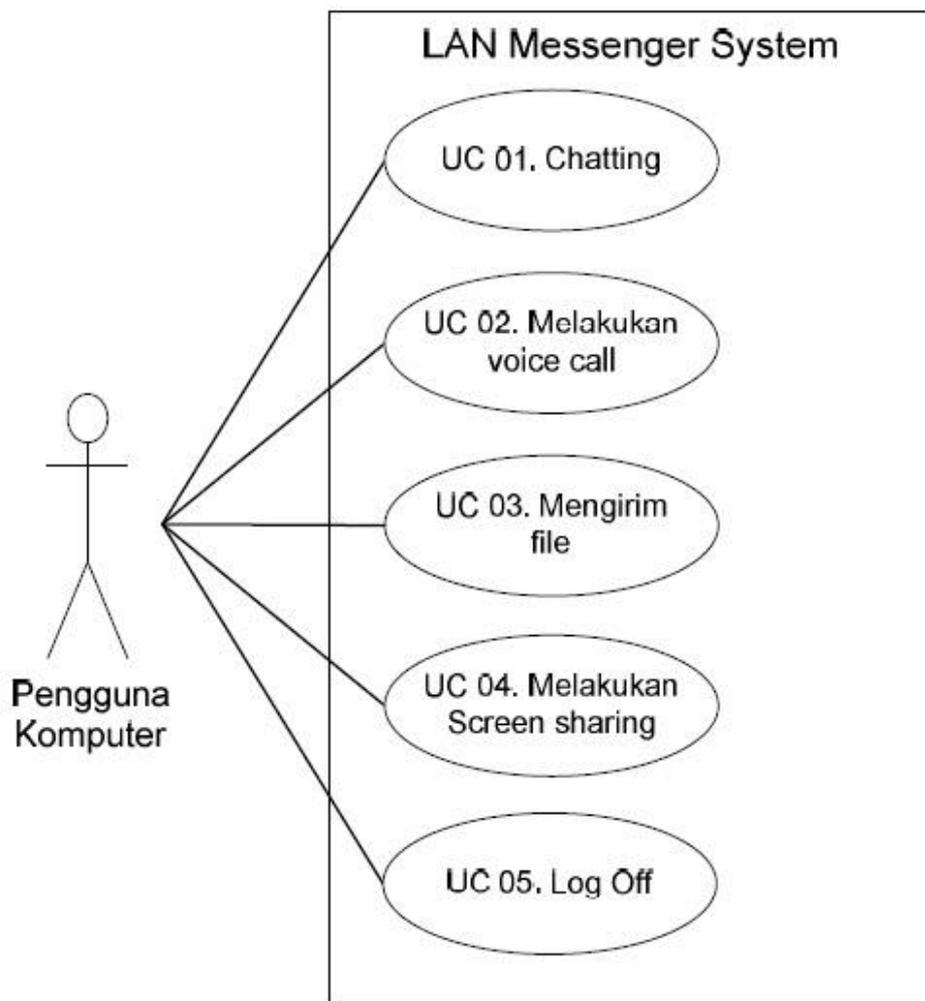
1. Meningkatkan produktivitas Kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reusable*).
2. Kecepatan pengembangan Sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.
3. Kemudahan pemeliharaan pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.
4. Adanya konsistensi Sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan, maupun pengkodean.
5. Meningkatkan kualitas perangkat lunak Pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

### **2.13.2. Unified Modelling Language (UML)**

UML adalah sebuah tools atau alat yang banyak digunakan di dunia industri untuk mendefinisikan kebutuhan, membuat analisis & design, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [18].

### 2.13.3. Use Case Diagram

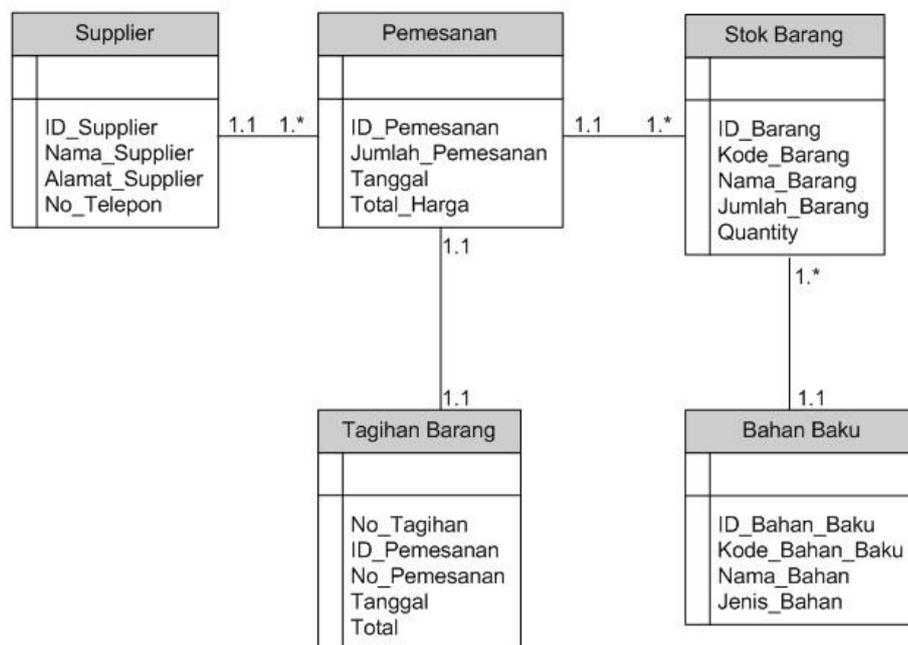
*Use case diagram* merupakan pemodelan untuk mengetahui tingkah laku (behaviour) sistem yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Use case, digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan sistem fungsi-fungsi itu. Syarat penamaan pada use case adalah nama didefinisikan sesingkat dan sejelas mungkin agar dapat dipahami. Contoh tampilan *use case diagram* yaitu seperti pada gambar 2.2.



**Gambar 2.2** Tampilan *Use Case Diagram*

#### 2.13.4. Class Diagram

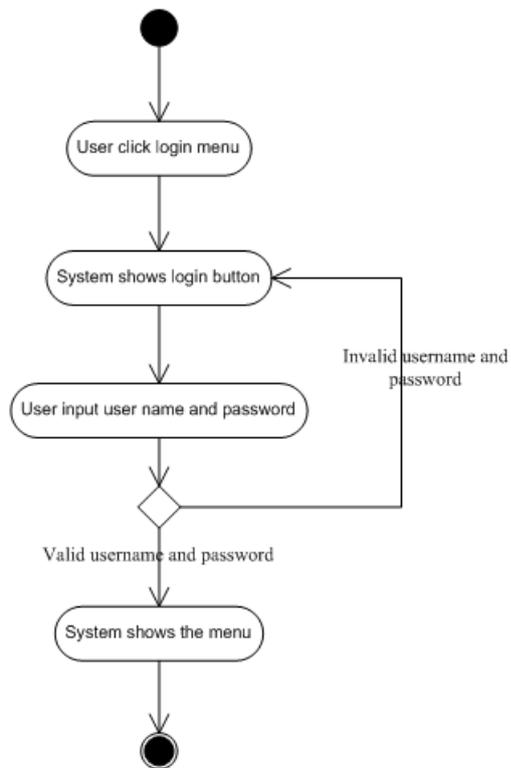
Class diagram atau kelas diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode. Diagram kelas dibuat agar pembuat program atau programmer membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak dapat sinkron. Contoh tampilan *class diagram* yaitu seperti pada gambar 2.3.



**Gambar 2.3 Tampilan Class Diagram**

#### 2.13.5. Activity Diagram

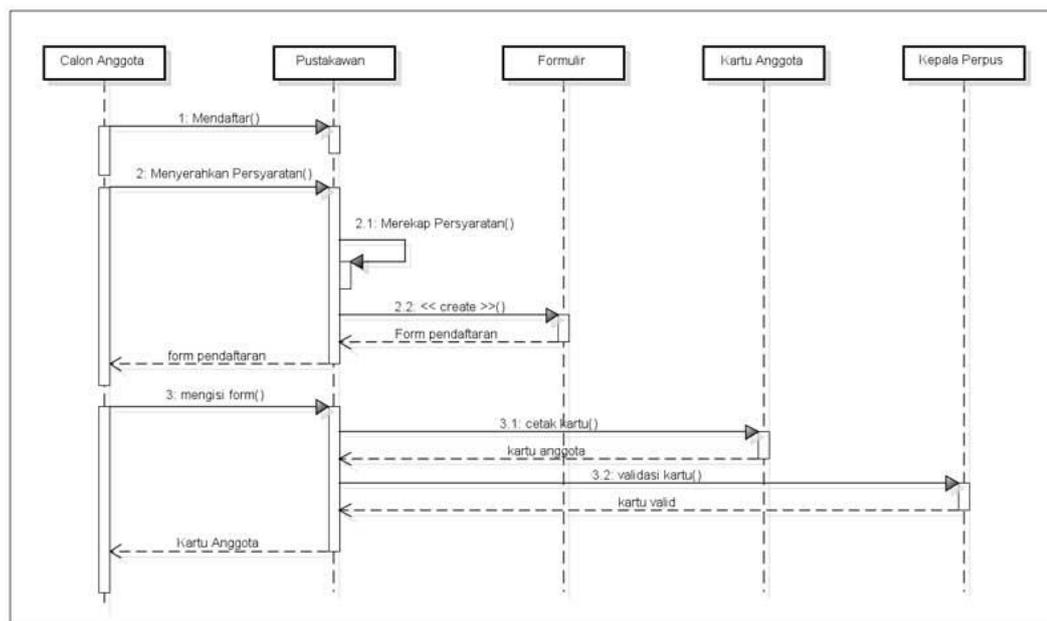
*Activity diagram* atau diagram aktivitas menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perlu diperhatikan bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Contoh tampilan *activity diagram* yaitu seperti pada gambar 2.4.



**Gambar 2.4** Tampilan *Activity Diagram*

#### 2.13.6. *Sequence Diagram*

Sequence diagram atau diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Untuk menggambarkan diagram sekuen harus diketahui metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Contoh tampilan *sequence diagram* yaitu seperti pada gambar 2.5.



**Gambar 2.5** Tampilan *Sequence Diagram*

## 2.14. Bahasa Pemrograman

Bahasa pemrograman adalah *software* bahasa komputer yang digunakan dengan cara merancang atau membuat program sesuai dengan struktur dan metode yang dimiliki oleh bahasa program itu sendiri. Komputer mengerjakan transformasi data berdasarkan kumpulan perintah program yang telah dibuat oleh program. Kumpulan perintah ini harus dimengerti oleh komputer, berstruktur tertentu (*syntax*), dan bermakna. Bahasa pemrograman merupakan notasi untuk memberikan secara tepat program komputer.

Pada penelitian ini bahasa pemrograman yang digunakan adalah PHP karena bahasa pemrograman PHP tidak memerlukan *Compile* dalam penggunaannya.

### 2.14.1. PHP (*Personal Home Page*)

PHP adalah kependekan dari PHP *Hypertext Preprocessor*, bahasa *interpreter* yang mirip dengan bahasa C dan Perl yang memiliki kesederhanaan dalam perintah, PHP dapat digunakan bersama dengan HTML sehingga memudahkan dalam pembangunan aplikasi web dengan cepat. PHP dapat digunakan untuk meng-*update* basis data dan menciptakan basis data. *Interpreter*

adalah sebuah program yang digunakan untuk membaca *file* yang berisi kode program yang akan dijalankan kemudian *interpreter* tersebut akan meminta CPU untuk melakukan perintah yang diterimanya.

Seperti halnya program *open source* lainnya, PHP dibuat di bawah lisensi GNU, *General Public License*, yang dapat di download gratis melalui situs <http://www.php.net>. Awalnya, PHP diciptakan oleh Andi Gutmans untuk menghitung jumlah pengunjung yang mengakses *homepage* yang dbuatnya. PHP banyak mendukung basis data, seperti MySQL, PostgreSQL, *Interbase*, ODBC, mSQL, *Oracle*, dan *Sybase* [21].

Pada penelitian ini PHP digunakan sebagai bahasa pemrograman utama dalam membangun sistem *Typo Checking* dimana kode-kode PHP hanya akan dieksekusi di server kemudian hasilnya dikirim ke browser dalam format HTML.

#### **2.14.2. JavaScript**

*Javascript* adalah bahasa skrip yang ditempelkan pada kode HTML dan diproses di sisi klien. Dengan adanya bahasa ini, kemampuan dokumen HTML menjadi semakin luas. Sebagai contoh, dengan menggunakan *JavaScript* dimungkinkan untuk memvalidasi masukan- masukan pada formulir sebelum formulir dikirimkan ke server. *Javascript* bukanlah bahasa Java dan merupakan dua bahasa yang berbeda. *Javascript* diinterpretasikan oleh klien (kodenya bisa dilihat pada sisi klien), sedangkan kode Java dikompilasi oleh pemrogram dan hasil kompilasinyalah yang dijalankan oleh klien.

Pada penelitian ini *Javascript* diimplementasikan pada pembuatan antarmuka tambah data, apabila *user* ingin menambah data cerpen baru sehingga tampilan lebih interaktif dan dinamis

#### **2.15. Bahasa Markup**

Bahasa *Markup* merupakan informasi tambahan yang ditempatkan pada teks untuk menjelaskan bagaimana teks tersebut diinterpretasi. Dalam penelitian ini HTML digunakan sebagai tambahan dalam pengimplementasian program.

### **2.15.1. HTML**

HTML adalah sebuah bahasa *markup* berupa kode-kode tag yang menginstruksikan browser untuk menghasilkan tampilan sesuai dengan yang diinginkan. HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web dan HTML kini merupakan standar Internet yang saat ini dikendalikan oleh *World Wide Web Consortium (W3C)*.

## **2.16. Software Pendukung**

### **2.16.1. XAMPP**

XAMPP adalah perangkat lunak (*free software*) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain. Apache HTTP Server, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis. Pada penelitian ini XAMPP berguna sebagai web server.

### **2.16.2. Adobe Dreamweaver CS 6**

Adobe dreamweaver cs 6 merupakan HTML editor professional yang berfungsi mendesain, melakukan editing dan mengembangkan aneka website. Salah satu kelebihan dreamweaver cs 6 yaitu ruang kerja dreamweaver cs 6 beserta tools yang tersedia dapat digunakan dengan sangat mudah dan cepat sehingga anda bisa membangun suatu website dengan cepat dan tanpa harus melakukan coding. Fungsi coding dreamweaver cs 6 tidak hanya mendukung coding HTML tetapi juga CSS (*Cascading Style Sheet*), Javascript, Coldfusion, ASP (*Active Server Sheet*), JSP (*Java Server Page*), dan Dreamweaver juga memungkinkan anda membangun website dengan server berbahasa CFML (*Colfusion Markup Language*), ASP.net,

dan PHP. Pada penelitian ini Adobe Dreamweaver CS 6 digunakan sebagai tools dalam membantu membuat kode-kode program.

### **2.17. Web Browser**

*Web browser* adalah suatu program perangkat lunak yang berfungsi untuk menerima dan menyajikan sumber informasi di Internet. *Web browser* memudahkan setiap orang untuk menjelajah dan menemukan berbagai informasi di di dunia maya. Pada awalnya *web browser* hanya dapat menampilkan teks tetapi pada perkembangan berikutnya *web browser* sudah bisa menampilkan teks, video, dan suara [19].

Contoh *web browser* yang digunakan saat ini.

1. Internet Explorer
2. Mozilla Firefox
3. Safari
4. Google Chrome
5. Opera

Pada penelitian ini web browser yang dipilih untuk digunakan adalah google chrome untuk menampilkan dan memproses program *typo checking*

### **2.18. DBMS**

DBMS adalah singkatan dari “*Database Management System*” yaitu sistem penorganisasian dan sistem pengolahan Database pada komputer. DBMS atau *database management system* ini merupakan perangkat lunak (*software*) yang dipakai untuk membangun basis data yang berbasis komputerisasi. DBMS (*Database Management System*) ini juga dapat membantu dalam memelihara serta pengolahan data dalam jumlah yang besar, dengan menggunakan DBMS bertujuan agar tidak dapat menimbulkan kekacauan dan dapat dipakai oleh user sesuai dengan kebutuhan. DBMS ialah perantara untuk user dengan basis data, untuk dapat berinteraksi dengan DBMS dapat memakai bahasa basis data yang sudah di tentukan oleh perusahaan DBMS. Bahasa basis data umumnya terdiri dari berbagai macam instruksi yang diformulasikan sehingga instruksi tersebut dapat di proses

oleh DBMS. Pada penelitian ini DBMS yang digunakan adalah MySQL karena sintaksnya lebih mudah dipahami.

### 2.18.1. MySQL

MySQL termasuk dalam kategori *database management system*, yaitu suatu database yang terstruktur dalam pengolahan dan penampilan datanya. MySQL merupakan *database* yang bersifat *client server*, di mana data diletakkan di *server* yang bisa diakses melalui komputer *client*. Pengaksesan dapat dilakukan apabila komputer telah terhubung dengan *server*. Berbeda dengan *database* desktop, di mana segala pemrosesan data harus dilakukan pada komputer yang bersangkutan.

Bila diinstal pada sistem operasi Microsoft Windows, maka MySQL berlisensi *shareware*, tetapi tidak mempunyai *expired date* (batas waktu). Sedangkan bila diinstal pada sistem operasi lainnya, maka berlisensi *free* sesuai dengan *General Public Licence* (GPL).

MySQL merupakan *database* yang dikembangkan dari bahasa SQL (*Structured Query Language*). SQL merupakan bahasa terstruktur yang digunakan untuk interaksi antara script program dengan *database server* dalam hal pengolahan data. Dengan SQL, maka dapat dibuat tabel yang akan diisi data, memanipulasi data seperti menambah, menghapus dan meng-update data, serta membuat suatu perhitungan berdasarkan data yang ditemukan [20].

Setiap RDBMS (*Relation Database Management System*) seperti Oracle, SQL Server, MySQL dan lain-lain, pasti memiliki tool yang dapat digunakan untuk mempermudah pengoperasian database. Oracle memiliki TOAD. SQL Server memiliki *Enterprise Manager* dan *SQL Query Analyzer*. Sedangkan MySQL memiliki tool atau aplikasi yang disebut Php My Admin [20].