

BAB II

LANDASAN TEORI

2.1. Logika *Fuzzy*

Konsep tentang logika *Fuzzy* diperkenalkan oleh Prof. Lotfi Astor Zadeh pada 1962. Logika *Fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang cocok untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded sistem*, jaringan PC, *multi channel* atau *work station* berbasis akuisisi data, dan sistem kontrol. Metodologi ini dapat diterapkan pada perangkat keras, perangkat lunak, atau kombinasi keduanya. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah hanya mempunyai dua kemungkinan, “Ya” atau “Tidak”, “Benar” atau “Salah”, “Baik” atau “Buruk”, dan lain-lain. Oleh karena itu semua itu dapat mempunyai nilai keanggotaan 0 atau 1. Akan tetapi dalam logika *fuzzy* memungkinkan nilai keanggotaan berada diantara 0 dan 1. Artinya bisa saja suatu keadaan mempunyai dua nilai “Ya dan Tidak”, “Benar dan Salah”, “Baik dan Buruk” secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya (Sutojo dkk, 2011).

Logika *fuzzy* dapat juga diartikan suatu cara yang tepat untuk memetakan suatu ruang *input* ke dalam suatu ruang *output* yang memiliki toleransi terhadap data-data yang tidak tepat (Wardhani dkk, 2011). Tidak seperti logika *Boolean*, logika *fuzzy* memiliki banyak nilai, *fuzzy* membaginya dalam derajat keanggotaan dan derajat kebenaran, yaitu sesuatu yang dapat menjadi sebagian benar dan sebagian salah pada waktu yang sama (Kusumadewi, 2003).

Ada tujuh alasan digunakannya Logika *fuzzy* (Kusumadewi, 2003) :

- 1) Konsep Logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.
- 2) Logika *fuzzy* sangat fleksibel.
- 3) Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat .
- 4) Logika *fuzzy* mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks.



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

- 5) Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui pelatihan.
- 6) Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
- 7) Logika *fuzzy* didasarkan pada bahasa alami.

Dalam memahami logika *fuzzy*, ada dua atribut dalam konsep himpunan *fuzzy* (Sutojo dkk, 2011) :

- 1) Linguistik, yaitu nama suatu kelompok yang mewakili suatu keadaan tertentu dengan menggunakan bahasa alami. Misalnya dingin, sejuk dan panas mewakili variabel temperatur.
- 2) Numeris, yaitu suatu nilai yang menunjukkan ukuran dari suatu variabel. Misalnya 20, 30, dan sebagainya.

2.2. Pencocokan *String* (*String Matching*)

Pengertian *string* menurut *Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology* (NIST) adalah susunan dari karakter-karakter (angka, alfabet atau karakter yang lain) dan biasanya direpresentasikan sebagai struktur data *array*. *String* dapat berupa kata, frase, atau kalimat.

Pencocokan *string* menjadi bagian penting dari sebuah proses pencarian *string* (*string searching*) dalam sebuah *database*. Hasil dari pencarian sebuah *string* dalam *database* tergantung dari teknik atau cara pencocokan *string* yang digunakan.

Pencocokan *string* (*String Matching*) menurut *Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology* (NIST), diartikan sebagai sebuah permasalahan untuk menemukan pola susunan karakter *string* di dalam *string* lain atau bagian dari isi teks.

Pencocokan *string* secara garis besar dapat dibedakan menjadi dua (Syaroni dkk, 2005), yaitu:

- 1) *Exact String Matching*, merupakan pencocokan *string* secara tepat dengan susunan karakter dalam *string* yang dicocokkan memiliki jumlah maupun



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

urutan karakter yang sama. Contoh: kata sistem akan menunjukkan kecocokan hanya dengan kata sistem.

- 2) *Inexact String Matching* atau *Fuzzy String Matching*, merupakan pencocokan *string* secara samar, maksudnya pencocokan *string* dimana *string* yang dicocokkan memiliki kemiripan yang bersifat samar, yaitu: 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*).

Exact String Matching bermanfaat jika pengguna ingin mencari *string* dalam dokumen atau teks yang sama persis dengan *string* masukan. Tetapi jika pengguna menginginkan pencarian *string* yang mendekati dengan *string* masukan atau terjadi kesalahan penulisan *string* masukan maupun dokumen objek pencarian, maka *inexact String Matching* yang lebih bermanfaat. Beberapa algoritma *exact String Matching* antara lain: algoritma *brute force*, *knuth-morrispratt*, *boyer-moore* dan yang lainnya. Beberapa algoritma *inexact String Matching* antara lain: *levenshtein edit distance*, *soundex*, *metaphone*, *caverphone* dan yang lainnya.

Dari contoh yang diberikan, sebenarnya *phonetic String Matching* juga dapat dimanfaatkan untuk *approximate String Matching* dengan batasan dua *string* yang dicocokkan masih memiliki kemiripan ucapan. *Phonetic String Matching* sering juga dimanfaatkan untuk *approximate String Matching* karena *phonetic String Matching* lebih mudah diimplementasikan. *Phonetic String Matching* banyak digunakan dalam bahasa Inggris karena dalam bahasa Inggris terdapat perbedaan antara penulisan dan pengucapan. Pencarian teks dapat dilakukan dengan menggunakan metode pencarian *string* (*string searching*). Proses pencocokan *string* (*String Matching*) yang merupakan bagian utama dalam proses pencarian *string* memegang peranan penting untuk mendapatkan teks yang relevan. Dengan demikian hasil dari pencarian sebuah *string* dalam teks tergantung dari teknik atau cara pencocokan *string* yang digunakan.



2.3. *Inexact String Matching (Fuzzy String Matching)*

Metode *Fuzzy String Matching* diarahkan untuk mencari nilai dari beberapa *string* yang mendekati dan tidak hanya menghasilkan cocok atau tidak cocok (Syaroni dkk, 2005).

Konsep *Fuzzy String Matching* (Dewanto dkk, 2009)

- 1) *Fuzzy String Matching* adalah salah satu metode pencarian *string* yang menggunakan proses pendekatan terhadap pola dari *string* yang dicari.
- 2) Melakukan pencarian terhadap *string* yang sama dan juga *string* yang mendekati dengan *string* lain yang terkumpul dalam sebuah penampung atau kamus.
- 3) Kunci dari konsep pencarian ini adalah bagaimana memutuskan bahwa sebuah *string* yang dicari memiliki kesamaan dengan *string* tertampung di kamus, meskipun tidak sama persis dalam susunan karakternya. Untuk memutuskan 'kesamaan' ini dipergunakan sebuah fungsi yang diistilahkan sebagai *similarity function*. Fungsi ini akan bertugas memutuskan *string* hasil pencarian jika ditemukan *string* hasil pendekatan (aproksimasi).

2.4. *Approximate String Matching*

Approximate String Matching merupakan pencocokan *string* dengan dasar kemiripan dari segi penulisannya (jumlah karakter dan susunan karakter), tingkat kemiripan ditentukan dengan jauh tidaknya beda penulisan dua buah *string* yang dibandingkan tersebut (Haryanto, 2011).

Operasi mengubah *string* tersebut dapat berupa mengubah satu huruf ke huruf yang lain, menghapus satu huruf dari *string*, atau memasukkan satu huruf ke dalam *string*. Operasi-operasi ini digunakan untuk menghitung jumlah perbedaan yang diperlukan untuk pertimbangan kecocokan suatu *string* dengan *string* sumber, jumlah perbedaan tersebut diperoleh dari penjumlahan semua pengubahan yang terjadi dari masing-masing operasi.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Operasi untuk mengubah *string* tersebut ada 3 operasi (Adiwidya, 2009),

yaitu :

1) Operasi Penghapusan

Operasi ini dilakukan dengan menghapus karakter pada suatu kata agar menyesuaikan *string* sumber (S) dengan *string* target (T), misalnya kata yang ingin dicari (sumber) adalah opera, namun pada target terdapat operasi. Maka pada karakter s pada lokasi ke-6 dan karakter i pada lokasi ke-7 akan dihapus. Perubahan yang terjadi dapat dilihat pada ilustrasi berikut :

	1	2	3	4	5	6	7
T =	o	p	e	r	a	-	-
S =	o	p	e	r	a	s	i

Gambar 2.1. Operasi Penghapusan
(Sumber : Data Primer,2015)

2) Operasi Penyisipan

Operasi ini dilakukan dengan menyisipkan karakter pada indeks tertentu untuk menyamakan *string* sumber (S) dengan *string* target (T), misalnya S= program dan T= pemrograman. Mekanisme penyisipan dilakukan dengan menyisipkan e pada posisi 2, menyisipkan m pada posisi 3, dan menyisipkan a pada posisi 10 dan menyisipkan n pada posisi 11. Dapat dilihat pada ilustrasi ini :

	1	2	3	4	5	6	7	8	9	10	11
T =	p	e	m	r	o	g	r	a	m	a	n
S =	p			r	o	g	r	a	m		
		e	m						a	n	

Gambar 2.2. Operasi Penyisipan
(Sumber : Data Primer,2015)

3) Operasi Penukaran

Operasi ini dilakukan dengan menukar suatu karakter untuk menyamakan *string* sumber (S) dengan *string target* (T), misalnya S= camera dan T =

1. Diarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Diarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

kamera. *String* S dirubah menjadi T dengan operasi penukaran, dimana c diganti k. Dapat diilustrasikan sebagai berikut :

	1	2	3	4	5	6	7	8
T =	k	o	m	p	u	t	e	r
S =	c	o	m	p	u	t	e	r

Gambar 2.3. Operasi Penukaran
(Sumber : Data Primer,2015)

2.5. Konsep Algoritma

Dalam catatan sejarah, Abu Ja'far Muhammad Ibnu Musa Al-Kwarizmi, penulis buku *Aljabar wal Muqabala* pada abad ke-9, dianggap sebagai pencetus awal algoritma karena didalam bukunya dijelaskan langkah-langkah dalam menyelesaikan berbagai persoalan aritmatika (aljabar). Kemungkinan besar nama algoritma diambil dari nama beliau Al-Kwarizmi yang berubah menjadi *Algorism*, selanjutnya menjadi *Algoritm*.

Menurut Kamus Besar Bahasa Indonesia (Kamus Besar Bahasa Indonesia, 2008) algoritma adalah prosedur sistematis untuk memecahkan masalah matematis di langkah-langkah terbatas. Algoritma adalah urutan langkah-langkah yang dinyatakan dengan jelas dan tidak rancu untuk memecahkan suatu masalah (jika ada pemecahannya) dalam waktu tertentu. Dapat diartikan setiap langkah harus dapat dikerjakan dan mempunyai efek tertentu. Langkah-langkah yang dapat dikerjakan dan mempunyai efek tertentu. Langkah-langkah yang tidak dapat dikerjakan dan tidak menghasilkan efek tertentu, tidak dapat dikatakan algoritma. Efek-efek setiap langkah pada akhirnya akan memecahkan masalah secara keseluruhan (Wahid, 2004).

Definisi algortima dapat juga berarti (Suarga, 2006) :

- 1) Teknik penyusunan langkah-langkah penyelesaian masalah dalam bentuk kalimat dengan jumlah kata terbatas, tetapi tersusun secara logis dan sistematis.
- 2) Suatu prosedur yang jelas untuk menyelesaikan suatu persoalan dengan menggunakan langkah-langkah tertentu dan terbatas jumlahnya.



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Algoritma juga memiliki enam sifat, diantaranya (Suarga, 2006):

- 1) *Input* : suatu algoritma memiliki *input* atau kondisi awal sebelum algoritma dilaksanakan dan bisa berupa nilai-nilai pengubah yang diambil dari himpunan khusus.
- 2) *Output* : suatu algoritma akan menghasilkan *output* setelah dilaksanakan, atau algoritma akan mengubah kondisi awal menjadi kondisi akhir, dimana nilai *output* diperoleh dari nilai *input* yang telah diproses melalui algoritma.
- 3) *Definiteness* : langkah-langkah yang dituliskan dalam algoritma terdefinisi dengan jelas sehingga mudah dilaksanakan oleh pengguna algoritma.
- 4) *Finiteness* : suatu algoritma harus memberi kondisi akhir atau output setelah melakukan sejumlah langkah yang terbatas jumlahnya untuk setiap kondisi awal atau input yang diberikan.
- 5) *Effectiveness* : setiap langkah dalam algoritma bisa dilaksanakan dalam suatu selang waktu tertentu sehingga pada akhirnya member solusi sesuai yang diharapkan.
- 6) *Generality* : langkah-langkah algoritma berlaku untuk setiap himpunan input yang sesuai dengan persoalan yang akan diberikan, tidak hanya untuk himpunan tertentu.

2.6. Algoritma Levenshtein Distance

Algoritma *Levenshtein distance* ditemukan oleh ilmuwan asal Rusia bernama Vladimir *Levenshtein* pada tahun 1963 (Junaedy S., 2014), algoritma ini juga disebut dengan algoritma *Edit Distance* (Husain, 2013). Perhitungan *edit distance* didapatkan dari matriks yang digunakan untuk menghitung jumlah perbedaan *string* antara dua *string*, sebagai contoh hasil penggunaan algoritma ini, *string* “komputer” dan “computer” memiliki *distance* 1 karena hanya perlu dilakukan satu operasi saja untuk mengubah satu *string* ke *string* yang lain. Dalam kasus dua *string* di atas, *string* “computer” dapat menjadi “komputer” hanya dengan melakukan satu penukaran karakter “c” menjadi “k” (Andhika, 2010).



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

Algoritma *Levenshtein distance* digunakan secara luas dalam berbagai bidang, misalnya mesin pencari, pengecek ejaan (*spell checking*), pengenalan pembicaraan (*speech recognition*), pengucapan dialek, analisis DNA, pendeteksi pemalsuan, dan lain-lain. Algoritma ini menghitung jumlah operasi *string* paling sedikit yang diperlukan untuk mentransformasikan suatu *string* menjadi *string* yang lain (Adiwidya, 2009). Algoritma *Levenshtein distance* bekerja dengan menghitung jumlah minimum pentransformasian suatu *string* menjadi *string* lain yang meliputi penghapusan, penyisipan, dan penukaran (Husain, 2013).

Selisih perbedaan antar *string* dapat diperoleh dengan memeriksa apakah suatu *string* sumber sesuai dengan *string* target. Nilai selisih perbedaan ini disebut juga *edit distance* atau jarak *Levenshtein*. Jarak *Levenshtein* antar *string* “s” dan *string* “t” tersebut adalah fungsi D yang memetakan (s,t) ke suatu bilangan *real nonnegatif*, sebagai contoh diberikan dua buah *string* $s = s(1)s(2)s(3), \dots, s(m)$ dan $t = t(1)t(2)t(3), \dots, t(n)$ dengan $|s| = m$ dan $|t| = n$ sepanjang alfabet V berukuran r sehingga “s” dan “t” anggota dari V^* . $s(j)$ adalah karakter pada posisi ke-j pada *string* “s” dan $t(i)$ adalah karakter pada posisi ke-i pada *string* “t”. Sehingga jarak *Levenshtein* dapat didefinisikan sebagai (Harahap, 2013).

$$D(s, t) = d(s_1, t_1) + d(s_2, t_2) + \dots + d(s_l, t_l)$$

$$D(s, t) = \sum_{i=1}^l d(s_i, t_i) \quad (2.1)$$

dimana : $s_i, t_i \in V$ untuk $i = 1, 2, \dots, l$

$d(s_i, t_i) = 0$ jika $s_i = t_i$ dan

$d(s_i, t_i) = 1$ jika $s_i \neq t_i$

$D(s, t)$ adalah banyaknya operasi minimum dari operasi penghapusan, penyisipan dan penukaran untuk menyamakan *string* s dan t. Pada implementasi pencocokan antar *string*, ketiga operasi tersebut dapat dilakukan sekaligus untuk menyamakan *string* sumber dengan *string* target seperti pada contoh berikut ini. Jika diberikan *string* sumber (s) = “algoritma” dan (t) = “ program” merupakan *string* target, dengan $|s| = 9$, $|t| = 7$, maka operasi pencocokan *string* dapat diilustrasikan sebagai berikut :

	1	2	3	4	5	6	7	8	9
T =	p	r	o	g	r	a	m	-	-
S =	a	l	g	o	r	i	t	m	a
	a	l	g	o		i	t		

Gambar 2.4. Ilustrasi perhitungan *Levenshtein*
(Sumber : Data Primer,2015)

Pada contoh di atas terlihat bahwa proses penukaran karakter “a” pada indeks ke-1, “l” pada indeks ke-2, “g” pada indeks ke-3, “o” pada indeks ke-4, “i” pada indeks ke-6, “t” pada indeks ke-7, dan operasi penghapusan karakter “m” pada indeks ke-8 dan “a” pada indeks ke-9. Maka jarak *Levenshtein* antara sumber (s) dan target (t) adalah sebagai berikut :

$$\begin{aligned}
 D(s,t) &= \sum_{i=1}^9 d(s_i, t_i) \\
 &= d(s_1, t_1) + d(s_2, t_2) + d(s_3, t_3) + d(s_4, t_4) + d(s_5, t_5) + d(s_6, t_6) + \\
 &\quad d(s_7, t_7) + d(s_8, t_8) + d(s_9, t_9) \\
 &= d(a, p) + d(r, l) + d(o, g) + d(g, o) + d(r, r) + d(i, a) + d(t, m) + \\
 &\quad d(m, -) + d(a, -) + \\
 &= 1+1+1+1+0+1+1+1+1 \\
 &= 8
 \end{aligned}$$

Sehingga jarak *Levenshtein* antara string s = “algoritma” dan t = “program” adalah D(s, t) = 8.

2.7. Model Analisis dan Perancangan Berbasis Objek (*Object-Oriented Analysis and Design / OOAD*)

Metode OOAD melakukan pendekatan terhadap masalah dari perspektif obyek, tidak pada perspektif fungsional seperti pada pemrograman terstruktur. Akhir-akhir ini penggunaan OOAD meningkat dibandingkan dengan penggunaan metode pengembangan software dengan metode tradisional. Hal tersebut untuk memenuhi peningkatan kebutuhan akan pendekatan berorientasi obyek pada aplikasi bisnis. Tahap perancangan dimulai dengan hasil keluaran yang dihasilkan tahapan analisis dan aktivitas yang dilakukan adalah secara perlahan bergeser tekanannya dari domain aplikasi/persoalan/masalah menuju ke domain komputasi.



OOAD adalah metode pengembangan sistem yang lebih menekankan objek dibandingkan dengan data atau proses. Metodologi pengembangan sistem berorientasi objek mempunyai tiga karakteristik utama, yaitu:

1) *Encapsulation*

Encapsulation merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya.

2) *Inheritance*

Inheritance adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung. Atribut dan metode dari objek dari objek induk diturunkan kepada anak objek,

3) *Polymorphism*

Polimorfisme yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda. Kemampuan objek-objek yang berbeda untuk melakukan metode yang pantas dalam merespon *message* yang sama.

2.8. *Unified Modeling Language (UML)*

UML adalah sebuah alat bantu yang sangat handal di dunia pengembangan sistem berorientasi objek. Hal ini di sebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku, mudah di mengerti serta dilengkapi dengan mekanisme efektif untuk berbagi dan mengkomunikasikan rancangan mereka yang lain (Nugroho ,2005)

UML adalah bahasa untuk memvisualisasi, menspesifikasikan dan mendokumentasikan artifak-artifak sistem perangkat lunak. UML merupakan sistem notasi yang membantu pemodelan sistem menggunakan konsep berorientasi objek. Dalam proyek pengembangan sistem apapun, fokus utama dalam analisis dan perancangan adalah model. Hal ini berlaku umum tidak hanya untuk perangkat lunak. Dengan model kita dapat mempresentasikan sesuatu (Nugroho ,2005) karena:

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

- 1) Model mudah dan cepat dibuat
- 2) Model bisa digunakan sebagai simulasi untuk mempelajari lebih detail tentang sesuatu
- 3) Model bisa dikembangkan sejalan dengan pemahaman kita tentang sesuatu
- 4) Konsep bisa mewakili sesuatu yang nyata maupun tidak nyata
- 5) Konsep-konsep yang diterapkan UML adalah suatu model berisikan informasi mengenai sistem(domain) model-model berisi elemen-elemen model seperti kelas, simpul-simpul, paket-paket, dan sebagainya. Satu diagram menunjukkan satu pandangan tertentu dari model.

Tujuan utama perancangan UML adalah:

- 1) Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti.
- 2) Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep inti.
- 3) Mendukung spesifikasi independen bahasa pemograman dan proses pengembangan tertentu.
- 4) Menyediakan basis formal untuk pemahaman bahasa pemodelan.
- 5) Mendorong pertumbuhan pasar kakas berorientasi objek.
- 6) Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, *framework* dan *pattern*.

2.8.1. Use Case Diagram

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendiskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Tujuan utama pemodelan *use case* adalah :

- 1) Memutuskan dan mendiskripsikan kebutuhan-kebutuhan fungsional sistem.



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

- 2) Memberikan deskripsi jelas dan konsisten dari apa yang seharusnya dilakukan, sehingga model *use case* digunakan diseluruh proses pengembangan untuk komunikasi dan menyediakan basis untuk pemodelan berikutnya yang mengacu sistem harus memberikan fungsionalitas yang dimodelkan pada *use case*.
- 3) Menyediakan basis untuk melakukan pengujian sistem yang memverifikasi sistem. Menguji apakah sistem telah memberikan fungsionalitas yang diminta.
- 4) Menyediakan kemampuan melacak kebutuhan fungsionalitas menjadi kelas-kelas dan operasi-operasi aktual di sistem. Untuk menyederhanakan perubahan dan ekstensi ke sistem dengan mengubah model *use case* dan kemudian melacak *use case* yang dipengaruhi ke perancangan dan implementasi sistem.

Syarat penamaan *use case* adalah nama didefinisikan sesederhana mungkin dan dapat dipahami, ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

- 1) Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan di buat diluar sistem informasi yang akan dibuat sendiri, jadi walaupun simbol dari aktor adalah gambar orang tapi aktor belum tentu orang.
- 2) *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Dalam *use case* juga dikenal dengan hubungan antar *use case* yang merupakan generalisasi antara *use case* yaitu:

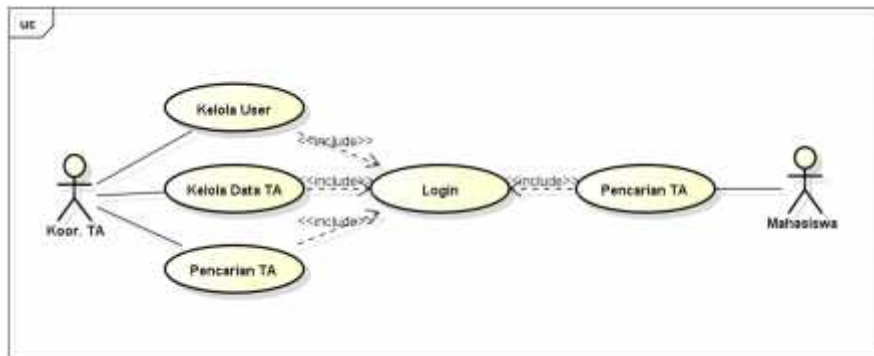
- 1) *Include*, yaitu perilaku *use case* merupakan bagian dari *use case* yang lain.
- 2) *Extend*, yaitu perilaku *use case* memperluas perilaku *use case* yang lain.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

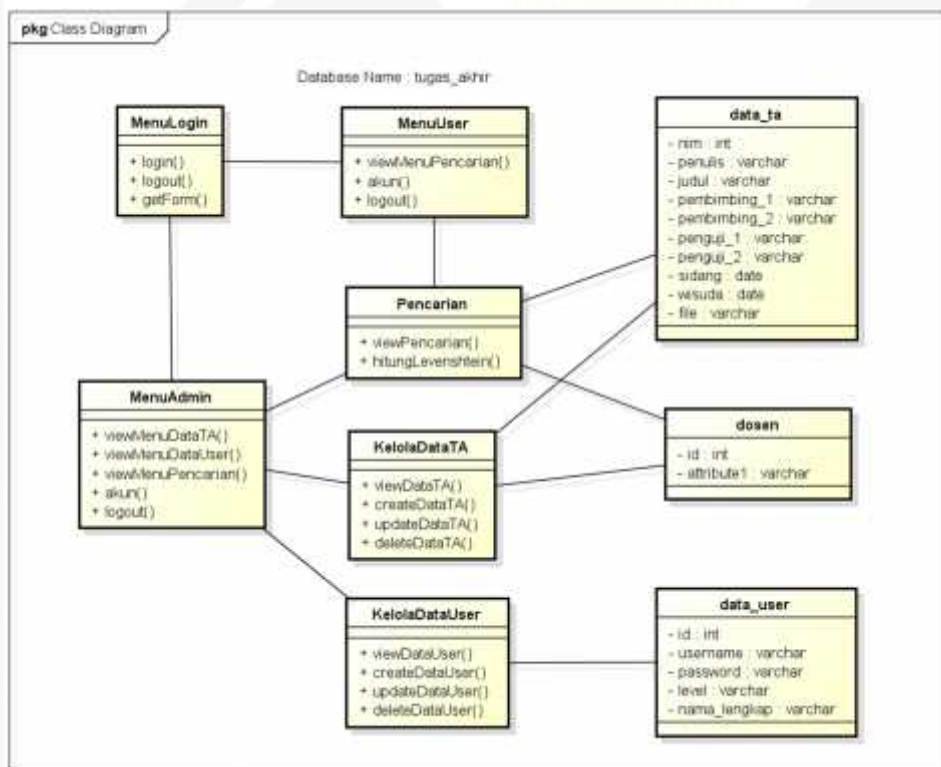


Gambar 2.5. Contoh *Use case* Diagram
(Sumber : Data Primer, 2015)

2.8.2. Class Diagram

Diagram kelas atau *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.

- 1) Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
- 2) Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.



Gambar 2.6. Contoh *Class Diagram*
(Sumber : Data Primer, 2015)

2.8.3. Object Diagram

Object Diagram menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggungjawabkan.

Untuk apa mendefinisikan sebuah kelas sedangkan pada jalannya sistem, objeknya tidak pernah dipakai. Hubungan link pada diagram objek merupakan hubungan memakai dan dipakai dimana dua buah objek akan dihubungkan oleh link jika ada objek yang dipakai oleh objek lainnya.



Gambar 2.7. Contoh *Object Diagram*
(Sumber : Data Primer, 2015)

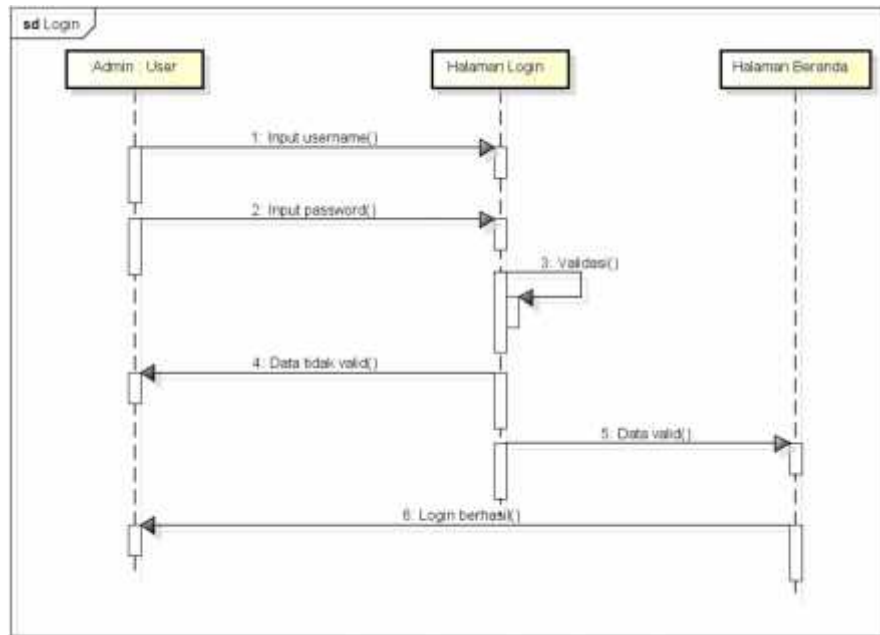
2.8.4. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



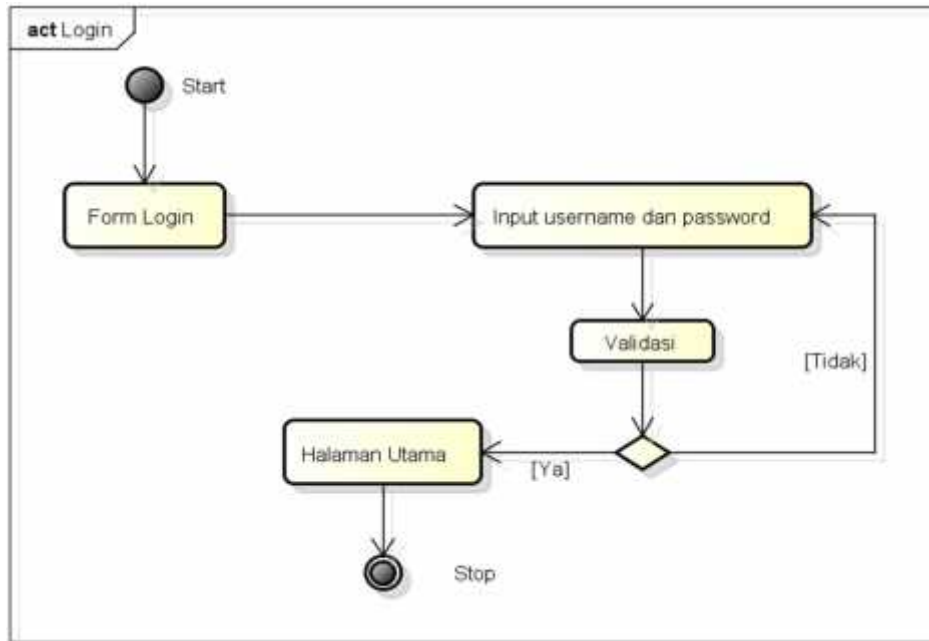
Gambar 2.8. Contoh *Sequence Diagram*
(Sumber : Data Primer, 2015)

2.8.5. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- 1) Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- 2) Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- 3) Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.



Gambar 2.9. Contoh Activity Diagram
(Sumber : Data Primer, 2015)

2.9. Aplikasi Web

Web merupakan salah satu sumber daya internet yang berkembang pesat. Pendistribusian informasi web dilakukan melalui pendekatan *hyperlink*, yang memungkinkan suatu teks, gambar, ataupun objek yang lain menjadi acuan untuk membuka halaman-halaman yang lain. Melalui pendekatan ini, seseorang dapat memperoleh informasi dengan beranjak dari satu halaman ke halaman lain (Kadir, 2005).

Pada awalnya aplikasi web dibangun hanya dengan menggunakan bahasa yang disebut HTML (*Hyper Text Markup Language*) dan protokol yang digunakan dinamakan HTTP (*Hyper Text Transfer Protocol*). Pada perkembangan berikutnya, sejumlah *script* dan objek yang dikembangkan untuk memperluas kemampuan HTML. Pada saat ini, banyak *script* seperti itu antara lain PHP dan ASP, sedangkan contoh yang berupa objek antara lain adalah *applet* (*Java*) (Kadir, 2005).

Sehingga awalnya web merupakan ruang informasi dalam internet, dengan menggunakan teknologi *hypertext*, pemakai dituntun menemukan informasi dengan mengikuti *link* yang tersedia dalam dokumen web yang ditampilkan dalam

web browser. Situs/web dapat dikategorikan menjadi dua (Kustianingsih dkk, 2011) yaitu :

1) Web Statis

Web statis adalah web yang menampilkan informasi-informasi yang sifatnya statis (tetap). Disebut statis karena pengguna tidak dapat berinteraksi dengan web tersebut. Untuk melihat suatu web bersifat statis atau dinamis dapat dilihat dari tampilannya. Jika suatu web hanya berhubungan dengan halaman web lain dan berisi suatu informasi yang teteap maka web tersebut dinamakan statis.

2) Web Dinamis

Web dinamis adalah web yang menampilkan informasi serta dapat berinteraksi dengan pengguna. Web dinamis memungkinkan pengguna untuk berinteraksi menggunakan form sehingga dapat mengolah informasi yang ditampilkan. Web dinamis bersifat interaktif, tidak kaku dan terlihat lebih indah.

Sehingga dapat diketahui bahwa kelebihan dari aplikasi web adalah dapat diakses kapan pun dan dimana pun selama ada internet. Dan dapat diakses dengan menggunakan *web browser* yang tersedia pada perangkat komputer atau *mobile*. Sedangkan untuk kekurangan yang ada aplikasi web adalah antarmuka yang dibuat terbatas sesuai spesifikasi standar untuk membuat dokumen web dan keterbatasan kemampuan *web browser* untuk menampilkannya. Dan terbatasnya kecepatan internet mungkin membuat respon aplikasi menjadi lambat (Kustianingsih dkk, 2011).

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



2.10. Pengenalan PHP (*Hypertext Preprocessor*)

PHP merupakan bahasa pemrograman yang di desain untuk web. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. jika dilihat dari versi pertamanya bahwa PHP terdiri dari sekumpulan *script* PERL yang digunakan untuk mengolah data *form* dan *website*. Pada awalnya PHP bernama FI (*Form Interpreted*). Nama PHP ada setelah Rasmus melepaskan *source code*-nya dan sejak itulah PHP bersifat *open source*.

Pada November 1997, PHP versi 2.0 berhasil dirilis dengan mengimplementasikannya dalam bahasa C. Pada tahun itu juga sebuah perusahaan bernama Zend menulis ulang *interpreter* PHP menjadi lebih bersih, lebih baik dan lebih cepat. Pada bulan Juni 1998 Zend kembali merilis *interpreter* baru untuk PHP dan meresmikan nama rilis tersebut menjadi PHP 3.0. Dilanjutkan pada pertengahan tahun 1999, Zend merilis kembali *interpreter* PHP baru dengan nama PHP 4.0. PHP 4.0 ini memiliki kemampuan untuk membangun aplikasi *website* yang kompleks, namun tetap stabil dalam kecepatan proses dan stabilitas yang tinggi, sehingga menjadi populer dikalangan *programmer website*.

Mulai pada tahun 2004 bulan Juni, PHP 5.0 kembali di rilis oleh Zend. Versi adalah versi mutakhir dari PHP. Pada versi ini juga dikenalkan pada model pemrograman berorientasi objek baru untuk menjawab perkembangan bahasa pemrograman kearah pemrograman berbasis objek. Saat ini versi terbaru dari PHP adalah PHP 5.6.6 yang dirilis pada 19 Februari 2015.

2.11. MySQL

Sejarah MySQL merupakan hasil buah pikir dari Michael Monty Widenius, David Axmark, dan Alian Larson pada tahun 1995. Mereka bertiga kemudian mendirikan perusahaan bernama MySQL AB di Swedia.

MySQL menggunakan bahasa standar SQL (*Structure Query Language*) sebagai bahasa interaktif dalam mengelola data. Perintah SQL sering juga disebut dengan *query*. Karena menggunakan bahasa standar yang sama, maka tidak akan menjadi kendala besar apabila berganti dengan *database* selain MySQL dengan sedikit perbedaan sintaks antara paket program SQL tersebut (Arbie, 2004).



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Database MySQL merupakan sistem manajemen basis data SQL yang sangat populer dan bersifat *open source*. MySQL dapat didefinisikan sebagai sistem manajemen *database*. *Database* sendiri merupakan struktur penyimpanan data. Untuk menambah, mengakses dan memproses data yang disimpan dalam sebuah *database* komputer, diperlukan sistem manajemen *database* seperti MySQL Server. Selain itu MySQL dapat dikatakan sebagai basis data terhubung (*Relational Database Management Sistem (RDBMS)*). *Database* terhubung menyimpan data pada tabel-tabel terpisah. Hal tersebut akan menambah kecepatan dan fleksibilitasnya (Kustianingsih dkk, 2011).

MySQL memiliki beberapa keistimewaan (Agus, 2010), antara lain:

- 1) Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga dan masih banyak lagi.
- 2) Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
- 3) *Multi-user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
- 4) *Performance tuning*, MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
- 5) Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed/unsigned integer, float, double, char, text, date, timestamp* dan lain-lain.
- 6) Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
- 7) Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
- 8) Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

- 9) Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan *protokol* TCP/IP, *Unix socket* (UNIX), atau *Named Pipes* (NT).
- 10) Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.
- 11) Antar Muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
- 12) Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
- 13) Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.12. Bootstrap

Bootstrap diciptakan oleh dua orang programmer di Twitter, yaitu Mark Otto dan Jacob Thornton pada tahun 2011. Pada saat itu para programmer di twitter menggunakan berbagai macam *tool* dan *library* yang mereka kenal dan suka untuk melaksanakan pekerjaan mereka, sehingga tidak ada standarisasi dan akibatnya sulit untuk dikelola sehingga Mark Otto dan Jacob Thornton tergerak untuk menciptakan satu *tool* ataupun *framework* yang dapat digunakan bersama di lingkungan internal twitter

Bootstrap merupakan *Framework* ataupun *Tools* untuk membuat aplikasi web ataupun situs web responsif secara cepat, mudah dan gratis. Bootstrap terdiri dari CSS, HTML untuk menghasilkan *Grid*, *Layout*, *Typography*, *Table*, *Form*, *Navigation*, dan lain-lain. Di dalam Bootstrap juga sudah terdapat *jQuery plugins* untuk menghasilkan komponen UI yang cantik seperti *Transitions*, *Modal*,



Hak Cipta Dilindungi Undang-Undang

1. Diarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

2. Diarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Dropdown, Scrollspy, Tooltip, Tab, Popover, Alert, Button, Carousel dan lain-lain.

Bootstrap dibangun dengan menggunakan *vanilla* CSS, akan tetapi *source code*-nya memanfaatkan dua CSS *preprocessor* yang paling populer yaitu Less dan Sass. Dengan bantuan *Bootstrap*, dapat dibuat website yang responsif dengan cepat dan mudah dan dapat berjalan pada *browser* baik itu di ponsel, tablet dan juga *desktop* (Bootstrap, 2015).

Walaupun nama resminya hanyalah Bootstrap, namun terkenal di kalangan *developer* sebagai Twitter Bootstrap. Saat laporan ini ditulis, versi terbaru dari bootstrap adalah versi 3.3.5.

2.13. Profil Lembaga

2.13.1. Sejarah Singkat

Program Studi Sistem Informasi mulai beroperasi pada bulan Juli tahun 2002 yang direkomendasi Gubernur Riau pada 7 Oktober 2003 berdasarkan SK Gubernur No. Kpts. 521/X/2002 Tanggal 24 Oktober 2002, rekomendasi DPRD Propinsi Riau No. 12/KPTS/PIMP/DPRD/2002 Tanggal 25 September 2002, dan rekomendasi dari Menteri Agama RI kepada Menteri Pendidikan Nasional dengan No : MA/266/2002 Tanggal 19 Juli 2002 serta PERPRES No. 2 Tanggal 11 Februari 2005 tentang perubahan status IAIN SUSQA menjadi UIN SUSKA RIAU, kemudian disusul dengan dikeluarkannya secara resmi izin pendirian Program Studi Sistem Informasi dengan SK Pendirian DJ.II/26/2006 (Jurusan Sistem Informasi, 2015).

2.13.2. Visi dan Misi Program Studi Sistem Informasi

1) Visi

Rumusan visi Program Studi Sistem Informasi sejalan dengan visi dari Fakultas Sains dan Teknologi.

a. Visi Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau:

Terwujudnya Fakultas Sains dan Teknologi sebagai fakultas yang menyelenggarakan pendidikan integrasi sains dan Islam, penelitian, dan pengabdian kepada masyarakat, yang berkualitas pada tahun 2033.



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

b. Visi Program Studi Sistem Informasi :

Menjadi program studi yang menghasilkan lulusan Sistem Informasi yang handal dalam pendidikan, penelitian dan pengabdian masyarakat dengan integrasi nilai-nilai keislaman pada tahun 2033.

Penyusunan dan penetapan visi Program Studi Sistem Informasi dibuat konsisten dengan visi fakultas dan universitas pada integrasi sains dan Islam. Visi ini sangat jelas karena fokus pada Tri Dharma Perguruan Tinggi yaitu menghasilkan lulusan melalui proses pendidikan, penyelenggaraan penelitian dan pengabdian kepada masyarakat, sehingga lulusan yang dihasilkan adalah sarjana Sistem Informasi yang handal dalam bidang Sistem Informasi dengan nilai-nilai ke-Islaman. Upaya mencapai kriteria lulusan tersebut sudah tergambar secara kongkrit di dalam kurikulum.

2) Misi

a. Misi Fakultas Sains dan Teknologi UIN Suska Riau:

- 1) Menyelenggarakan pendidikan dan pengajaran berkualitas yang mengintegrasikan antara sains, teknologi, dan Islam.
- 2) Menyelenggarakan penelitian yang berkualitas di bidang sains dan teknologi, yang mampu mendukung pada kegiatan pelayanan kepada masyarakat yang berkualitas.
- 3) Menyelenggarakan pengabdian dan pelayanan kepada masyarakat yang berkualitas, sehingga mampu memberikan pendampingan kepada masyarakat, terutama pada penyelesaian permasalahan keislaman.

b. Misi Program Studi Sistem Informasi adalah :

- 1) Melaksanakan pendidikan dan pengajaran yang handal dalam bidang teknologi informasi, manajemen sistem informasi dan bisnis informasi yang terintegrasi dengan nilai-nilai keislaman.
- 2) Melaksanakan penelitian yang inovatif dan berkualitas dibidang sistem informasi.



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

- 3) Melaksanakan pengabdian, pendampingan dan pemberdayaan masyarakat dalam bidang teknologi informasi dan komunikasi.

Misi Program Studi Sistem Informasi diturunkan dan diselaraskan dengan misi Fakultas Sains dan Teknologi dan Universitas Islam Negeri Sultan Syarif Kasim Riau yang pada prinsipnya mencakup tiga unsur utama Tri Dharma Perguruan Tinggi yang terdiri dari pendidikan, penelitian, dan pengabdian kepada masyarakat.

2.13.3. Tujuan Program Studi Sistem Informasi

Tujuan utama dari Program Studi Sistem Informasi yang akan dicapai pada tahun 2033 adalah :

- 1) Menghasilkan lulusan yang memiliki kemampuan akademis dan praktis di bidang sistem informasi yang berwawasan ke-Islaman.
- 2) Menghasilkan penelitian dan karya ilmiah yang dipublikasikan dalam skala nasional dan internasional.
- 3) Menghasilkan produk sistem informasi yang bermanfaat bagi masyarakat, institusi dan *stakeholder*.
- 4) Menghasilkan kemitraan yang strategis dengan lembaga pemerintah maupun non pemerintah tingkat lokal, nasional dan internasional.

Tujuan Program Studi Sistem Informasi dibuat merujuk pada tujuan Fakultas dan tujuan Universitas. Secara prinsip, tujuan Fakultas dan Universitas mencakup Tri Dharma Perguruan Tinggi. Tujuan Program Studi Sistem Informasi merupakan turunan dari Misi Program Studi Sistem Informasi yang juga mencakup Tri Dharma Perguruan Tinggi.



2.13.4. Struktur Organisasi Jurusan Sistem Informasi

Berikut merupakan struktur Jurusan Sistem Informasi :

© Hak cipta milik UIN Suska Riau

State Islamic University of Sultan Syarif Kasim Riau

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

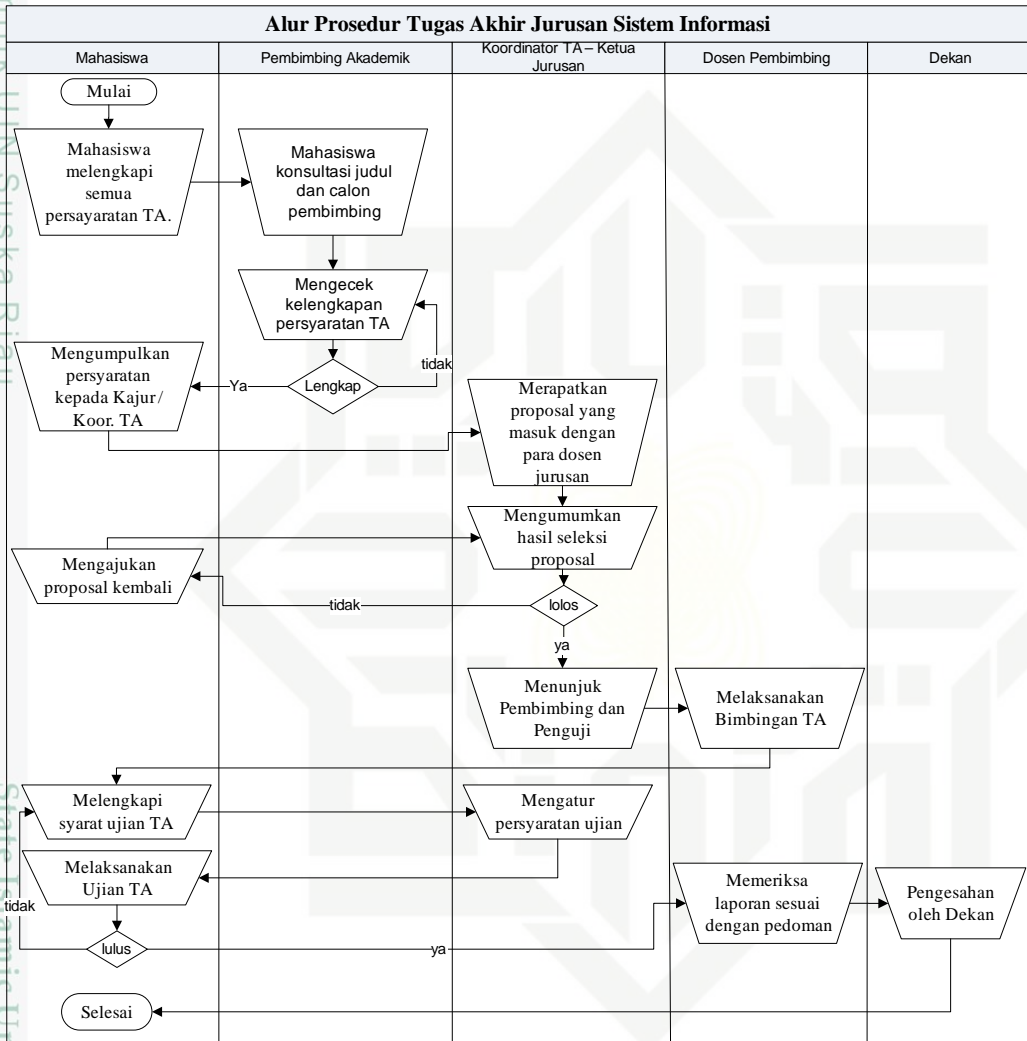
Gambar 2.10. Struktur Organisasi Jurusan Sistem Informasi
(Sumber : Dokumentasi Akreditasi Jurusan SI, 2015)

2.14. Prosedur Tugas Akhir Jurusan Sistem Informasi

Berikut beberapa beberapa prosedur tugas akhir jurusan Sistem Informasi :

2.14.1. Alur Prosedur Tugas Akhir

Dibawah ini merupakan gambaran alur prosedur tugas akhir pada jurusan Sistem Informasi :



Gambar 2.11. Alur Prosedur Tugas Akhir Jurusan Sistem Informasi
(Sumber : Data Primer, 2015)

2.14.2. Persyaratan Pengajuan Sidang Tugas Akhir

Syarat-syarat pengajuan sidang tugas akhir adalah sebagai berikut:

- 1) Mahasiswa harus terdaftar dan mencantumkan matakuliah Tugas Akhir di KRS

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

- 2) Surat pengajuan sidang TA (Form TA 04) yang diisi lengkap dan disetujui oleh semua pembimbing (jika lebih dari satu) dan disetujui oleh Ketua Jurusan
- 3) Tidak dibenarkan terdapat nilai E didalam transkrip sementara dan tidak terdapat nilai D pada Mata Kuliah Keahlian (MKK) dan nilai D tidak lebih 10% dari total SKS. Transkrip nilai telah diperiksa oleh Kasubag Akademik, dan disahkan oleh Pembantu Dekan I.
- 4) Laporan TA harus sudah di setujui oleh semua pembimbing (jika lebih dari satu) (Form B3) dan laporan harus sesuai dan mengikuti format penulisan TA yang sudah ditetapkan
- 5) Kartu bimbingan TA (Form B3) yang telah diisi sepuluh kali selama proses bimbingan dari bab satu hingga bab kesimpulan.
- 6) Asli KRS semester terakhir.
- 7) Asli Transkrip nilai (khusus TA) yang ditandatangani Pembantu Dekan I.
- 8) Asli Sertifikat Kompetensi Basic Kesilaman
- 9) Asli Sertifikat Penanaman Nilai Dasar Keislaman (PNDK)
- 10) Asli Sertifikat Pembinaan Akademik dan Kemahasiswaan (PAK).
- 11) Surat bebas Laboratorium yang ditanda tangani Kepala Laboratorium Jurusan (Form B2)
- 12) Mengisi formulir data mahasiswa dan pasfoto ukuran 4x6 cm dua lembar
- 13) Menyerahkan *fotocopy* laporan TA sebanyak pembimbing dan penguji yang ditetapkan jurusan masing-masing dan telah diserahkan tiga hari sebelum hari pelaksanaan sidang TA (teknis distribusi diserahkan kepada pihak jurusan masing-masing). Laporan harus dalam kondisi lengkap, mulai dari abstrak, daftar isi hingga riwayat hidup, dan masing-masing dimasukkan kedalam map.
- 14) Telah memasukkan minimal 10 judul buku dan/atau jurnal terbaru didalam daftar pustaka. Menyerahkan poster hasil penelitian TA yang telah di paraf pembimbing (sisi belakang poster) dengan



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

menggunakan kertas A2 (*double folio*) dengan format tulisan bebas dan dicetak Menyerahkan makalah/*paper* kepada Ketua Jurusan sebelum mendapatkan SKL (Surat Keterangan Lulus) dengan format mengikuti ketentuan Jurnal Sitekin.

2.14.3. Tata Cara Persetujuan Laporan Tugas Akhir

Dibawah ini merupakan tata cara persetujuan laporan tugas akhir :

- 1) Laporan hasil sidang TA yang sudah dinyatakan lulus akan diperiksa kembali oleh pembimbing sebelum ke penguji berdasarkan catatan didalam berita acara sidang TA.
- 2) Laporan yang sudah disetujui pembimbing dan penguji selanjutnya akan diperiksa Ketua Jurusan dengan penekanan pada standarisasi penulisan laporan secara lengkap berdasarkan Pedoman Sistem dan Prosedur Tugas Akhir.
- 3) Setelah diperbaiki dan disetujui Ketua Jurusan, selanjutnya laporan dapat diserahkan kepada Dekan untuk disahkan dan ditandatangani.
- 4) Memenuhi kelengkapan persyaratan wisuda, maka setiap mahasiswa wajib menyerahkan minimal dua laporan tugas akhir dalam bentuk *hardcopy* sesuai ketentuan (masing-masing untuk fakultas dan universitas) dan *softcopy* dalam bentuk CD yang berisi : laporan TA dengan format dokumen dan *pdf*, *source code* serta poster ukuran A3

2.15. Penelitian Terdahulu

Penggunaan metode *Fuzzy String Matching* dan atau algoritma *Levenshtein distance* ada didalam beberapa jurnal dan sumber pendukung, seperti:

- 1) Dalam jurnal Pelita Informatika Budi Dharma, vol. VII, hal 161-164, oleh Richard Junedy S. Pada tahun 2014 dengan judul Perancangan Aplikasi Deteksi Kemiripan Isi Dokumen Teks Dengan Menggunakan Metode *Levenshtein distance*, seperti abstrak dibawah ini.

Teknologi informasi yang berkembang pesat membawa dampak positif dan negatif bagi kehidupan Salah satu dampak negatif yang ditimbulkan adalah plagiarisme. Plagiarisme adalah tindakan menjiplak karya orang lain dan mengakui sebagai hasil karya



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

pribadinya. Oleh karena itu pendeteksian plagiarisme perlu dilakukan untuk mengurangi penjiplakan terhadap hasil karya orang lain. Algoritma *Levenshtein distance* adalah salah satu algoritma pencocokan *string* yang dapat digunakan untuk mendeteksi plagiarisme. Skripsi ini bertujuan untuk mendeteksi kemiripan dokumen teks menggunakan algoritma *Levenshtein distance* sehingga dapat digunakan untuk membantu menentukan plagiarisme. Untuk meningkatkan kerja algoritma *Levenshtein distance* maka sebelumnya dilakukan *preprocessing* terhadap dokumen teks yang akan diproses. Dari ujicoba yang dilakukan, algoritma *Levenshtein distance* mampu mendeteksi kemiripan dokumen teks, ini dibuktikan dengan nilai *similarity* yang tinggi untuk dokumen yang plagiat. *Preprocessing* terbukti dapat meningkatkan kinerja algoritma *Levenshtein distance* meskipun menambah waktu untuk pemrosesan.

- 2) Dalam jurnal Konferensi Nasional Sistem dan Informatika 2011, Hal 76-82, oleh Edy Victor Haryanto pada tahun 2011 dengan judul Rancang Bangun *Prototype* Mesin Pencari *String* Menggunakan Metode *Fuzzy String Matching*, seperti abstrak dibawah ini.

Development of computer technology causes the information created more easily. Information can be obtained from anywhere and anytime. The form can be either written or oral. The amount of information often makes people confused. This is because humans have limited capabilities, accuracy, and speed in conducting the search process of the information contained in a document. To resolve this problem, a search method comparing the string contained in a document is used. It is known as the search method that uses a Fuzzy String Matching algorithm or the Knuth-Morris- Pratt search engine. Implementing the method, the searches can be done easier and more accurate.

- 3) Dalam jurnal oleh Benisius pada tahun 2010 dengan judul Sistem Pengkoreksian Kata Kunci Dengan Menggunakan Metode

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Levenshtein distance (Studi Kasus Pada Website Universitas Halmahera), seperti abstrak dibawah ini.

Sejumlah penelitian terhadap mesin pencari (*search engine*) menyimpulkan bahwa rata-rata kesalahan pengejaan kata kunci yang dilakukan pengguna cukup tinggi. Oleh karenanya diperlukan suatu sistem yang dapat melakukan pengkoreksian kata kunci pada aplikasi pencarian kata berbasis web. Penelitian ini bertujuan mengimplementasikan salah satu metode *approximate String Matching*, yakni *Levenshtein distance* dalam mengatasi permasalahan tersebut. Pembahasan mengenai proses penghitungan jarak antar *string*, proses kerja mesin pencari berbasis web yang meliputi proses *crawling*, *indexing* sampai dengan menampilkan hasil pencarian beserta saran kata kunci adalah fokus pada penelitian ini. Pengujian dilakukan dengan menggunakan beberapa jenis inputan kata kunci serta mengamati bagaimana hasil pencarian dan saran kata kunci yang dihasilkan sistem diperoleh.

- 4) Dalam penelitian yang dilakukan oleh Bernardino Madaharsa Dito Adiwidya pada tahun 2009 dengan judul *Algoritma Levenshtein Dalam Pendekatan Approximate String Matching*, seperti abstrak dibawah ini.

Algoritma Levenshtein merupakan algoritma yang digunakan untuk mencari jumlah operasi *string* yang paling sedikit untuk mentransformasikan suatu *string* menjadi *string* yang lain. Algoritma ini digunakan dalam pencarian *string* dengan pendekatan perkiraan (*Approximate String Matching*). Dengan pendekatan perkiraan ini, pencarian *string* target menjadi tidak harus sama persis dengan yang ada di dalam *string* sumber. Tentu saja pendekatan tersebut menjadi bahan analisis untuk pengaplikasiannya dalam DNA, pendeteksi pemalsuan, dan lain-lain. Ada tiga buah operasi *string* yang digunakan yaitu operasi penghapusan, penyisipan, dan penggantian. Jumlah operasi minimal dari ketiga operasi tersebut terhadap dua buah *string*



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

disebut dengan *Levenshtein distance* atau *edit distance*. Dengan algoritma ini, tidak perlu dilakukan seluruh percobaan kemungkinan penghapusan, penyisipan, atau penggantian terhadap *string*. Oleh karena itu, algoritma ini dapat digolongkan sebagai program dinamis. Nilai *edit distance* dapat diperoleh hanya dengan menyediakan matriks *cost* yang ukurannya tergantung panjang kedua *string*.

