

**PENERAPAN METODE BOYER MOORE PADA APLIKASI KAMUS
ISTILAH TEKNOLOGI INFORMASI BERBASIS WEB
(STUDI KASUS: PT. EREFKA TIGA PILAR UTAMA)**

SKRIPSI

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar
Sarjana Komputer (S.Kom)**



Disusun Oleh:

RIKZA MUNTAZARI

1112091000104

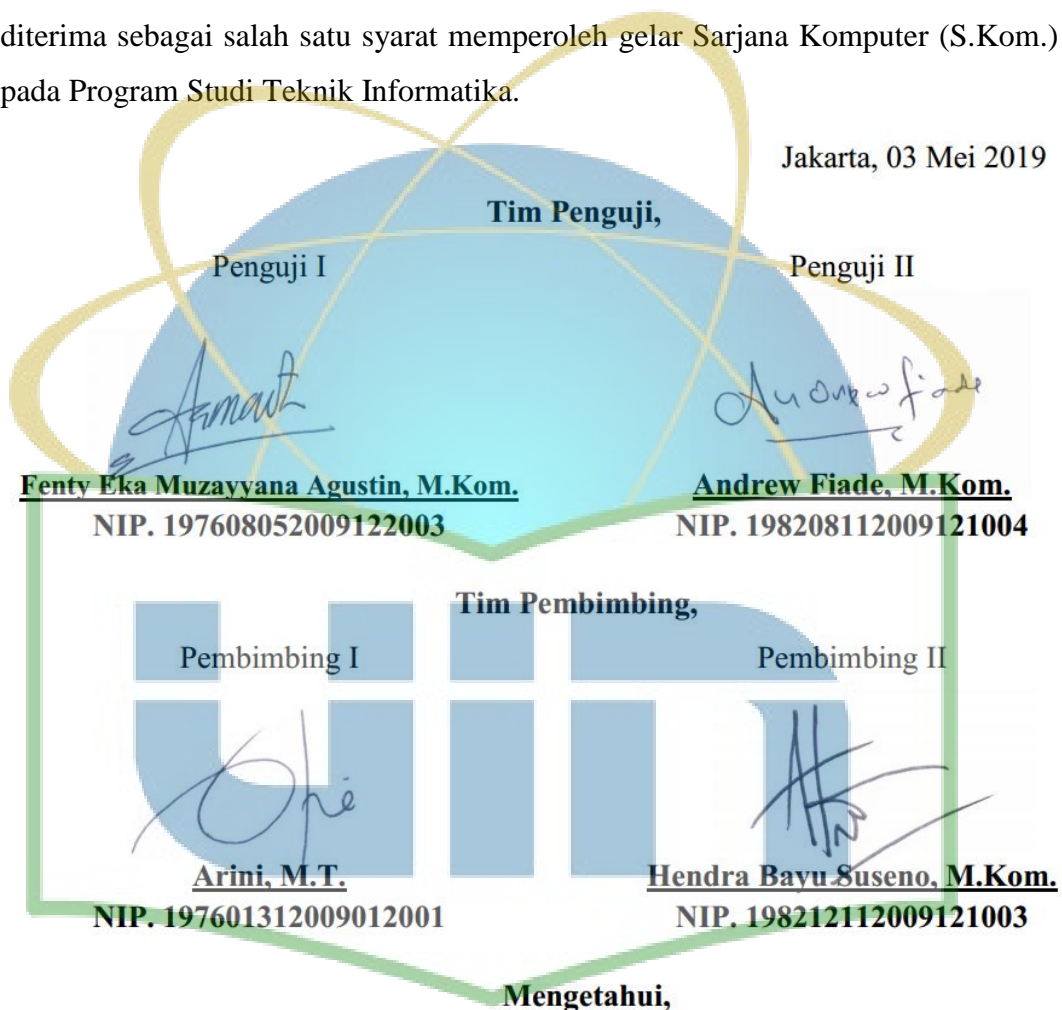
**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH
JAKARTA**

2019 M / 1440 H

PENGESAHAN UJIAN

Skripsi berjudul “Penerapan Metode Boyer Moore Pada Aplikasi Kamus Istilah Teknologi Informasi Berbasis Web (Studi Kasus: PT. Erefka Tiga Pilar Utama)” telah diujikan dalam sidang Munaqasyah Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta pada 03 Mei 2019. Skripsi ini telah diterima sebagai salah satu syarat memperoleh gelar Sarjana Komputer (S.Kom.) pada Program Studi Teknik Informatika.

Jakarta, 03 Mei 2019



Dekan Fakultas Sains dan Teknologi

Ketua Program Studi

Prof. Dr. Lilv Suravva Eka Putri, M.Env.Stud.
NIP. 196904042005012005

Arini, M.T.
NIP. 197601312009012001

PERSETUJUAN PEMBIMBING

PENERAPAN METODE BOYER MOORE PADA APLIKASI KAMUS ISTILAH TEKNOLOGI INFORMASI BERBASIS WEB (STUDI KASUS: PT. EREFKA TIGA PILAR UTAMA)

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar
Sarjana Komputer (S.Kom)

Disusun Oleh:

Rikza Muntazari

1112091000104

Menyetujui

Pembimbing I

Arini, M.T.

NIP. 197601312009012001

Pembimbing II

Hendra Bayu Suseno, M.Kom.

NIP. 198212112009121003

Mengetahui

Ketua Program Studi Teknik Informatika

Arini, M.T.

NIP. 197601312009012001

PERNYATAAN ORISINALITAS

Dengan ini saya menyatakan bahwa:

1. Skripsi ini merupakan hasil karya asli saya yang diajukan untuk memenuhi salah satu persyaratan memperoleh gelar strata 1 UIN Syarif Hidayatullah Jakarta.
2. Semua sumber yang saya gunakan dalam penulisan ini telah saya cantumkan sesuai dengan ketentuan yang berlaku di UIN Syarif Hidayatullah Jakarta.
3. Jika dikemudian hari terbukti bahwa karya ini bukan hasil asli karya saya atau merupakan hasil jiplakan dari karya orang lain, maka saya bersedia menerima sanksi yang berlaku di UIN Syarif Hidayatullah Jakarta.



HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik UIN Syarif Hidayatullah Jakarta, saya yang bertanda tangan di bawah ini:

Nama : Rikza Muntazari

NIM : 1112091000104

Program Studi : Teknik Informatika

Fakultas : Fakultas Sains dan Teknologi

Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Islam Negeri Syarif Hidayatullah Jakarta Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul:

“Penerapan Metode Boyer Moore Pada Aplikasi Kamus Istilah Teknologi Informasi Berbasis Web (Studi Kasus: PT. Erefka Tiga Pilar Utama)”

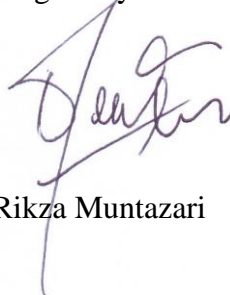
Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Islam Negeri Syarif Hidayatullah Jakarta berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jakarta

Pada tanggal : 03 Mei 2019

Yang Menyatakan



Rikza Muntazari

ABSTRAK

Rikza Muntazari (1112091000104), Penerapan Metode Boyer Moore Pada Aplikasi Kamus Istilah Teknologi Informasi Berbasis Web (Studi Kasus: PT. Erefka Tiga Pilar Utama). Dibimbing oleh **Arini, M.T.** dan **Hendra Bayu Suseno, M.Kom.**

Munculnya banyak istilah baru dalam ilmu teknologi informasi karena pesatnya perkembangan teknologi membuat masyarakat khususnya orang-orang yang menekuni bidang IT harus mengetahui dan mempelajari istilah-istilah yang berkaitan dengan ilmu teknologi informasi. Oleh karena itu dibutuhkan ketersediaan kamus teknologi informasi. Kamus istilah teknologi informasi yang beredar banyak dalam bentuk media cetak/buku. Namun demikian, kamus dalam bentuk media cetak memiliki kelemahan berupa proses pencarian yang memakan waktu lama dan tidak dapat di *update* setiap saat, maka dari itu kamus dalam bentuk digital/elektronik ada untuk mengatasi kelemahan tersebut. Aplikasi kamus istilah teknologi informasi dengan menerapkan algoritma *boyer moore* dalam pencarian istilah kata dirasa dapat membantu hal tersebut. Algoritma *boyer moore* menerapkan prinsip *good suffix* dimana karakter yang dicari sejajar dengan karakter yang menyerupainya, serta prinsip *bad character* dimana jika karakter tidak memiliki kemiripan maka langsung di eliminasi. *Field* yang dipakai dalam pencarian ini adalah istilah kata, dengan memasukan istilah kata yang dicari sebagai *input* pada kolom pencarian, kemudian kita dapat mengetahui arti dari istilah kata tersebut. Hasil dari aplikasi ini menampilkan keseluruhan *pattern* yang *match* dengan teks, dengan total waktu pencarian dibawah 300 *milisecond*.

Kata Kunci : pencocokan *string*, kamus teknologi informasi, algoritma *boyer moore*

Jumlah Pustaka : 3 Buku + 8 Jurnal + 4 Website

Jumlah Halaman : VI Bab + xvi Halaman + 120 Halaman + 81 Gambar + 36 Tabel + 3 Lampiran

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji dan syukur penulis panjatkan kehadiran Allah Subhanahu wa Ta'ala atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penelitian dan penulisan skripsi ini. Shalawat dan Salam selalu tersampaikan kepada Baginda Nabi Besar Muhammad Shallallahu 'alaihi wa Sallam, keluarga, para sahabat dan para pengikutnya hingga akhir zaman.

Skripsi penulis yang berjudul “Penerapan Metode Boyer Moore Pada Aplikasi Kamus Istilah Teknologi Informasi Berbasis Web (Studi Kasus: PT. Erefka Tiga Pilar Utama)” disusun untuk memenuhi salah satu syarat dalam menyelesaikan Program Strata Satu (S1) pada Program Studi Teknik Informatika, Universitas Islam Negeri Syarif Hidayatullah Jakarta.


Suatu kebanggan bagi penulis apabila skripsi ini dapat bermanfaat bagi pihak-pihak yang membutuhkannya. Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada pihak-pihak yang telah membantu penulis, baik secara moril maupun materil, sehingga penulisan ini terlaksana dengan baik. Secara khusus penulis mengucapkan terimakasih kepada:

1. Ibu Prof. Dr. Lily Surayya Eka Putri, M.Env.Stud., selaku Dekan Fakultas Sains dan Teknologi.
2. Ibu Arini, M.T., selaku Ketua Program Studi Teknik Informatika dan Bapak Feri Fahrianto, M.Sc., selaku Sekretaris Jurusan Program Studi Teknik Informatika, yang selalu memberikan motivasi dan arahan kepada penulis.
3. Ibu Arini, M.T. selaku Dosen Pembimbing I dan Bapak Hendra Bayu Suseno, M.Kom. selaku Dosen Pembimbing II, yang telah memberikan bimbingan dan arahan kepada penulis.
4. Bapak Nugraha, selaku pemilik PT. Erefka Tiga Pilar Utama yang sudah membantu untuk memperoleh data yang diperlukan dan meluangkan waktu untuk wawancara.

5. Kedua Orang Tua, Ibunda tercinta Zuriati, S.H. dan Ayahanda tercinta DR. H. Hasan Basri, M.A., yang telah memberikan moril maupun materil dan selalu setia mendukung serta mendo'akan penulis untuk tetap semangat dan bersungguh-sungguh dalam menyelesaikan skripsi ini.
6. Seluruh Dosen Program Studi Teknik Informatika, Fakultas Sains dan Teknologi. Yang telah banyak memberikan bimbingan dan pengarahan kepada penulis selama menjalankan aktifitas perkuliahan.
7. Kepada sahabat-sahabat seperjuangan Rizky Ivan Dharmawan, Galang Ardian Sugianto, Muhammad Faishal, Irfandi Ade Setiawan, Muhammad Arif Billah, Muhammad Fadhlhan, Yuda Masputra, Revan Julian, Renov Julian, Muhammad Rizqi Azhari dan lainnya yang tidak dapat disebut satu-persatu telah banyak memberikan masukan dan arahan kepada penulis untuk menyelesaikan skripsi ini.
8. Terimakasih kepada rekan-rekan seperjuangan di UIN Syarif Hidayatullah Jakarta, yang telah memberikan dukungan dan semangat serta saran-saran yang berguna hingga akhir penulisan skripsi ini.
9. Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini, yang tidak dapat penulis sebutkan satu-persatu. Semoga Allah Subhanahu wa Ta'ala memberikan berkah dan karunia-Nya serta membalas kebaikan mereka. Aamiin.

Penulis menyadari bahwa dalam penulisan tugas akhir ini masih jauh dari kata sempurna. Oleh karena itu, saran dan kritik yang membangun senantiasa terbuka untuk perbaikan tugas akhir ini. Harapan penulis tugas akhir ini dapat bermanfaat bagi para pembaca sekalian.

Jakarta, 03 Mei 2019



Rikza Muntazari

DAFTAR ISI

PENGESAHAN UJIAN.....	ii
PERSETUJUAN PEMBIMBING	iii
PERNYATAAN ORISINALITAS.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI	v
ABSTRAK	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xv
BAB I.....	1
PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	4
1.6. Metodologi Penelitian	5
1.6.1. Metodologi Pengumpulan Data	5
1.6.2. Metode Pengembangan Sistem.....	6
1.7. Sistematika Penulisan.....	6
BAB II	8
TINJAUAN PUSTAKA DAN LANDASAN TEORI	8
2.1. Tinjauan Pustaka	8
2.2. Landasan Teori	11
2.2.1. Kamus	11
2.2.2. Teknologi Informasi	12
2.2.3. <i>String Matching</i>	13
2.2.5. Algoritma <i>Boyer Moore</i>	14
2.2.6. <i>Browser Developer Tools</i>	23

2.2.7. <i>Business Process Modeling Notation (BPMN)</i>	24
2.2.8. <i>Pengertian Unified Modeling Language (UML)</i>	27
2.2.9. <i>Metode Rapid Application Development (RAD)</i>	30
2.2.10. <i>Fase Dalam Rapid Application Development (RAD)</i>	31
2.2.11. <i>Alasan Penggunaan Rapid Application Development (RAD)</i>	32
2.2.12. <i>Pengujian Sistem</i>	32
BAB III.....	34
METODOLOGI PENELITIAN	34
3.1. <i>Metode Pengumpulan Data</i>	34
3.1.1. <i>Studi Lapangan</i>	34
3.1.2. <i>Studi Pustaka</i>	35
3.2. <i>Metode Pengembangan Sistem</i>	35
3.3. <i>Kerangka Berfikir</i>	38
BAB IV	40
ANALISIS, PERANCANGAN SISTEM, IMPLEMENTASI DAN PENGUJIAN SISTEM	40
4.1. <i>Tahap Pengumpulan Data</i>	40
4.1.1. <i>Observasi</i>	40
4.1.2. <i>Wawancara</i>	40
4.2. <i>Tahap Pengembangan Sistem</i>	41
4.2.1. <i>Fase Requirement Planning</i>	41
4.2.2. <i>Fase Workshop Design</i>	44
4.2.3. <i>Fase Construction</i>	84
4.2.4. <i>Fase Implementation</i>	85
BAB V.....	96
HASIL DAN PEMBAHASAN	96
5.1. <i>Aplikasi Kamus Teknologi Informasi dan Komunikasi Berbasis Web.</i> 96	
5.2. <i>Hasil Dan Pembahasan</i>	107
BAB VI.....	109
PENUTUP.....	109
6.1. <i>Kesimpulan</i>	109

6.2. Saran	109
DAFTAR PUSTAKA	111
LAMPIRAN.....	114



DAFTAR GAMBAR

Gambar 2.1. Good-suffix shift, u terjadi karena karakter c berbeda dari a.	18
Gambar 2.2. Good-suffix shift, suffixu terjadi pada pattern x	18
Gambar 2.3. Bad-character shift, b terdapat di pattern x	19
Gambar 2.4. Bad-character shift, b tidak ada di pattern x.....	19
Gambar 2.5 Flow Object - Event	25
Gambar 2.6 Flow Object - Activity	25
Gambar 2.7 Flow Object - Gateaway	25
Gambar 2.8 Connection Object - Sequence Flow	25
Gambar 2.9 Connection Object - Message Flow	26
Gambar 2.10 Connection Object - Association Flow	26
Gambar 2.11 Swimlanes - Pool dan Lane	26
Gambar 2.12 Atrifacts - Group.....	26
Gambar 2.13 Atrifacts - Annotation.....	27
Gambar 3.1 Kerangka Berfikir	39
Gambar 4.1 Proses Bisnis Pencarian Kata Pada Kamus Istilah Teknologi Informasi.....	41
Gambar 4.2 Proses Bisnis Pencarian Kata Pada Aplikasi Kamus Istilah Teknologi Informasi.....	42
Gambar 4.3 Flowchart Alur Pencarian Algoritma <i>Boyer Moore</i>	47
Gambar 4.4 Flowchart Pencarian Kata (<i>Pattern</i>) Pada Kamus.....	48
Gambar 4.5 Usecase Diagram Admin.....	50
Gambar 4.6 Usecase Diagram User	51
Gambar 4.8 Activity Diagram Admin Ubah Data Kamus.....	53
Gambar 4.9 Activity Diagram Admin Lihat Data Kamus.....	54
Gambar 4.10 Activity Diagram Admin Hapus Data Kamus	55
Gambar 4.11 Activity Diagram Admin Tambah Data Kategori.....	57
Gambar 4.12 Activity Diagram Admin Ubah Data Kategori.....	58
Gambar 4.13 Activity Diagram Admin Lihat Data Kategori	59
Gambar 4.14 Activity Diagram Admin Hapus Data Kategori	60

Gambar 4.15 Activity Diagram Admin Lihat Data Pesan	62
Gambar 4.16 Activity Diagram Admin Hapus Data Pesan	63
Gambar 4.17 Activity Diagram User Cari Data Kamus.....	64
Gambar 4.18 Activity Diagram User Tambah Data Pesan	65
Gambar 4.19 Sequence Diagram Admin Tambah Data Kamus	66
Gambar 4.20 Sequence Diagram Admin Ubah Data Kamus	67
Gambar 4.21 Sequence Diagram Admin Lihat Data Kamus	67
Gambar 4.22 Sequence Diagram Admin Hapus Data Kamus.....	68
Gambar 4.23 Sequence Diagram Admin Tambah Data Kategori	69
Gambar 4.24 Sequence Diagram Admin Ubah Data Kategori.....	69
Gambar 4.25 Sequence Diagram Admin Lihat Data Kategori.....	70
Gambar 4.26 Sequence Diagram Admin Hapus Data Kategori.....	71
Gambar 4.27 Sequence Diagram Admin Lihat Data Pesan.....	71
Gambar 4.28 Sequence Diagram Admin Hapus Data Pesan.....	72
Gambar 4.29 Sequence Diagram User Mencari Data Kamus Metode Boyer Moore	73
Gambar 4.29 Sequence Diagram User Menambah Data Pesan.....	74
Gambar 4.30 Class Diagram Aplikasi Kamus Istilah Teknologi Informasi..	77
Gambar 4.31 Halaman Pencarian Kata Pada Kamus.....	78
Gambar 4.32 Halaman Login Admin.....	78
Gambar 4.33 Halaman Beranda Admin.....	79
Gambar 4.34 Halaman Admin Data Kamus.....	79
Gambar 4.35 Halaman Admin Data Kategori	80
Gambar 4.36 Halaman Admin Data Pengguna	80
Gambar 4.37 Halaman Admin Data Pesan	81
Gambar 4.38 Pencarian Istilah Binary	88
Gambar 4.39 Hasil Dari Pencarian Istilah Binary.....	88
Gambar 4.40 Pengujian Karakter “bi” Aplikasi Kamus Istilah Teknologi Informasi.....	91
Gambar 4.41 Pengujian Karakter “bin” Aplikasi Kamus Istilah Teknologi Informasi.....	91

Gambar 4.42 Pengujian Karakter “bina” Aplikasi Kamus Istilah Teknologi Informasi.....	91
Gambar 4.43 Pengujian Karakter binar Aplikasi Kamus Istilah Teknologi Informasi.....	91
Gambar 4.44 Pengujian Karakter “binary” Aplikasi Kamus Istilah Teknologi Informasi.....	92
Gambar 5.1 Menu <i>User</i> Pencarian Istilah Kata Teknologi Informasi.....	96
Gambar 5.2 <i>Form Login Admin</i>	97
Gambar 5.3 <i>Menu Admin</i>	97
Gambar 5.4 <i>Menu Kamus</i>	98
Gambar 5.5 <i>Form Tambah Data Kamus</i>	98
Gambar 5.6 <i>Form Edit Data Kamus</i>	99
Gambar 5.7 <i>Form Hapus Data Kamus</i>	99
Gambar 5.8 <i>Form Print Data Kamus</i>	100
Gambar 5.9 <i>Form Import Data Kamus</i>	100
Gambar 5.10 <i>Menu Kategori</i>	101
Gambar 5.11 <i>Form Pencarian Data Kategori</i>	101
Gambar 5.12 <i>Form Tambah Data Kategori</i>	102
Gambar 5.13 <i>Form Edit Data Kategori</i>	102
Gambar 5.14 <i>Form Hapus Data Kategori</i>	103
Gambar 5.15 <i>Menu User</i>	103
Gambar 5.16 <i>Form Tambah Data User</i>	104
Gambar 5.17 <i>Form Edit Data User</i>	104
Gambar 5.18 <i>Form Edit Data User</i>	105
Gambar 5.19 <i>Form Hapus Data User</i>	105
Gambar 5.20 <i>Menu Pesan</i>	106
Gambar 5.21 <i>Form Tambah Data Pesan</i>	106
Gambar 5.22 <i>Form Hapus Data Pesan</i>	107
Gambar 5.23 Hasil Pengujian Sistem Kamus Istilah Teknologi Informasi.	108

DAFTAR TABEL

Tabel 2.1 Tabel Tinjauan Pustaka.....	8
Tabel 2.2 Tabel Perbandingan dengan Literatur Sejenis.....	10
Tabel 2.3 Contoh Algoritma <i>Boyer Moore</i>	15
Tabel 2.4 Contoh Pergeseran Algoritma <i>Boyer Moore</i>	15
Tabel 2.5 Perbandingan Kinerja Algoritma.....	17
Tabel 2.6 Contoh <i>Occurence Heuristic</i>	20
Tabel 2.7 Contoh Hasil Pencarian <i>Occurence Heuristic</i>	21
Tabel 2.8 Contoh Hasil Akhir Pencarian <i>Occurence Heuristic</i>	21
Tabel 2.9 Contoh <i>Match Heuristic</i>	22
Tabel 2.10 Contoh Pencocokan <i>Match Heuristic</i>	22
Tabel 2.11 Contoh Hasil Akhir Pencarian <i>Match Heuristic</i>	22
Tabel 2.12 Contoh Nilai <i>Occurence Heuristic</i> dan <i>Match Heuristic</i>	23
Tabel 2.13 Contoh implementasi pencarian pada teks proses ke-1.....	23
Tabel 2.14 Contoh implementasi pencarian pada teks proses ke-2.....	23
Tabel 2.15 Komponen <i>Use Case Diagram</i>	27
Tabel 2.16 Komponen <i>Sequence Diagram</i>	28
Tabel 2.17 Komponen <i>Activity Diagram</i>	29
Tabel 2.18 Komponen <i>Class Diagram</i>	30
Tabel 4.1 Persyaratan Sistem.....	44
Tabel 4.2 Tabel <i>Database Kamus</i>	82
Tabel 4.3 Tabel <i>Database Kategori</i>	82
Tabel 4.4 Tabel <i>Database Kategori Kamus</i>	82
Tabel 4.5 Tabel <i>Database User</i>	83
Tabel 4.6 Tabel <i>Database Roles</i>	83
Tabel 4.7 Tabel <i>Database Pesan</i>	83
Tabel 4.8 Tabel Pengujian Sistem	86
Tabel 4.9 Tabel Pengujian Contoh Istilah Kata Algoritma <i>Boyer Moore</i>	87
Tabel 4.10 Tabel Pengujian Pencarian Kata Algoritma <i>Boyer Moore</i>	89
Tabel 4.11 Tabel Pengujian Kinerja <i>Boyer Moore</i>	90

Tabel 4.12 Tabel Pengujian Kinerja Boyer Moore Kata (Antivirus).....	92
Tabel 4.13 Tabel Pengujian Kinerja Boyer Moore Kata (Binary)	92
Tabel 4.14 Tabel Pengujian Kinerja Boyer Moore Kata (Hashtag).....	93
Tabel 4.15 Tabel Pengujian Kinerja Boyer Moore Kata (Domain Name)	93
Tabel 4.16 Tabel Pengujian Kinerja Boyer Moore Kata (Email).....	93
Tabel 4.17 Tabel Pengujian Kinerja Boyer Moore Kata (PDF)	93
Tabel 4.18 Tabel Pengujian Kinerja Boyer Moore Kata (HTTPS).....	94



BAB I

PENDAHULUAN

1.1. Latar Belakang

Teknologi informasi secara umum dapat dipahami sebagai pengelolaan informasi yang berbasis pada teknologi komputer yang saat ini teknologinya terus berkembang *sehubungan* perkembangan teknologi lain yang dapat dikoneksikan dengan komputer itu sendiri (Supriyanto, 2005). Perkembangan teknologi informasi ini berjalan cepat seiring dengan lahirnya *internet* dan kepopuleran *world wide web*. Hal ini didukung dan ditandai penuh oleh transformasi konsep *web* dari *web 1.0* bergerak menjadi *web 2.0* yang digunakan pada teknologi *www* (*world wide web*) dan *web* desain yang memfasilitasi kreatifitas dalam sebuah komunitas berbasis *web sharing* informasi, dan kolaborasi atau diskusi antar pengguna (Wahono, 2008).

Kebutuhan masyarakat terhadap layanan teknologi berbasis IT sangat bervariasi, salah satunya kebutuhan akan ketersediaan kamus, kamus memiliki berbagai macam jenis mulai dari kamus yang bersifat umum seperti kamus bahasa hingga kamus istilah-istilah khusus seperti kamus politik, kamus ekonomi, kamus bahasa dan sebagainya (Halim, 2016).

Munculnya banyak istilah baru dalam teknologi informasi karena pesatnya perkembangan teknologi membuat orang-orang yang menekuni bidang teknologi harus mengetahui dan mempelajari hal-hal yang berkaitan dengan ilmu teknologi informasi. Kendala yang penulis temukan dari observasi dan wawancara dengan Bapak Nugraha pada PT. EREFKA TIGA PILAR UTAMA, masih terdapat beberapa pegawai dan mahasiswa magang yang tidak terlalu paham tentang istilah-istilah dalam teknologi informasi. Didasari hal tersebut, kemudian dibuatlah kamus istilah teknologi informasi dalam bentuk media cetak/buku. Namun demikian, kamus dalam bentuk media cetak memiliki kelemahan berupa proses pencarian yang memakan waktu lama, maka dari itu kamus digital/elektronik ada untuk mengatasi kelemahan tersebut (Renanda, 2012).

Adaptasi kamus istilah teknologi informasi dari bentuk cetak ke media digital/elektronik khususnya *web* masih sangat terbatas dalam hal kuantitas. Dari hasil penelusuran menggunakan mesin pencari di *internet* hanya terdapat sedikit layanan diantaranya *website* www.pnri.go.id/IstilahKomputer.aspx. *Website* ini pun masih menggunakan teknologi HTML konvensional yang mengharuskan pengguna memuat ulang seluruh halaman untuk mendapatkan informasi yang diperlukan.

Kamus teknologi informasi merupakan penunjang yang dapat membantu seseorang untuk memahami istilah-istilah penting dalam ilmu teknologi informasi. Di dalamnya memuat istilah-istilah penting untuk diketahui. Oleh sebab itu, sangat bermanfaat jika kamus istilah teknologi informasi dimiliki oleh setiap orang khususnya yang berprofesi di bidang IT. Akan tetapi, jika dilihat dari istilah-istilah teknologi informasi sangat banyak dan membutuhkan banyak waktu sehingga mengakibatkan proses pencarian kata dalam kamus semakin lambat dan tidak efisien.

Untuk mengatasi permasalahan tersebut, dibutuhkan suatu aplikasi kamus teknologi informasi yang dapat mempermudah dan mempersingkat waktu dalam penggunaannya. Untuk mempercepat dan mempermudah dalam proses pencarian, dibutuhkan suatu algoritma yang dapat memaksimalkan waktu proses pencarian. Algoritma adalah langkah-langkah logis dalam menyelesaikan suatu masalah yang disusun secara sistematis. Algoritma pencarian pada saat ini semakin berkembang, ada algoritma yang proses pencocokan *string* dari arah kanan ke kiri dan dari kiri ke kanan. Pencarian arti kata (*string*) merupakan hal yang sangat dibutuhkan dalam sebuah kamus terutama kamus istilah teknologi informasi, dengan demikian maka digunakanlah algoritma pencarian *string* sebagai penyelesaian untuk mengatasi hal tersebut.

Pada penelitian ini, penulis berinisiatif menggunakan algoritma *Boyer Moore*. Algoritma *Boyer Moore* melakukan pencarian kata dimulai dari posisi kanan hingga akhirnya sampai pada posisi paling kiri. Langkah ini berbeda dengan algoritma pencarian *string* sejenis yang memulai pencarian kata dari kiri. Algoritma ini menerapkan prinsip *good-suffix* (dimana karakter yang dicari

disejajarkan dengan karakter yang menyerupainya) serta prinsip *bad character* (dimana jika karakter tidak memiliki kemiripan akan langsung dieliminasi). Dengan kedua prinsip ini, informasi *string* yang diperoleh dalam pencarian akan semakin banyak sehingga *output* yang dihasilkan akan menjadi lebih baik (Ramadhansyah, 2013). Algoritma *boyer moore* mampu menghasilkan *output* dengan baik. Hal ini sesuai dengan hasil penelitian yang dilakukan oleh Vina Sagita dan Maria Irmira Prasetiyowati yang menyatakan bahwa algoritma *boyer moore* mampu menghasilkan *output* paling baik dibandingkan dengan algoritma *string matching* lainnya.

Berdasarkan uraian diatas penulis tertarik untuk mengambil judul **“PENERAPAN METODE BOYER MOORE PADA APLIKASI KAMUS ISTILAH TEKNOLOGI INFORMASI BERBASIS WEB”**.

1.2. Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat diidentifikasi masalah yang terjadi adalah sebagai berikut:

1. Bagaimana penerapan algoritma *boyer moore* pada aplikasi kamus istilah teknologi informasi?
2. Bagaimana kinerja algoritma *boyer moore* pada aplikasi kamus istilah teknologi informasi berdasarkan *running time* atau waktu dari hasil pencarian kata?

1.3. Batasan Masalah

Adapun batasan masalah yang digunakan dalam penelitian ini, diantaranya sebagai berikut:

1. Metode pengembangan sistem menggunakan metode RAD (*Rapid Application Development*) menurut teori Gary B. Shelly, Harry J. Rosenblatt, 2011. Dengan empat tahapan, yaitu *requirement planning*, *workshop design*, *construction*, dan *implementation*.
2. Perancangan UML yang digunakan (*Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*).

3. Algoritma yang digunakan pada penelitian ini adalah algoritma *Boyer Moore*.
4. Bahasa pemrograman yang digunakan adalah PHP serta menggunakan *tool editor Sublime Text* dan MySQL sebagai *database*.
5. Pada penelitian ini tidak membahas mengenai kemanan data.
6. Pada penelitian ini tidak membahas lebih dalam tentang kinerja algoritma *Boyer Moore*.
7. Mengingat banyaknya istilah teknologi informasi didalam kamus, maka penulis membatasi jumlah istilah kata pada penelitian ini yaitu 200 istilah kata teknologi informasi.
8. Kinerja algoritma *Boyer Moore* pada penelitian ini hanya sebatas pada *running time* atau waktu dari hasil pencarian kata.
9. Algoritma *Boyer Moore* yang di terapkan pada aplikasi ini di letakkan pada fungsi pencarian istilah kata pada aplikasi kamus istilah teknologi informasi.

1.4. Tujuan Penelitian

Penelitian ini memiliki tujuan sebagai berikut:

1. Merancang aplikasi kamus istilah teknologi informasi berbasis web.
2. Menerapkan metode *boyer moore* untuk pencarian kata pada aplikasi kamus istilah teknologi informasi yang akan dibangun.
3. Untuk mengetahui kinerja algoritma *boyer moore* pada aplikasi kamus istilah teknologi informasi berdasarkan waktu dari hasil pencarian kata.

1.5. Manfaat Penelitian

Sesuai dengan permasalahan dan tujuan penelitian yang sudah disebutkan, banyak manfaat yang dapat di petik dalam penyusunan skripsi ini, diantaranya adalah:

1. Bagi Pengguna
 - a. Memperkaya wawasan kosa kata teknologi informasi terkini.

- b. Sebagai bahan rujukan dalam mencari definisi suatu istilah teknologi informasi.
- c. Sebagai solusi bagi masyarakat dengan mewadahi segala informasi istilah teknologi informasi terkini dalam satu tempat.

2. Bagi Penulis

- a. Menerapkan ilmu-ilmu yang telah diperoleh selama perkuliahan.
- b. Menambah pengetahuan penulis tentang algoritma *string matching* yaitu algoritma *Boyer Moore*.
- c. Untuk memenuhi salah satu syarat kelulusan strata satu (S1), Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta.

3. Bagi Universitas

- a. Mengetahui kemampuan mahasiswa dalam menguasai materi pelajaran yang diperoleh dibangku kuliah.
- b. Mengetahui kemampuan mahasiswa dalam menerapkan ilmunya dan sebagai bahan evaluasi.
- c. Memberikan gambaran tentang kesiapan mahasiswa dalam menghadapi dunia kerja yang sebenarnya.

1.6. Metodologi Penelitian

1.6.1. Metodologi Pengumpulan Data

Teknik pengumpulan data yang dilakukan:

1. Studi Lapangan

Penelitian ini dilakukan dengan mendatangi lokasi penelitian (survey) untuk memperoleh informasi serta data yang diperlukan. Adapun teknik yang ditempuh adalah:

- a. Observasi, merupakan proses pengamatan dan pencatatan secara sistematis mengenai objek yang ingin diteliti.
- b. Wawancara adalah komunikasi dua arah untuk mendapatkan informasi dari responden.

2. Studi Pustaka

Mengumpulkan dan mempelajari dari beberapa buku, dan referensi dari internet yang dapat dijadikan bahan acuan pembahasan dalam penelitian ini.

3. Studi Literatur Sejenis

Melihat dan membandingkan aplikasi ataupun jurnal penelitian sejenis dari penelitian yang sebelumnya telah dilakukan.

1.6.2. Metode Pengembangan Sistem

SDLC (*Software Development Life Cycle*) merupakan salah satu metode pengembangan sistem. SDLC adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik) (Shalahuddin & Rosa, 2013).

Ada beberapa model SDLC yang dapat digunakan, salah satunya yaitu RAD. Pada penelitian ini, model *Rapid Application Development* (RAD) digunakan sebagai metode pengembangan sistem, terdapat empat fase dalam pengembangan sistem dengan menggunakan metode RAD (Shelly & Harry, 2011) yaitu, *requirement planning*, *workshop design*, *construction*, dan *implementation*.

1.7. Sistematika Penulisan

BAB I

PENDAHULUAN

Pada bab ini berisikan latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metodologi penelitian, serta sistematika penulisan.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

Pada bab ini, berisikan literatur siapa saja yang dijadikan acuan serta menguraikan teori-teori yang

terkait dengan sistem dan konsep algoritma pencarian untuk mendukung pembahasan bab selanjutnya.

BAB III

METODOLOGI PENELITIAN

Pada bab ini menjelaskan tentang metode-metode apa saja yang digunakan dalam sistem tersebut.

BAB IV

ANALISIS, PERANCANGAN SISTEM, IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini menjelaskan tentang analisa perancangan, implementasi dan pengujian sistemnya.

BAB V

HASIL DAN PEMBAHASAN

Pada bab ini akan membahas implementasi antarmuka dari perancangan sistem yang telah dirancang pada bab sebelumnya.

BAB VI

PENUTUP

Bab ini merupakan bab terakhir dari skripsi, yang terdiri dari kesimpulan dari apa yang telah diuraikan pada bab sebelumnya beserta dengan kritik dan saran untuk pengembangan selanjutnya.

DAFTAR PUSTAKA

Daftar pustaka merupakan rujukan yang digunakan dalam penulisan skripsi ini. Cara penulisan daftar pustaka, termasuk juga sitasi (*citation*) harus mengikuti cara penulisan dari Badan Standardisasi Nasional Indonesia tahun 2005, SNI.06-6989.35-2005.

LAMPIRAN

Lampiran dipakai untuk menempatkan data atau keterangan lain yang berfungsi untuk melengkapi uraian yang telah disajikan dalam bagian utama skripsi.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Berikut adalah beberapa jurnal yang berkaitan dengan algoritma *boyer moore*, dari beberapa jurnal tersebut diambil beberapa kesimpulan seperti algoritma yang digunakan, kelebihan, dan kekurangan dari penelitian pada jurnal tersebut. Berikut merupakan tinjauan pustaka dari beberapa jurnal penelitian tersebut:

Tabel 2.1 Tabel Tinjauan Pustaka

No.	Nama	Judul	Kelebihan	Kekurangan	Basis Aplikasi
1.	Vina Sagita, Maria Irminda Prasetyowati (2013)	STUDI PERBANDINGAN IMPLEMENTASI ALGORITMA BOYER-MOORE, TURBO BOYER MOORE, DAN TUNED BOYER MOORE DALAM PENCARIAN STRING	Terdapat tiga perbandingan metode terhadap varian dari algoritma boyer moore	Aplikasi yang dibuat bersifat offline	Desktop
2.	Muhammad Muz Fakhri (2018)	IMPLEMENTASI ALGORITMA BOYER MOORE PADA SISTEM PENGAJUAN BIMBINGAN SKRIPSI DAN PKL	Terdapat laporan honor kepada dosen secara real time ketika nilai sudah dimasukkan	Sistem ini masih memiliki tingkat kemanan yang rendah	Web

3.	Kencana Wulan Argakusumah, Seng Hansun (2014)	IMPLEMENTASI ALGORITMA BOYER MOORE PADA APLIKASI KAMUS KEDOKTERAN BERBASIS ANDROID	Adanya fitur bookmark kata pada aplikasi	Tidak terdapat fitur gambar pada tiap kategori penyakit	Andorid
4.	Rizky Ivan Darmawan (2017)	IMPLEMENTASI ALGORITMA BOYER MOORE PADA APLIKASI KAMUS KEBIDANAN	Pencarian kata dapat dengan mudah dilakukan oleh pengguna	Tidak terdapat fitur export dan import data kamus	Web
5.	Darmawan Utomo, Eric Wijaya Harjo, Handoko (2013)	PERBANDINGAN ALGORITMA STRING SEARCHING BRUTE FORCE, KNUTH MORRIS PARTT, BOYER MOORE, DAN KARP RABIN PADA TEKS ALKITAB BAHASA INDONESIA	Terdapat beberapa metode algoritma pencarian pada penelitian yang dilakukan	Tidak adanya implementasi pada aplikasi yang dibangun pada penelitian	Tidak Berbentuk Aplikasi

Hasil analisa dari literatur sejenis diatas adalah:

1. Pada literature pertama diatas merupakan aplikasi berbasis *desktop*, merupakan aplikasi *standalone*, aplikasi yang dibangun tidak dapat digunakan untuk beberapa pengguna, karena aplikasi yang dibangun bersifat *offline*.
2. Pada literatur kedua, merupakan aplikasi berbasis web untuk pencarian data mahasiswa, namun fitur dan tingkat keamanan yang ditawarkan masih kurang mendukung untuk digunakan.

3. Pada literatur ketiga diatas merupakan aplikasi berbasis *mobile android*, maka dari itu penulis membangun aplikasi berbasis *web* dikarenakan perkembangan teknologi *website* yang terus berkembang, dan aplikasi *web* yang dapat berjalan pada beberapa perangkat *device* seperti, *desktop*, *tablet*, dan *smartphone* karena hanya membutuhkan aplikasi *browser* untuk menjalankannya.
4. Pada literatur kelima diatas, merupakan beberapa perbandingan *string searching* namun pada penelitian tersebut tidak adanya penerapan aplikasi yang dibangun, hanya berfokus pada perbandingan metode.

Berikut ini merupakan tabel keunikan dari perbandingan sistem yang dibangun penulis dengan literatur sejenis seperti pada tabel 2.1 diatas:

Tabel 2.2 Tabel Perbandingan dengan Literatur Sejenis

No.	Nama	Maksimal Input Kata (Pattern)	Data Pada Database	Kata Sambung dan Simbol	Dukungan Autocomplete	Export / Import Data	Dukung an file .txt .doc
1.	Vina Sagita, Maria Irina Prasetyowati (2013)	< 15	Hanya mencari pattern	✓		✓	✓
2.	Muhammad Muz Fakhri (2018)	> 15	< 50	✓			
3.	Kencana Wulan Argakusumah, Seng Hansun (2014)	> 15	< 200	✓			
4.	Rizky Ivan Darmawan (2017)	> 15	> 200	✓	✓		
5.	Darmawan Utomo, Eric Wijaya Harjo, Handoko (2013)	> 15	Hanya mencari pattern	✓			
6.	Penulis	> 15	> 200	✓	✓	✓	

2.2. Landasan Teori

Berikut merupakan landasan teori yang digunakan dalam penelitian ini:

2.2.1. Kamus

Secara etimologi, kata kamus berasal dari kata dalam bahasa Arab, yaitu *qamus* (bentuk jamaknya *qawamus*). Bahasa Arab *qamus* (قاموس) menyerap kata kamus dari kata dalam bahasa Yunani kuno, *okeanos* yang berarti lautan. Tentu menjadi pertanyaan, bagaimana kata kamus yang berurusan dengan kosakata berasal dari bahasa Yunani kuno *okeanos* yang berarti lautan. Kalau kita mencoba untuk memahami sejarah kata tersebut, jelaslah bahwa kata kamus memiliki makna dasar wadah pengetahuan, khususnya pengetahuan bahasa yang tidak terhingga dalam dan luasnya, seluas dan sedalam lautan (Chaer, 2007).

Kamus adalah sejenis buku rujukan yang menerangkan sebuah kata yang mengandung makna. Berfungsi untuk membantu mengenal perkataan baru. Selain menjelaskan sebuah kata, kamus juga mempunyai pedoman atau sebutan, asal usul (etimologi) sesuatu perkataan dan juga contoh penggunaan bagi sesuatu perkataan.

Menurut Chaer (2007:179), padanan kata kamus dalam bahasa Inggris adalah *dictionary*, mulai digunakan pada karya tulis pada tahun 1526 dan berasal dari kata dalam bahasa Latin, yaitu *dictionary*. Kata ini diturunkan dari kata *dictio* yang berarti kata atau berkata. Padanannya dalam bahasa Belanda adalah *woordenboek* yang dibedakan dari *woordenschat* yang dalam bahasa Indonesia dipadankan dengan pembendaharan atau kosakata.

Selain pengertian kamus yang telah disebutkan di atas, Chaer (2007:179) juga menyebutkan pengertian kamus yang dikemukakan oleh beberapa para ahli.

1. Kridalaksana menyebutkan bahwa kamus adalah buku referensi yang memuat daftar kata atau gabungan kata dengan keterangan mengenai pelbagai segi maknanya dan penggunaannya dalam bahasa, biasanya disusun menurut abjad.
2. Dalam *American Every Dictionary* disebutkan bahwa kamus adalah sebuah buku berisi kata-kata dari sebuah bahasa, biasanya disusun

secara alfabetis, disertai keterangan akan artinya ucapannya, ejaannya, dsb.

3. Labrousse (1977) menyebutkan bahwa kamus adalah buku berisi kumpulan kata-kata sebuah bahasa yang disusun secara alfabetis diikuti dengan definisi atau terjemahannya dalam bahasa lain.
4. Keraf (1984) mendefinisikan kamus sebagai sebuah buku referensi, memuat daftar kata-kata yang terdapat dalam sebuah bahasa, disusun secara alfabetis, disertai keterangan cara menggunakan kata itu.

2.2.2. Teknologi Informasi

Teknologi terkait dengan ide atau pikiran yang tidak akan pernah berakhir, keberadaan teknologi bersama dengan keberadaan budaya umat manusia. Teknologi merupakan kreasi manusia sehingga tidak alami dan bersifat buatan (*artificial*). Teknologi merupakan himpunan dari pikiran (*set of means*) sehingga teknologi dapat dibatasi atau bersifat universal, tergantung dari sudut pandang analisis. Teknologi bertujuan memfasilitasi ikhtiar manusia (*human endeavor*) sehingga harus mampu meningkatkan performa kemampuan manusia. Berikut beberapa definisi tentang teknologi informasi:

- Menurut Martin (1999:2), Teknologi Informasi tidak hanya terbatas pada teknologi komputer (perangkat keras dan perangkat lunak) yang digunakan untuk memproses dan menyimpan informasi, melainkan juga mencakup teknologi komunikasi untuk mengirimkan informasi.
- Menurut Williams dan Sawyer (2003:2), Teknologi Informasi adalah teknologi yang menggabungkan komputasi (*computer*) dengan jalur komunikasi berkecepatan tinggi yang membawa data, suara dan video.
- Menurut David Goetch (2006:2), teknologi yaitu suatu produk yang dilakukan oleh manusia dengan memanfaatkan peralatan (*tools*), proses dan sumber daya (*resources*).
- Menurut Senn (2006:2), teknologi informasi yaitu suatu item yang bermacam-macam dan kemampuan yang digunakan dalam pembuatan, penyimpanan, dan penyebaran data serta informasi. Komponen utamanya

ada tiga yaitu komputer (*computer*), komunikasi (*communication*), dan keterampilan (*skill*). Tiga hal yang terkandung di dalam teknologi yaitu keterampilan (*skill*), logika berpikir (*algorithm*) dan perangkat keras (*hardware*). Dalam pandangan manajemen teknologi (*management of technology*), teknologi diartikan,

1. Teknologi sebagai makna untuk memenuhi suatu maksud didalamnya terkandung apa saja yang dibutuhkan untuk mengubah sumber daya (*resources*) ke suatu produk atau jasa.
2. Teknologi tidak ubahnya sebagai pengetahuan, sumber daya yang diperlukan untuk mencapai suatu tujuan.
3. Teknologi adalah suatu tubuh dari ilmu pengetahuan dan rekayasa yang dapat diaplikasikan pada perancang produk dan atau proses pada penelitian untuk mendapatkan pengetahuan baru

Susunan hierarki informasi mulai dari data/fakta, kemudian diseleksi dan diolah menjadi sesuatu yang berguna. Suatu sistem tanpa informasi akan tidak berguna, karena suatu sistem yang kurang mendapatkan informasi akan mengalami kemacetan dan akhirnya berhenti. Dengan demikian informasi sangat penting bagi suatu sistem. Mendefinisikan informasi sebagai data yang telah diproses sehingga mempunyai arti dan dapat meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Jadi sumber informasi adalah data yang merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu, kesatuan nyata (*fact and entity*) berupa objek nyata seperti tempat, benda, dan orang yang betul-betul ada dan terjadi.

2.2.3. String Matching

String matching adalah pencarian sebuah *pattern* pada sebuah teks (Kurnaedi, Effendi, & Diana, 2012). Algoritma *string matching* adalah algoritma yang ditujukan untuk melakukan *string matching*. Prinsip kerja algoritma *string matching* adalah sebagai berikut:

1. Melakukan *scan* teks dengan bantuan sebuah *window* yang ukurannya sama dengan panjang *pattern*.
2. Menempatkan *window* pada awal teks.
3. Membandingkan karakter pada *window* dengan karakter dari *pattern*. Setelah pencocokan (baik hasilnya cocok atau tidak cocok), dilakukan *shift* ke kanan pada *window*. Prosedur ini dilakukan berulang-ulang sampai *window* berada pada akhir teks. Mekanisme ini disebut mekanisme *sliding-window*.

Algoritma-algoritma *string matching* mempunyai tiga komponen, yaitu: *pattern*, teks dan alfabet dengan asumsi sebagai berikut:

1. *Pattern*, yaitu deretan karakter yang dicocokkan dengan teks, dinyatakan dengan $x[0..m-1]$, panjang *pattern* dinyatakan dengan m .
2. Teks, yaitu tempat pencocokan *pattern* dilakukan, dinyatakan dengan $y[0..n-1]$, panjang teks dinyatakan dengan n .
3. Alfabet, yang berisi semua simbol yang digunakan oleh bahasa pada teks dan *pattern*, dinyatakan dengan Σ dengan ukuran dinyatakan dengan $ASIZE$.

2.2.5. Algoritma *Boyer Moore*

Algoritma *boyer moore* adalah salah satu algoritma pencarian *string*, dipublikasikan oleh Robert S. Boyer, dan J. Strother Moore pada tahun 1977. Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Tidak seperti algoritma pencarian *string* yang ditemukan sebelumnya, algoritma *boyer moore* mulai mencocokkan karakter dari sebelah kanan *pattern*. Ide di balik algoritma ini adalah bahwa dengan memulai pencocokan karakter dari kanan, dan bukan dari kiri, maka akan lebih banyak informasi yang didapat. Algoritma *boyer moore* termasuk algoritma *string matching* yang paling efisien dibandingkan algoritma-algoritma *string matching* lainnya. Karena sifatnya yang efisien, banyak dikembangkan algoritma *string matching* dengan bertumpu pada konsep algoritma *boyer moore*.

2.2.5.1. Kelebihan Algoritma *Boyer Moore*

Tidak seperti pencarian *string* lainnya *Brute Force*, *Knuth Morris Pratt* yang mempunyai cara kerja membandingkan satu-persatu karakter dari kiri ke kanan, *boyer moore* membandingkan karakter dari kanan ke kiri dan memiliki loncatan karakter yang besar sehingga mempercepat pencarian *string* karena dengan hanya memeriksa sedikit karakter, dapat langsung diketahui bahwa *string* yang dicari tidak ditemukan dan dapat digeser ke posisi berikutnya.

Tabel 2.3 Contoh Algoritma *Boyer Moore*

Teks	G	R	A	C	E				
Pattern	R	I	K	Z	A				

Pada tabel 2.2, dengan melakukan pencocokan dari posisi paling akhir/kanan *pattern* dapat dilihat bahwa karakter “A” pada *pattern* “RIKZA” tidak cocok dengan karakter “E” pada teks “GRACE”, dan karakter “E” tidak pernah ada dalam *pattern* “RIKZA” yang dicari sehingga *pattern* “RIKZA” dapat digeser melewati teks “GRACE” sehingga posisinya menjadi:

Tabel 2.4 Contoh Pergeseran Algoritma *Boyer Moore*

Teks	G	R	A	C	E						
Pattern						R	I	K	Z	A	

Tabel 2.3 menunjukkan bahwa algoritma *boyer moore* memiliki pergeseran karakter yang besar sehingga mempercepat pencarian *pattern* karena dengan hanya memeriksa sedikit karakter, dapat langsung diketahui bahwa *pattern* yang dicari tidak ditemukan dan dapat digeser ke posisi berikutnya (Ginting, 2014).

2.2.5.2. Kelemahan Algoritma *Boyer Moore*

Algoritma *boyer moore* mencocokkan *pattern* dari kanan ke kiri oleh sebab itu kelemahan dari algoritma ini adalah ketika semua karakter memiliki kesamaan atau cocok dan hanya karakter terakhir atau karakter paling kiri

yang berbeda maka pencarian ini akan memerlukan waktu yang sedikit lama (Utomo D. , 2008).

2.2.5.3. Deskripsi Kerja Algoritma *Boyer Moore*

Algoritma *boyer moore* menggunakan dua buah tabel untuk mengolah informasi saat terjadi kegagalan pencocokan *pattern*. Tabel pertama disebut *bad character shift* juga sering disebut *occurrence heuristic* (OH). Tabel kedua disebut dengan istilah *good suffix shift* juga disebut *match heuristic* (MH) (Charras & Lecroq, 2014).

Secara sistematis, langkah-langkah yang dilakukan algoritma *boyer moore* pada saat mencocokkan *pattern* adalah:

1. Algoritma *boyer moore* mulai mencocokkan *pattern* pada karakter paling akhir atau kanan.
2. Dari kanan ke kiri, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
 - a. Karakter di *pattern* dan di teks yang dibandingkan tidak cocok (*mismatch*).
 - b. Semua karakter di *pattern* cocok, kemudian algoritma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian menggeser *pattern* dengan mengambil nilai terbesar dari penggeseran *good-suffix* dan penggeseran *bad-character*, lalu mengulangi langkah 2 sampai *pattern* berada di ujung teks. (Ramadhansyah, 2013).

2.2.5.4. Kinerja Algoritma *Boyer Moore*

Dari hasil penelitian yang dilakukan oleh Darmawan Utomo, Eric Wijaya Haris dan Handoko, dengan judul “Perbandingan Algoritma *String Searching Brute Force*, *Knuth Morris Pratt*, *Boyer Moore*, dan *Karp Rabin* pada teks Alkitab Bahasa Indonesia” terdapat beberapa kesimpulan antara lain:

1. Pada algoritma *Brute Force* semakin panjang pola yang dicari waktu pencariannya tetap dengan waktu rata-rata pola 0,98 detik.
2. Pada algoritma *Knuth Morris Pratt* semakin panjang pola yang dicari waktu pencariannya cenderung meningkat dengan waktu rata-rata pola 0,99 detik.
3. Pada algoritma *Boyer Moore* semakin panjang polanya waktu pencarian semakin singkat dengan waktu rata-rata pola 0,92 detik.
4. Pada algoritma *Karp Rabin* semakin panjang pola waktu pencariannya cenderung meningkat dengan waktu rata-rata pola 3,46 detik karena ada penggunaan operator mod.
5. Untuk proses pencarian pola dalam teks Alkitab (lebih dari 26 alphabet) algoritma *Boyer Moore* yang paling cepat dibandingkan dengan KR, KMP, dan BF.

Dengan jumlah perbandingan kinerja hasil pencarian, pada kasus terbaik dan kasus terburuk sebagai berikut:

Tabel 2.5 Perbandingan Kinerja Algoritma

Algoritma	Jumlah Perbandingan	Best Case	Worse Case
Brute Force	5.050.601	m+n	m.n
		4.861.378	48.613.680
Knuth Morris Pratt	5.022.319	m+n	m+n
		4.861.378	4.861.378
Boyer Moore	1.269.844	N / m	m+n
		486.137	4.861.378
Karp Rabin	4.568.997	m+n	m.n
		4.861.378	48.613.680

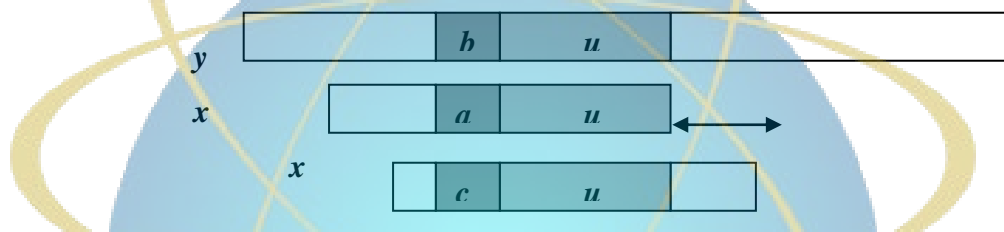
Beberapa hasil penelitian lain terkait perbandingan algoritma *boyer moore* yang dilakukan oleh Vina Sagita, Maria Irmina Prasetyowati dengan judul “Studi Perbandingan Implementasi Algoritma *Boyer Moore*, *Turbo Boyer Moore*, Dan *Tuned Boyer Moore* Dalam Pencarian *String*” dengan hasil

kesimpulan penelitian bahwa algoritma *boyer moore* memiliki waktu pencarian tercepat dari tiga varian *boyer moore* yang telah ditentukan. Algoritma *turbo boyer moore* merupakan algoritma tercepat kedua dan yang terakhir adalah *tuned boyer moore*.

2.2.5.5. Good Suffix Shift

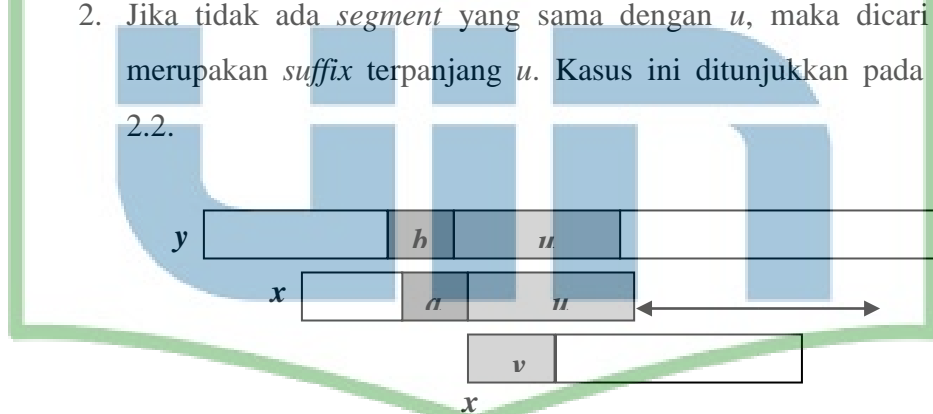
Konsep dari fungsi *good suffix shift* adalah sebagai berikut:

1. *Good-suffix shift* adalah pergeseran yang dibutuhkan dari $x[i]=a$ ke karakter lain yang letaknya lebih kiri dari $x[i]$ dan terletak di sebelah kiri segmen u . Kasus ini ditunjukkan pada Gambar 2.1.



Gambar 2.1. Good-suffix shift, u terjadi karena karakter c berbeda dari a

2. Jika tidak ada *segment* yang sama dengan u , maka dicari u yang merupakan *suffix* terpanjang u . Kasus ini ditunjukkan pada Gambar 2.2.



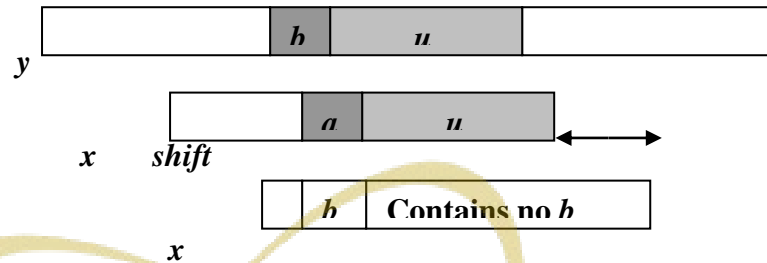
Gambar 2.2. Good-suffix shift, suffixu terjadi pada pattern x

2.2.5.6. Bad Character Shift

Berdasarkan contoh kasus di atas, *bad-character* adalah karakter pada teks yaitu $y[i+j]$ yang tidak cocok dengan karakter pada *pattern*.

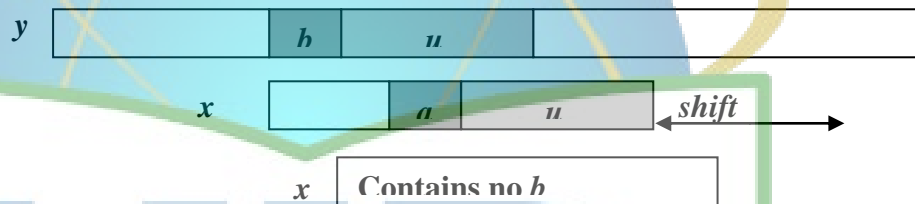
Konsep dari fungsi *bad-character shift* adalah sebagai berikut:

1. Jika *bad-character* $y[i+j]$ terdapat pada *pattern* di posisi terkanan k yang lebih kiri dari $x[i]$ maka *pattern* digeser ke kanan sejauh $i-k$. Kasus ini ditunjukkan pada Gambar 2.3.



Gambar 2.3. Bad-character shift, b terdapat di pattern x

2. Jika *bad-character* $y[i+j]$ tidak ada pada *pattern* sama sekali, maka *pattern* digeser ke kanan sejauh i . Kasus ini ditunjukkan pada gambar 2.4.



Gambar 2.4. Bad-character shift, b tidak ada di pattern x

3. Jika *bad-character* $y[i+j]$ terdapat pada *pattern* di posisi terkanan k yang lebih kanan dari $x[i]$ maka *pattern* seharusnya digeser sejauh $i-k$ yang hasilnya negatif (*pattern* digeser kembali ke kiri). Maka bila kasus ini terjadi, akan diabaikan. Pada kasus ketidakcocokan di atas, algoritma akan membandingkan langkah yang diambil oleh fungsi *good-suffix shift* dan *bad-character shift* di mana langkah yang paling besar yang akan digunakan.

2.2.5.7. Pencarian Dengan Algoritma Boyer Moore

1. Buat tabel pergeseran *pattern* yang dicari (P) dengan pendekatan *Match Heuristic* (MH) dan *Occurence Heuristic* (OH), untuk

menentukan jumlah pergeseran yang akan dilakukan jika mendapat karakter tidak cocok pada proses pencocokan dengan teks (T).

2. Jika dalam proses perbandingan terjadi ketidakcocokan antara pasangan karakter pada *pattern* dan karakter teks, pergeseran dilakukan dengan memilih salah satu nilai pergeseran dari dua tabel, dan memiliki nilai pergeseran paling besar dari tabel *Match Heuristic* dan *Occurence Heuristic*.
3. Dua kemungkinan penyelesaian dalam melakukan pergeseran *pattern*, Jika karakter yang tidak cocok, tidak ada pada *pattern* maka pergeseran adalah sebanyak jumlah karakter pada *pattern*. dan jika karakter yang tidak cocok, ada pada *pattern*, maka banyaknya pergeseran bergantung dari nilai pada tabel *Match Heuristic* dan *Occurence Heuristic*.
 - Jika karakter pada teks yang sedang dibandingkan cocok dengan karakter pada *pattern*, maka posisi karakter pada *pattern* dan teks diturunkan sebanyak 1 posisi, kemudian lanjutkan dengan pencocokan pada posisi tersebut dan seterusnya.
 - Jika kemudian terjadi ketidakcocokan karakter *pattern* dan teks, maka pilih nilai pergeseran terbesar dari tabel *match heuristic* dan nilai tabel *occurence heuristic*.
4. Jika semua karakter telah cocok, artinya *pattern* telah ditemukan di dalam teks. (Ramadhansyah, 2013).

Cara menghitung tabel *occurence heuristic*:

Contoh *pattern*: MOORE

Panjang karakter: 5

Tabel 2.6 Contoh *Occurence Heuristic*

Index	0	1	2	3	4		
Pattern	M	O	O	R	E		
Occurence Heuristic							

Langkah-langkah pemberian nilainya adalah sebagai berikut:

1. Lakukan perhitungan, **OH = (length-1-index)**
length = panjang karakter= 5
2. Karakter pertama adalah “M” dengan Index = 0
OH = (5 - 1 - 0 = 4) maka nilai karakter “M” = 4
3. Karakter kedua adalah “O” dengan index = 1
OH = (5 - 1 - 1 = 3) maka nilai karakter “O” = 3
4. Karakter ketiga adalah “O” dengan index = 2
OH = (5 - 1 - 2 = 2) maka nilai karakter “O” = 2
5. Karakter keempat adalah “R” dengan index = 3
OH = (5 - 1 - 3 = 1) maka nilai karakter “R” = 1
6. Karakter kelima adalah “E” dengan index = 4
OH = (5 - 1 - 4 = 0) maka nilai karakter “E” = 0

Tabel 2.7 Contoh Hasil Pencarian Occurence Heuristic

Index	0	1	2	3	4		
Pattern	M	O	O	R	E		
Occurence Heuristic	4	3	2	1	0		

7. Jika ada karakter yang berulang ambil nilai OH terkecil, dalam kasus ini ada karakter “O” yang bernilai 3 dan 2, maka jadikan karakter “O” bernilai 2. Dapat dilihat pada tabel 2.5.

Tabel 2.8 Contoh Hasil Akhir Pencarian Occurence Heuristic

Index	0	1	2	3	4		
Pattern	M	O	O	R	E		
Occurence Heuristic	4	2	2	1	0		

Cara menghitung tabel *match heuristic*:

Contoh pattern: MOORE

Panjang karakter: 5

Tabel 2.9 Contoh *Match Heuristic*

Index	0	1	2	3	4		
Pattern	M	O	O	R	E		
Match Heuristic							

Langkah-langkah pemberian nilainya adalah sebagai berikut:

1. Berikan nilai 1 (nilai *default* MH) pada index terbesar, dalam kasus ini index terbesar adalah “E”.
2. Bandingkan karakter kedua, jika karakter “E” sudah ditemukan/sudah cocok, tetapi karakter sebelum “E” bukan “R”, maka cek apakah karakter yang sudah cocok pada teks ada pada *pattern* berikutnya, Jika ada sejajarkan karakter tersebut dengan karakter yang sudah cocok, dan beri nilai pergeseranya ke tabel MH, pada kasus ini “E” sudah cocok.

Tabel 2.10 Contoh Pencocokan *Match Heuristic*

Index	0	1	2	3	4		
Text	M	O	O	L	E		
Pattern	M	O	O	R	E		
Match Heuristic							

3. Lanjutkan langkah ke dua untuk karakter berikutnya sampai akhir *pattern*, ketidakcocokan *pattern* maka bandingkan karakter yang sudah cocok dengan karakter yang belum dibandingkan pada *pattern*, jika ada maka sejajarkan karakter tersebut, jika tidak maka pergeseranya sejauh panjang karakter, dan beri nilai pergeserannya ke tabel MH.

Tabel 2.11 Contoh Hasil Akhir Pencarian *Match Heuristic*

Index	0	1	2	3	4		
Text	M	O	O	L	E		
Pattern	M	O	O	R	E		
Match Heuristic	5	5	5	5	1		

Contoh pencarian dengan algoritma *boyer moore*:

1. Jika karakter yang dibanding tidak ada pada *pattern* maka nilai pergeserannya sepanjang jumlah *pattern*.
2. Untuk pergeseran bandingkan nilai OH dan MH ambil nilai terbesar sebagai keputusan pergeseran.
3. Jika semua karakter telah cocok, artinya *pattern* telah ditemukan di dalam teks.

Tabel 2.12 Contoh Nilai Occurrence Heuristic dan Match Heuristic

Index	0	1	2	3	4		
Pattern	M	O	O	R	E		
Occurrence Heuristic	4	2	2	1	0		
Match Heuristic	5	5	5	5	1		

Tabel 2.13 Contoh implementasi pencarian pada teks proses ke-1

Text	M	O	O	L	I	M	O	O	R	E		
Pattern	M	O	O	R	E							
Index	0	1	2	3	4							

OH = I (Teks) = 5

MH = *index* [4] (Pattern) = 1

5 > 1, dengan begitu *pattern* digeser 5 karakter

Tabel 2.14 Contoh implementasi pencarian pada teks proses ke-2

Text	M	O	O	L	I	M	O	O	R	E		
Pattern						M	O	O	R	E		
Index						0	1	2	3	4		

Semua karakter telah cocok, artinya *pattern* telah ditemukan di dalam teks.

2.2.6. Browser Developer Tools

Browser Developer Tools merupakan fitur yang tersedia pada browser yang digunakan sebagai alat untuk menulis dan melakukan fungsi *debug web* yang menjadi bawaan pada sebuah *browser*, khususnya *browser Google Chrome*. Fitur *developer tools* biasa digunakan untuk mengiterasikan, melakukan *debug*

dan memprofilkan suatu situs *web*. Terdapat beberapa fitur pada *developer tools* yang ada pada sebuah *browser*, salah satunya adalah fitur yang terdapat pada panel *network* yang berfungsi untuk mengukur kinerja jaringan pada sebuah halaman *web*. *Panel network* mencatat informasi tentang operasi setiap jaringan di suatu halaman, termasuk detail data *timing*, permintaan dan *respons* HTTP *header*, *cookie*, dan lainnya. Untuk mengetahui kinerja dari algoritma *boyer moore* pada penelitian ini, penulis memanfaatkan fitur yang terdapat dalam *panel network* pada *developer tools* untuk mengetahui waktu yang didapat pada pencarian istilah kata teknologi informasi. Kinerja algoritma *boyer moore* diambil berdasarkan tiga waktu, yaitu:

1. *Request Sent/Sending*, yaitu proses pengiriman data yang di *input client* untuk *server*.
2. *Waiting (Time To First Byte)*, lamanya waktu dari proses pengiriman sampai proses pengambilan data dari *server*.
3. *Downloading*, merupakan proses pengambilan data dari *server* ke *client*.

2.2.7. *Business Process Modeling Notation (BPMN)*

Business Process Modeling Notation (BPMN) merupakan sebuah standar untuk memodelkan *web service* dan proses *web service*, diinisiasi oleh *Business Process Management Initiative (BPMI)*. BPMN terdiri atas sebuah diagram, yaitu *Business Process Diagram (BPD)*. Menurut Bintang Eka Putera (2014) tujuan utama dari BPMN adalah untuk menyediakan notasi yang dapat dimengerti oleh pengguna proses bisnis, dari analis bisnis yang membuat draft awal proses, sampai kepada teknisi pengembangan yang bertanggung jawab, dalam pelaksanaan teknologi yang menggunakan proses tersebut.

Diagram dari BPMN terdiri atas beberapa elemen. Elemen ini terbagi atas empat kategori, yaitu *Flow Object*, *Connection Object*, *Swimlanes* dan *Atrifacts*. Berikut ini merupakan penjelasan dari masing-masing elemen BPMN.

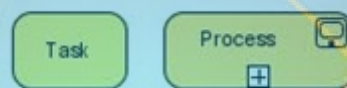
1. *Flow Object*

- a. *Event* direpresentasikan dalam bentuk lingkaran dan menjelaskan apa yang terjadi. Terdapat tiga jenis *event* yaitu *start*, *intermediate*, *end*. Masing-masing mewakili kejadian dimulainya proses, interupsi proses dan berakhirnya proses bisnis.



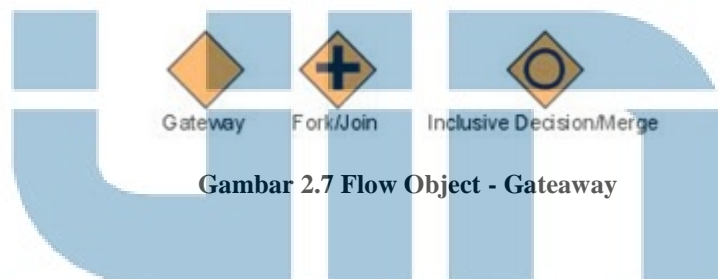
Gambar 2.5 Flow Object - Event

- b. *Activity* merepresentasikan pekerjaan (*task*) yang harus diselesaikan.



Gambar 2.6 Flow Object - Activity

- c. *Gateway* merepresentasikan pemecahan alur yang terdapat pada proses bisnis.



Gambar 2.7 Flow Object - Gateway

2. Connection Object

- a. *Sequence Flow* merepresentasikan pilihan *default* untuk menjalankan proses.



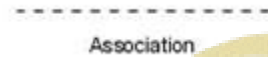
Gambar 2.8 Connection Object - Sequence Flow

- b. *Message Flow* merepresentasikan aliran pesan antar proses.



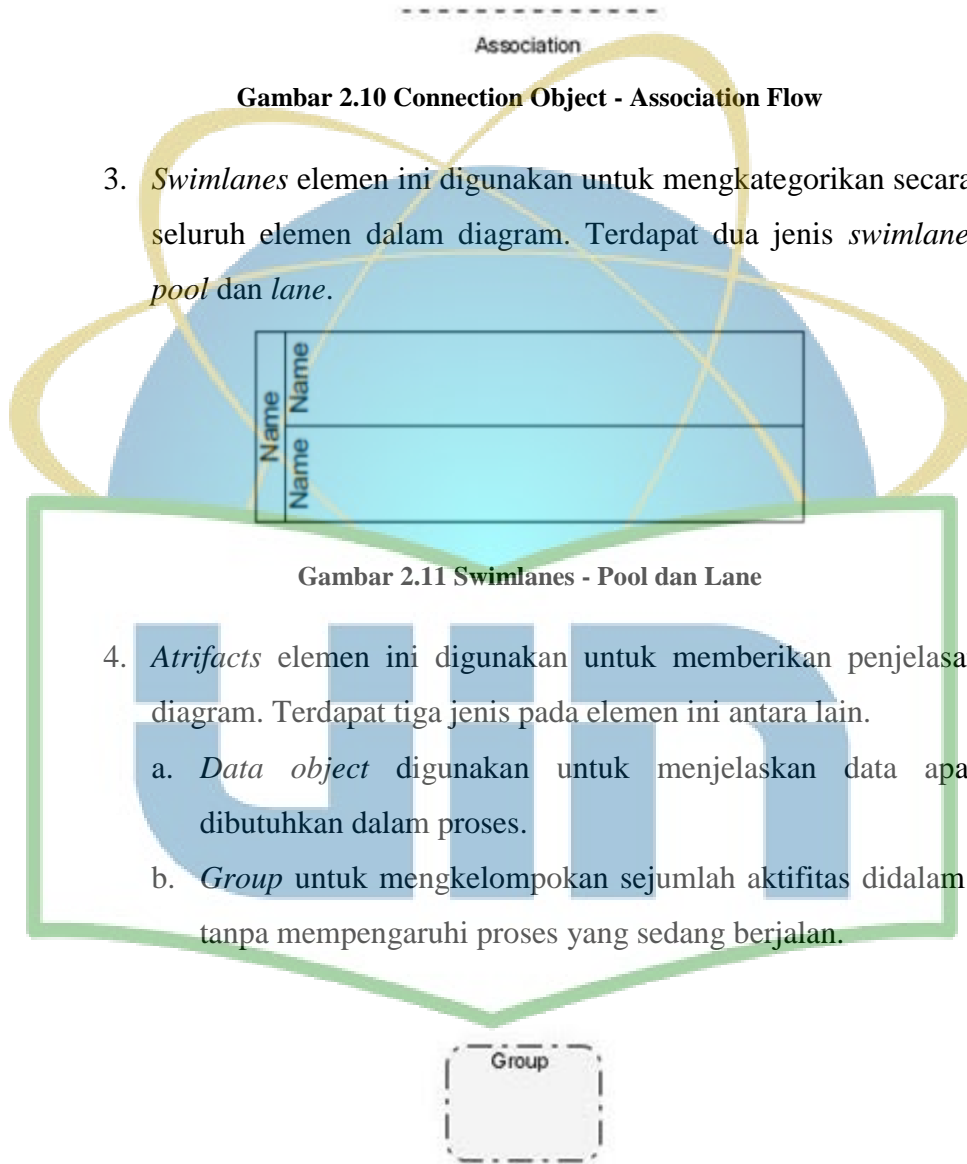
Gambar 2.9 Connection Object - Message Flow

- c. *Association Flow* digunakan untuk menghubungkan elemen dengan *artifact*.



Gambar 2.10 Connection Object - Association Flow

3. *Swimlanes* elemen ini digunakan untuk mengkategorikan secara visual seluruh elemen dalam diagram. Terdapat dua jenis *swimlanes* yaitu *pool* dan *lane*.



Gambar 2.11 Swimlanes - Pool dan Lane

4. *Atrifacts* elemen ini digunakan untuk memberikan penjelasan pada diagram. Terdapat tiga jenis pada elemen ini antara lain.
- Data object* digunakan untuk menjelaskan data apa yang dibutuhkan dalam proses.
 - Group* untuk mengelompokkan sejumlah aktifitas didalam proses tanpa mempengaruhi proses yang sedang berjalan.



Gambar 2.12 Atrifacts - Group

- c. *Annotation* digunakan untuk memberikan catatan agar diagram mudah dimengerti.



Gambar 2.13 Atrifacts - Annotation

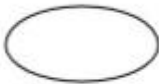
2.2.8. Pengertian *Unified Modeling Language* (UML)


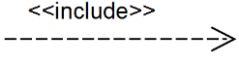
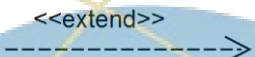
Menurut (Shalahuddin & Rosa, 2013) UML (*Unified Modeling Language*) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram. UML (*Unified Modeling Language*) muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak yang berbasis objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. Bagian-bagian utama dari UML adalah *View*, *Diagram*, model *element*, dan *general mechanism*. Adapun jenis Diagram dalam UML antara lain:

1. *Use case Diagram*

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

Tabel 2.15 Komponen *Use Case Diagram*

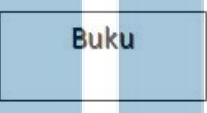


Nama	Gambar	Keterangan
Usecase		Gambaran fungsional dari suatu sistem

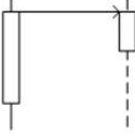
Association		Interaksi antara aktor dan use case
<<include>>		Relasi terhadap 2 buah use case, dimana use case dasar sebagai base use case akan selalu membutuhkan use case yang di include. Arah panah include dari use case utama menuju use case pendukung.
<<extend>>		Relasi antar 2 buah use case, dimana use case dasar membutuhkan use case pendukung jika diperlukan. Arah panah extend dari use case pendukung menuju ke use case utama.

2. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. *Sequence* diagram menunjukkan urutan *event* kejadian dalam satu waktu.

Tabel 2.16 Komponen Sequence Diagram






Nama	Gambar	Keterangan
Object		Komponen berbentuk kotak yang mewakili sebuah class
Actors		Komponen berbentuk <i>stick figure</i> , yang mewakili seorang pengguna berinteraksi dengan sistem
Lifeline		Berbentuk garis putus – putus, berfungsi untuk menggambarkan aktifitas dari <i>object</i>

Object Message		Hubungan antar object yang menunjukkan urutan kejadian yang terjadi
----------------	---	---

3. Activity Diagram

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

Tabel 2.17 Komponen Activity Diagram

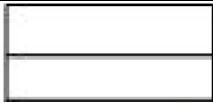


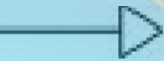
Nama	Gambar	Keterangan
Activity		Memperlihatkan bagaimana masing-masing class antarmuka saling berinteraksi
Start State		Bagaimana objek dibentuk diawal
End State		Bagaimana objek dibentuk diakhir
Decision		Menggambarkan suatu keputusan yang diambil dari kondisi tertentu
Line Connector		Digunakan untuk menghubungkan simbol dengan simbol lainnya

4. Class Diagram

Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat *program* atau *programmer* membuat

kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak bisa sesuai.

Tabel 2.18 Komponen Class Diagram

Nama	Gambar	Keterangan
Class		Himpunan dari objek-objek
Dependency		Hubungan dimana perubahan yang terjadi pada suatu element mandiri
Association		Hubungan antara objek satu dengan yang lainnya
Generalization		Relasi antar class dengan makna Generalisasi-spesialisasi

2.2.9. Metode *Rapid Application Development* (RAD)

Menurut Martin (Pressman, 2002) adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan pada siklus perkembangan yang sangat pendek. Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier di mana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen jika kebutuhan dipahami dengan baik, proses RAD memungkinkan tim pengembangan menciptakan “sistem fungsional yang utuh” dalam periode waktu yang relatif singkat (kira-kira 60-90 hari).

Pada buku *Systems Analysis and Design Ninth Edition*, Shelly Cashman Series yang penulis baca tentang *Rapid Application Development* (RAD) adalah teknik berbasis tim yang mempercepat informasi pembangunan sistem dan menghasilkan sistem informasi yang aktif. Seperti JAD, RAD menggunakan pendekatan kelompok. Produk akhir dari RAD adalah sistem informasi yang baru, sementara produk akhir dari JAD adalah persyaratan model. RAD adalah metodologi lengkap, dengan siklus hidup empat fase yang sejalan dengan fase

SDLC tradisional. Sebuah perusahaan menggunakan RAD untuk mengurangi biaya dan waktu pengembangan, dan meningkatkan probabilitas keberhasilan. RAD sangat bergantung pada *prototyping* dan keterlibatan terhadap pengguna. Proses RAD memungkinkan pengguna untuk memeriksa model kerja sedini mungkin, menentukan apakah memenuhi kebutuhan mereka, dan menyarankan perubahan yang diperlukan. Berdasarkan *input* pengguna, prototipe dimodifikasi dan proses interaktif berlanjut sampai sistem sudah benar-benar selesai dan pengguna puas terhadap sistem tersebut. Sebuah tim dalam proyek menggunakan alat CASE untuk membangun prototipe dan membuat dokumentasi berlanjut.

2.2.10. Fase Dalam *Rapid Application Development (RAD)*

Terdapat empat fase dalam RAD yang melibatkan penganalisis dan pengguna dalam tahap perencanaan kebutuhan, perancangan, pengembangan, dan penerapan (Shelly & Harry, 2011).

1. Fase Perencanaan Kebutuhan (*Requirement Planning*)

Tahap perencanaan kebutuhan menggabungkan unsur-unsur dari sistem perencanaan dan analisis sistem. Pengguna, manajer, dan anggota staf IT membahas dan menyepakati kebutuhan bisnis, lingkup proyek, kendala, dan persyaratan sistem. Tahap perencanaan kebutuhan berakhir ketika tim setuju pada isu-isu yang ada dan memperoleh manajemen otorisasi.

2. Fase *Workshop Design*

Selama fase *workshop design*, pengguna berinteraksi dengan sistem analisis dan mengembangkan model dan prototipe yang mewakili semua sistem *input*, proses dan *output*. Kelompok RAD atau sub kelompok biasanya menggunakan kombinasi teknik JAD dan CASE tools untuk menerjemahkan kebutuhan pengguna ke dalam model kerja. desain pengguna merupakan proses yang berkesinambungan, interaktif yang memungkinkan pengguna untuk memahami, memodifikasi, dan akhirnya menyetujui sebuah model kerja dari sistem yang memenuhi kebutuhan.

3. *Construction*

Tahap konstruksi berfokus pada program dan pengembangan aplikasi. Dalam RAD, pengguna terus berpartisipasi dan masih dapat menyarankan perubahan atau perbaikan laporan pengembangan yang tepat.

4. *Implementation*

Tahap *cutover* merupakan tugas akhir dalam pelaksanaan, seperti konversi data, pengujian, perubahan ke sistem baru, dan pelatihan pengguna. Dibandingkan dengan metode tradisional, seluruh proses dikompresi. Akibatnya, sistem baru dibangun, disampaikan, dan ditempatkan dalam operasi lebih cepat.

2.2.11. Alasan Penggunaan *Rapid Application Development* (RAD)

Berikut ini merupakan alasan yang mendorong digunakannya metode *Rapid Application Development* pada penelitian tugas akhir ini. Antara lain:

1. Aplikasi yang dirancang dan dikembangkan, merupakan aplikasi yang dikembangkan dalam waktu relatif singkat, karena aplikasi yang dibangun merupakan aplikasi yang memiliki tingkat kompleksitas yang relatif sederhana. Hal ini menjadi alasan utama sesuai dengan pernyataan dari Kenneth E. Kendal dan Julie E. Kendall (2008) bahwa metode ini cocok digunakan untuk mempersingkat waktu antara perancangan dan penerapan sebuah sistem atau aplikasi.
2. Aplikasi ini juga merupakan aplikasi yang berdiri sendiri dan tidak terintegrasi dengan sistem lain. Hal ini berdasarkan pada pernyataan George Marakas (2006) dalam bukunya, "*Management Information System*" bahwa metode ini baik dipertimbangkan oleh pengembangan apabila aplikasi yang dibuat merupakan aplikasi perangkat lunak yang akan berjalan sendiri tanpa ada interaksi dengan sistem lain.

2.2.12. Pengujian Sistem

Pengujian *Black Box* adalah tipe pengujian yang memperlakukan perangkat lunak yang diketahui kinerja internalnya. Sehingga para penguji memandang perangkat lunak seperti layaknya sebuah "kotak hitam" yang tidak

penting dilihat isinya, tapi cukup di uji bagian luar (Soetam, 2011). Sebagai contoh, jika terdapat sebuah perangkat lunak yang merupakan sebuah sistem informasi *inventory* di sebuah perusahaan. Pada jenis pengujian *Black Box*, perangkat lunak tersebut akan dieksekusi kemudian dites apakah telah memenuhi kebutuhan pengguna yang didefinisikan pada tahap awal tanpa harus membongkar programnya.

Berikut ini merupakan keuntungan yang diperoleh dari jenis pengujian *Black Box* (Soetam, 2011):

1. Anggota tim pengujian tidak harus dari seseorang yang memiliki kemampuan teknis di bidang pemrograman.
2. Kesalahan dari perangkat lunak ataupun *bug* seringkali ditemukan oleh komponen tester yang berasal dari pengguna.
3. Hasil dari pengujian *black box* dapat memperjelas kontradiksi ataupun kerancuan yang mungkin timbul dari eksekusi sebuah perangkat lunak.
4. Proses pengujian dapat dilakukan lebih cepat dibandingkan pengujian *white box*.



BAB III

METODOLOGI PENELITIAN

3.1. Metode Pengumpulan Data

Dalam penyusunan skripsi ini diperlukan data-data serta informasi yang lengkap sebagai bahan yang dapat mendukung kebenaran materi uraian dan pembahasan. Oleh karena itu, sebelum penyusunan skripsi ini dilakukan riset atau penelitian terlebih dahulu untuk mendapatkan data dan informasi yang terkait. Teknik pengumpulan data yang dilakukan untuk skripsi ini adalah sebagai berikut.

3.1.1. Studi Lapangan

a. Observasi

Inti dari observasi adalah adanya perilaku yang tampak dan tujuan yang ingin dicapai. Perilaku yang tampak dapat berupa perilaku yang dapat dilihat langsung oleh mata, dapat didengar, dapat dihitung, dan dapat diukur (Herdiansyah, 2010). Pada penelitian ini, penulis melakukan observasi pada PT. Erefka Tiga Pilar Utama pada tanggal 20 Oktober 2018 dan 10 November 2018.

b. Wawancara

Menurut Gordon yang dikutip oleh Haris Herdiansyah dalam bukunya yang berjudul “Metodologi Penelitian Kualitatif” definisi wawancara adalah wawancara merupakan percakapan antara dua orang yang salah satunya bertujuan untuk menggali dan mendapatkan informasi untuk tujuan tertentu (Herdiansyah, 2010). Pada penelitian yang dilakukan penulis di PT. Erefka Tiga Pilar Utama, yang menjadi narasumber penulis untuk memperoleh data yaitu Bapak Nugraha sebagai pemilik dari PT. Erefka Tiga Pilar Utama, wawancara dilakukan pada tanggal 20 November 2018.

3.1.2. Studi Pustaka

a. Referensi Buku

Beberapa buku atau literatur yang digunakan oleh penulis dalam merancang dan menyusun laporan penelitian ini antara lain:

- Rekayasa Perangkat Lunak (Shalahuddin & Rosa, 2013).
- Membangun Website Dinamis Interaktif dengan PHP-MySQL (Windows & Linux) Untuk Tingkat Pemula dan Tingkat Lanjut (Sukarno, 2006).
- Mastering Ajax dan PHP (Kadir, 2009).

b. Literatur Sejenis

Pada studi literatur sejenis yang telah dibahas pada bab sebelumnya, penulis mengumpulkan data dari lima jurnal sejenis yang berkaitan dengan penelitian ini, dalam mengumpulkan informasi tentang masalah yang hampir sama, diperlukan pembandingan yang berkaitan dalam hal ini agar sistem yang dibuat dapat lebih baik lagi.

c. Penelusuran Data Online

Pada tahapan ini penulis melakukan proses pencarian informasi dengan mempelajari setiap konsep yang berhubungan dengan metode *boyer moore* dan implementasinya pada studi kasus lain, UML (*Unified Modelling Language*), dan RAD (*Rapid Application Development*). Penelusuran secara *online* juga dimanfaatkan untuk pemecahan masalah yang berhubungan dengan *source code* pada pembuatan aplikasi kamus istilah teknologi informasi ini.

3.2. Metode Pengembangan Sistem

Metode pengembangan sistem yang penulis gunakan untuk mengembangkan sistem ini yaitu metode pengembangan RAD (*Rapid Application Development*) menggunakan teori Gary B. Shelly, Harry J. Rosenblatt, 2011. Penulis memilih menggunakan RAD karena memiliki kelebihan antara lain menurut George Marakas (2006):

1. Penghematan waktu dalam keseluruhan fase proyek dapat dicapai
2. RAD sangat membantu pengembangan aplikasi yang berfokus pada waktu penyelesaian proyek.

Sedangkan *tools* yang digunakan adalah notasi UML (*Unified Modelling Language*) menggunakan teori Salahuddin. Pada metode RAD Gary B. Shelly, Harry J. Rosenblatt, 2011 terdiri dari empat pengembangan yaitu:

1. Perencanaan Kebutuhan (*Requirement Planning*)

Pada tahap ini, *user* dan *analyst* (penulis) melakukan semacam pertemuan untuk melakukan analisis masalah dari kebutuhan aplikasi, dan melakukan analisis kebutuhan data dan informasi kebutuhan aplikasi. Pada tahap ini hal terpenting adalah. Mendapatkan data tentang ilmu teknologi informasi dan komunikasi seperti macam-macam perangkat *hardware*, *software*, *brainware* lain sebagainya. Cara yang penulis gunakan untuk tahap ini adalah wawancara dan observasi. Pada proses perencanaan terdapat beberapa proses yang penulis buat yaitu:

- a. Kebutuhan Bisnis

Pada tahap ini akan dijelaskan mengenai bisnis proses yang terjadi dan pembuatan diagram pada bisnis proses pada aplikasi yang akan dibangun, kebutuhan bisnis yang dimaksud lebih mencakup dalam proses dimana *user* mencari data kamus dalam aplikasi kamus istilah teknologi informasi berbasis *web*.

- b. Kendala

Pada tahap ini akan di jelaskan berbagai macam permasalahan yang terjadi dari proses pengumpulan data yang penulis dapat.

- c. Persyaratan Sistem

Pada tahap ini penulis menjelaskan beberapa persyaratan data, *software*, *hardware* yang dibutuhkan dalam pengembangan aplikasi yang dibuat.

2. Proses Desain (*Workshop Design*)

Tahap ini penulis melakukan proses desain dan melakukan perbaikan apabila masih terdapat ketidaksesuaian desain antara *user* dan *analyst* (penulis).

a. Desain Algoritma

Desain algoritma ini berisi tentang penggambaran alur kerja sistem yang akan dibuat seperti penggambaran langkah-langkah algoritma *boyer moore*, *pseudocode* dan *flowchart*.

b. Desain UML

Desain UML ini berisi tentang rancangan model dari sistem yang akan dibuat dengan menggunakan diagram UML (*Unified Modeling Language*) bertujuan agar sistem yang dibuat dapat dikembangkan ke tahap selanjutnya, model UML yang dibuat dalam penelitian ini meliputi

- *Use case Diagram*

Use case diagram menggambarkan interaksi antara *user* (pengguna) dengan sistem melalui sebuah cerita bagaimana sebuah sistem dapat dipakai.

- *Activity Diagram*

Merupakan teknik mendeskripsikan logika prosedural, proses bisnis dan aliran kerja.

- *Sequence Diagram*

Menjelaskan interaksi dari setiap urutan langkah yang terjadi dari sistem yang dibangun.

- *Class Diagram*

Class Diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan interaksi antara mereka.

c. Desain Interface

Desain *Interface* ini berisi tentang gambaran rancangan sebelum pembuatan *interface* yang nyata dalam aplikasi, memberikan

kemudahan dalam pembuatan interface dari aplikasi yang akan dikembangkan dikemudian hari.

d. *Desain Database*

Desain *database* ini berisi data-data yang dibutuhkan dalam aplikasi, data-data tersebut antara lain data kamus, data kategori, data pengguna dan data pesan.

Selanjutnya, Perancangan antar muka merupakan mekanisme komunikasi antara pengguna dengan sistem. *Interface* dapat memberikan informasi kepada *user* untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan suatu solusi.

3. *Construction*

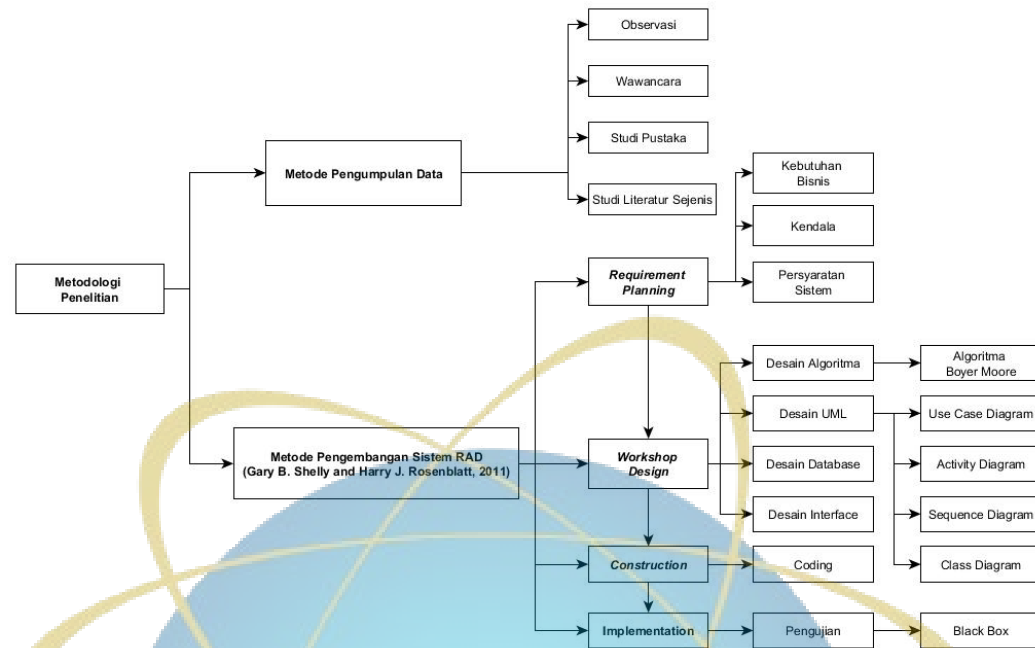
Pada tahap ini *programmer* (penulis) melakukan konversi dari desain yang sudah dibuat menjadi sebuah program dengan melakukan *coding* pada sistem sehingga menghasilkan aplikasi. *Software* yang digunakan adalah XAMPP yang meliputi *Apache* sebagai *web server*, PHP dan HTML sebagai bahasa pemrograman, MySQL sebagai *database*-nya, dan *Sublime Text* sebagai *software text editor*.

4. *Implementation*

Tahap implementasi sistem merupakan tahap meletakkan sistem agar siap untuk dioperasikan, yaitu dengan menyediakan perangkat keras apa saja yang dibutuhkan dan *install software* apa saja yang dibutuhkan hingga aplikasi berjalan. Tahap ini juga terdapat pengujian sistem yang merupakan pengujian sistem yang bertujuan melihat apakah aplikasi berjalan dengan baik atau tidak.

3.3. Kerangka Berfikir

Berdasarkan landasan teori yang telah dijabarkan pada bab sebelumnya dan metodologi yang dijelaskan pada bab ini maka dibuatlah suatu kerangka berfikir sebagai alur berfikir dalam penyusunan penelitian skripsi ini. Kerangka berfikir tersebut digambarkan seperti gambar dibawah ini:



Gambar 3.1 Kerangka Berfikir

Metode RAD yang sudah di jelaskan pada bab sebelumnya penulis terapkan pada perancangan sistem ini terlihat seperti pada gambar 3.1 diatas, terdapat beberapa tahapan seperti pengumpulan data informasi yang dibutuhkan atau *requirement planning*, *workshop design*, *construction* dan *implementation*.

BAB IV

ANALISIS, PERANCANGAN SISTEM, IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Tahap Pengumpulan Data

Berikut ini merupakan beberapa pengumpulan data yang penulis lakukan:

4.1.1. Observasi

Dalam penelitian ini, penulis melakukan observasi pada:

Lokasi

Nama Instansi : PT. EREFKA TIGA PILAR UTAMA

Alamat : Jl. H. Mawi Gg. Patung Hansip, Blok Melati No.5,
Bojong Indah, Parung, Kab. Bogor, Jawa Barat

Waktu Pelaksanaan : 20 Oktober 2018 dan 10 November 2018

Observasi ini dilakukan untuk memperoleh data yang dibutuhkan dalam penelitian.

4.1.2. Wawancara

Metode ini dilakukan dengan cara mewawancarai seseorang yang ahli dalam bidangnya atau yang mengerti tentang materi bahasan agar menjadi pendukung dalam penyusunan skripsi ini. Penggunaan metode wawancara ini karena memiliki beberapa kekuatan dalam pencarian datanya, seperti memudahkan pengaplikasian dan penerapannya, murah, dan dapat mengetahui kebutuhan pihak terkait dalam mempelajari dan memahami ilmu teknologi informasi. Dalam metode wawancara ini penulis melakukan wawancara kepada Bapak Nugraha sebagai pemilik dari PT. EREFKA TIGA PILAR UTAMA pada tanggal 20 Oktober 2018 untuk memperoleh data.

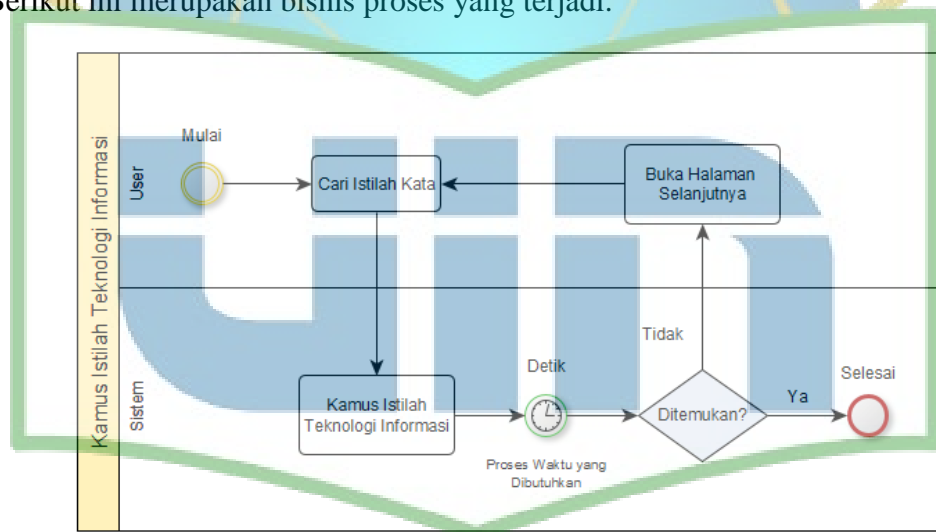
4.2. Tahap Pengembangan Sistem

Pada aplikasi kamus istilah teknologi informasi berbasis web yang dibuat ini, penulis menggunakan metode *Rapid Application Development* (RAD) dengan empat fase antara lain fase *Requirement Planning*, *Workshop Design*, *Construction* dan *Implementation* yang dibuat oleh Gary B. Shelly, Harry J. Rosenblatt sudah dijelaskan pada bab sebelumnya.

4.2.1. Fase *Requirement Planning*

4.2.1.1. Kebutuhan Bisnis

Sebelum membuat aplikasi kamus istilah teknologi informasi, penulis melakukan rancangan diagram proses bisnis yang terjadi sebelum adanya aplikasi kamus istilah teknologi informasi, rancangan proses yang terjadi meliputi *user* melakukan proses pencarian kata pada buku kamus istilah teknologi informasi. Berikut ini merupakan bisnis proses yang terjadi.

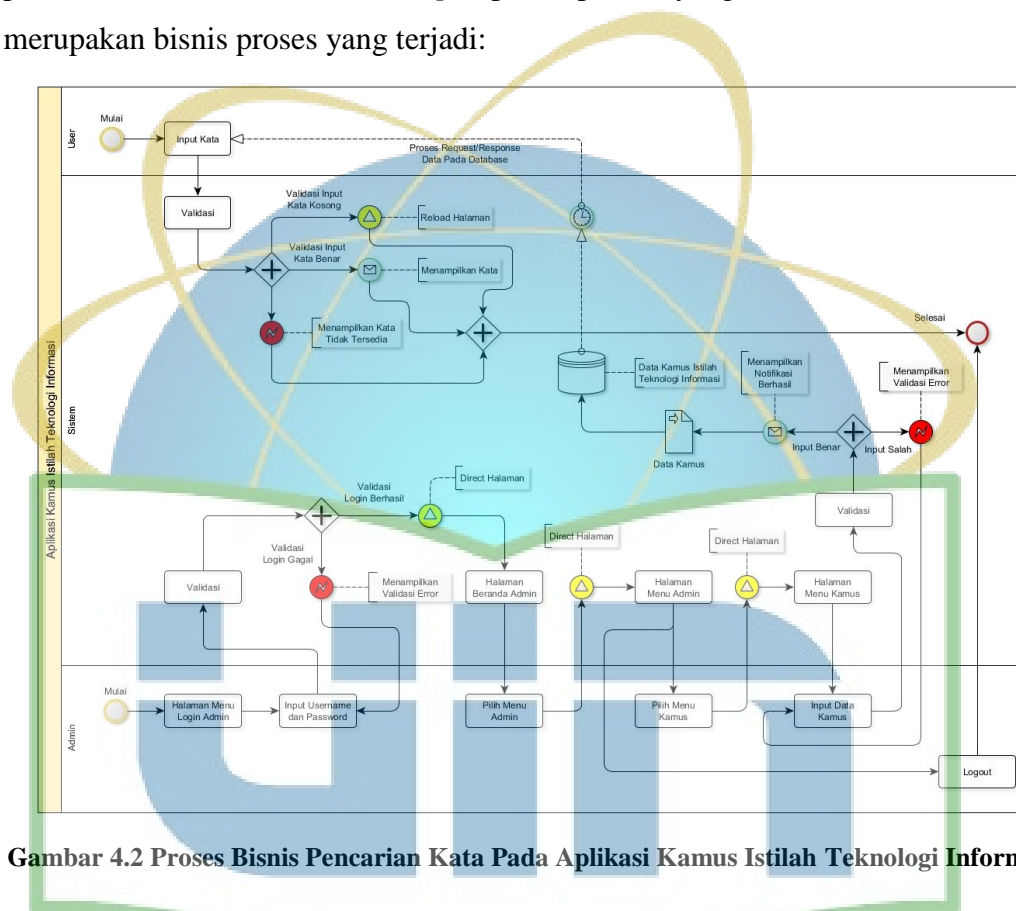


Gambar 4.1 Proses Bisnis Pencarian Kata Pada Kamus Istilah Teknologi Informasi

Pada gambar 4.1 menjelaskan bagaimana pengguna melakukan proses pencarian kata pada buku kamus istilah teknologi informasi. Pengguna tersebut menjalankan proses pencarian kata. Selanjutnya pengguna akan melakukan proses pembukaan kamus istilah teknologi informasi, proses yang dilakukan saat pencarian pada kamus istilah teknologi informasi membutuhkan waktu beberapa

detik hingga kata yang diinginkan dapat ditemukan, proses pembukaan halaman selanjutnya akan terus berulang jika kata yang diinginkan belum ditemukan, jika kata sudah didapat maka proses akan dihentikan.

Selanjutnya merupakan rancangan dari sistem yang penulis buat pada sistem yang ingin dibangun, rancangan ini menjelaskan tentang proses bisnis pencarian kata dan melakukan *input* pada aplikasi yang akan dibuat. Berikut ini merupakan bisnis proses yang terjadi:



Gambar 4.2 Proses Bisnis Pencarian Kata Pada Aplikasi Kamus Istilah Teknologi Informasi

Pada gambar 4.2 menjelaskan bagaimana pengguna melakukan proses pencarian kata pada aplikasi kamus istilah teknologi informasi. Pengguna tersebut menjalankan proses *input* kata, *input* yang dimasukkan pengguna harus melebihi satu karakter, proses pencocokan kata akan dilakukan jika karakter yang di masukkan melebihi satu karakter, proses tersebut terhubung ke dalam *database* pada aplikasi kamu istilah teknologi informasi, proses pencarian kata pada *database* membutuhkan beberapa detik hingga kata yang diinginkan ditemukan.

Selanjutnya sistem akan melakukan proses validasi, jika kata yang di masukkan tersedia maka akan menampilkan istilah dari kata yang di cari pada

aplikasi, jika kata yang di masukkan kosong maka aplikasi akan memuat ulang halaman *web*, jika kata yang di masukkan tidak tersedia maka aplikasi akan memuat ulang halaman *web* dan menampilkan bahwa kata yang di masukkan tidak tersedia pada *database*.

4.2.1.2. Kendala

Dari beberapa informasi yang didapat oleh penulis saat melakukan penelusuran pada mesin pencarian di internet, observasi dan wawancara, penulis menemukan beberapa permasalahan, diantaranya informasi pada kamus istilah teknologi informasi dari beberapa *website* di *internet* masih sedikit ketersediaan istilah atau kosakata yang terkumpul dalam sebuah *website*, seperti pada *website*:

- <http://nuysputra.blogspot.com/2011/04/kamus-tik.html>
- <https://ridhaharwan.wordpress.com/2012/10/09/kamus-istilah-komputer-teknologi-informasi/>

Sehingga membuat pencarian terhadap kata istilah teknologi informasi sulit ditemukan karena masih sedikit dari istilah atau kosakata yang ada pada *website* tersebut dan masih menggunakan sebuah *web blog* sebagai medianya.

Permasalahan selanjutnya adalah dari beberapa pegawai maupun mahasiswa magang belum sepenuhnya memahami tentang istilah dalam ilmu teknologi informasi karena banyaknya istilah yang ada, permasalahan ini dapat dilihat dari observasi yang penulis lakukan pada lampiran dari penelitian ini, sehingga menyulitkan bagi para pegawai maupun mahasiswa magang untuk memahami istilah-istilah tersebut, dari permasalahan yang telah di jelaskan maka dibuatlah buku kamus istilah teknologi informasi untuk mengatasi masalah yang terjadi. Namun dalam buku kamus istilah teknologi informasi pun masih memiliki beberapa kelemahan diantaranya, pencarian pada kamus yang memakan waktu lama karena pengguna harus mencari halaman per halaman dari buku kamus tersebut, pembawaan buku kamus yang tidak efektif, dan pembaharuan kata dalam kamus yang tidak bisa dilakukan secara cepat, karena ilmu teknologi informasi yang semakin berkembang.

4.2.1.3. Persyaratan Sistem

Pembuatan aplikasi kamus istilah teknologi informasi berbasis *web* ini harus memenuhi beberapa persyaratan yang meliputi kelengkapan *hardware* dan *software*. Kelengkapan *hardware* dan *software* yang digunakan penulis untuk membuat dan merancang aplikasi kamus istilah teknologi informasi berbasis *web* ini antara lain sebagai berikut:

Tabel 4.1 Persyaratan Sistem

<u>Hardware</u>	<u>Software</u>
Laptop Asus A43S, Processor Core i5 2,4 Ghz, RAM 4GB, Harddisk 320GB.	Windows 7, Notepad++ atau Sublime Text 3, Laragon 2.1.5, Browser Google Chrome, Git.

4.2.2. Fase Workshop Design

4.2.2.1. Perancangan Algoritma Boyer Moore

Pada tahap ini penulis mencoba membuat perancangan algoritma *boyer moore*, dengan adanya rancangan algoritma ini tahapan-tahapan dalam penerapan algoritma dari penelitian ini dapat dengan mudah diterapkan pada sistem yang dibangun.

4.2.2.1.1. Langkah-Langkah Algoritma Boyer Moore

Berikut ini adalah gambaran dari langkah-langkah algoritma *boyer moore* pada pencarian kamus yang terjadi ketika *user* menginputkan kata "*list*".

Teks (Y) : Mailing list

Pattern (X) : list

Bc : Bad Character

index	0	1	2	3	4	5	6	7	8	9	10	11
Y	M	a	i	l	i	n	g		l	i	s	t
X	l	i	s	t								
Bc	3	2	1	4								

Pada langkah pertama penyesuaian karakter akan dimulai pada karakter *pattern* “t” dimana karakter “t” tidak sama dengan karakter di atasnya pada teks (*miss match*) yaitu “l”, maka *pattern* akan bergeser sejumlah *bad character* “l” pada teks yaitu 3 langkah.

index	0	1	2	3	4	5	6	7	8	9	10	11
Y	M	a	i	l	i	n	g		l	i	s	t
X				l	i	s	t					
Bc				3	2	1	4					

Pada langkah kedua penyesuaian akan dimulai kembali pada karakter “t” karakter paling kanan *pattern* dimana karakter “t” tidak sama dengan karakter di atasnya pada teks yaitu “g”, maka *pattern* akan bergeser sejumlah *bad character* “g” pada teks. Karena “g” tidak tersedia pada *bad character pattern*, maka yang diambil adalah jumlah dari *pattern* yaitu 4 karakter, maka akan bergeser sebanyak 4 langkah.

index	0	1	2	3	4	5	6	7	8	9	10	11
Y	M	a	i	l	i	n	g		l	i	s	t
X								l	i	s	t	
Bc								3	2	1	4	

Pada langkah ketiga, penyesuaian akan dimulai kembali pada karakter “t” karakter paling kanan *pattern* dimana karakter “t” tidak sama dengan karakter di atasnya pada teks yaitu “s”, maka *pattern* akan bergeser sejumlah *bad character* “s” pada teks, yaitu 1 langkah.

index	0	1	2	3	4	5	6	7	8	9	10	11
Y	M	a	i	l	i	n	g		l	i	s	t
X									l	i	s	t
Bc									3	2	1	4

Pada langkah keempat penyesuaian akan dimulai kembali pada karakter “t” karakter paling kanan *pattern*, dimana karakter “t” sudah sesuai dengan

karakter pada teks, begitupun karakter “s”, “i”, dan “l” pada *pattern* sudah sesuai dengan teks, sehingga *pattern* sudah ditemukan di dalam teks pada *database* aplikasi kamus istilah teknologi informasi dan akan mengeluarkan hasil berupa istilah dari kata yang sudah ditemukan dalam pencarian yaitu kata *list*.

4.2.2.1.2. Pseudocode Algoritma Boyer Moore

Berikut ini merupakan bagian dari *pseudocode* algoritma *boyer moore* pada fungsi pencarian yang digunakan penulis dalam pembuatan aplikasi kamus istilah teknologi informasi berbasis *web*.

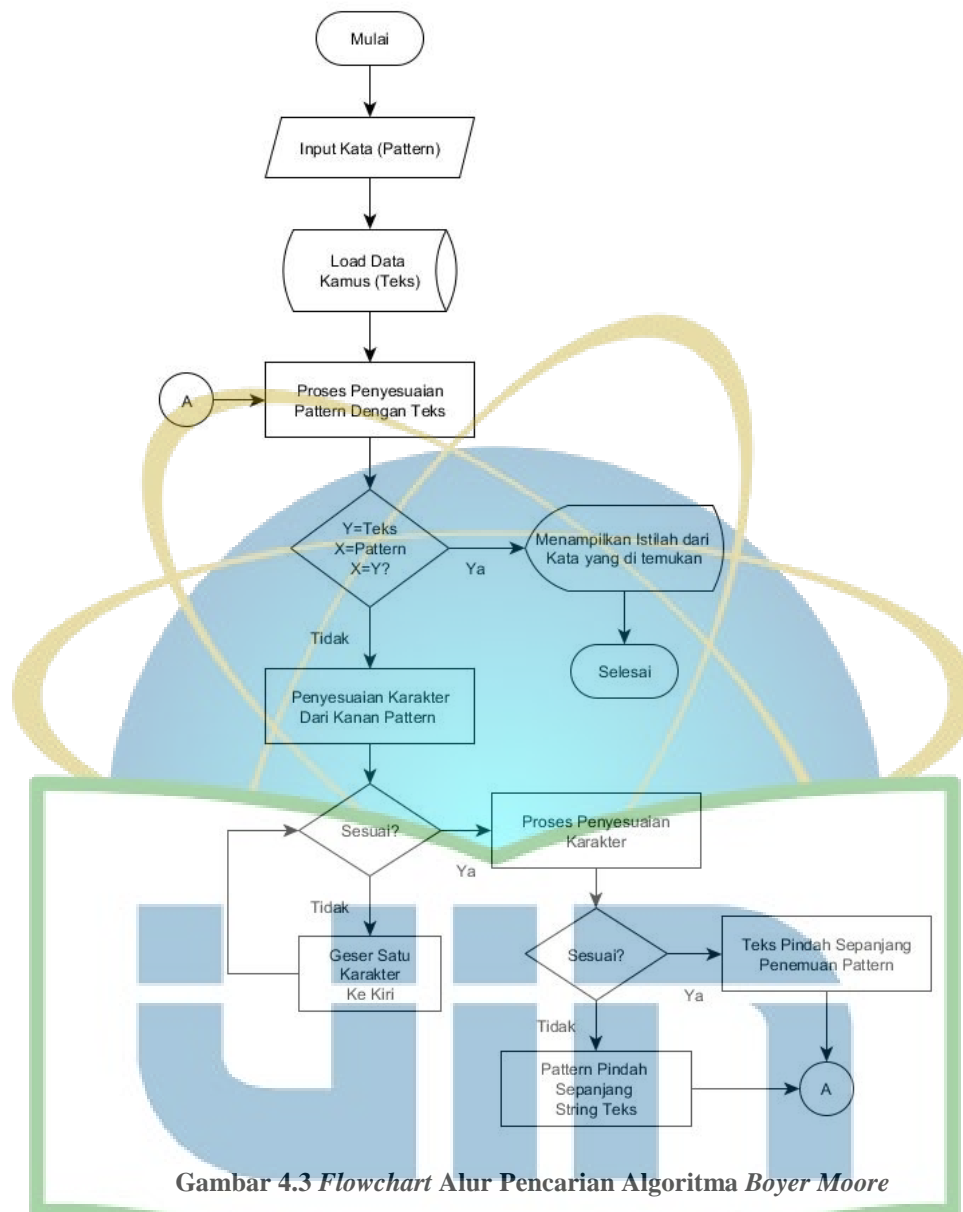
```

procedure BoyerMooreSearch(
  input m, n : integer,
  input P : array[0..n-1] of char,
  input T : array[0..m-1] of char,
  output ketemu : array[0..m-1] of boolean
)
Deklarasi:
i, j, shift, bmBcShift, bmGsShift: integer
BmBc : array[0..255] of integer
BmGs : array[0..n-1] of integer
Algoritme:
preBmBc(n, P, BmBc)
preBmGs(n, P, BmGs)
i:=0
while (i<= m-n) do
  j:=n-1
  while (j >=0 n and T[i+j] = P[j]) do
    j:=j-1
  endwhile
  if(j < 0) then
    ketemu[i]:=true;
  endif
  bmBcShift:= BmBc[chartoint(T[i+j])]-n+j+1
  bmGsShift:= BmGs[j]
  shift:= max(bmBcShift, bmGsShift)
  i:= i+shift

```

4.2.2.1.3. Flowchart Algoritma Boyer Moore

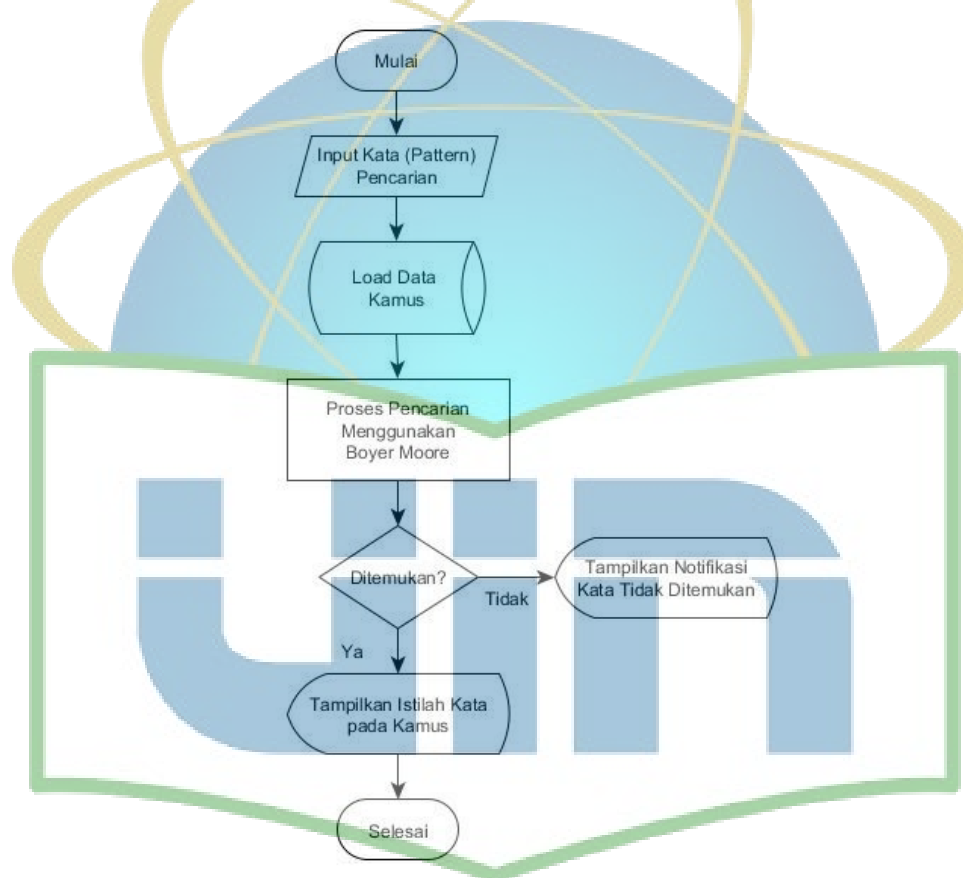
Pada tahap ini penulis melakukan beberapa penjelasan langkah-langkah dari metode algoritma *boyer moore* yang akan digambarkan dengan menggunakan diagram alur (*Flowchart*) sebagai berikut:



Pada gambar 4.3 merupakan alur dari algoritma *boyer moore* pada sistem. Pada tahap awal sistem akan melakukan pengambilan data pada *database* kamus istilah teknologi informasi, lalu akan mencocokkan dengan kata (*pattern*) yang dimasukan oleh *user*, tahap selanjutnya adalah penyesuaian dari kata yang dimasukan oleh *user* jika kata yang dimasukan sesuai maka alur yang dibuat akan selesai yang berarti kata yang di masukan oleh *user* sudah ditemukan, jika kata yang dimasukan tidak sesuai maka penyesuaian karakter akan dilakukan dari kanan *pattern* pada kata yang dicari, yang berarti dari akhir kata, jika tidak sesuai

maka karakter akan bergeser ke kiri, ke karakter selanjutnya, penyesuaian dilakukan sesuai dengan panjang karakter yang di inputkan *user*, jika sesuai teks akan berpindah sepanjang penemuan karakter, jika tidak sesuai teks akan berpindah sepanjang *string* teks hingga karakter teks sesuai dengan karakter kata yang di inputkan *user*.

Pada *flowchart* selanjutnya adalah rancangan alur dari pencarian kata (*pattern*) pada sistem kamus istilah teknologi informasi yang penulis coba buat, berikut ini adalah gambar dari rancangan tersebut:



Gambar 4.4 Flowchart Pencarian Kata (*Pattern*) Pada Kamus

Pada gambar 4.4 merupakan alur dari pencarian pada aplikasi kamus istilah teknologi informasi, pada tahap awal user harus melakukan *input* kata yang diinginkan, setelah melakukan *input*, sistem akan melakukan pengambilan data pada *database*, selanjutnya tahap proses metode *boyer moore* dilakukan jika

ditemukan maka data yang dimasukkan akan tampil pada sistem dan jika tidak maka akan menampilkan bahwa data yang dicari tidak ditemukan.

4.2.2.2. UML (*Unified Modeling Language*)

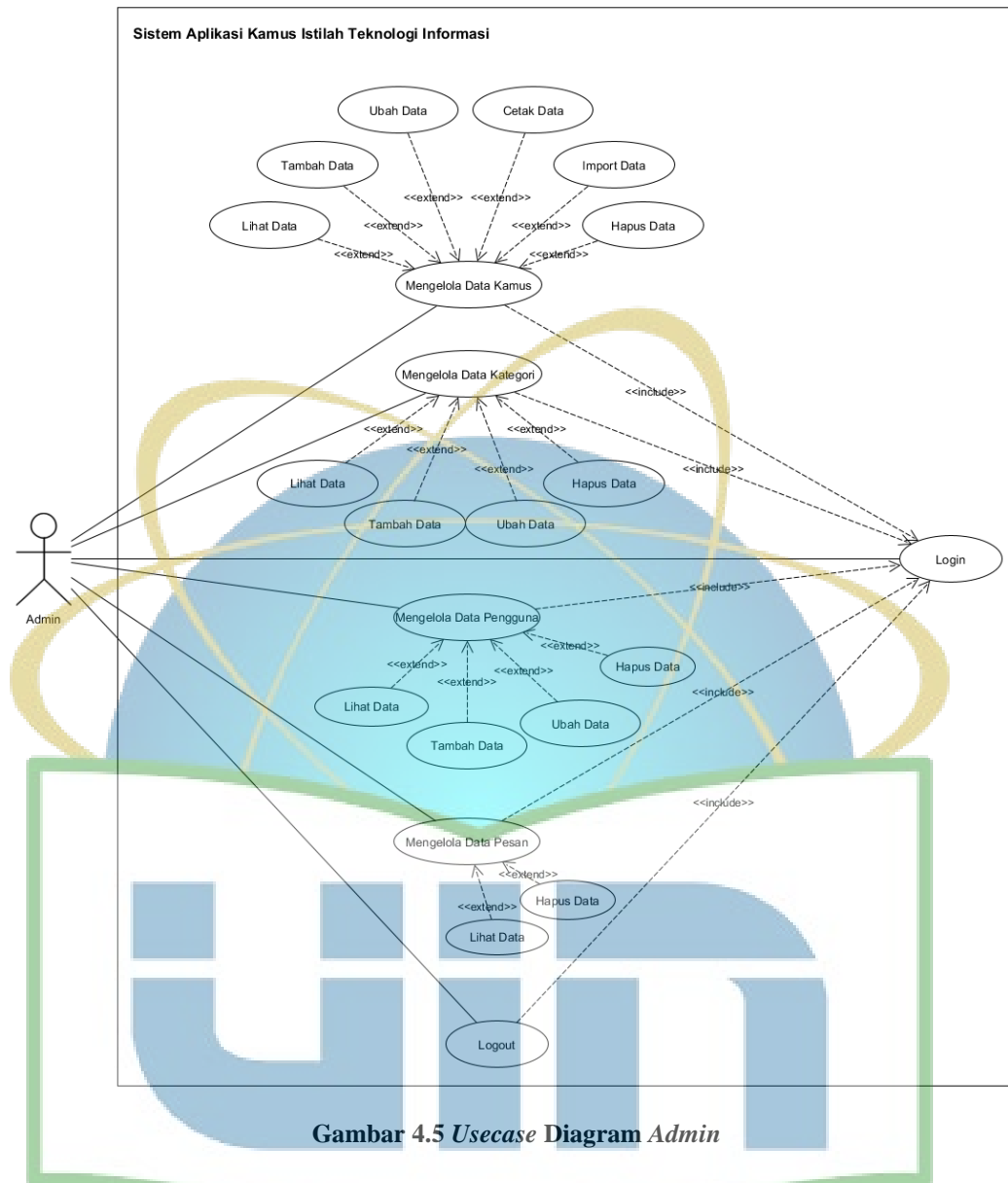
Pada tahap ini penulis mencoba membuat beberapa rancangan *Unified Modeling Language*, karena dengan adanya rancangan *Unified Modeling Language* pembuatan dan pengembangan sistem menjadi lebih mudah, berikut ini beberapa rancangan *Unified Modeling Language* yang terdapat pada aplikasi kamus teknologi informasi berbasis *web*.

4.2.2.2.1. *Usecase Diagram*

Berikut ini merupakan penjelasan dari *usecase* diagram dari aplikasi kamus istilah teknologi informasi berbasis *web* yang penulis uraikan dari level *admin* dan *user*:

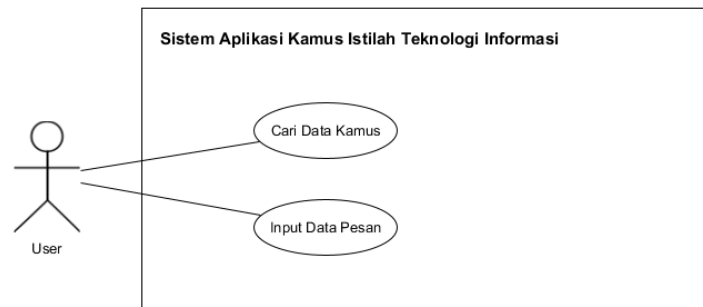
1. *Admin*

Pada level ini, *admin* dapat mengelola data kamus yang di dalamnya bisa menambah data pada kamus, mengubah data pada kamus, melihat data pada kamus dan menghapus data pada kamus. Kemudian *admin* dapat mengelola data kategori yang di dalamnya bisa menambah data kategori, mengubah data kategori, melihat data kategori dan menghapus data kategori. Kemudian *admin* dapat mengelola data pesan yang di dalamnya bisa melihat data pesan yang dikirim oleh *user* dan menghapus data pesan yang dikirim oleh *user* pada sistem, sebagaimana yang terdapat pada gambar 4.5 dibawah.



2. User

Pada level ini, *user* dapat mencari data kamus pada aplikasi dan menambah data pesan untuk dikirim kepada *admin*, dapat dilihat pada gambar 4.6 dibawah.



Gambar 4.6 Usecase Diagram User

4.2.2.2.2. Activity Diagram

Activity diagram pada aplikasi kamus teknologi informasi ini dibuat untuk menggambarkan alur dari aktivitas suatu sistem dan proses berjalannya sistem. Berikut ini adalah *activity* diagram dari aplikasi kamus teknologi informasi berbasis web.

1. Admin Mengelola Data Kamus

Di bawah ini merupakan penjelasan dari *activity* diagram untuk *admin* dalam mengelola data kamus pada sistem kamus teknologi informasi berbasis web untuk memberikan gambaran aktivitas dalam melakukan penambahan data, ubah data, hapus data, lihat data.

a. Tambah Data Kamus

Aktor pada level *admin* yang berinteraksi dengan sistem dalam proses penambahan data kamus, terlebih dahulu *admin* melakukan *login* pada aplikasi dengan memilih halaman login kemudian memasukkan *username* dan *password*, memilih menu *admin*, memilih menu kamus, memilih tombol tambah data kamus, mengisi *form* yang telah disediakan pada sistem dan kemudian memilih tombol *save* sehingga data yang dimasukkan dapat disimpan kedalam *database*, rancangan tersebut dapat dilihat pada gambar 4.7.

b. Ubah Data Kamus

Aktor pada level *admin* yang berinteraksi dengan sistem dalam proses pengubahan data kamus, terlebih dahulu *admin* melakukan *login* pada aplikasi dengan memilih halaman *login* kemudian memasukkan *username* dan *password*, memilih menu *admin*, memilih menu kamus, memilih data kamus pada tabel

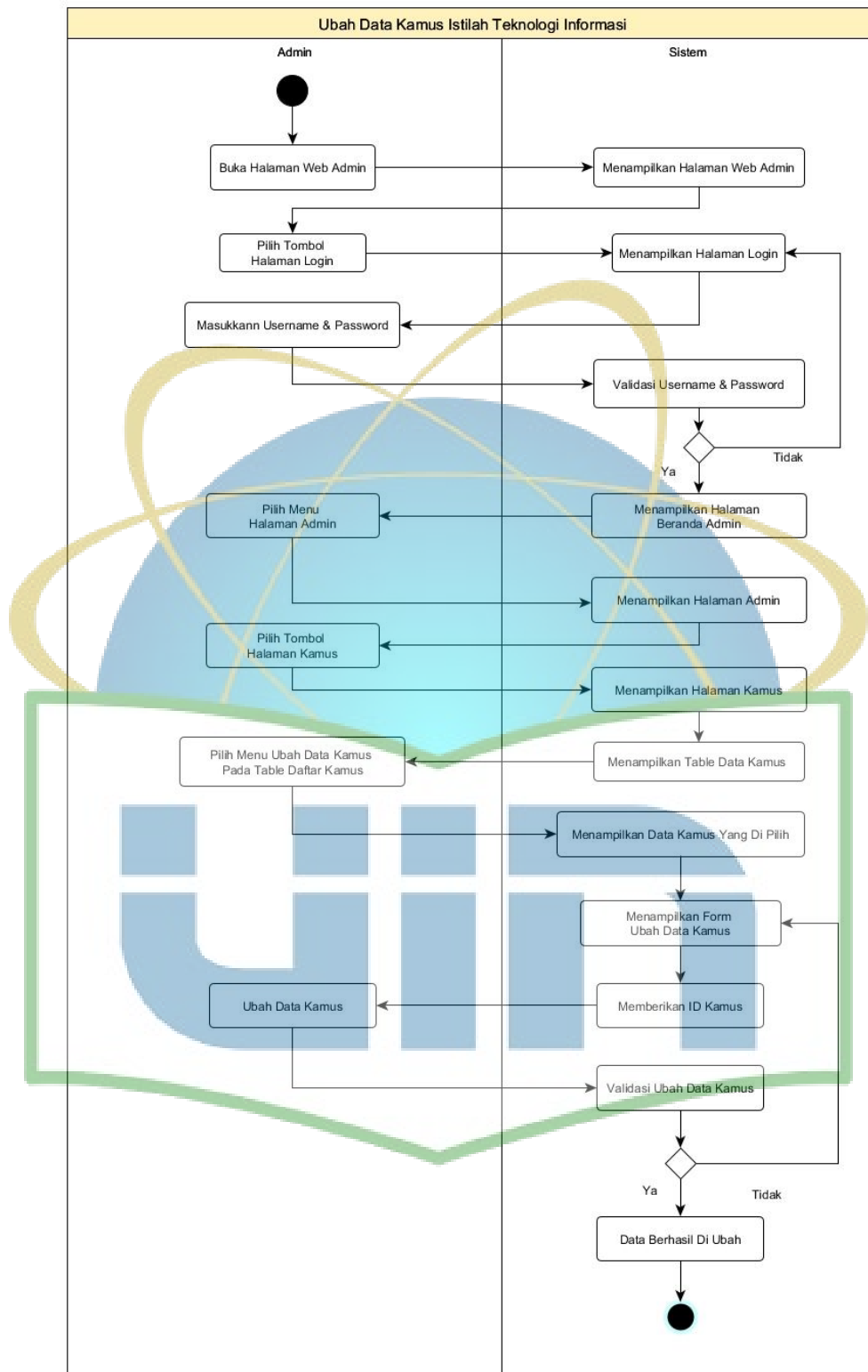
kamus, mengisi *form edit* yang telah disediakan pada sistem dan kemudian memilih tombol *edit* sehingga data yang dimasukkan dapat diubah kedalam *database*, rancangan tersebut dapat dilihat pada gambar 4.8.

c. Lihat Data Kamus

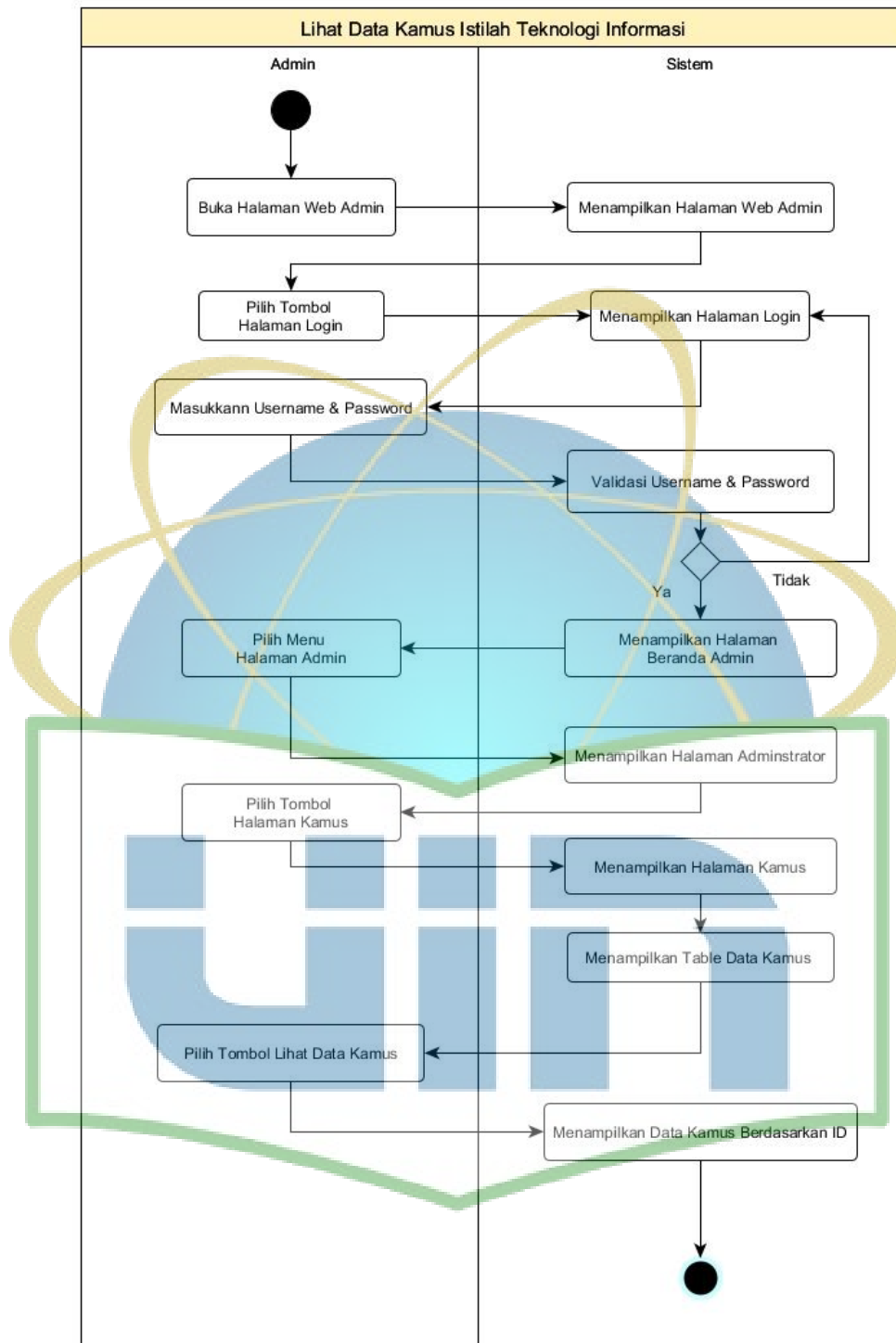
Aktor pada level *admin* yang berinteraksi dengan sistem dalam proses melihat data kamus, terlebih dahulu admin melakukan *login* pada aplikasi dengan memilih halaman *login* kemudian memasukkan *username* dan *password*, memilih menu *admin*, memilih menu kamus, dan menu halaman kamus ditampilkan pada sistem sehingga *admin* dapat melihat data kamus, rancangan tersebut dapat dilihat pada gambar 4.9.

d. Hapus Data Kamus

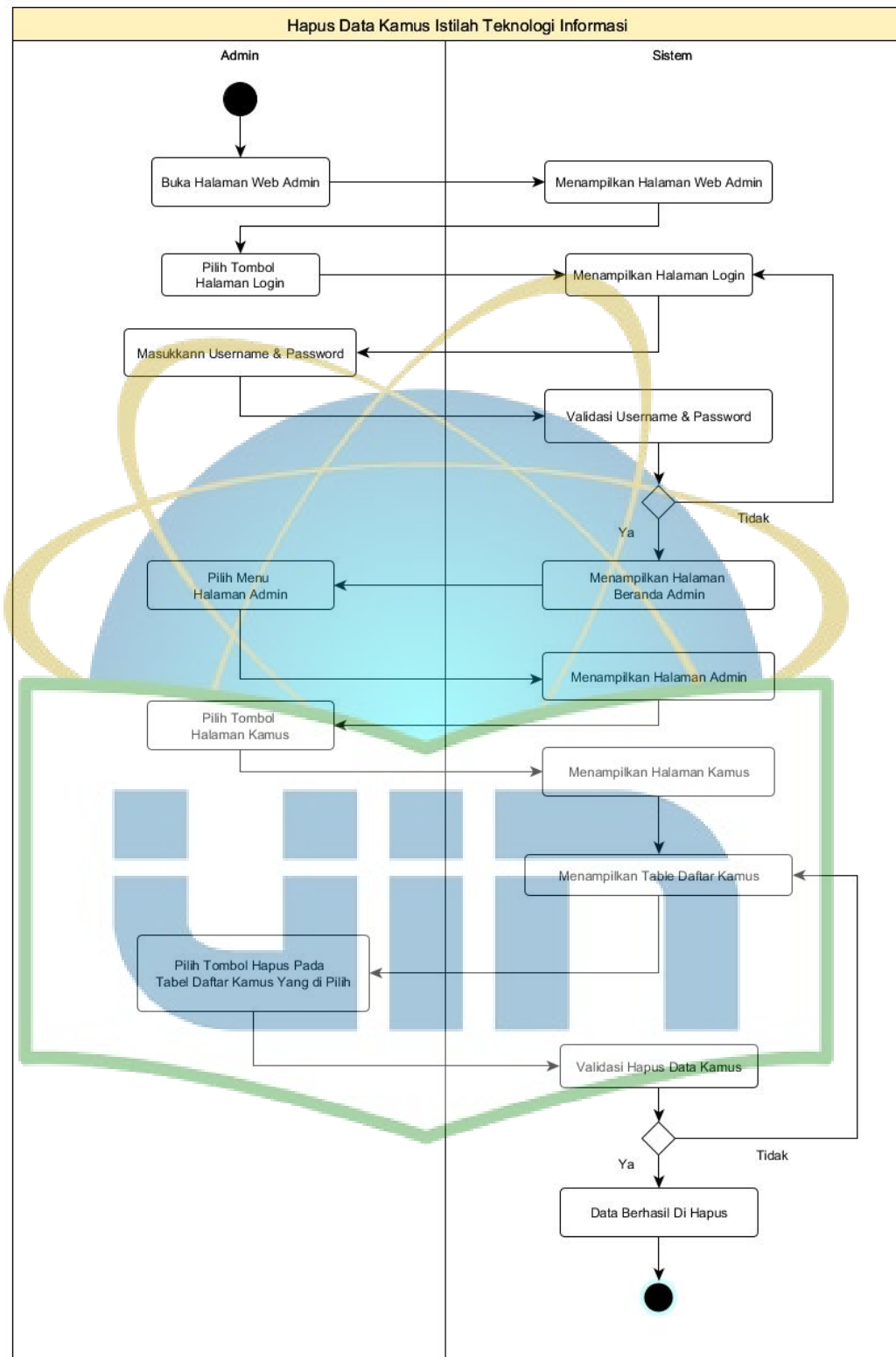
Aktor pada level admin yang berinteraksi dengan sistem dalam proses pengubahan data kamus, terlebih dahulu *admin* melakukan *login* pada aplikasi dengan memilih halaman *login* kemudian memasukkan *username* dan *password*, memilih menu admin, memilih menu kamus, memilih data kamus yang ingin dihapus pada *table* kamus, memilih tombol hapus pada *table* kamus, sehingga data yang dipilih dapat dihapus kedalam *database*, rancangan tersebut dapat dilihat pada gambar 4.10.



Gambar 4.8 Activity Diagram Admin Ubah Data Kamus



Gambar 4.9 Activity Diagram Admin Lihat Data Kamus



Gambar 4.10 Activity Diagram Admin Hapus Data Kamus

2. Admin Mengelola Data Kategori

Di bawah ini merupakan *activity* diagram untuk *admin* mengelola data kategori pada sistem aplikasi kamus istilah teknologi informasi berbasis *web* memberikan gambaran aktivitas untuk melakukan penambahan data, lihat data, ubah data, hapus data.

a. Tambah Data Kategori

Aktor *admin* yang berinteraksi dengan sistem dalam proses penambahan data kategori, terlebih dahulu *admin* melakukan *login* pada aplikasi dengan memilih halaman *login* kemudian memasukkan *username* dan *password*, memilih menu *admin*, memilih menu kategori, memilih tombol tambah data kategori, mengisi *form* yang telah disediakan pada sistem dan kemudian memilih tombol *save* sehingga data yang dimasukkan dapat disimpan kedalam *database*, rancangan tersebut dapat dilihat pada Gambar 4.11.

b. Ubah Data Kategori

Aktor *admin* yang berinteraksi dengan sistem dalam proses pengubahan data kategori, terlebih dahulu *admin* melakukan *login* pada aplikasi dengan memilih halaman *login* kemudian memasukkan *username* dan *password*, memilih menu *admin*, memilih menu kategori, memilih data kategori pada *table* kategori, mengisi *form edit* yang telah disediakan pada sistem dan kemudian memilih tombol *edit* sehingga data yang diinputkan dapat diubah kedalam *database*, rancangan tersebut dapat dilihat pada gambar 4.12.

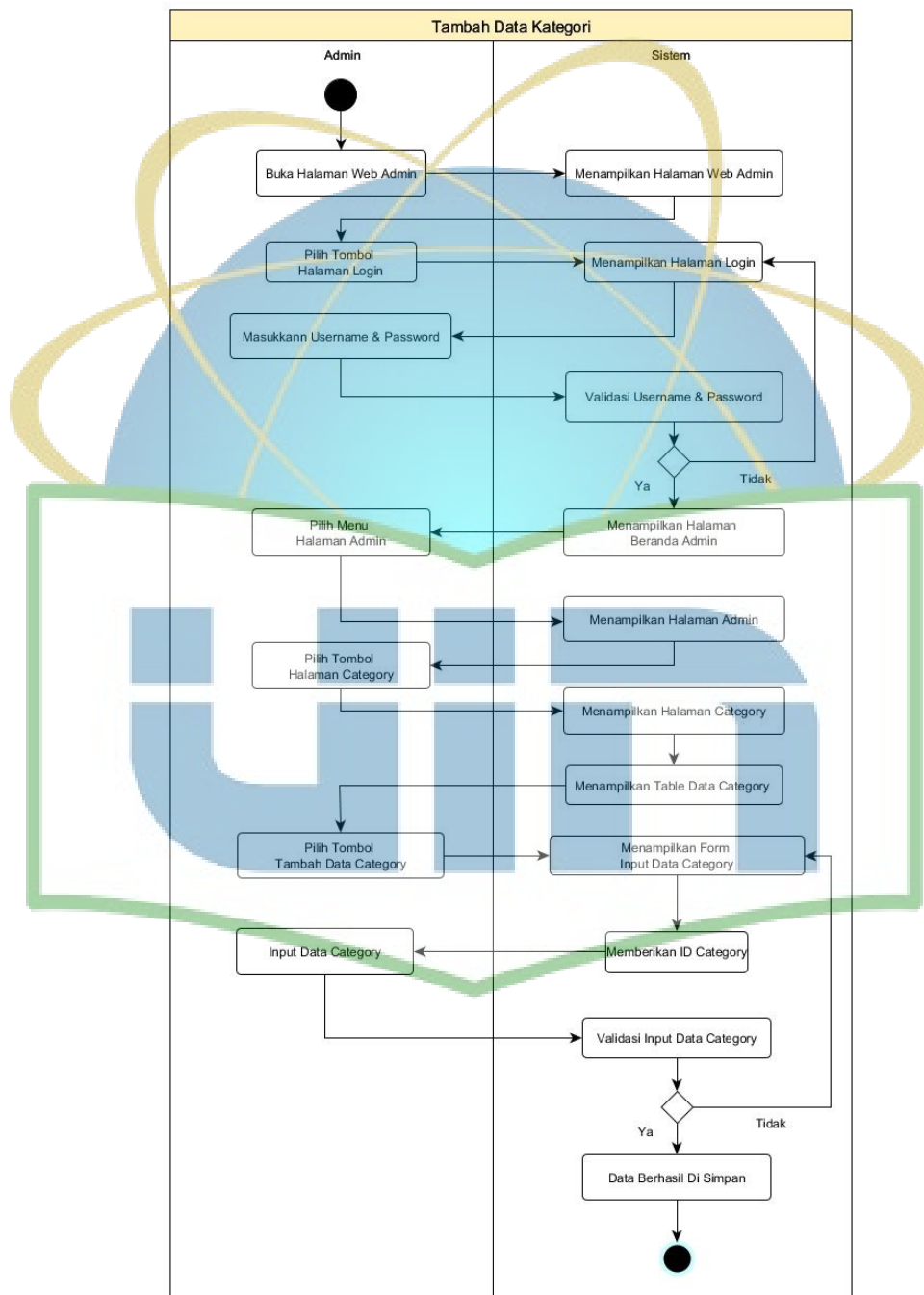
c. Lihat Data Kategori

Aktor *admin* yang berinteraksi dengan sistem dalam proses melihat data kategori, terlebih dahulu *admin* melakukan *login* pada aplikasi dengan memilih halaman *login* kemudian memasukkan *username* dan *password*, memilih menu *admin*, memilih menu kategori, dan menu halaman kategori ditampilkan pada sistem sehingga *admin* dapat melihat data kategori, rancangan tersebut dapat dilihat pada gambar 4.13.

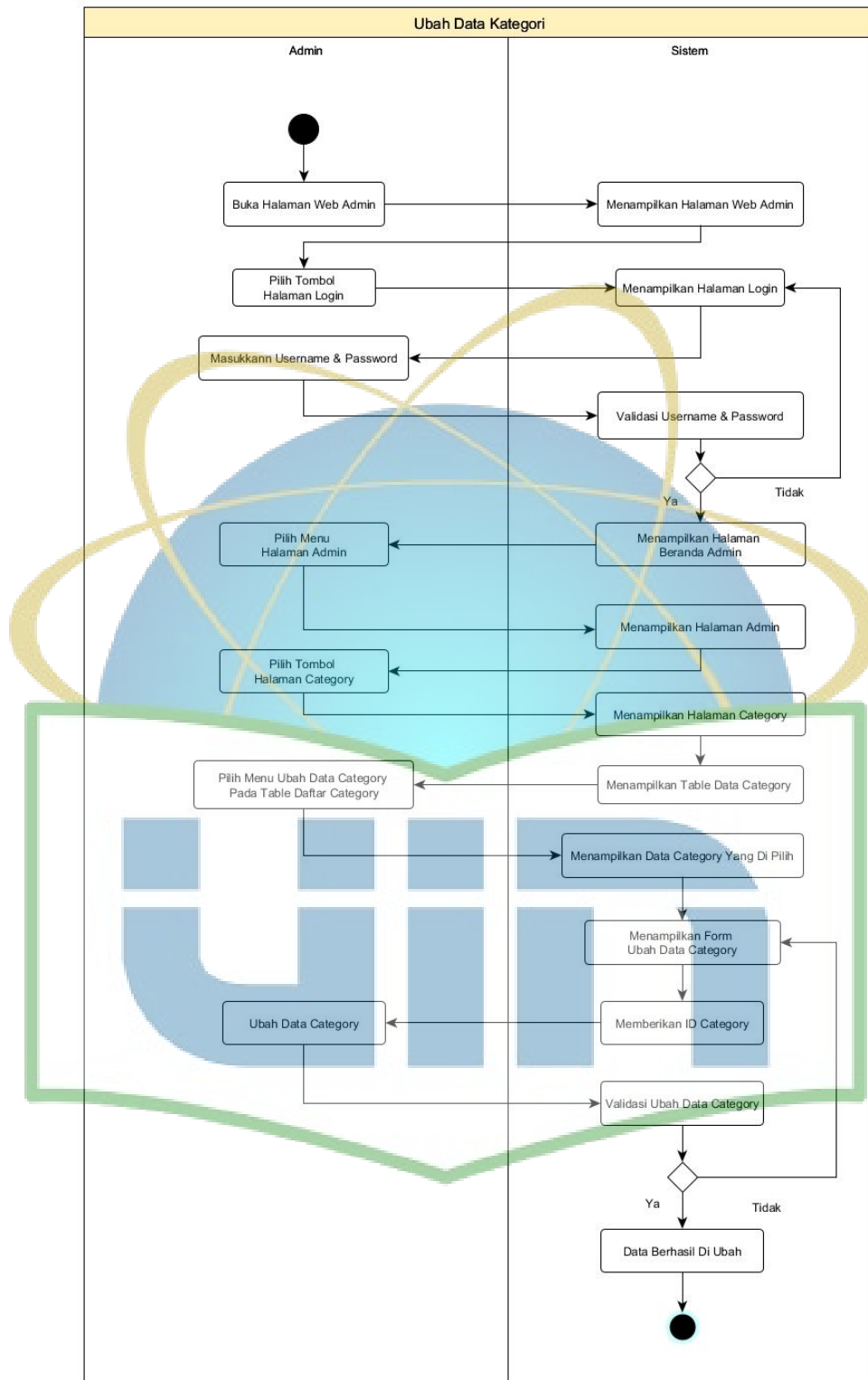
d. Hapus Data Kategori

Aktor *admin* yang berinteraksi dengan sistem dalam proses pengubahan data kamus, terlebih dahulu *admin* melakukan *login* pada aplikasi dengan memilih

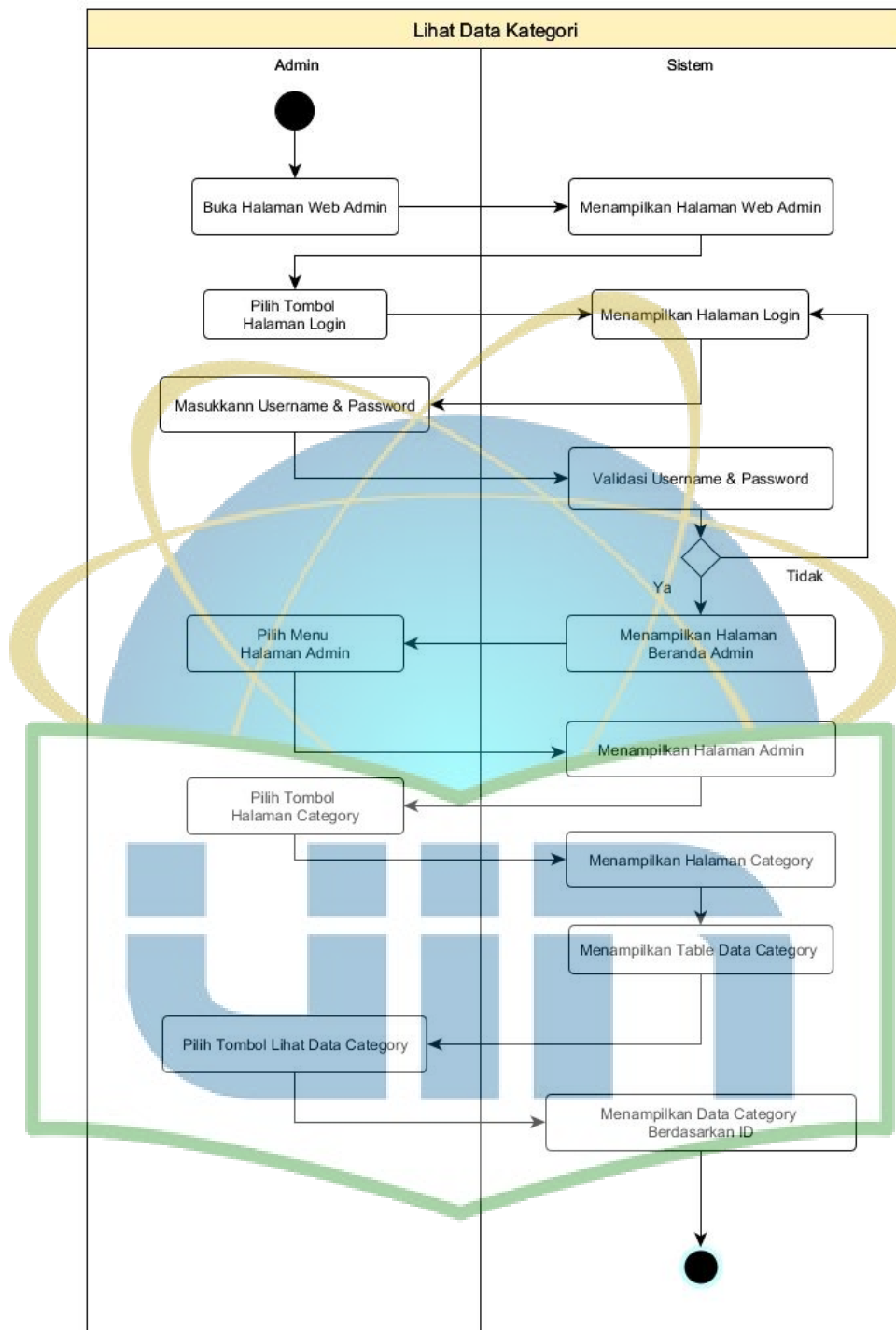
halaman *login* kemudian memasukan *username* dan *password*, memilih menu *admin*, memilih menu kategori, memilih data kategori yang ingin dihapus pada *table* kategori, memilih tombol hapus pada *table* kategori, sehingga data yang dipilih dapat dihapus kedalam *database*, rancangan tersebut dapat dilihat pada gambar 4.14.



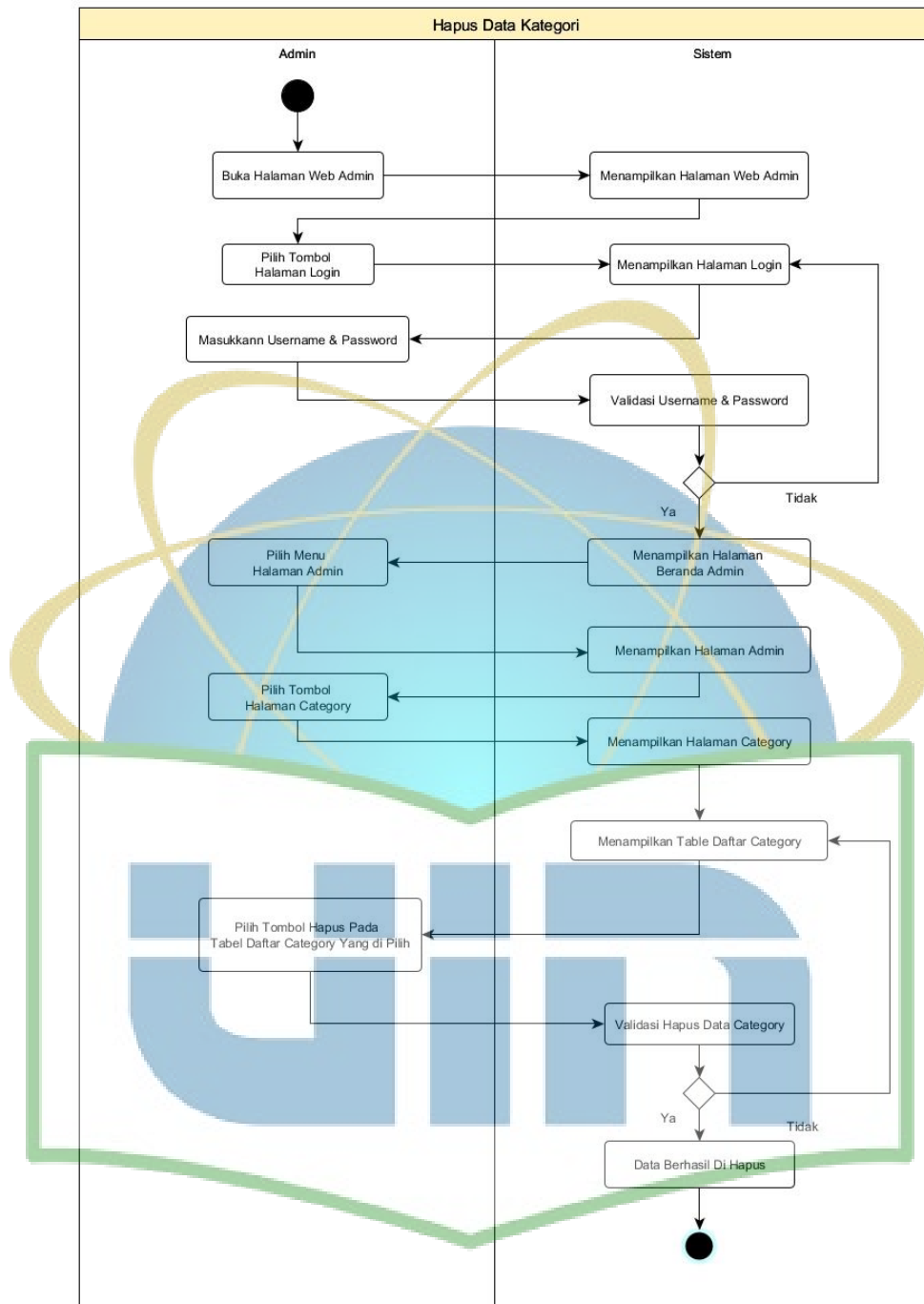
Gambar 4.11 Activity Diagram Admin Tambah Data Kategori



Gambar 4.12 Activity Diagram Admin Ubah Data Kategori



Gambar 4.13 Activity Diagram Admin Lihat Data Kategori



Gambar 4.14 Activity Diagram Admin Hapus Data Kategori

3. Admin Mengelola Data Pesan

Di bawah ini merupakan *activity* diagram untuk *admin* mengelola data pesan pada sistem aplikasi kamus teknologi informasi berbasis *web* untuk

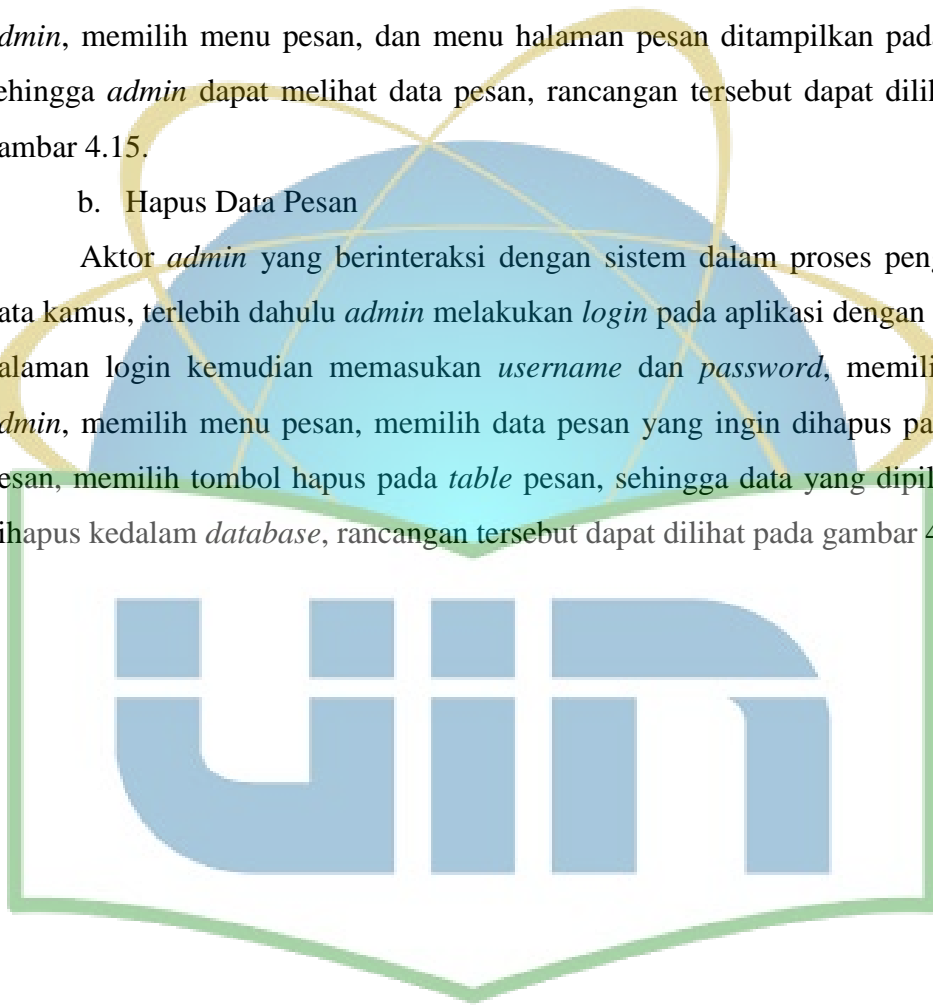
memberikan gambaran aktivitas untuk melakukan lihat data pesan dari pengguna, dan hapus data pesan dari pengguna.

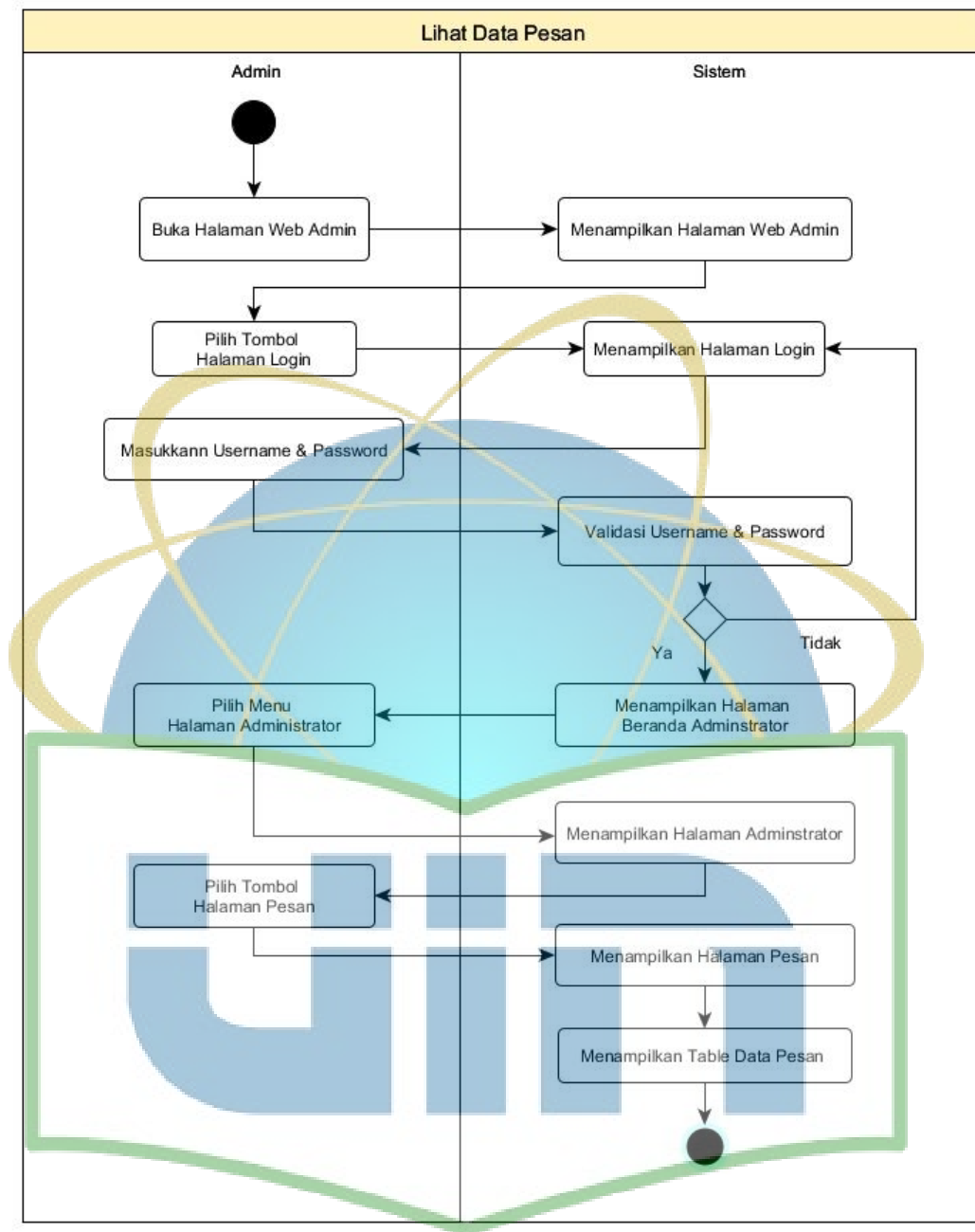
a. Lihat Data Pesan

Aktor *admin* yang berinteraksi dengan sistem dalam proses melihat data pesan, terlebih dahulu admin melakukan *login* pada aplikasi dengan memilih halaman login kemudian memasukkan *username* dan *password*, memilih menu *admin*, memilih menu pesan, dan menu halaman pesan ditampilkan pada sistem sehingga *admin* dapat melihat data pesan, rancangan tersebut dapat dilihat pada gambar 4.15.

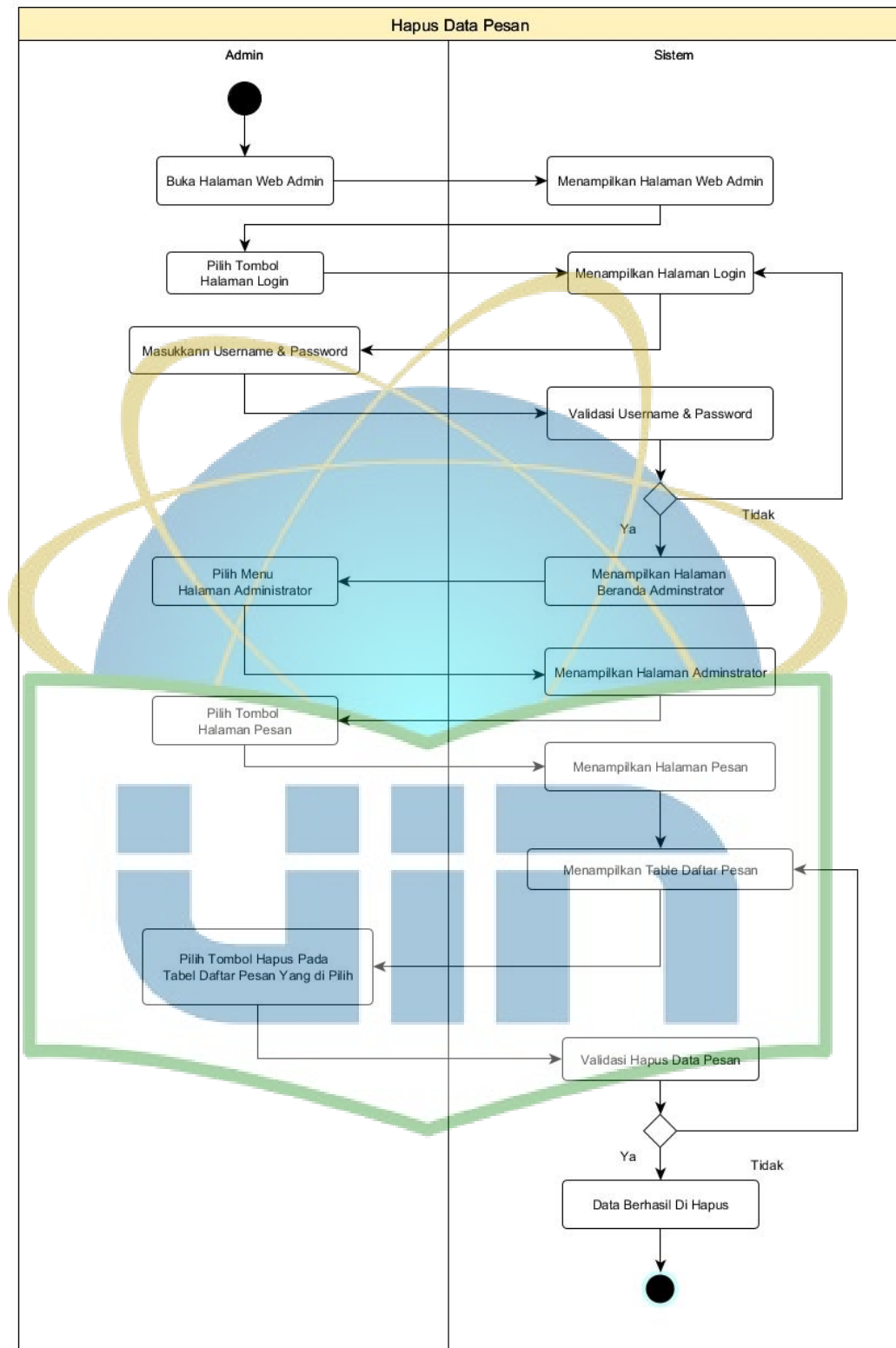
b. Hapus Data Pesan

Aktor *admin* yang berinteraksi dengan sistem dalam proses pengubahan data kamus, terlebih dahulu *admin* melakukan *login* pada aplikasi dengan memilih halaman login kemudian memasukkan *username* dan *password*, memilih menu *admin*, memilih menu pesan, memilih data pesan yang ingin dihapus pada *table* pesan, memilih tombol hapus pada *table* pesan, sehingga data yang dipilih dapat dihapus kedalam *database*, rancangan tersebut dapat dilihat pada gambar 4.16.





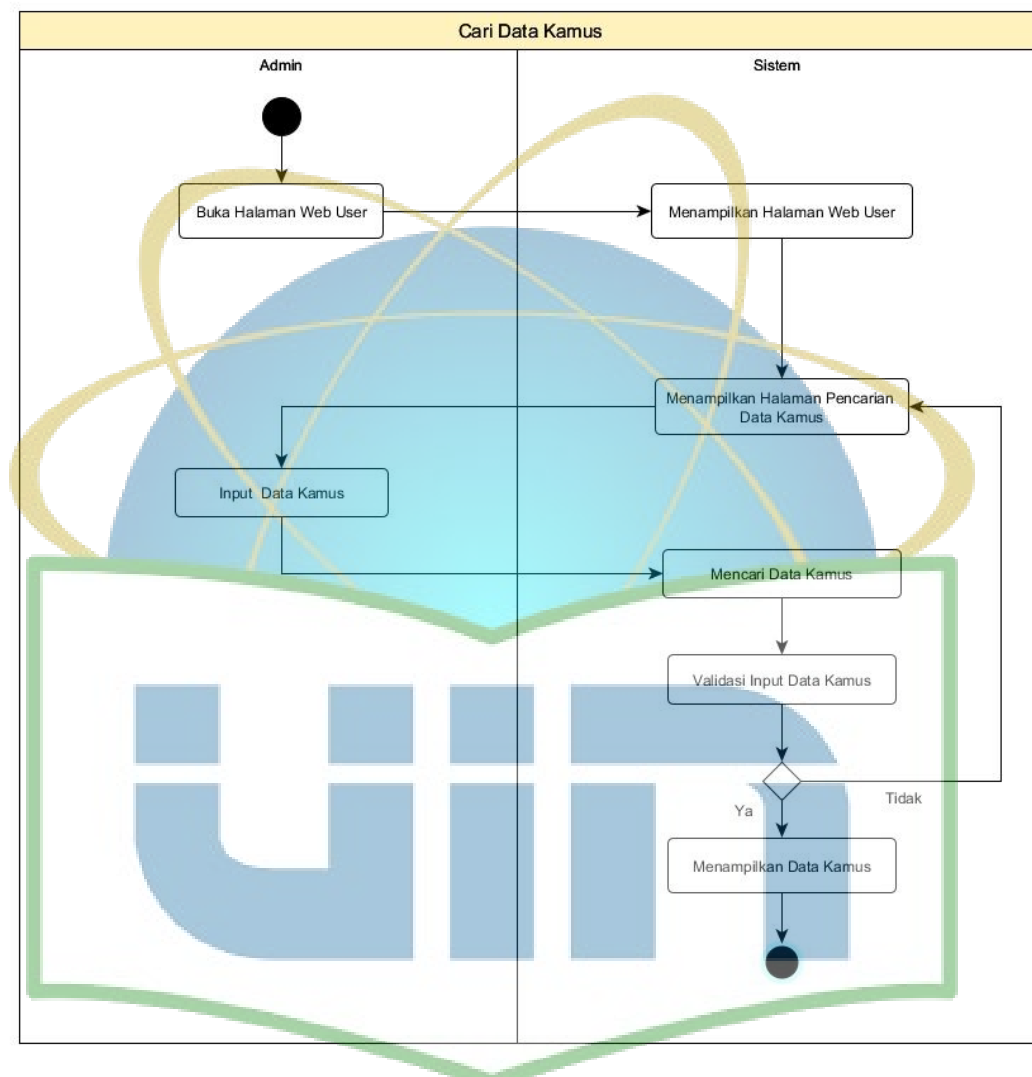
Gambar 4.15 Activity Diagram Admin Lihat Data Pesan



Gambar 4.16 Activity Diagram Admin Hapus Data Pesan

4. User Mencari Data Kamus

Di bawah ini merupakan *activity* diagram untuk user mengelolah *data* kamus pada sistem aplikasi kamus teknologi informasi berbasis *web* untuk memberikan gambaran aktifitas melakukan pencarian data kamus.



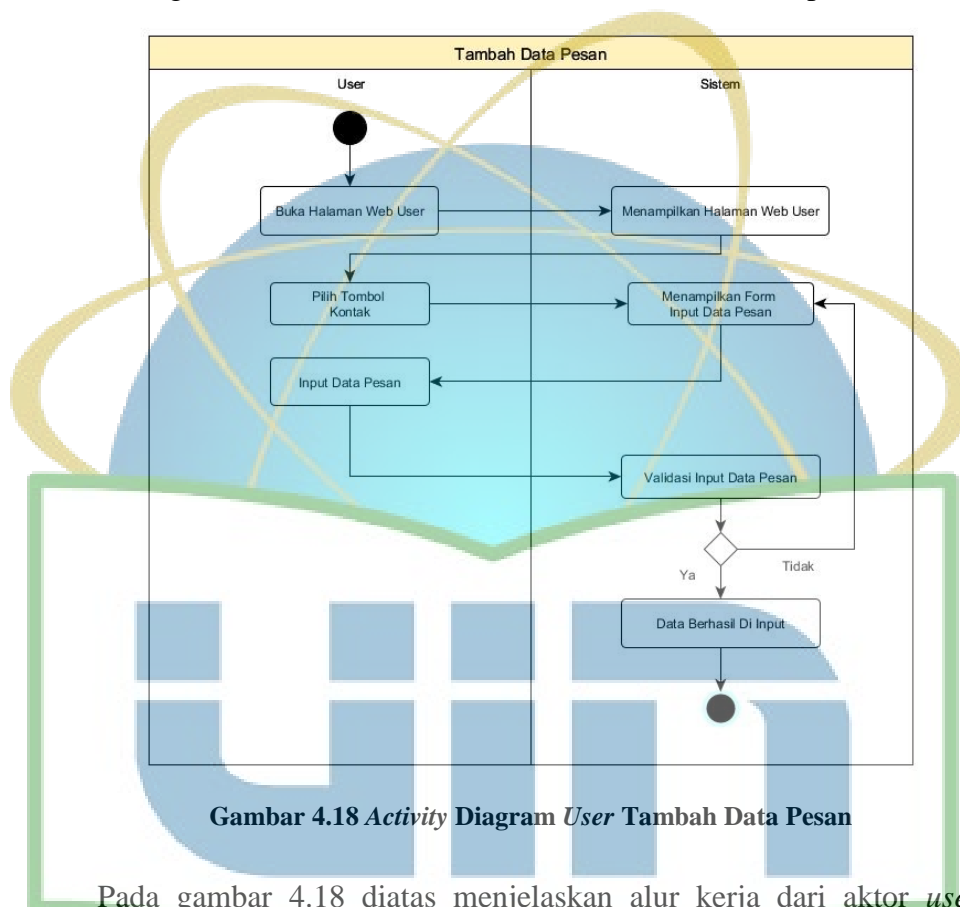
Gambar 4.17 Activity Diagram User Cari Data Kamus

Pada gambar 4.17 diatas menjelaskan alur kerja dari aktor *user* yang berinteraksi dengan sistem dalam proses pencarian data kamus, metode yang digunakan pada aplikasi kamus istilah teknologi informasi yaitu metode *boyer moore*, kemudian *user* melakukan *input* pada halaman pencarian, sistem akan melakukan pencarian data yang diinputkan user dan menjalankan validasi, jika input yang dimaksud ada maka akan muncul istilah dari kata yang dicari,

sedangkan jika kata yang dicari tidak tersedia, maka akan dikembalikan ke halaman pencarian untuk mencari kata yang lain.

5. *User Menambah Data Pesan*

Di bawah ini merupakan activity diagram untuk user mengelolah data kamus pada sistem aplikasi kamus teknologi informasi berbasis web untuk memberikan gambaran aktifitas untuk melakukan tambah data pesan.



Pada gambar 4.18 diatas menjelaskan alur kerja dari aktor *user* yang berinteraksi dengan sistem dalam proses penambahan data pesan, terlebih dahulu user memilih tombol tambah data pesan, mengisi *form* yang telah disediakan pada sistem dan kemudian memilih tombol kirim sehingga data yang telah dimasukkan dapat disimpan kedalam *database*.

4.2.2.2.3. *Sequence Diagram*

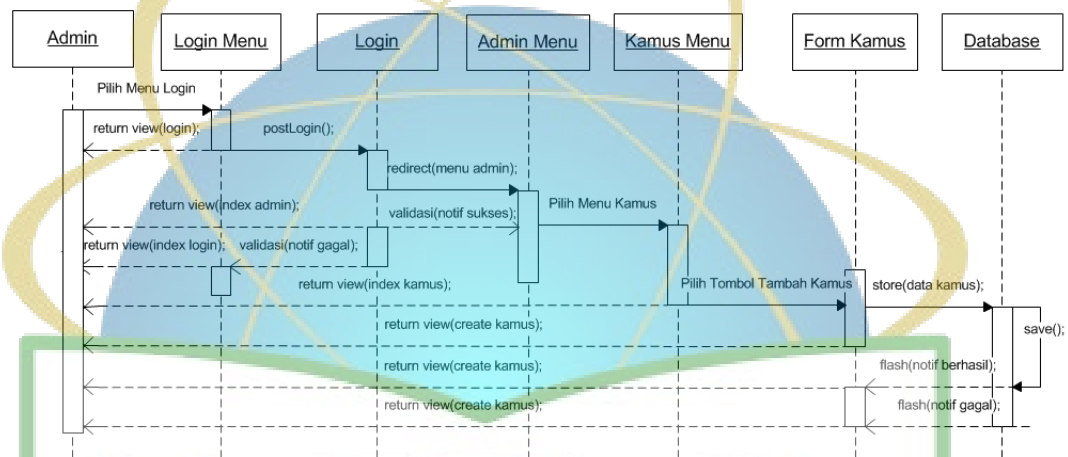
Sequence diagram pada aplikasi kamus istilah teknologi informasi ini dibuat untuk menjelaskan interaksi dari setiap urutan langkah yang terjadi dari

sistem yang dibangun. Berikut ini merupakan *sequence* diagram pada aplikasi kamus istilah teknologi informasi dengan level *admin*.

1. Admin Mengelola Data Kamus

Di bawah ini merupakan *sequence* diagram untuk *admin* mengelola data kamus pada sistem kamus teknologi informasi berbasis *web* menjelaskan interaksi user terhadap sistem untuk melakukan penambahan data, ubah data, hapus data, lihat data.

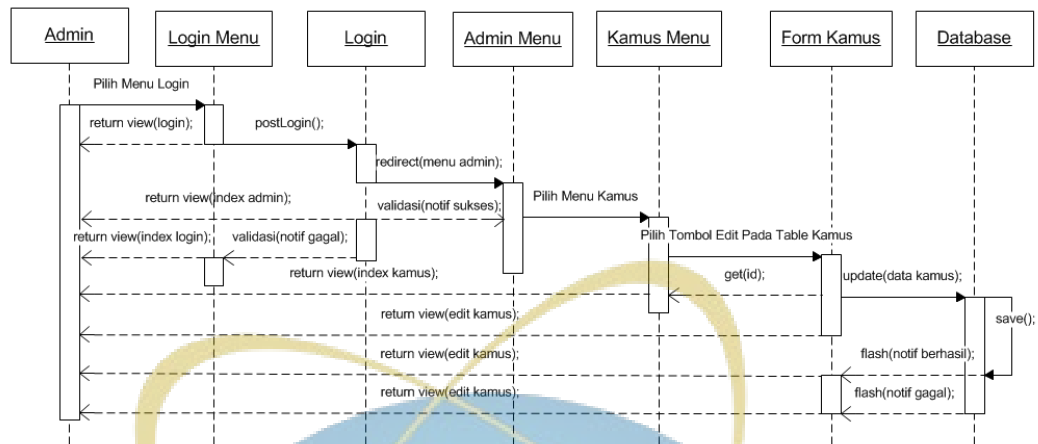
a. Tambah Data Kamus



Gambar 4.19 Sequence Diagram Admin Tambah Data Kamus

Pada gambar 4.19 diatas menjelaskan alur kerja dari *admin* yang berinteraksi dengan sistem dalam proses penambahan data kamus, *admin* terlebih dahulu melakukan *login* kedalam sistem, jika gagal maka akan menampilkan notifikasi dan akan melakukan pemuatan ulang halaman *login admin*, dan jika berhasil maka akan diarahkan ke dalam menu *admin*, lalu *admin* memilih menu kamus, dan pilih tombol tambah data kamus, melakukan pengisian *form* yang telah disediakan pada sistem dan kemudian memilih tombol *save* sehingga data yang telah dimasukkan dapat di simpan kedalam *database*.

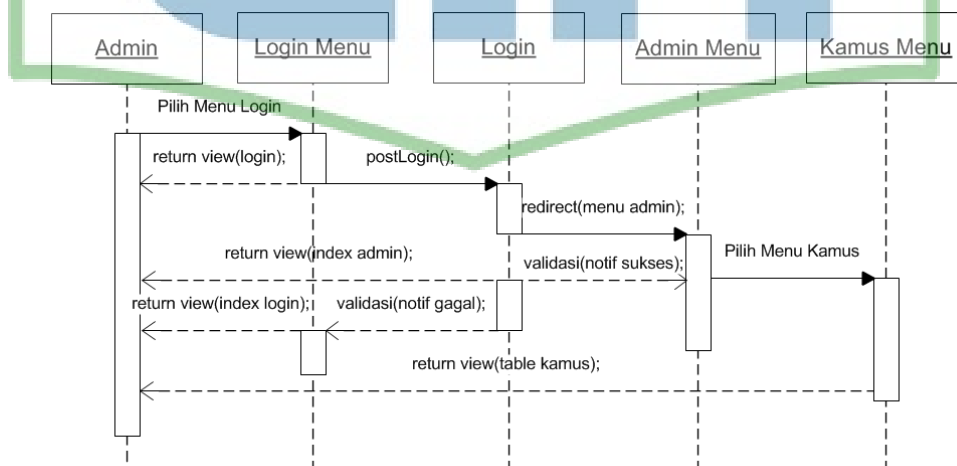
b. Ubah Data Kamus



Gambar 4.20 Sequence Diagram Admin Ubah Data Kamus

Pada gambar 4.20 diatas menjelaskan alur kerja dari *admin* yang berinteraksi dengan sistem dalam proses pengubahan data kamus, *admin* terlebih dahulu melakukan *login* kedalam sistem, jika gagal maka akan menampilkan notifikasi dan akan melakukan pemuatan ulang halaman *login admin*, dan jika berhasil maka akan diarahkan ke dalam menu *admin*, lalu *admin* memilih menu kamus, dan memilih tombol *edit* pada *table* kamus, melakukan pengisian *form* yang telah disediakan pada sistem dan kemudian memilih tombol *edit* sehingga data yang telah dimasukkan dapat disimpan kedalam *database*.

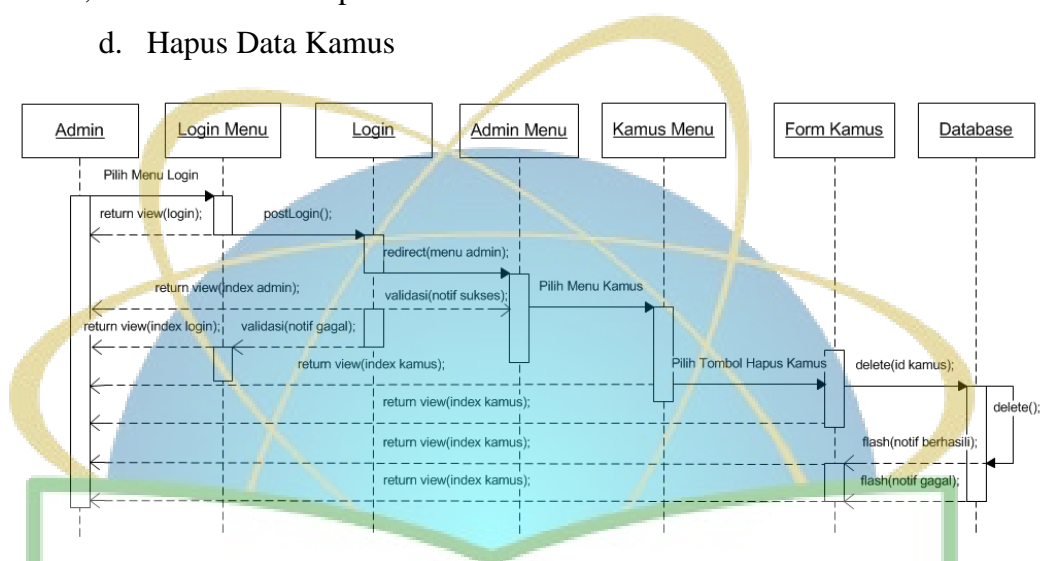
c. Lihat Data Kamus



Gambar 4.21 Sequence Diagram Admin Lihat Data Kamus

Pada gambar 4.21 diatas menjelaskan alur kerja dari *admin* yang berinteraksi dengan sistem dalam proses melihat data kamus, *admin* terlebih dahulu melakukan *login* kedalam sistem, jika gagal maka akan menampilkan notifikasi dan akan melakukan pemuatan ulang halaman *login admin*, dan jika berhasil maka akan diarahkan ke dalam menu *admin*, lalu *admin* memilih menu kamus, dan sistem menampilkan *table* kamus.

d. Hapus Data Kamus



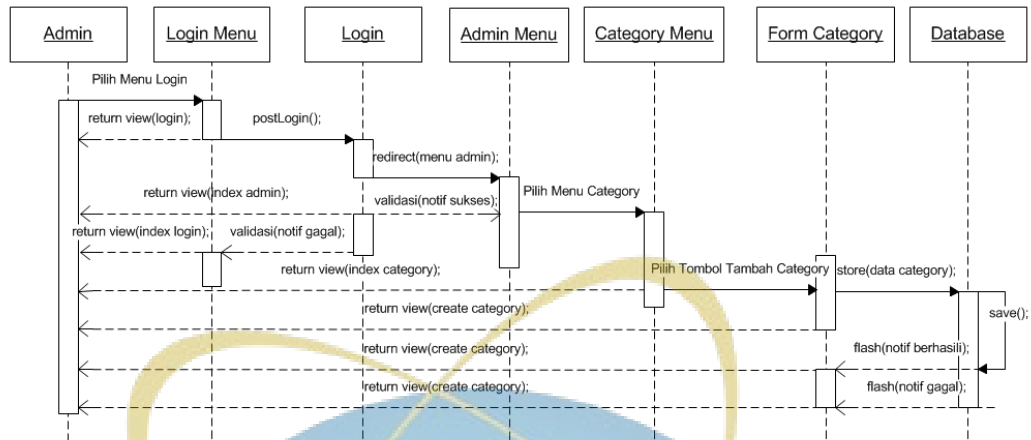
Gambar 4.22 Sequence Diagram Admin Hapus Data Kamus

Pada gambar 4.22 diatas menjelaskan alur kerja dari *admin* yang berinteraksi dengan sistem dalam proses penghapusan data kamus, *admin* terlebih dahulu melakukan *login* kedalam sistem, jika gagal maka akan menampilkan notifikasi dan akan melakukan pemuatan ulang halaman *login admin*, dan jika berhasil maka akan diarahkan ke dalam menu *admin*, lalu *admin* memilih menu kamus, dan memilih tombol hapus pada *table* kamus, sehingga data yang dipilih dapat dihapus pada *database*.

2. Admin Mengelola Data Kategori

Di bawah ini merupakan *sequence* diagram untuk *admin* mengelola data pengguna pada sistem kamus teknologi informasi berbasis *web* menjelaskan interaksi *admin* terhadap sistem untuk melakukan penambahan data kategori, lihat data kategori, ubah data kategori, hapus data kategori.

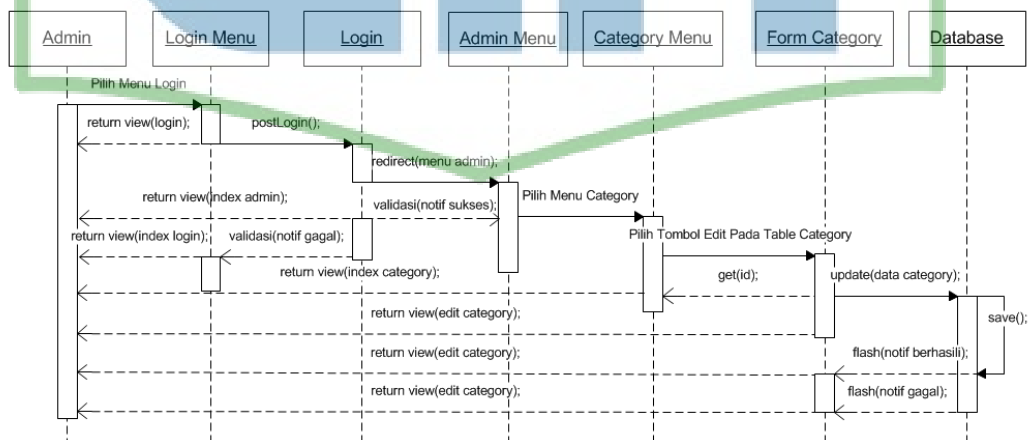
a. Tambah Data Kategori



Gambar 4.23 Sequence Diagram Admin Tambah Data Kategori

Pada gambar 4.23 diatas menjelaskan alur kerja dari *admin* yang berinteraksi dengan sistem dalam proses penambahan data kategori, *admin* terlebih dahulu melakukan *login* kedalam sistem, jika gagal maka akan menampilkan notifikasi dan akan melakukan pemuatan ulang halaman *login admin*, dan jika berhasil maka akan diarahkan ke dalam menu *admin*, lalu *admin* memilih menu kategori, dan pilih tombol tambah data kategori, melakukan pengisian *form* yang telah disediakan pada sistem dan kemudian memilih tombol *save* sehingga data yang dimasukkan dapat disimpan kedalam *database*.

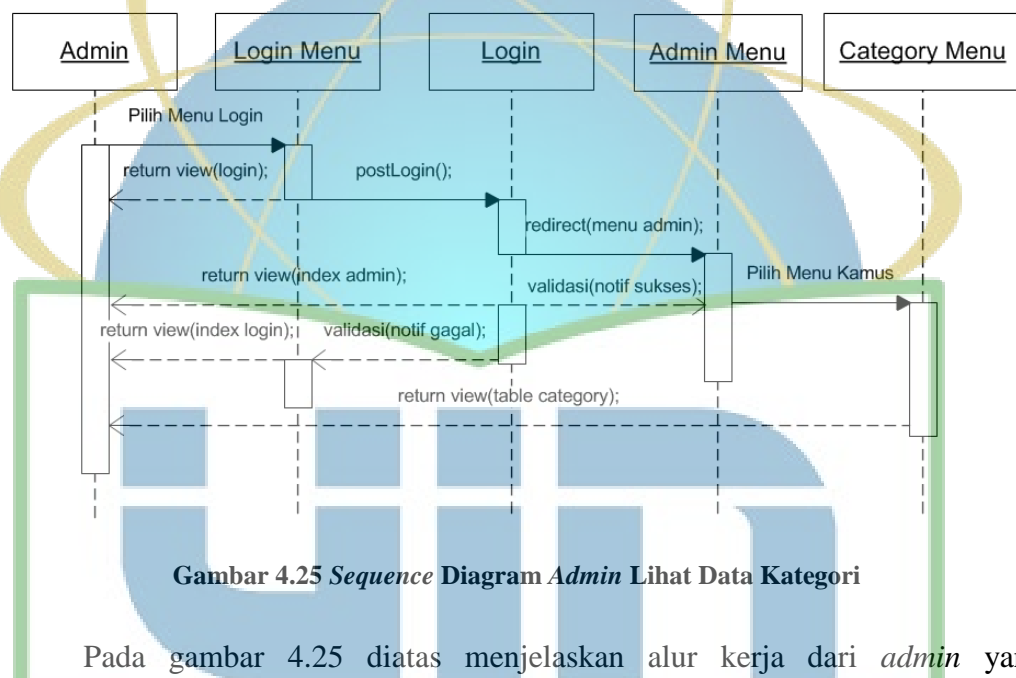
b. Ubah Data Kategori



Gambar 4.24 Sequence Diagram Admin Ubah Data Kategori

Pada gambar 4.24 diatas menjelaskan alur kerja dari *admin* yang berinteraksi dengan sistem dalam proses pengubahan data kategori, *admin* terlebih dahulu melakukan *login* kedalam sistem, jika gagal maka akan menampilkan notifikasi dan akan melakukan pemuatan ulang halaman *login admin*, dan jika berhasil maka akan diarahkan ke dalam menu *admin*, lalu admin memilih menu kategori, dan memilih tombol *edit* pada *table* kategori, melakukan pengisian *form* yang telah disediakan pada sistem dan kemudian memilih tombol *edit* sehingga *data* yang dimasukkan dapat disimpan kedalam *database*.

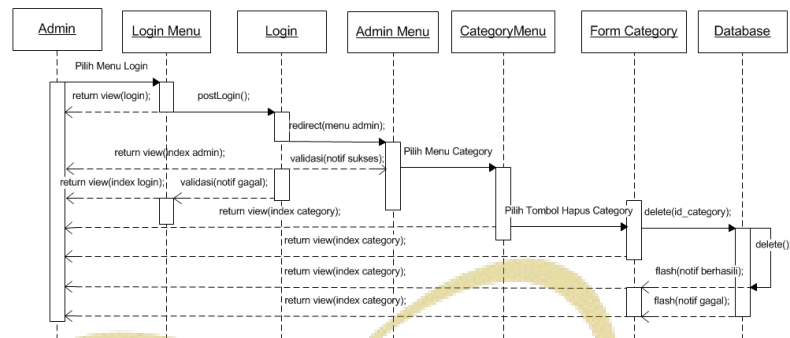
c. Lihat Data Kategori



Gambar 4.25 Sequence Diagram Admin Lihat Data Kategori

Pada gambar 4.25 diatas menjelaskan alur kerja dari *admin* yang berinteraksi dengan sistem dalam proses melihat data kategori, *admin* terlebih dahulu melakukan *login* kedalam sistem, jika gagal maka akan menampilkan notifikasi dan akan melakukan pemuatan ulang halaman *login admin*, dan jika berhasil maka akan diarahkan ke dalam menu *admin*, lalu admin memilih menu kategori, dan sistem menampilkan *table* kategori.

d. Hapus Data Kategori



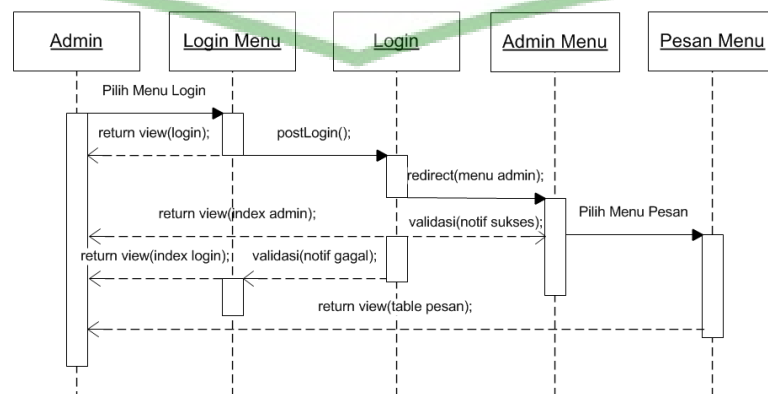
Gambar 4.26 Sequence Diagram Admin Hapus Data Kategori

Pada gambar 4.26 diatas menjelaskan alur kerja dari *admin* yang berinteraksi dengan sistem dalam proses penghapusan data kategori, *admin* terlebih dahulu melakukan *login* kedalam sistem, jika gagal maka akan menampilkan notifikasi dan akan melakukan pemuatan ulang halaman *login admin*, dan jika berhasil maka akan diarahkan ke dalam menu *admin*, lalu *admin* memilih menu kategori, dan memilih tombol hapus pada *table* kategori, sehingga data yang dipilih dapat dihapus pada *database*.

3. Admin Mengelola Data Pesan

Di bawah ini merupakan *sequence* diagram untuk *admin* mengelola data pesan pada sistem kamus istilah teknologi informasi berbasis *web* menjelaskan interaksi *admin* terhadap sistem untuk melakukan lihat data pesan, dan hapus data pesan.

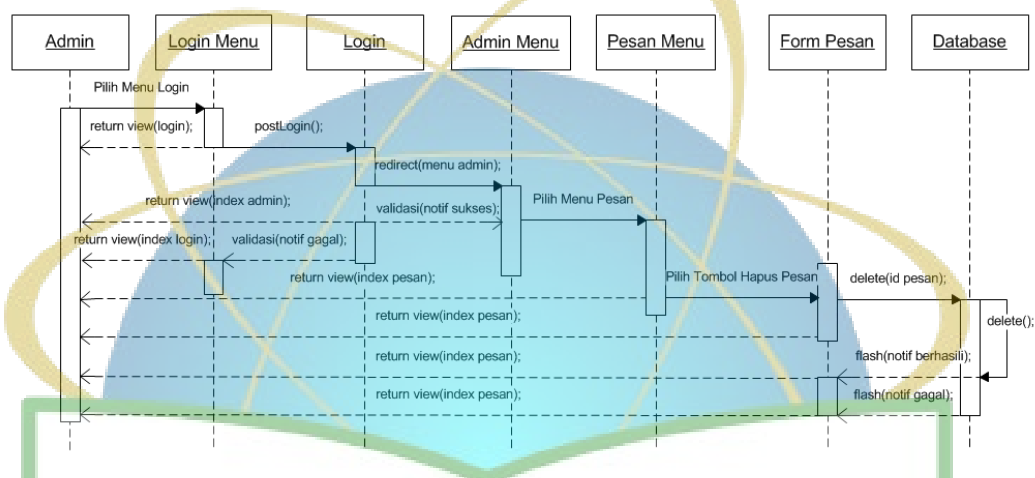
a. Lihat Data Pesan



Gambar 4.27 Sequence Diagram Admin Lihat Data Pesan

Pada gambar 4.27 diatas menjelaskan alur kerja dari admin yang berinteraksi dengan sistem dalam proses melihat data pesan, *admin* terlebih dahulu melakukan *login* kedalam sistem, jika gagal maka akan menampilkan notifikasi dan akan melakukan pemuatan ulang halaman *login admin*, dan jika berhasil maka akan diarahkan ke dalam menu *admin*, lalu admin memilih menu pesan, dan kemudian sistem akan menampilkan *table* pesan.

b. Hapus Data Pesan

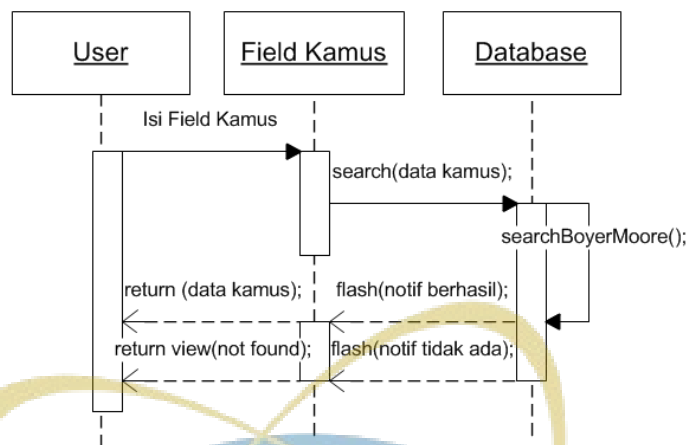


Gambar 4.28 Sequence Diagram Admin Hapus Data Pesan

Pada gambar 4.28 diatas menjelaskan alur kerja dari *admin* yang berinteraksi dengan sistem dalam proses penghapusan data pesan, *admin* terlebih dahulu melakukan *login* kedalam sistem, jika gagal maka akan menampilkan notifikasi dan akan melakukan pemuatan ulang halaman *login admin*, dan jika berhasil maka akan diarahkan ke dalam menu *admin*, lalu *admin* memilih menu pesan, dan memilih tombol hapus pada *table* pesan, sehingga data yang dipilih dapat dihapus pada *database*.

4. User Mencari Data Kamus

Di bawah ini merupakan *sequence* diagram untuk user mencari data kamus pada sistem aplikasi kamus istilah teknologi informasi berbasis *web* menjelaskan interaksi *user* terhadap sistem untuk melakukan pencarian data kamus.

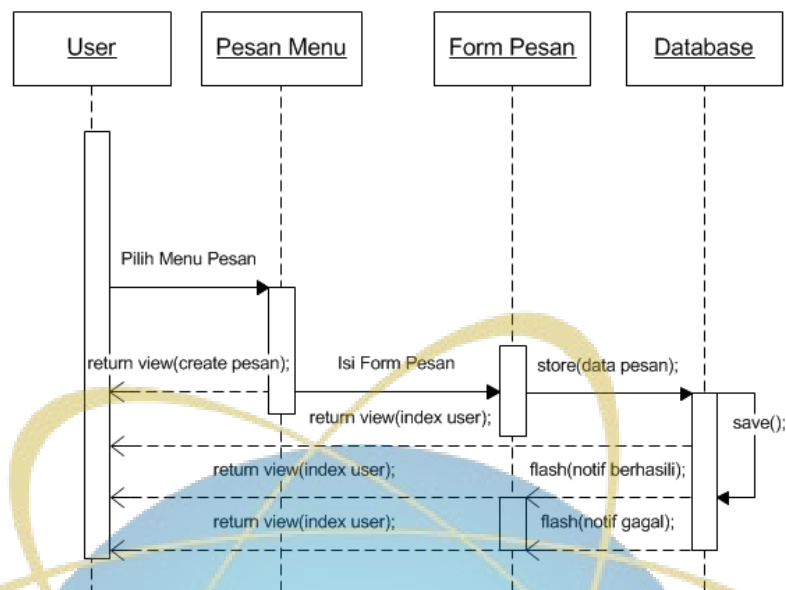


Gambar 4.29 Sequence Diagram User Mencari Data Kamus Metode Boyer Moore

Pada gambar 4.29 diatas menjelaskan alur kerja dari *user* yang berinteraksi dengan sistem dalam proses pencarian data kamus menggunakan metode *boyer moore*, terlebih dahulu user mengisi *field* yang disediakan untuk mencari istilah kata yang di inginkan pada sistem kemudian tekan tombol *search* pada halaman pencarian, sistem melakukan pengecekan pada *database* apakah data yang dimasukkan ada, jika ada maka sistem akan menampilkan data kamus yang dicari, jika tidak maka akan menampilkan bahwa data tidak ditemukan.

5. User Menambah Data Pesan

Di bawah ini merupakan *sequence* diagram untuk *user* menambah data pesan pada sistem aplikasi kamus istilah teknologi informasi berbasis *web* menjelaskan interaksi *user* terhadap sistem untuk melakukan tambah data pesan



Gambar 4.29 Sequence Diagram User Menambah Data Pesan

Pada gambar 4.29 diatas menjelaskan alur kerja dari *user* yang berinteraksi dengan sistem dalam proses penambahan data pesan, terlebih dahulu user memilih tombol tambah pesan pada halaman *user*, dan sistem menampilkan *form* tambah pesan, kemudian user memasukkan pesan yang diinginkan dan data akan disimpan kedalam *database*.

4.2.2.2.4. Class Diagram

Class diagram pada aplikasi kamus istilah teknologi informasi ini dibuat untuk menggambarkan kelas-kelas dan operasi pada kelas yang mempunyai relasi dari setiap kelas yang saling berhubungan satu dengan lainnya. Hal ini bertujuan agar mempermudah pembuatan aplikasi kamus teknologi informasi ini. berikut merupakan penjelasan dari *class diagram* pada aplikasi kamus istilah teknologi informasi beserta relasinya.

1. Kelas Kamus

Di dalam kelas ini berisi model beserta nama tipe data dari model tersebut yang berhubungan dengan kelas kamus_category dan kamus repository, terdapat beberapa operasi dalam kelas ini antara lain *get_category_id()*, *__construct()*, *getKamus()*, *index()*, *tambah()*, *ubah()*,

updateSeen(), *updateActive()*, *lihat()*, *hapus()*, *cari()*, *downloadExcelKamus()*, *importExcelkamus()* yang bertujuan untuk mengolah data kamus.

2. Kelas Kamus *Repository*

Di dalam kelas ini berisi tambahan operasi-operasi yang berhubungan pada kelas kamus, operasi-operasi tersebut antara lain *__construct()*, *saveKamus()*, *index()*, *store()*, *update()*, *indexFront()*, *hapus()*, *getTagById()*, *updateSeen()*, *updateActive()*, *GetByIdWithCategory()*, *queryActiveWithUserOrderByDate()*.

3. Kelas *Category*

Di dalam kelas ini berisi model beserta nama tipe data dari model tersebut yang berhubungan dengan kelas *kamus_category* terdapat beberapa operasi dalam kelas ini antara lain *get_kamus_id()*, *__construct()*, *getKamus()*, *index()*, *tambah()*, *ubah()*, *updateSeen()*, *updateActive()*, *lihat()*, *hapus()*, *cari()*, *downloadExcelKamus()*, *importExcelkamus()* yang bertujuan untuk mengolah data *category*.

4. Kelas *Category_Kamus*

Di dalam kelas ini berisi model beserta nama tipe data dari model tersebut yang berhubungan dengan kelas kamus dan kelas *category*, tidak ada operasi pada kelas ini.

5. Kelas *User* (Pengguna)

Di dalam kelas ini berisi model beserta nama tipe data dari model tersebut yang berhubungan dengan kelas kamus, *roles*, *user repository* terdapat beberapa operasi dalam kelas ini antara lain *get_role_id()*, *__construct()*, *index()*, *indexSort()*, *tambah()*, *ubah()*, *updateSeen()*, *lihat()*, *hapus()*, *getRoles()*, *postRoles()* yang bertujuan untuk mengolah data data *user*.

6. Kelas *User Repository*

Di dalam kelas ini berisi tambahan operasi-operasi yang berhubungan pada kelas *user*, operasi-operasi tersebut antara lain

__construct(), *save ()*, *index()*, *count()*, *counts()*, *store()*, *update()*, *getStatus()*, *valid()* *hapus()*, *confirm()*.

7. Kelas *Roles*

Di dalam kelas ini berisi model beserta nama tipe data dari model tersebut yang berhubungan dengan kelas *user* dan *role repository* terdapat beberapa operasi dalam kelas ini antara lain *edit()* yang bertujuan untuk mengolah data *role*.

8. Kelas *Role Repository*

Di dalam kelas ini berisi tambahan operasi-operasi yang berhubungan pada kelas *user* dan *role*, operasi-operasi tersebut antara lain *__construct()*, *all()*, *update()*, *getAllSelect*.

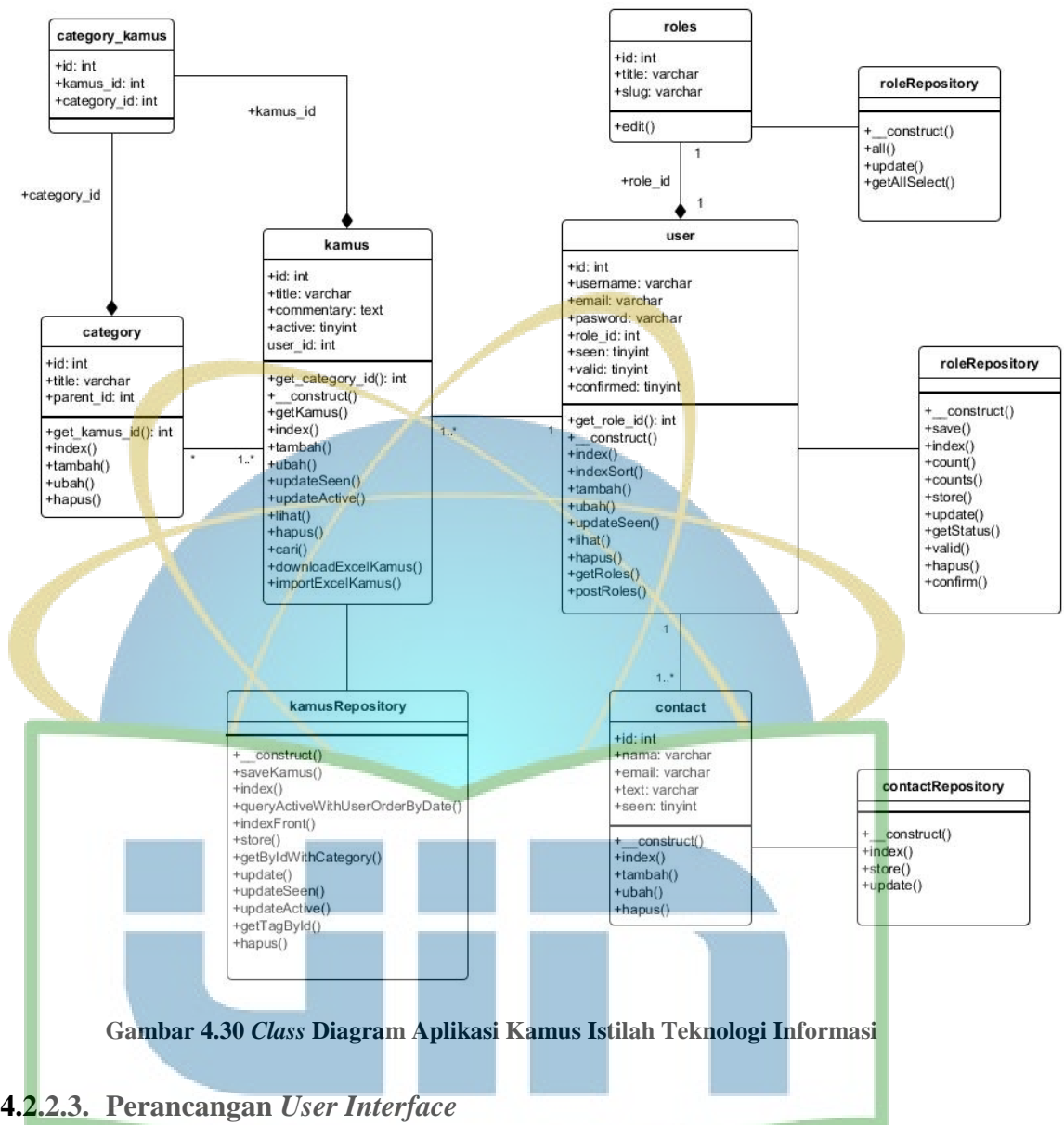
9. Kelas *Contact* (Pesan)

Di dalam kelas ini berisi model beserta nama tipe data dari model tersebut yang berhubungan dengan kelas *user* terdapat beberapa operasi dalam kelas ini antara lain *__construct()*, *index()*, *tambah()*, *ubah()*, *hapus()* yang bertujuan untuk mengolah data *contact/pesan*.

10. Kelas *Contact Repository*

Di dalam kelas ini berisi tambahan operasi-operasi yang berhubungan pada kelas *contact*, operasi-operasi tersebut antara lain *__construct()*, *index()*, *store()*, *update()*.

Berikut merupakan rancangan dari *class* diagram pada aplikasi kamus istilah teknologi informasi beserta relasinya.



Gambar 4.30 Class Diagram Aplikasi Kamus Istilah Teknologi Informasi

4.2.2.3. Perancangan User Interface

Berikut ini adalah rancangan *user interface* yang digunakan oleh penulis untuk membuat aplikasi kamus istilah teknologi informasi berbasis web.

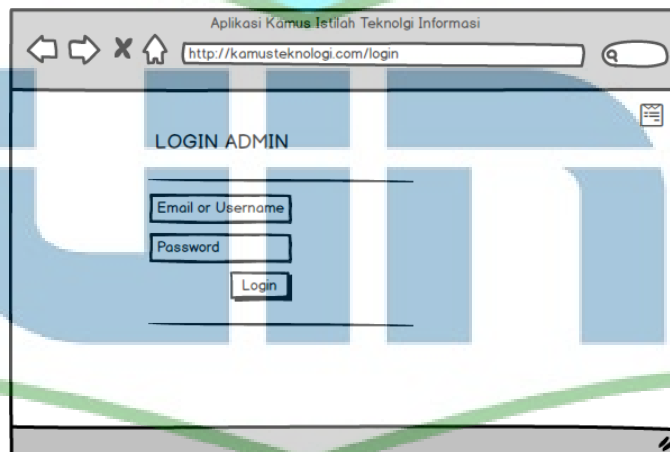
1. Halaman *user* untuk pencarian istilah kata



Gambar 4.31 Halaman Pencarian Kata Pada Kamus

Halaman user untuk mencari kata istilah tentang teknologi informasi, dimana seluruh aktifitas user yang ingin melakukan pencarian kamus dilakukan di sini, terdapat beberapa menu lainnya di halaman *user* ini seperti *user* bisa mengirim pesan kepada *admin* dengan klik tombol pesan.

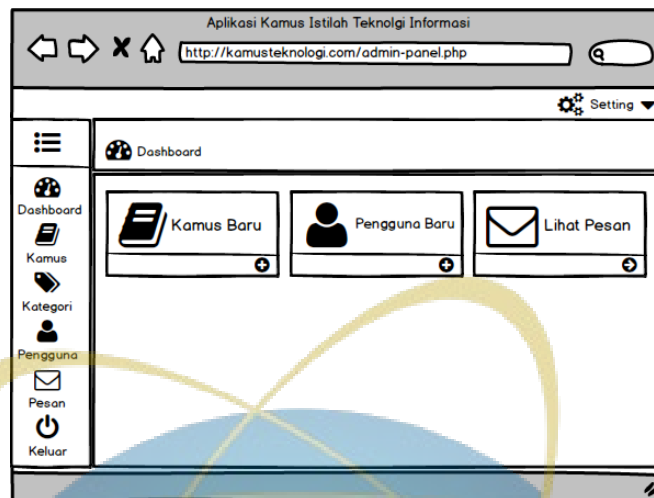
2. Halaman *login admin*



Gambar 4.32 Halaman Login Admin

Halaman *login* ditampilkan ketika *admin* ingin melakukan akses masuk pada aplikasi kamus teknologi informasi, dimana *admin* yang ingin masuk ke dalam sistem harus melewati halaman ini terlebih dahulu dengan memasukkan *email* dan *password* yang di miliki.

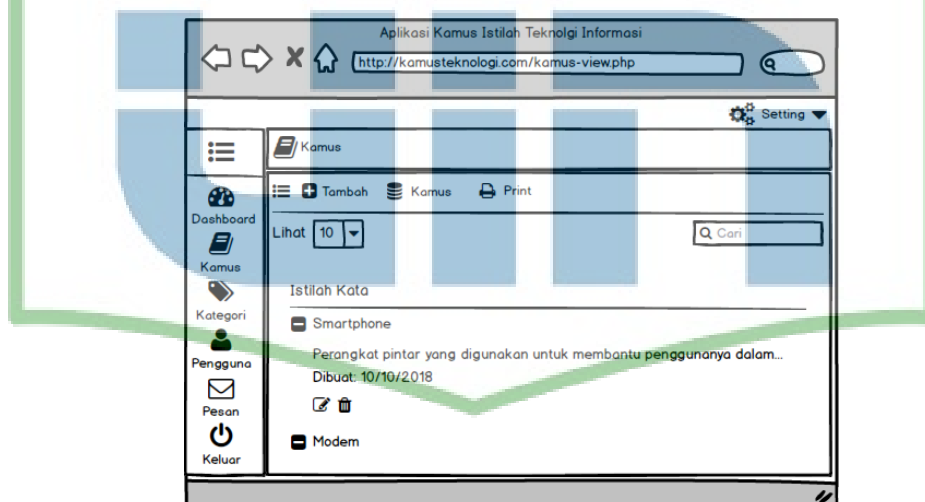
3. Halaman beranda *admin*



Gambar 4.33 Halaman Beranda Admin

Halaman beranda *admin* ini dapat di akses jika admin sudah melakukan login perbedaan dengan menu staf adalah tidak adanya menu untuk melihat data pengguna, di halaman ini terdapat beberapa menu seperti menu kamus, menu kategori, menu pengguna dan menu pesan.

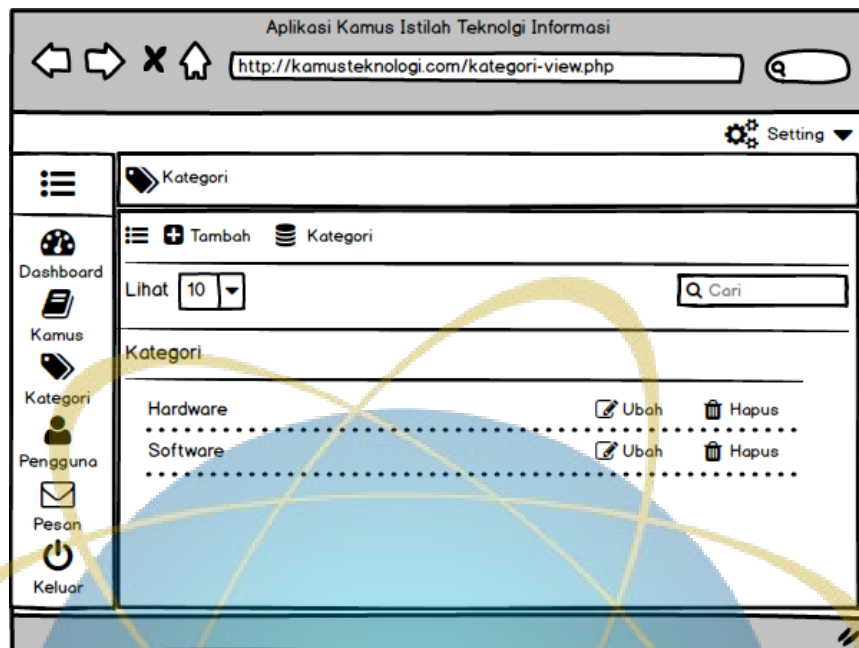
4. Halaman data kamus



Gambar 4.34 Halaman Admin Data Kamus

Halaman untuk melihat data kamus ini berisi beberapa menu, diantaranya terdapat menu tambah kamus, lihat kamus, ubah kamus dan hapus kamus.

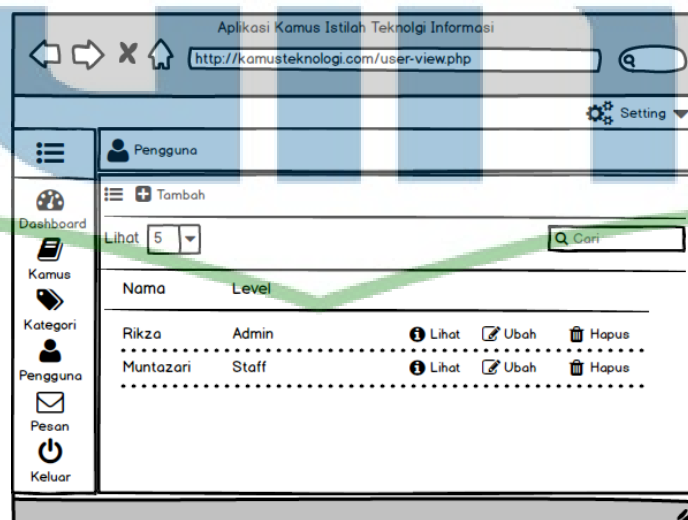
5. Halaman Data Kategori



Gambar 4.35 Halaman Admin Data Kategori

Halaman untuk melihat data kategori ini berisi beberapa menu, diantaranya terdapat menu tambah kategori, lihat kategori, ubah kategori dan hapus kategori.

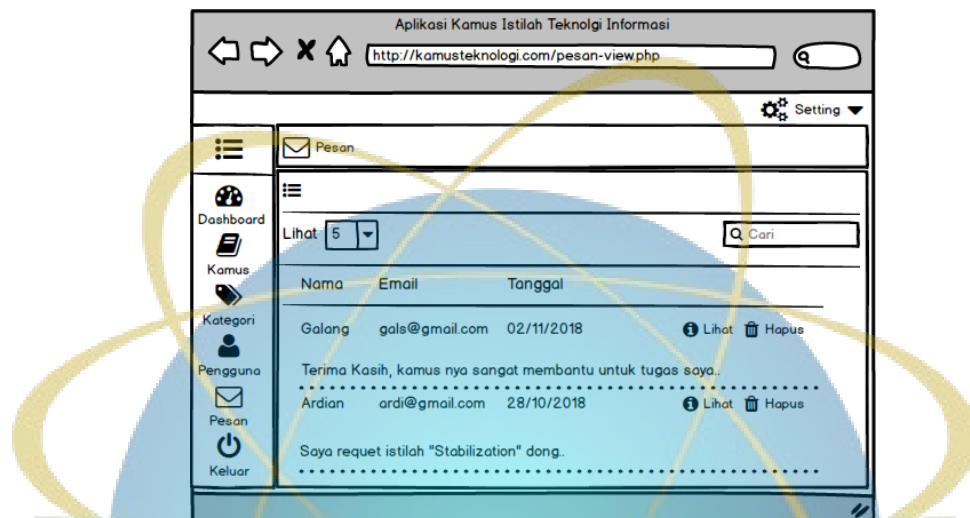
6. Halaman data pengguna



Gambar 4.36 Halaman Admin Data Pengguna

Halaman pengguna hanya bisa diakses oleh pengguna dengan *role* atau level *admin*, pada halaman ini berisi beberapa menu, diantaranya terdapat menu tambah pengguna, lihat pengguna, *edit* pengguna dan hapus pengguna, dan menu tambahan lainnya seperti mengganti *role* atau level pengguna.

7. Halaman data pesan



Gambar 4.37 Halaman Admin Data Pesan

Pada halaman ini berisi pesan dari para *user*, didalam pesan yang dikirim oleh *user* bisa berupa kontribusi dalam kosa kata kamus teknologi informasi atau pesan kritik dan saran pada *web* aplikasi kamus istilah teknologi informasi untuk meningkatkan pelayanan bagi para penggunanya.

4.2.2.4. Perancangan Tabel *Database*

Berikut ini adalah rancangan *table database* yang penulis buat dalam membangun aplikasi kamus istilah teknologi informasi berbasis *web*, dengan menggunakan MySQL sebagai databasenya:

1. Kamus

Nama Tabel : kamus

Primary Key : *id*

Tabel 4.2 Tabel Database Kamus

Nama Field	Tipe Data
<i>id</i>	Integer
<i>title</i>	Varchar
<i>commentary</i>	Text
<i>seen</i>	Tiny Integer
<i>active</i>	Tiny Integer
<i>user_id</i>	Integer

2. Kategori

Nama Tabel : *category*Primary Key : *id*

Tabel 4.3 Tabel Database Kategori

Nama Field	Tipe Data
<i>id</i>	Integer
<i>title</i>	Varchar
<i>parent_id</i>	Integer

3. Kategori Kamus

Nama Tabel : *kamus_category*Primary Key : *id*

Tabel 4.4 Tabel Database Kategori Kamus

Nama Field	Tipe Data
<i>id</i>	Integer
<i>kamus_id</i>	Integer
<i>category_id</i>	Integer

4. User

Nama Tabel : *user*Primary Key : *id*

Tabel 4.5 Tabel Database User

Nama Field	Tipe Data
<i>id</i>	Integer
<i>username</i>	Varchar
<i>email</i>	Varchar
<i>password</i>	Varchar
<i>role_id</i>	Integer
<i>seen</i>	Tiny Integer
<i>valid</i>	Tiny Integer
<i>confirmed</i>	Tiny Integer
<i>confirmation_code</i>	Varchar

5. Roles

Nama Tabel : *roles*Primary Key : *id*

Tabel 4.6 Tabel Database Roles

Nama Field	Tipe Data
<i>id</i>	Integer
<i>title</i>	Varchar
<i>slug</i>	Varchar

6. Pesan/Kontak

Nama Tabel : *contacts*Primary Key : *id*

Tabel 4.7 Tabel Database Pesan

Nama Field	Tipe Data
<i>Id</i>	Integer
<i>Name</i>	Varchar
<i>Email</i>	Varchar
<i>Text</i>	Text
<i>Seen</i>	Tiny Integer

4.2.3. Fase Construction

Dalam tahap pengkodean (*coding*) digunakan bahasa pemrograman PHP, berikut ini merupakan *source code* algoritma *boyer moore* yang digunakan pada aplikasi kamus istilah teknologi informasi berbasis web:

```
public function boyer_moore($term, $text){
    $n = strlen($text);
    $m = strlen($term = $_request>get('term'));
    $goodSuffixes = array();
    $badCharacters = array();
    $this->suffixesgoodSuffixes($term,
    $goodSuffixes);
    $this->suffixesbadCharacters($term,
    $badCharacters);
    $j = 0;
    while ($j < $n - $m)
    {
        for ($i = $m - 1; $i >= 0 && $term[$i]
        == $text[$i + $j]; $i--);
        if ($i < 0)
        {
            echo $j;
            $j += $goodSuffixes[0];
        }
        else
        {
            $j += max($goodSuffixes[$i],
            $badCharacters[$text[$i + $j]] - $m
            + $i + 1);
        }
    }
}
```

Di dalam pembuatan kode pada aplikasi yang dibangun terdapat beberapa urutan modul-modul yang penulis buat dalam aplikasi kamus istilah teknologi informasi berbasis web ini yang memiliki fungsi dan menampilkan *user interface* yang berbeda-beda sesuai dengan kode yang di tulis. Modul-modul yang digunakan dalam aplikasi kamus istilah teknologi informasi berbasis web ini antara lain:

1. Modul *Login*

Modul ini terdiri dari form *login*, proses *login* dan proses *logout*.

2. Modul Kamus

Modul ini terdiri dari menu data kamus, *form* tambah data kamus, *form edit* data kamus, tombol hapus data kamus, tombol lihat data kamus, *form* cari data kamus, menu *print* data kamus, menu *import* data kamus.

3. Modul Kategori

Modul ini terdiri dari menu data kategori, *form* tambah data kategori, *form edit* data kategori, tombol hapus data kategori, *form* cari data kategori.

4. Modul *User*

Modul ini terdiri dari menu data *user*, *form* tambah data *user*, *form edit* data *user*, tombol hapus data *user*, tombol lihat data *user*.

5. Modul *Role*

Modul ini terdiri dari *form edit* data *role*.

6. Modul Pesan

Modul ini terdiri dari menu data pesan, tombol hapus data pesan.

4.2.4. Fase *Implementation*

4.2.4.1. Pengujian

Tahap pengujian ini merupakan tahap untuk mengetahui apakah setiap fungsi yang ada pada aplikasi kamus istilah teknologi informasi berbasis *web* yang telah di program sudah berfungsi sesuai dengan rancangan yang telah dibuat. Pengujian yang dilakukan ini menggunakan pengujian dengan teknik *black box*, penulis memilih menggunakan teknik pengujian *black box* dikarenakan pengujian ini lebih memfokuskan pada keperluan fungsional dari sistem yang dibuat, berikut ini merupakan beberapa kategori dari pengujian *black box* dalam menemukan kesalahan pada sistem:

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan performa
5. Kesalahan inisialisasi dan terminasi

4.2.4.2. Pengujian Sistem

Berikut ini merupakan proses pengujian sistem pada aplikasi kamus istilah teknologi informasi berbasis web yang penulis lakukan dengan menggunakan metode *black box*, terdiri dari beberapa kolom, antara lain kolom deskripsi berisi penjelasan aktifitas pengguna pada sistem, kolom hasil yang diharapkan berisi penjelasan harapan pada aktifitas sistem yang diinginkan, kolom hasil sebenarnya berisi penjelasan tentang hasil nyata pada sistem setelah dilakukan pengujian.

Tabel 4.8 Tabel Pengujian Sistem

No	Deskripsi	Hasil yang Diharapkan	Hasil Sebenarnya
1.	<i>Admin Melakukan Login</i>	<i>Admin dapat melakukan login pada aplikasi</i>	Sesuai
2.	<i>Admin Create, Read, Update, Delete Data Kamus</i>	<i>Admin dapat melakukan create, read, update, delete data kamus pada aplikasi</i>	Sesuai
3.	<i>Admin Cetak Data Kamus</i>	<i>Admin dapat melakukan cetak data kamus pada aplikasi</i>	Sesuai
4.	<i>Admin Import Data Kamus</i>	<i>Admin dapat melakukan import data dalam format excel pada aplikasi</i>	Sesuai
5.	<i>Admin Create, Read, Update, Delete Data Kategori</i>	<i>Admin dapat melakukan create, read, update, delete data category pada aplikasi</i>	Sesuai
6.	<i>Admin Delete Data Pesan</i>	<i>Admin dapat melakukan delete data pesan pada aplikasi</i>	Sesuai
7.	<i>Admin Logout</i>	<i>Admin dapat logout pada aplikasi</i>	Sesuai
8.	<i>User Mencari Data Kamus</i>	<i>User dapat melakukan pencarian data pada aplikasi</i>	Sesuai
9.	<i>User Mengirim Pesan</i>	<i>User dapat melakukan input pesan dalam aplikasi</i>	Sesuai

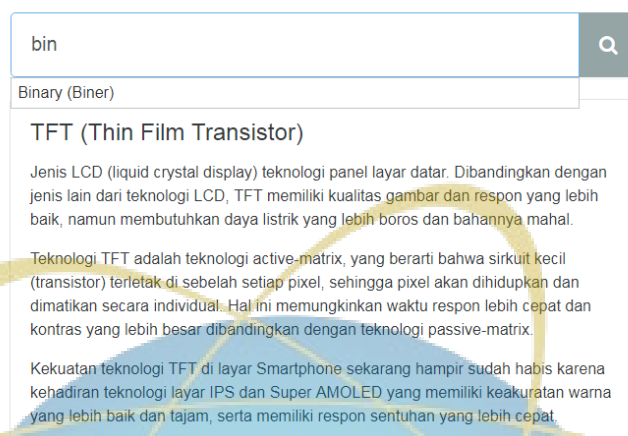
4.2.4.3. Pengujian Algoritma Boyer Moore

Berikut ini merupakan pengujian dari algoritma *boyer moore* yang penulis lakukan untuk mencari kata dengan menggunakan algoritma *boyer moore* pada aplikasi kamus istilah teknologi informasi, dengan mengambil enam contoh istilah kata pada *database* kamus istilah teknologi informasi, enam contoh istilah kata tersebut antara lain sebagai berikut:

Tabel 4.9 Tabel Pengujian Contoh Istilah Kata Algoritma Boyer Moore

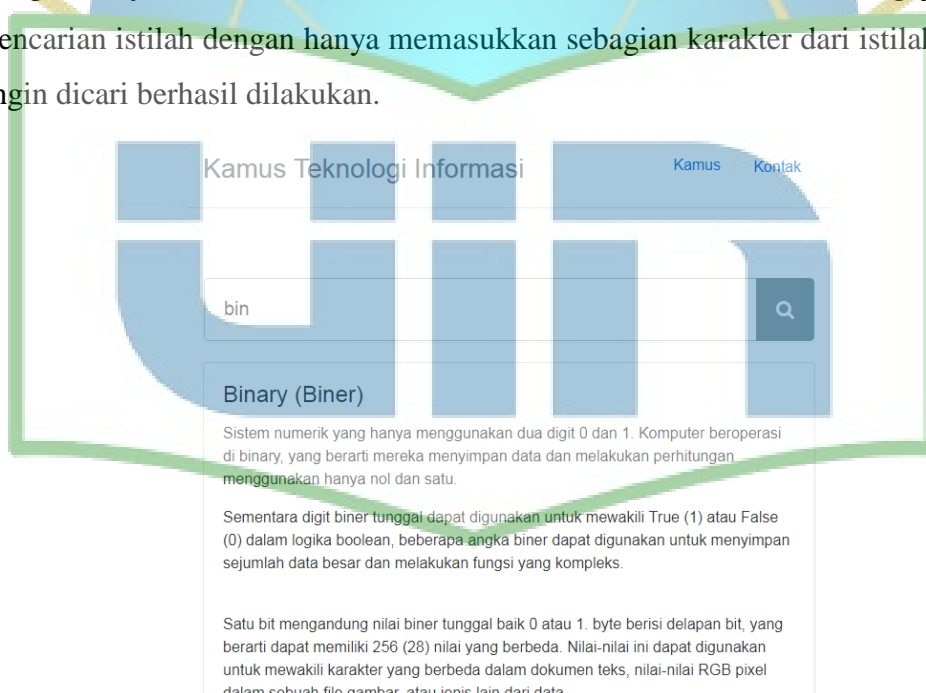
No	Kata	Istilah
1	<i>Antivirus</i>	Utilitas yang digunakan untuk melindungi, memindai dan menghapus <i>virus</i> dari komputer maupun perangkat <i>device</i>
2	<i>Binary</i>	Sistem numerik yang hanya menggunakan dua digit 0 dan 1
3	<i>Domain Name</i>	Nama unik yang mengidentifikasi sebuah situs web
4	<i>Email</i>	Surat elektronik yang memungkinkan Anda untuk mengirim dan menerima pesan dari siapa saja dengan alamat email
5	PDF	Format file <i>multi-platform</i> yang dikembangkan oleh Adobe <i>Systems</i> . Sebuah file PDF menampilkan teks dokumen, <i>font</i> , gambar, dan bahkan format dokumen dari berbagai aplikasi
6	HTTPS	Sama seperti HTTP, tetapi menggunakan <i>secure socket layer</i> (SSL) untuk tujuan keamanan. Beberapa contoh situs yang menggunakan HTTPS seperti perbankan dan website investasi, situs <i>e-commerce</i> , dan sebagian besar situs yang mengharuskan Anda untuk <i>login</i>
7	Hashtag (#)	Hashtag digunakan untuk pengelompokan konten pada sebuah postingan berupa teks, foto, video, event, dan lain-lain.

Setelah ditentukan contoh istilah kata, selanjutnya penulis akan melakukan pengujian algoritma *boyer moore* pada aplikasi kamus istilah teknologi informasi berbasis *web*, berikut ini merupakan gambar dari hasil pengujian yang penulis lakukan.



Gambar 4.38 Pencarian Istilah Binary

Pada gambar 4.38 pengujian untuk mencari istilah kata “binary” dilakukan dengan hanya memasukkan karakter “bin”, hal ini dilakukan untuk menguji apakah pencarian istilah dengan hanya memasukkan sebagian karakter dari istilah kata yg ingin dicari berhasil dilakukan.



Gambar 4.39 Hasil Dari Pencarian Istilah Binary

Pada gambar 4.39 pengujian untuk mencari istilah dengan hanya memasukkan sebagian karakter dari istilah kata yang ingin dicari berhasil dilakukan. Maka dapat disimpulkan bahwa pencarian kata menggunakan

algoritma *boyer moore* berhasil dilakukan dengan melakukan *input* kata pada kolom *textfield* pencarian, kemudian pencarian akan di proses kedalam algoritma *boyer moore* dan menghasilkan *output* yang dimaksud.

Berikut ini beberapa hasil pengujian dengan menggunakan algoritma *boyer moore* dari tujuh contoh istilah kata pada tabel 4.9 di atas:

Tabel 4.10 Tabel Pengujian Pencarian Kata Algoritma Boyer Moore

No.	Data Masukkan	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
1.	Pencarian dengan kata “Antivirus”	Menampilkan hasil pencarian dengan keyword “Antivirus”	Sistem menampilkan hasil pencarian dengan keyword “Antivirus”	Sesuai
2.	Pencarian dengan kata “Binary”	Menampilkan hasil pencarian dengan keyword “Binary”	Sistem menampilkan hasil pencarian dengan keyword “Binary”	Sesuai
3.	Pencarian dengan kata “Domain Name”	Menampilkan hasil pencarian dengan keyword “Domain Name”	Sistem menampilkan hasil pencarian dengan keyword “Domain Name”	Sesuai
4.	Pencarian dengan kata “Email”	Menampilkan hasil pencarian dengan keyword “Email”	Sistem menampilkan hasil pencarian dengan keyword “Email”	Sesuai
5.	Pencarian dengan kata “PDF”	Menampilkan hasil pencarian dengan keyword “PDF”	Sistem menampilkan hasil pencarian dengan keyword “PDF”	Sesuai
6.	Pencarian dengan kata “HTTPS”	Menampilkan hasil pencarian dengan keyword “HTTPS”	Sistem menampilkan hasil pencarian dengan keyword “HTTPS”	Sesuai
7.	Pencarian dengan kata “Hashtag (#)”	Menampilkan hasil pencarian dengan keyword “Hashtag (#)”	Sistem menampilkan hasil pencarian dengan keyword “Hashtag (#)”	Sesuai

4.2.4.4. Kinerja Algoritma Boyer Moore Menggunakan Browser Developer Tools

Pembahasan selanjutnya merupakan pengujian dari kinerja algoritma *boyer moore* pada aplikasi kamus istilah teknologi informasi. Untuk mengetahui kinerja dari algoritma *boyer moore* pada penelitian ini, penulis memanfaatkan

fitur dari *browser* yang terdapat dalam *panel network* pada *developer tools* untuk mengetahui waktu yang di dapat pada pencarian istilah kata teknologi informasi.

Kinerja algoritma *boyer moore* diambil berdasarkan tiga waktu, yaitu:

1. *Request Sent/Sending*, yaitu proses pengiriman data yang di *input client* untuk *server*.
2. *Waiting (Time To First Byte)*, mengetahui lamanya waktu dari proses pengiriman sampai proses pengambilan data dari *server*.
3. *Downloading*, merupakan proses pengambilan data dari *server* ke *client*.

Pengujian dilakukan dengan cara melakukan *input* data pada kolom pencarian dengan beberapa karakter sebagai berikut:

Tabel 4.11 Tabel Pengujian Kinerja Boyer Moore

No.	Teks yang diketik	Time (ms)	Timeline Request/Response (Milisecond)		
			<i>Sending</i>	<i>Waiting (TTFB)</i>	<i>Downloading</i>
1	Bi	238	0.16	213.38	7.17
2	Bin	239	0.14	229.69	6.99
3	Bina	239	0.15	210.09	7.37
4	Binar	251	0.22	229.91	5.75
5	Binary	256	0.16	225.85	9.19

Dari pengujian yang dilakukan beberapa kali oleh penulis pada penginputan karakter diatas, dapat di ambil kesimpulan bahwa semakin banyak karakter yang ada maka waktu yang dibutuhkan akan sedikit lebih lama, dikarenakan lamanya proses pada pengecekan panjang *pattern* pada kolom *input*, begitu pula sebaliknya jika *pattern* yang di inputkan sedikit maka hasil dari waktu yang dibutuhkan akan semakin cepat, namun ada beberapa faktor lainnya juga yang mempengaruhi kinerja algoritma *boyer moore* ketika di terapkan pada sebuah aplikasi, seperti banyaknya data, penulisan *source code* aplikasi, kinerja perangkat keras yang digunakan, dan keadaan trafik jaringan pada saat pencarian. Akan tetapi algoritma *boyer moore* ini dapat memberikan solusi waktu yang relatif lebih cepat dibandingkan dengan algoritma pencarian lainnya hal ini sesuai

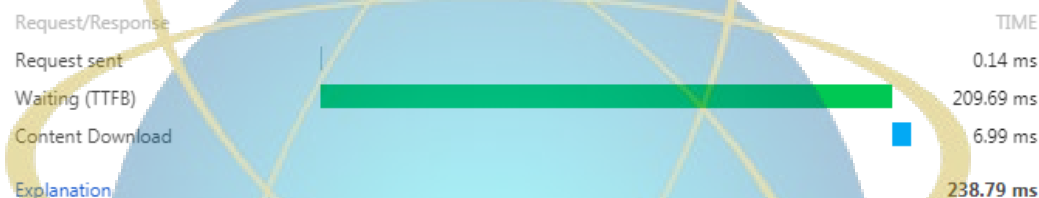
dengan rujukan dari studi literatur bab dua pada penelitian ini. Berikut ini merupakan gambar dari hasil pengujian algoritma yang penulis lakukan.

1. Gambar hasil pengujian karakter “bi”



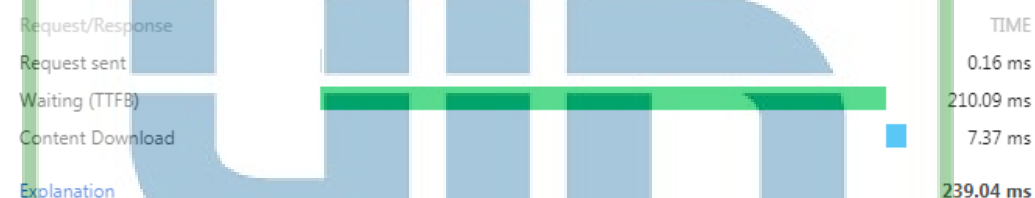
Gambar 4.40 Pengujian Karakter “bi” Aplikasi Kamus Istilah Teknologi Informasi

2. Gambar hasil pengujian karakter “bin”



Gambar 4.41 Pengujian Karakter “bin” Aplikasi Kamus Istilah Teknologi Informasi

3. Gambar hasil pengujian karakter “bina”



Gambar 4.42 Pengujian Karakter “bina” Aplikasi Kamus Istilah Teknologi Informasi

4. Gambar hasil pengujian karakter “binar”



Gambar 4.43 Pengujian Karakter binar Aplikasi Kamus Istilah Teknologi Informasi

5. Gambar hasil pengujian karakter “binary”



Gambar 4.44 Pengujian Karakter “binary” Aplikasi Kamus Istilah Teknologi Informasi

Berikut ini merupakan pengujian selanjutnya dari kinerja algoritma *boyer moore* pada aplikasi kamus istilah teknologi informasi, pengujian dilakukan dengan mengklasifikasikan beberapa data dari *database* kamus istilah teknologi informasi sebagai berikut.

1. Teks dengan kata Antivirus

Tabel 4.12 Tabel Pengujian Kinerja Boyer Moore Kata (Antivirus)

Teks yang diketik	Data Pada Database	Time (ms)	Timeline Request/Response (Milisecond)		
			Sending	Waiting (TTFB)	Downloading
Antivirus	70	228	0.16	212.72	8.74
	140	230	0.16	216.06	8.91
	200	265	0.18	242.27	9.56

2. Teks dengan kata Binary

Tabel 4.13 Tabel Pengujian Kinerja Boyer Moore Kata (Binary)

Teks yang diketik	Data Pada Database	Time (ms)	Timeline Request/Response (Milisecond)		
			Sending	Waiting (TTFB)	Downloading
Binary	70	224	0.16	209.32	7.81
	140	229	0.17	214.73	8.33
	200	240	0.19	229.23	9.21

3. Teks dengan kata Hashtag (#)

Tabel 4.14 Tabel Pengujian Kinerja Boyer Moore Kata (Hashtag)

Teks yang diketik	Data Pada Database	Time (ms)	Timeline Request/Response (Milisecond)		
			Sending	Waiting (TTFB)	Downloading
Hashtag (#)	70	223	0.15	208.75	7.62
	140	227	0.17	215.47	8.57
	200	236	0.18	228.16	9.83

4. Teks dengan kata Domain Name

Tabel 4.15 Tabel Pengujian Kinerja Boyer Moore Kata (Domain Name)

Teks yang diketik	Data Pada Database	Time (ms)	Timeline Request/Response (Milisecond)		
			Sending	Waiting (TTFB)	Downloading
Domain Name	70	227	0.15	214.68	8.24
	140	231	0.16	216.32	8.64
	200	237	0.18	222.27	9.56

5. Teks dengan kata Email

Tabel 4.16 Tabel Pengujian Kinerja Boyer Moore Kata (Email)

Teks yang diketik	Data Pada Database	Time (ms)	Timeline Request/Response (Milisecond)		
			Sending	Waiting (TTFB)	Downloading
Email	70	221	0.14	210.33	7.31
	140	225	0.16	213.59	7.54
	200	236	0.17	221.54	8.21

6. Teks dengan kata PDF

Tabel 4.17 Tabel Pengujian Kinerja Boyer Moore Kata (PDF)

Teks yang diketik	Data Pada Database	Time (ms)	Timeline Request/Response (Milisecond)		
			Sending	Waiting (TTFB)	Downloading
PDF	70	214	0.13	205.68	4.17

	140	220	0.15	212.51	5.43
	200	225	0.16	216.84	7.18

7. Teks dengan kata HTTPS

Tabel 4.18 Tabel Pengujian Kinerja Boyer Moore Kata (HTTPS)

Teks yang diketik	Data Pada Database	Time (ms)	Timeline Request/Response (Milisecond)		
			Sending	Waiting (TTFB)	Downloading
HTTPS	70	217	0.15	208.84	6.04
	140	220	0.15	213.09	6.27
	200	224	0.17	214.63	7.18

Dari hasil pengujian diatas dapat di ambil kesimpulan bahwa:

1. Perbedaan jumlah data pada *database* tidak terlalu mempengaruhi kinerja algoritma *boyer moore*, terbukti bahwa algoritma ini dapat menghasilkan waktu pengiriman pencarian kata dibawah 0.30 *milisecond*, waktu tunggu dibawah 250 *milisecond*, dan waktu pengambilan atau unduh data dibawah 10 *milisecond*, dengan total waktu pencarian dibawah 300 *milisecond*.
2. Dalam melakukan pencarian kata, aplikasi ini berhasil memberikan hasil pencarian informasi dengan baik.
3. Dalam melakukan pencarian kata kamus istilah teknologi informasi, aplikasi ini berhasil memberikan hasil pencarian informasi secara tepat dan cepat, dengan bantuan fitur *autocomplete* yang tersedia ketika memasukkan kata yang ingin dicari. Fitur ini akan muncul ketika potongan kata yang di masukkan oleh *user* tersedia pada *database*.
4. Pencarian kata pada aplikasi kamus istilah teknologi informasi mendukung pencarian dengan menggunakan kata sambung ataupun simbol sebagai *input* pada pencarian istilah kata selama kata tersebut tersedia pada *database* aplikasi.
5. Aplikasi kamus istilah teknologi informasi berbasis web ini berhasil memberikan kemudahan bagi pengguna untuk memperoleh informasi

tentang istilah dalam ilmu teknologi informasi, hasil yang diberikan dapat dilihat pada sub bab 5 pada penelitian ini.

6. Pada pengujian kedua yang terdapat di tabel 4.11 dapat disimpulkan bahwa kinerja algoritma *boyer moore* yang di terapkan pada aplikasi juga dipengaruhi oleh beberapa faktor, antara lain banyaknya data pada *database*, penulisan *source code* pada aplikasi, keadaan lalu lintas jaringan pada saat pencarian kata dan faktor kinerja pada perangkat keras yang digunakan.



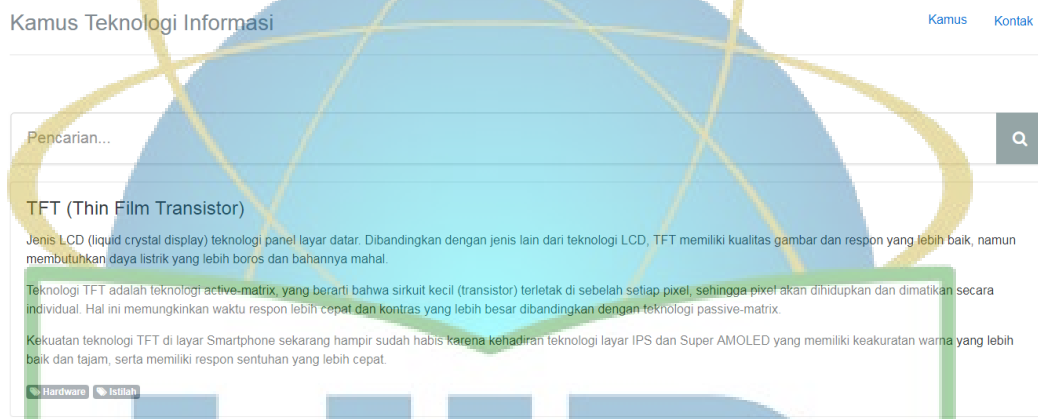
BAB V

HASIL DAN PEMBAHASAN

5.1. Aplikasi Kamus Teknologi Informasi dan Komunikasi Berbasis Web

Setelah seluruh tahapan pembuatan aplikasi kamus istilah teknologi informasi berbasis web telah diselesaikan, maka hasil tampilan dari aplikasi akan dijelaskan pada gambar dibawah ini:

1. Menu *user* untuk pencarian istilah kata teknologi informasi



Gambar 5.1 Menu *User* Pencarian Istilah Kata Teknologi Informasi

Gambar diatas merupakan menu *user* untuk pencarian istilah kata, agar dapat mencari kata pada aplikasi kamus istilah teknologi informasi, pengguna harus memasukan kata pada kolom pencarian yang telah disediakan. Dibawah kolom pencarian kata terdapat istilah beserta definisinya.

2. Menu *Login Admin*

Kembali Ke Situs Beranda Penghubung

Koneksi

Untuk login di situs ini Anda harus mengisi formulir ini:

Email atau nama pengguna Kata sandi

Kirim

Ingat saya

[Saya lupa kata sandi saya!](#)

© 2018-2019

Gambar 5.2 Form Login Admin

Gambar diatas merupakan *form login admin*, untuk dapat masuk ke dalam sistem pada aplikasi kamus istilah teknologi informasi pengguna harus memasukkan *email* dan *password* yang dimiliki.

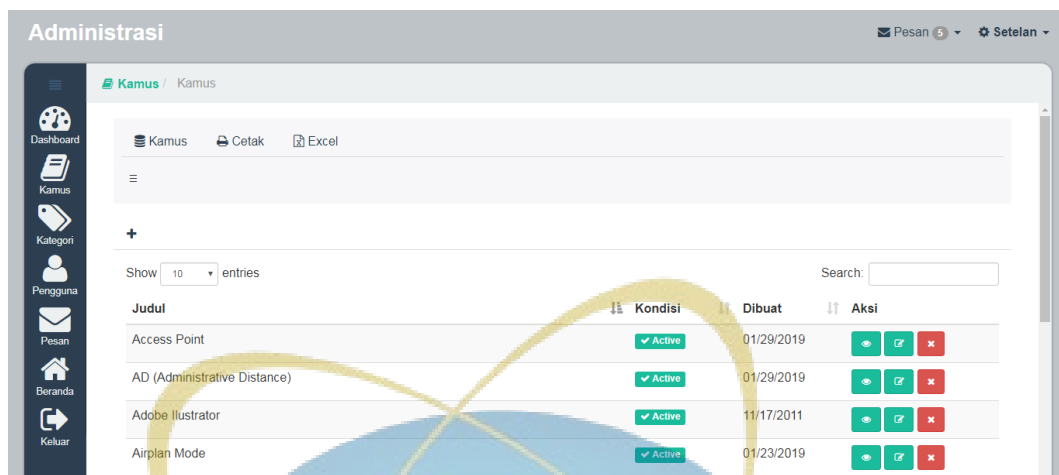
3. Menu *Admin*



Gambar 5.3 Menu Admin

Pada gambar diatas, merupakan tampilan menu *admin*, *admin* dapat melakukan pengolahan data kamus, data kategori, pengolahan data pengguna sistem dan data pesan.

4. Menu Kamus



Gambar 5.4 Menu Kamus

Gambar diatas merupakan tampilan dari menu kamus, admin dapat melakukan pengolahan data kamus, seperti tambah data, ubah data, lihat data, hapus data dan cari data.

5. Form Tambah Data Kamus



Gambar 5.5 Form Tambah Data Kamus

Gambar 5.5 merupakan tampilan dari *form* tambah data kamus, dimana *admin* yang memiliki hak akses dapat melakukan penambahan data kamus.

6. Form Edit Data Kamus

Gambar 5.6 Form Edit Data Kamus

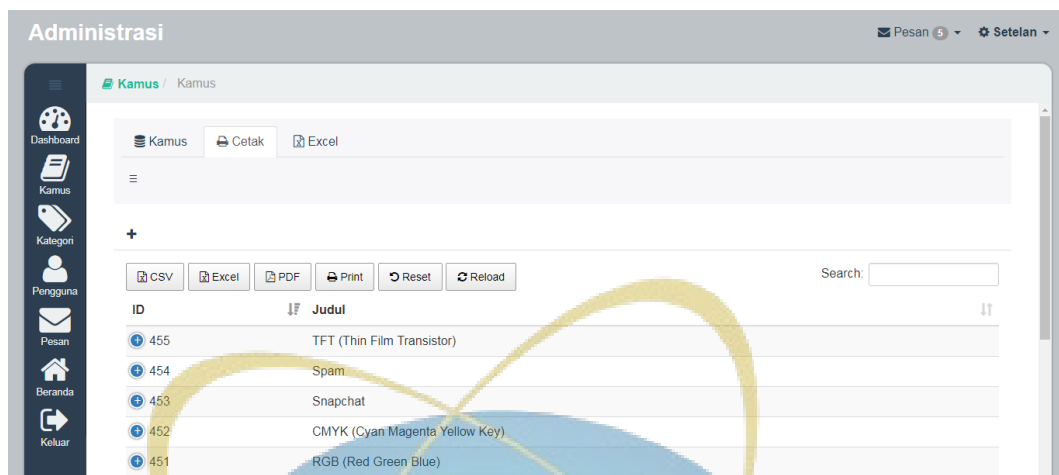
Gambar diatas merupakan tampilan dari *form edit* data kamus, dimana *admin* memiliki hak akses untuk dapat melakukan perubahan data kamus.

7. Form Hapus Data Kamus

Gambar 5.7 Form Hapus Data Kamus

Gambar 5.7 merupakan tampilan dari *form hapus* data kamus, dimana *admin* yang memiliki hak akses dapat melakukan penghapusan data kamus.

8. Form Print Data Kamus



Gambar 5.8 Form Print Data Kamus

Gambar diatas merupakan tampilan dari *form print* data kamus, dimana *admin* yang memiliki hak akses dapat melakukan pencetakan data kamus sesuai dengan format yang di inginkan.

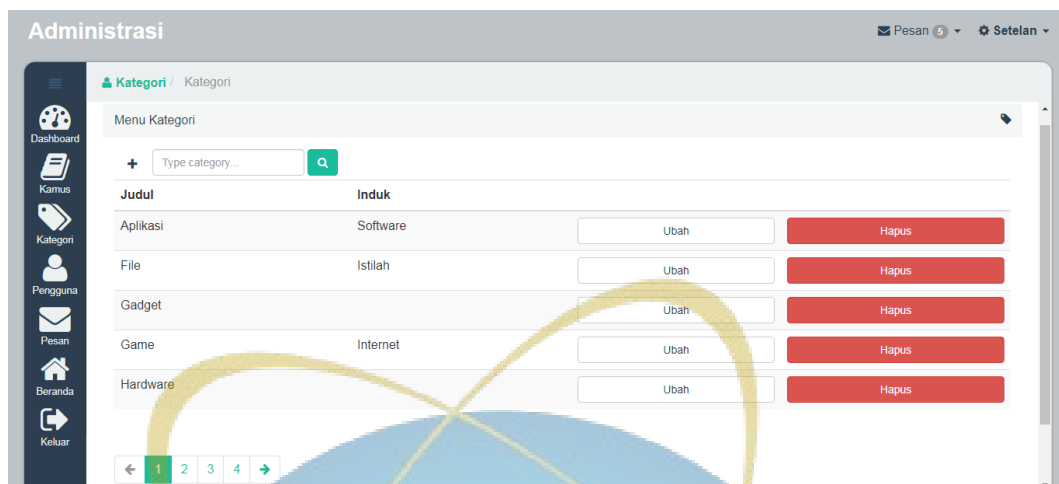
9. Form Import Data Kamus



Gambar 5.9 Form Import Data Kamus

Gambar 5.9 merupakan tampilan dari *form import* data kamus, dimana *admin* yang memiliki hak akses dapat melakukan *import* data kamus dengan format *excel*.

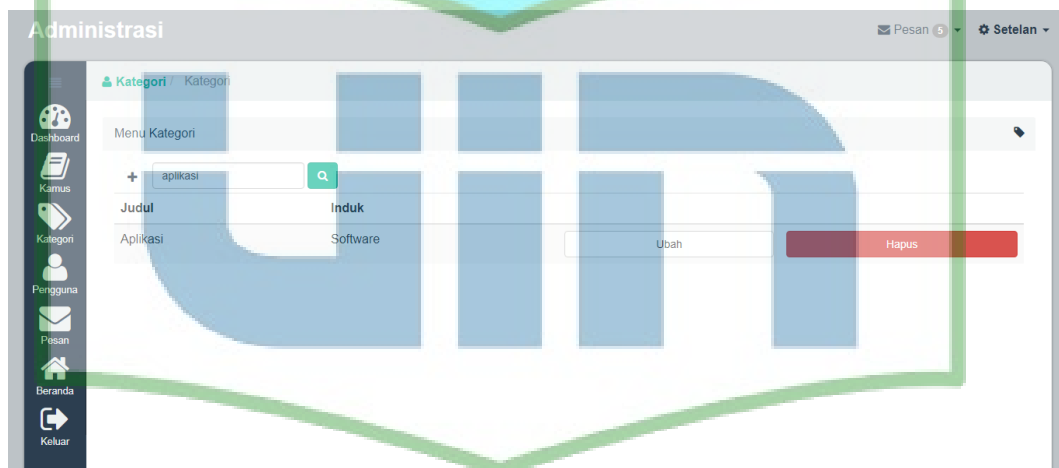
10. Menu Kategori



Gambar 5.10 Menu Kategori

Gambar diatas merupakan tampilan dari menu kategori, *admin* dapat melakukan pengolahan data kategori, seperti tambah data, *edit* data, hapus data dan cari data.

11. Form Pencarian Data Kategori



Gambar 5.11 Form Pencarian Data Kategori

Gambar 5.11 merupakan tampilan dari menu pencarian kategori, *admin* dapat melakukan pencarian data kategori.

12. Form Tambah Kategori

Gambar 5.12 Form Tambah Data Kategori

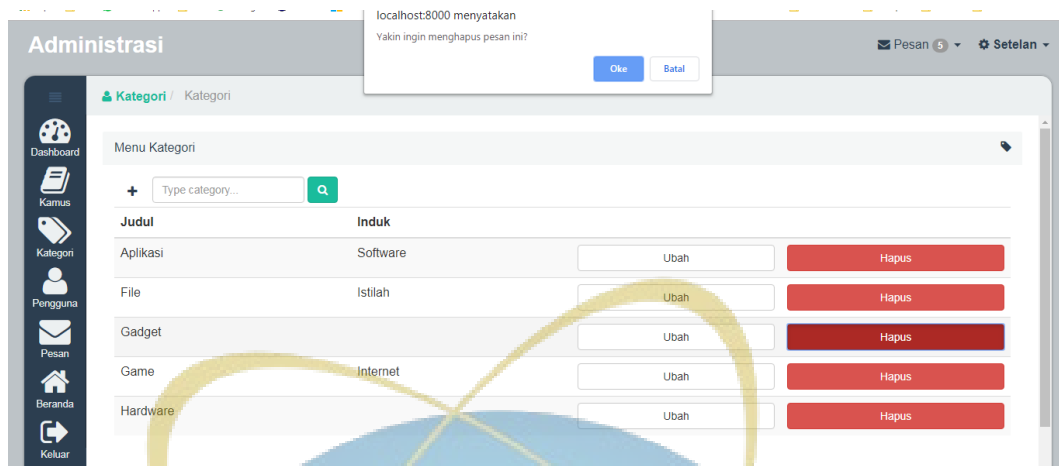
Gambar diatas merupakan tampilan dari *form* tambah data kategori, dimana *admin* yang memiliki hak akses dapat melakukan penambahan data kamus.

13. Form Edit Kategori

Gambar 5.13 Form Edit Data Kategori

Gambar 5.13 merupakan tampilan dari *form edit* data kategori, dimana *admin* yang memiliki hak akses dapat melakukan pengubahan data kategori.

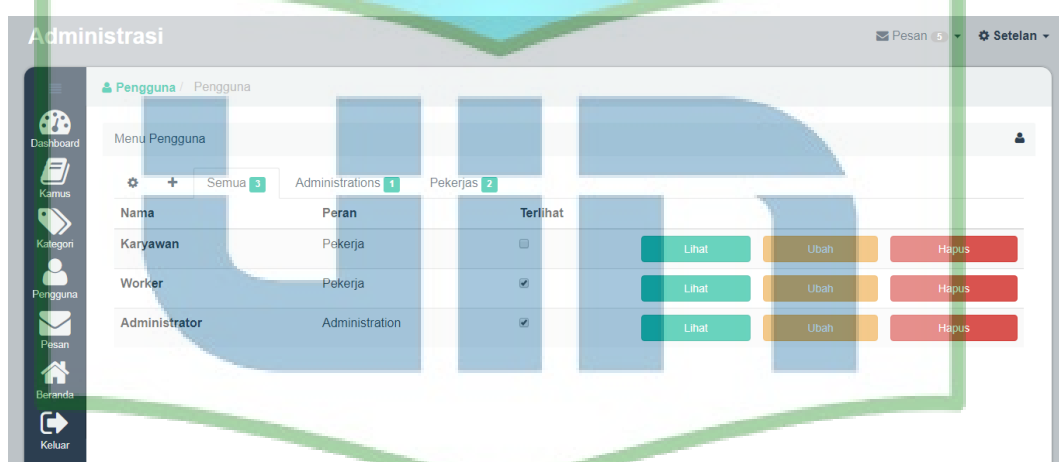
14. Form Hapus Kategori



Gambar 5.14 Form Hapus Data Kategori

Gambar diatas merupakan tampilan dari *form* hapus data kategori, dimana *admin* yang memiliki hak akses dapat melakukan penghapusan data kategori.

15. Menu User



Gambar 5.15 Menu User

Gambar 5.15 merupakan tampilan dari menu pengguna sistem, *admin* dapat melakukan pengolahan data pengguna sistem, seperti tambah data, *edit* data dan hapus data.

16. Form Tambah Data User

Gambar 5.16 Form Tambah Data User

Gambar diatas merupakan tampilan dari *form* tambah data pengguna pada sistem, dimana *admin* yang memiliki hak akses untuk dapat melakukan penambahan data pengguna baru.

17. Form Edit Data User

Gambar 5.17 Form Edit Data User

Gambar 5.17 merupakan tampilan dari *form edit* data pengguna sistem, dimana *admin* yang memiliki hak akses dapat melakukan pengubahan data pengguna pada sistem.

18. Form Edit Role User

Gambar 5.18 Form Edit Data User

Gambar 5.18 diatas merupakan tampilan dari form *edit* data pengguna sistem, dimana *admin* yang memiliki hak akses dapat melakukan pengubahan data pengguna sistem.

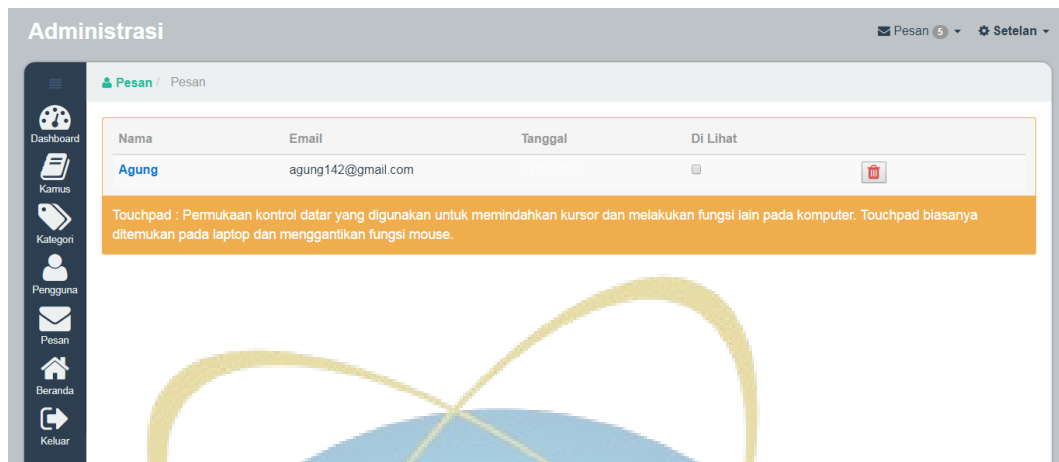
19. Form Hapus Data User

Nama	Peran	Terlihat	Lihat	Ubah	Hapus
Karyawan	Pekerja	<input type="checkbox"/>	Lihat	Ubah	Hapus
Worker	Pekerja	<input checked="" type="checkbox"/>	Lihat	Ubah	Hapus
Administrator	Administration	<input checked="" type="checkbox"/>	Lihat	Ubah	Hapus

Gambar 5.19 Form Hapus Data User

Gambar 5.19 merupakan tampilan dari *form* hapus data *user*, dimana *admin* yang memiliki hak akses dapat melakukan penghapusan data *user*.

20. Menu Pesan



Gambar 5.20 Menu Pesan

Gambar diatas merupakan tampilan dari menu pesan, *admin* dapat melakukan pengolahan data pesan, seperti penghapusan data.

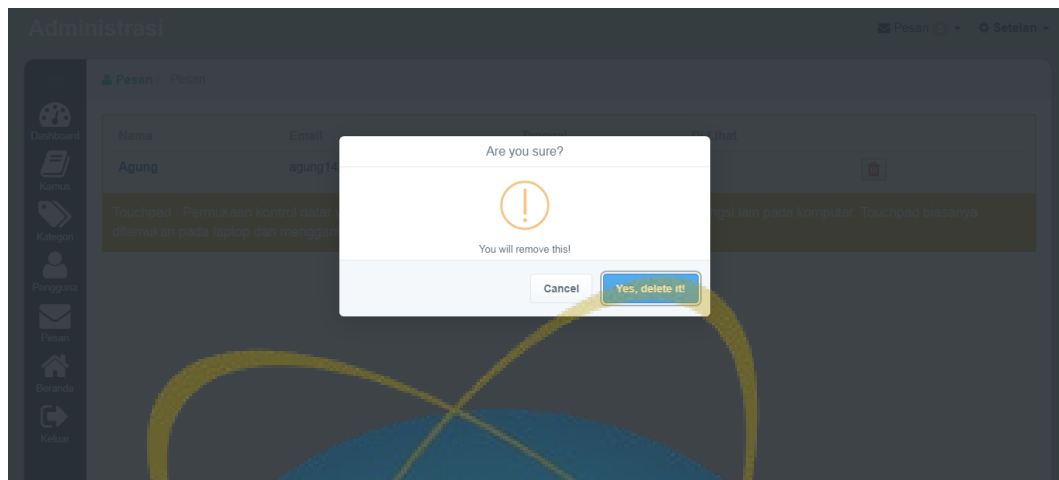
21. Form Tambah Data Pesan



Gambar 5.21 Form Tambah Data Pesan

Gambar 5.21 merupakan tampilan dari *form* tambah data pesan, dimana *user* yang melakukan hak akses pada menu utama kamus dapat melakukan penambahan data pesan.

22. Form Hapus Data Pesan



Gambar 5.22 Form Hapus Data Pesan

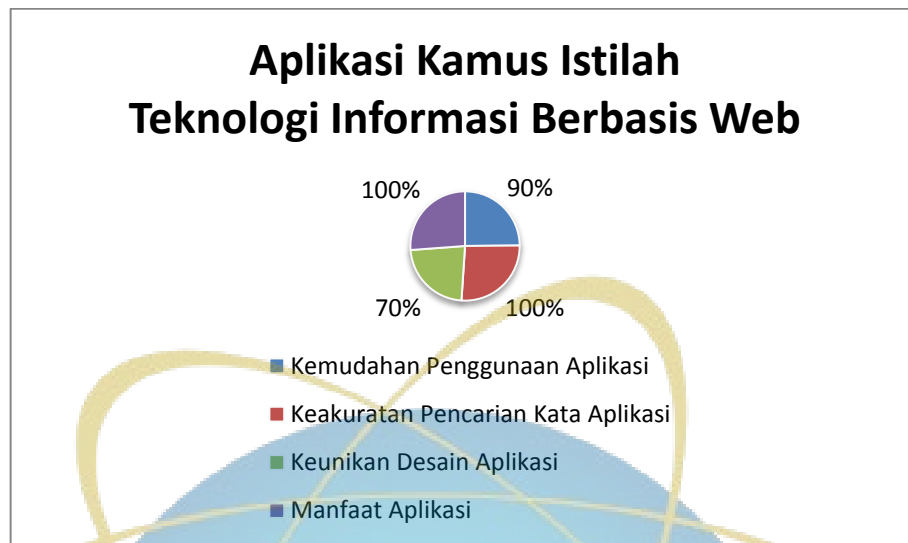
Gambar diatas merupakan tampilan dari *form* hapus data pesan, dimana *admin* yang memiliki hak akses dapat melakukan penghapusan data pesan.

5.2. Hasil Dan Pembahasan

Dari pengujian yang dilakukan oleh penulis dengan menggunakan pengujian *black box*, penulis mencoba melakukan pengujian dengan 10 responden yang berkaitan dengan ilmu teknologi informasi. Pengujian yang dilakukan terhadap responden untuk membuktikan apakah sistem yang dibangun sudah layak di publikasikan atau belum. Pengujian pada penelitian ini di nilai dari empat parameter yang penulis buat, empat parameter tersebut antar lain:

1. Kemudahan penggunaan aplikasi kamus istilah teknologi informasi berbasis *web*.
2. Keakuratan dari hasil proses pencarian kata.
3. Keunikan desain dari aplikasi kamus istilah teknologi informasi.
4. Manfaat yang diberikan dari aplikasi.

Dari empat parameter tersebut didapatkan hasil sebagai berikut:



Gambar 5.23 Hasil Pengujian Sistem Kamus Istilah Teknologi Informasi



BAB VI

PENUTUP

6.1. Kesimpulan

Berdasarkan penelitian yang dilakukan dengan beberapa tahap pengujian pencarian kata menggunakan metode *boyer moore*, bahwa penerapan metode *boyer moore* pada aplikasi kamus istilah teknologi informasi berhasil dilakukan, dapat dilihat dari beberapa penginputan kata dan menghasilkan *output* yang sesuai. Dalam melakukan *input* kata pada kolom *text field* pencarian, dengan menggunakan kata “*binary*” sebagai *pattern* pada pencarian, didapatkan total rentang waktu terendah 238 *milisecond* dengan melakukan *input* menggunakan 2 karakter (B, i), kemudian didapatkan waktu 256 *milisecond* dengan melakukan *input* menggunakan 6 karakter (B, i, n, a, r, y), sampai waktu tertinggi yang didapat 265 *milisecond* dengan melakukan *input* menggunakan 8 karakter (A, n, t, i, v, i, r, u, s), pencarian diproses kedalam algoritma *boyer moore* dan berhasil dilakukan. Pada pengujian yang dilakukan dengan mengklasifikasikan beberapa data dari *database* berdasarkan pada jumlah data pada *database* kamus istilah teknologi informasi, didapatkan hasil bahwa perbedaan jumlah data pada *database* sedikit mempengaruhi kinerja algoritma *boyer moore* yang di terapkan pada aplikasi.

6.2. Saran

Sebuah sistem tidak ada yang benar-benar sempurna, namun untuk mencapai hal tersebut, sebuah sistem harus terus dikembangkan untuk meningkatkan kinerja untuk menutupi kekurangan sistem tersebut. Maka dari itu penulis menyadari bahwa aplikasi kamus istilah teknologi informasi yang penulis bangun masih memiliki banyak kekurangan. Oleh karenanya, ada beberapa yang perlu dipertimbangkan untuk memperbaiki aplikasi kamus istilah teknologi informasi ini menjadi lebih baik, antara lain:

1. Pada pengembangan selanjutnya diharapkan adanya aplikasi sejenis yang berbasis pada sistem operasi mobile seperti Android dan IOS, yang mana data dari aplikasi ini dapat di sinkronisasikan ke dalamnya.
2. Diharapkan pada pengembangan selanjutnya terdapat fitur real time pada pencarian kata aplikasi kamus istilah teknologi informasi ini, sehingga ketika kata yang ingin di cari dimasukkan, akan langsung muncul hasil pencarian tanpa melakukan klik pada *button* pencarian.
3. Pada pengembangan selanjutnya diharapkan *user* bisa berkontribusi untuk dapat menambah kosakata yang belum tersedia pada aplikasi dengan persetujuan dari *admin*.
4. Dari segi keamanan aplikasi harus ditingkatkan.
5. Diharapkan pada pengembangan selanjutnya pengguna dan para ahli dalam ilmu teknologi informasi dapat berkomunikasi melalui aplikasi baik dalam *web* maupun *smartphone*.
6. Kosakata dalam kamus harus lebih diperbanyak lagi.



DAFTAR PUSTAKA

- Argakusumah, K., & Hansun, S. (2012). *Implementasi Algoritma Boyer Moore pada Aplikasi Kamus Kedokteran Berbasis Android*, Jurnal Ilmiah Universitas Multimedia Nusantara, Vol.VI: 70-78.
- Argakusumah, K., & Hansun, S. (2014). *Implementasi Algoritma Boyer Moore pada Aplikasi Kamus Kedokteran Berbasis Android*, Volume 6. No. 2. Hal: 1-9.
- Budiasa, R. M. (2010). *Pemanfaatan Algoritma Boyer Moore dalam Penyaringan Teks Halaman Website Sederhana*. Bandung: Institut Teknologi Bandung.
- Budiasa, R. M. (2010). *Pemanfaatan Algoritma Boyer Moore dalam Penyaringan Teks Halaman Website Sederhana*. Bandung: Institut Teknologi Bandung.
- Chaer, A. (2007). *Linguistik Umum*. Jakarta: Rineka Cipta.
- Charras, C., & Lecroq. (2014). *Handbook of Exact String Matching Algorithms (E-Book)*. English: Collage Publications.
- Darmawan, R. I. (2017). *Implementasi Algoritma Boyer Moore Pada Aplikasi Kamus Kebidanan Berbasis Web*.
- Efendi, D., & Kurnaedi, A. (2012). *Pengembangan Algoritma Boyer Moore pada Translator Bahasa Pemrograman*, Volume 8. No. 1. Hal : 13-21.
- Fakhri, M. M. (2018). *Implementasi Algoritma Boyer Moore Pada Sistem Bimbingan Skripsi Dan PKL Berbasis Web*.
- Halim, I. G. (2016). *Perbandingan Algoritma Boyer Moore Dan Algoritma Brute Force Dalam Pembuatan Kamus Bahasa Mandarin–Indonesia–Inggris Platform Android*.
- Hasanah, L., & Aprilianto, H. (2013). *Aplikasi Pencarian Data Produk Herbal Menggunakan Algoritma Boyer-Moore*. Banjarbaru: STMIK Banjarbaru.
- Herdiansyah, H. (2010). *Metodologi Penelitian Kualitatif*. Jakarta: Salemba Humanika.
- Kadir, A. (2009). *Mastering AJAX dan PHP*. Yogyakarta: Andi Publisher.

- Kurnaedi, A., Effendi, & Diana. (2012). *Pengembangan Algoritma Boyer Moore pada Translator Bahasa Pemrograman*. Bandung: Universitas Kristen Maranatha.
- Marakas, G. (2006). *Management Information System*.
- Marakas, G., M.O'Brien, & James A. (2005). *Sistem Informasi Manajemen*. Jakarta: Salemba Empat.
- Martin, E. (1999). *Managing Information Technology What Managers Need to Know (3rd ed.)*. New Jersey: Pearson Education International.
- Novitasari, S. (2015). *Penggunaan Algoritma Boyer Moore dalam Penentuan Artikel yang Aman bagi Anak*.
- Popov, S. (2012). *Algorithm of the Week: Boyer-Moore String Searching*. Dipetik 09 27, 2018, dari <http://java.dzone.com/articles/alogrithm-week-boyer-moore>
- Pressman. (2002). *Rekayasa Perangkat Lunak: Pendekatan. Praktisi (Buku I)*. Dipetik 10 02, 2018, dari <http://library.um.ac.id/free-contents/index.php/buku/detail/rekayasa-perangkat-lunak-pendekatan-praktisi-buku-ii-roger-s-pressman-19704.html>
- Rahmanita, E. (2014). *Pencarian String Menggunakan Algoritma Boyer Moore Pada Dokumen*.
- Ramadhansyah. (2013). *Perancangan Aplikasi Kamus Bahasa Gayo dengan Menggunakan Metode Boyer-Moore*, Jurnal Pelita Informatika Budi Darma IV(3):118-122.
- Renanda, C. (2012). *Rancang Bangun Aplikasi Kamus Percakapan Bahasa*. Surabaya: STIKOM.
- Rizal. (2015). *Permainan Tebak Kata Bahasa Aceh Menggunakan Algoritma Turbo Boyer*, Volume 6. No. 1. Hal: 1-20.
- Rosa, & Shalahudin. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.
- Sagita, V., & Prasetyowati, M. (2012). *Studi Perbandingan Implementasi Algoritma Boyer Moore, Turbo Boyer Moore dan Tuned Boyer Moore dalam Pencarian String*, Volume 6. No. 1. Hal: 1-7.

- Shalahuddin, M., & Rosa, A. (2013). *Rekayasa Perangkat Lunak*. Bandung: Informatika.
- Shelly, G., & Harry, J. (2011). *Systems Analysis and Design Ninth Edition*. United States of America: Course Technology.
- Simarmata, J. (2006). *Pengenalan Teknologi Komputer dan Informasi*. Yogyakarta: Penerbit Andi.
- Soetam. (2011). *Konsep Dasar Rekayasa Perangkat Lunak*. Dipetik 09 17, 2018, dari <https://openlibrary.telkomuniversity.ac.id/pustaka/21842/konsep-dasar-rekayasa-perangkat-lunak.html>
- Sukarno, M. (2006). *Membangun Website Dinamis Interaktif Dengan PHP-MySQL (Windows & Linux): Untuk Tingkat Pemula Dan Tingkat Lanjut*. Bekasi: Eska Media Press.
- Supriyanto, A. (2005). *Pengantar Teknologi Informasi*. Jakarta: Salemba Infotek.
- Utomo, D. (2008). *Perbandingan Algoritma String Searching Brute Force, Knuth Morris Pratt, Boyer Moore, dan Karp Rabin pada teks Alkitab Bahasa Indonesia*. Jurnal Techne Jurnal Ilmiah Elektronika.
- Utomo, D., Harjo, E. W., & Handoko. (2013). *Perbandingan Algoritma String Searching Brute Force , Knuth Morris Pratt, Boyer Moore, Dan Karp Rabin Pada Teks Alkitab Bahasa Indonesia*. Jurnal Techne Jurnal Ilmiah Elektronik.
- Wahono, R. (2008). *Definisi dan Komponen E-learning*. Dipetik Juli 02, 2018, dari <http://romisatriawahono.net/2008/01/23/meluruskan-salahkaprah-tentang-e-learning>
- Williams, B., & Stacey, S. (2003). *Using Information Technology*. London: United Kingdom: Greener books.

LAMPIRAN

Lampiran I

LEMBARAN OBSERVASI PEGAWAI DAN MAHASISWA MAGANG PT. EREFKA TIGA PILAR UTAMA

Tanggal: 20 Oktober, 10 November 2018

No	Nama Lengkap	Status	Aspek Yang Diamati		
			1	2	3
1	Firman Maulana	Pegawai	3	3	3
2	Haris Putra	Pegawai	2	3	2
3	Hafiz Alfi	Pegawai	3	2	2
4	Zainur Ikhwan	Mahasiswa	2	2	3
5	Ardian	Mahasiswa	2	2	1

Keterangan: Dari beberapa data observasi yang penulis dapat dari pegawai dan mahasiswa magang pada PT. Erefka Tiga Pilar Utama di atas dapat diambil kesimpulan bahwa pegawai pada nomor tiga dan mahasiswa pada nomor empat dan lima masih kurang paham dalam hal istilah pada ilmu teknologi informasi dapat dilihat dari skor yang didapat pada masing-masing mahasiswa.

No	Aspek	Skor	Kriteria Penilaian
1	Keaktifan	3	Sering bertanya dan memberi pendapat
		2	Pernah bertanya dan memberi pendapat
		1	Tidak bertanya dan memberi pendapat

2	Kemampuan Pemahaman Materi	3	Paham dengan baik terhadap materi dan baik dalam melakukan implementasi
		2	Kurang paham terhadap materi dan kurang baik dalam melakukan implementasi
		1	Tidak paham terhadap materi dan kurang dalam melakukan implementasi
3	Kemampuan Berkomunikasi	3	Cakap dan mampu berkomunikasi lisan dengan baik
		2	Kurang cakap dalam berkomunikasi lisan
		1	Tidak cakap dalam berkomunikasi



FORM WAWANCARA

Hari/Tanggal : Sabtu, 20 Oktober 2018

Narasumber : Bapak Nugraha

Jabatan : CEO (Chief Executive Officer) PT. Erefka Tiga Pilar Utama

Pertanyaan:

1. Dari manakah Anda biasanya mempelajari ilmu teknologi yang berhubungan dengan istilah-istilah teknologi informasi?

Jawaban: Dari media cetak dan internet.

Pertanyaan:

2. Apakah media cetak saat ini, khususnya buku yang membahas tentang teknologi cukup membantu Anda dalam mempelajari hal-hal yang baru yang berkaitan dengan ilmu teknologi informasi?

Jawaban: Sebagian media cetak yang beredar saat ini cukup membantu, namun dalam hal update seputar teknologi, media cetak tentu lambat dalam hal waktu jika dibandingkan dengan media internet yang lebih dapat diandalkan.

Pertanyaan:

3. Menurut Anda, seberapa penting untuk mengetahui istilah-istilah tentang teknologi informasi khususnya bagi orang yang menekuni bidang IT?

Jawaban: Sebagai salah satu orang yang menekuni bidang IT, menurut saya cukup penting untuk mengetahui istilah-istilah tersebut, bahkan bagi masyarakat umum sekalipun. Karena kedepannya arus teknologi informasi akan semakin pesat perkembangannya.

Pertanyaan:

4. Bagaimana menurut Anda jika terdapat aplikasi yang memungkinkan proses pembelajaran lebih mudah dalam hal istilah bidang ilmu teknologi informasi?

Jawaban: Jika dirasa dengan adanya aplikasi tersebut dapat mempermudah pembelajaran, maka pastinya sangat berguna jika ada aplikasi tersebut.

Pertanyaan:

5. Apa yang Anda harapkan dengan adanya aplikasi yang di dalamnya menyediakan data tentang istilah teknologi informasi?

Jawaban: Saya mengharapkan dengan adanya aplikasi tersebut dapat mempermudah masyarakat untuk bisa mengetahui istilah-istilah teknologi baik istilah yang sudah lama sampai dengan istilah terbaru saat ini, karena yang istilah yang lama saja sudah sangat banyak, namun tidak semuanya dapat di ketahui. Maka dengan aplikasi tersebut semoga bisa membantu untuk mencari istilah-istilah teknologi informasi.

Pertanyaan:

6. Hal apa yang membuat Anda tertarik pada bidang teknologi informasi?

Jawaban: Saya tertarik pada bidang pengembangan perangkat lunak, jadi saat ini saya masih suka bagian pemrograman, bidang lainnya seperti desain arsitektur, mungkin sebagian tentang desain editing foto dan video, dan lain sebagainya.

Pertanyaan:

7. Adakah dari beberapa ilmu teknologi informasi yang mungkin membuat Anda merasa tidak paham tentang ilmu tersebut?

Jawaban: karena teknologi saat ini berkembang lebih pesat, maka pastinya akan terus muncul hal-hal baru mengenai teknologi yang belum di

mengerti. Namun saya tetap berusaha untuk terus mempelajari hal-hal baru tersebut.

Pertanyaan:

8. Adakah mahasiswa yang pernah magang ditempat Anda? apakah dari setiap mahasiswa yang pernah magang ditempat anda dapat mengimplementasikan ilmunya dengan baik?

Jawaban: Saat ini ada mahasiswa yang sedang magang, namun hanya beberapa dari mereka yang dapat mengimplementasikan ilmunya dengan baik.

Tertanda Tangan Dibawah ini



FORM UJI COBA KAMUS ISTILAH TEKNOLOGI INFORMASI

Dalam uji coba aplikasi kamus istilah teknologi informasi berbasis web, yang dilakukan oleh 10 orang yang menekuni bidang teknologi informasi, yaitu 2 orang mahasiswa magang pada PT. Erefka Tiga Pilar Utama, 3 orang pegawai PT. Erefka Tiga Pilar Utama, dan 5 orang mahasiswa teknik informatika, dipaparkan empat parameter apakah aplikasi yang dibuat sudah sesuai dan layak untuk digunakan, empat parameter tersebut antara lain:

1. Kemudahan penggunaan aplikasi kamus istilah teknologi informasi berbasis *web*.
2. Keakuratan dari hasil proses pencarian kata.
3. Keunikan desain dari aplikasi kamus istilah teknologi informasi.
4. Manfaat yang diberikan dari aplikasi.

Berikut ini merupakan hasil dari empat parameter tersebut:

No.	Parameter	Ya	Tidak	Presentase
1	Kemudahan dalam menggunakan aplikasi kamus istilah teknologi informasi berbasis web	9	1	90%
2	Keakuratan kata dari hasil proses pencarian	10	0	100%
3	Keunikan desain yang dibuat dari aplikasi kamus istilah teknologi informasi	7	3	80%
4	Manfaat yang diberikan	10	0	100%

Dari tabel diatas dapat disimpulkan, bahwa pencarian algoritma *boyer moore* berhasil dilakukan dengan presentase 100% dari responden yang menyatakan bahwa aplikasi kamus istilah teknologi informasi berbasis *web* ini berhasil memberikan penjelasan yang sesuai dengan yang diharapkan.

