

SKRIPSI

**RANCANG BANGUN APLIKASI KAMUS KEPERAWATAN
MENGUNAKAN ALGORITMA BERBASIS *WEBSITE***



DISUSUN OLEH :

MARTALIA

DBC 115 017

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKARAYA
2019**

HALAMAN PERSEMBAHAN

Puji dan Syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena berkat limpahan Rahmat dan Karunia-Nya sehingga penulis dapat menyusun laporan skripsi ini dengan baik.

Adapun laporan hasil Mata Kuliah Tugas Akhir/Skripsi dengan judul **“Rancang Bangun Aplikasi Kamus Keperawatan Menggunakan Algoritma Berbasis Website”** ini diangkat dan dibahas, yaitu sepenuhnya untuk memenuhi tugas Mata Kuliah Tugas/Skripsi Akhir Jurusan Teknik Informatika Fakultas Teknik, Universitas Palangka Raya.

Selama penyusunan laporan ini, penulis sangat menyadari begitu besarnya peranan orang-orang disekitar penulis. Dengan diangkatnya laporan ini, penulis mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa, yang memberikan limpahan karunia dan penyertaan nya kepada penulis.
2. Kedua Orang Tua dan Saudara-Saudara yang telah memberikan semangat, serta berdoa atas kesuksesan penulis dalam menyelesaikan skripsi ini.
3. Nova Noor Kamala Sari, S.T., M.Kom dan Devi Karolita, S.Kom., M.Kom selaku Dosen Pembimbing yang telah sabar membimbing dan memberikan arahan serta masukkan kepada penulis guna menyempurnakan Laporan Skripsi ini.
4. Teman- teman (Rika, Nanda, Efi, Giezka, Adjeng, Rosya, Monica, Winny, Dede, Egy) yang telah memberikan semangat, do'a dan bantuan kepada penulis selama menyusun Skripsi ini.

Penulis menyadari bahwa laporan Skripsi ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan pendapat, kritik, maupun saran yang bersifat membangun demi penyempurnaan laporan Skripsi ini.

KATA PENGANTAR

Puji serta syukur Penulis panjatkan kehadiran Tuhan Yang Maha Kuasa karena atas rahmat dan perkenan-Nya lah, hingga tepat pada waktunya penulis dapat menyelesaikan laporan Skripsi ini sebagai salah satu mata kuliah syarat untuk menyelesaikan pendidikan Strata I Program Studi Teknik Informatika Fakultas Teknik Universitas Palangka Raya.

Laporan ini berisi tentang Rancang Bangun Aplikasi Kamus Keperawatan Menggunakan Algoritma Berbasis *Website*. Dengan adanya *website* ini, maka dalam mencari istilah keperawatan akan sangat mudah tanpa harus membawa banyak buku kamus.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan pendapat, kritik, maupun saran yang bersifat membangun demi penyempurnaan laporan ini. Akhir kata, semoga laporan ini dapat bermanfaat bagi kita semua khususnya bagi Mahasiswa Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya.

Palangka Raya, Januari 2020

Penulis

RANCANG BANGUN APLIKASI KAMUS KEPERAWATAN MENGUNAKAN ALGORITMA BERBASIS *WEBSITE*

MARTALIA (DBC 115 017)

Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya Kampus
Kampus Tanjung Nyaho Jl. Yos Sudarso, Palangka Raya 73112
Email : martalia@mhs.it.upr.ac.id

ABSTRAK

Kamus adalah sebuah buku berisi kata-kata dari sebuah bahasa, biasanya disusun secara alfabetis, disertai keterangan akan arti ucapannya, ejaannya, dan sebagainya. Masing-masing kamus memiliki kelebihan dan kekurangan. Kamus dalam bentuk buku memiliki kelebihan dalam hal jumlah kosa kata dan memiliki kelemahan dalam pencarian arti kata yang memakan waktu lama. Untuk mempercepat dan mempermudah suatu proses pencarian kata yang relevan, dibutuhkan suatu algoritma yang dapat memaksimalkan proses pencarian dan pembobotan kata tersebut. Untuk mempermudah pencarian dan pembobotan kata istilah keperawatan dalam *website*, maka dirancang suatu *website* dengan menggunakan Algoritma *Term Frequency Inverse Dokumen Frequency (TF-IDF)*.

Dalam merancang dan membangun Aplikasi Kamus Keperawatan Menggunakan Algoritma Berbasis *Website* ini, digunakan metodologi pengembangan perangkat lunak waterfall. Desain perancangannya dengan DFD (*Data Flow Diagram*) dan perancangan *database* menggunakan ERD (*Entity Relationship Diagram*), *coding* menggunakan PHP dan *MYSQL* untuk *database* dan *testing* menggunakan *blackbox*. *Black-Box Testing* fokus pada spesifikasi fungsional dari program.

Setelah melakukan *testing*, dapat dilihat hasil dari pembuatan situs ini bahwa *website* ini dapat menjalankan fungsinya yaitu membantu pengguna didalam mencari dan memperoleh informasi terkait istilah-istilah keperawatan. Pada *website* ini terdapat 2 fitur yaitu kelola kamus dan pencarian. Dalam *website* ini admin bertugas untuk mengelola kamus keperawatan. Tugas admin tersebut meliputi menambah, mengubah dan menghapus data kamus istilah pada *database*. Sedangkan pengunjung hanya dapat melakukan pencarian saja.

Kata Kunci : Kamus, Keperawatan, *TF-IDF*, *Waterfall*, *Website*.

**DESIGN OF NURSING DICTIONARY
APPLICATIONS USING WEBSITE- BASED ALGORITHM**

MARTALIA (DBC 115 017)

*Department of Informatics, Faculty of Engineering, Palangka Raya
University , Tanjung Nyaho Campus, Jl. Yos Sudarso , Palangka Raya 73112
Email : martalia@mhs.it.upr.ac.id*

ABSTRACT

A dictionary is a book containing words from a language, usually arranged alphabetically, accompanied by an explanation of the meaning of the words, spelling, and so on. Each dictionary has advantages and disadvantages. The dictionary in the form of a book has advantages in terms of the number of vocabulary and has a weakness in the search for the meaning of words that take a long time. To speed up and simplify a relevant word search process, we need an algorithm that can maximize the process of searching and weighting of these words. To facilitate the search and weighting of nursing terminology on the website, a website is designed using the Frequency Inverse Document Frequency Algorithm (TF-IDF).

In designing and building a Nursing Dictionary Application Using this Website Based Algorithm, the waterfall software development methodology is used. Design the design with DFD (Data Flow Diagrams) and database design using ERD (Entity Relationship Diagrams), coding using PHP and MYSQL for databases and testing using blackboxes. Black-Box Testing focuses on the functional specifications of the program.

After testing, it can be seen the results of making this site that this website can carry out its function, which is to help users find and obtain information related to nursing terms. On this website there are 2 features, namely managing dictionaries and searching. In webs admin is responsible for managing the nursing dictionary. The admin's tasks include adding, changing and deleting dictionary terms in the database. Whereas visitors can only search only.

Keywords: Dictionary, Nursing, TF-IDF, Waterfall, Website.

DAFTAR ISI

Halaman

HALAMAN JUDUL	
HALAMAN PERSETUJUAN	
HALAMAN PENGESAHAN	
HALAMAN PERNYATAAN	
HALAMAN RIWAYAT PENYUSUN	
HALAMAN PERSEMBAHAN	i
KATA PENGANTAR	ii
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI	v
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Sistematika Penulisan	3
1.7 Jadwal Kegiatan	5
BAB II LANDASAN TEORI	
2.1 Tinjauan Pustaka	6
2.2 Pengertian Kamus	11
2.3 Pengertian Keperawatan	11
2.4 <i>Text Mining</i>	12
2.5 <i>Processing Data</i>	13
2.5.1 <i>Tokenizing</i>	13

2.5.2	<i>Stopword Removal</i> atau <i>Filtering</i>	14
2.6	Pembobotan <i>Term Frequency Inverse Document Frequency</i> atau <i>TF-IDF</i>	14
2.7	<i>Website</i>	16
2.8	<i>Web Server</i>	17
2.9	<i>HyperText Markup Language (HTML)</i>	17
2.10	<i>Personal Home Page (PHP)</i>	18
2.11	Penggabungan <i>Script PHP dan HTML</i>	18
2.12	Data dan Basis Data	19
2.12.1	Pengertian Data	19
2.12.2	Basis Data.....	19
2.13	<i>MySQL</i>	20
2.14	<i>Internet</i>	21
2.15	<i>Web Browser</i>	21
2.16	<i>XAMPP</i>	21
2.17	Pengertian Dasar <i>Flowchart</i>	22
2.18	Data Flow Diagram (<i>DFD</i>)	27
2.18.1	Definisi Data Flow Diagram	27
2.18.2	Komponen <i>DFD</i>	27
2.19	<i>Entity Relationship Diagram (ERD)</i>	29
2.20	<i>Notepad++</i>	32
2.21	<i>Edraw Max</i>	32
2.22	<i>Blackbox Testing</i>	33

BAB III METODOLOGI PENELITIAN

3.1	Metodologi Pengembangan Perangkat Lunak	34
3.1.1	<i>Requirements Definition</i> (Definisi Kebutuhan).....	35
3.1.1.1	Alat dan Bahan.....	36
3.1.1.2	Metode Pengumpulan Data.....	36
3.1.1.3	Sasaran Pengguna	39
3.1.1.4	Analisis Sistem Lama	39

3.1.1.5 Kesimpulan Analisis Sistem Lama	39
3.1.1.6 Rekomendasi Sistem Baru	40
3.1.1.7 Bisnis Proses Sistem Baru	40
3.1.1.8 Kesimpulan Bisnis Proses Sistem Baru	41
3.1.2 <i>System dan Software Design</i> (Desain Sistem dan Software).....	42
3.1.2.1 Diagram Konteks	42
3.1.2.2 <i>DFD</i> Level 1	42
3.1.2.3 <i>DFD</i> Level 2 Proses 1 (Proses Login)	43
3.1.2.4 <i>DFD</i> Level 2 Proses 2 (Proses Kelola Kamus).....	44
3.1.2.5 <i>DFD</i> Level 2 Proses 3 (Proses Pencarian).....	44
3.1.2.6 <i>ERD (Entity Relationship Diagram)</i>	45
3.1.2.7 Desain Basis Data (Database)	45
3.1.2.8 Desain Struktur Navigasi	46
3.1.2.9 Desain <i>User Interface</i>	47
3.2 Perhitungan Manual Algoritma <i>Term Frequency Inverse Document Frequency (TF-IDF)</i>	50

BAB IV HASIL DAN PEMBAHASAN

4.1 <i>Implemetation and Unit Testing</i> (Implementasi dan Pengujian Unit)	53
4.1.1 Implementasi Basis Data	53
4.1.2 Implementasi Antarmuka.....	53
4.1.2.1 Halaman <i>Login</i> admin.....	54
4.1.2.2 Halaman <i>Dashboard</i> Admin dan Pengunjung	54
4.1.2.3 Halaman Kelola Kamus pada Admin	55
4.1.2.4 Halaman Tambah Data Untuk Admin	55
4.1.2.5 Halaman Ubah Kamus Untuk Admin.....	56
4.1.2.6 Tampilan Hapus Data Untuk Admin	56
4.1.2.7 Halaman Pencarian Admin dan Pengunjung	57

4.2	<i>Integration and System Testing (Integrasi dan Pengujian Sistem)</i>	58
4.2.1	Pengujian Fitur Admin	59
4.2.1.1	<i>Blackbox Testing</i> Pada Halaman <i>Login</i>	59
4.2.1.2	<i>Black Box Testing</i> Pada Halaman Kelola Kamus	59
4.2.1.3	<i>Black Box Testing</i> pada Halaman Pencarian	60
4.2.2	Pengujian Fitur Pengunjung	60
4.3	Kuesioner Skala Likert	60
4.3.1	Perhitungan Skala Likert	61
4.3.2	Kuesioner Skala Likert	61

BAB V PENUTUP

5.1	Kesimpulan	64
5.2	Saran	65

DAFTAR PUSTAKA

DAFTAR TABEL

	Halaman
Tabel 1.1 Jadwal Tugas Akhir.....	5
Tabel 2.1 Tabel Perbandingan Penelitian yang Relevan.....	8
Tabel 2.2 <i>Flow Direction Symbols</i>	24
Tabel 2.3 <i>Storage Symbols</i>	25
Tabel 2.4 Simbol-simbol <i>Entity Relationship Diagram (ERD)</i>	32
Tabel 3.1 Tabel <i>Admin</i>	45
Tabel 3.2 Tabel Kamus.....	46
Tabel 3.3 Tabel Nilai <i>tf</i>	50
Tabel 3.4 Tabel Nilai <i>df</i>	50
Tabel 3.5 Tabel Nilai <i>D/df</i>	51
Tabel 3.6 Tabel Nilai <i>IDF</i>	51
Tabel 3.7 Lengkap Perhitungan Manual Algoritma <i>Term</i> <i>Frequency Inverse Document</i>	52
Tabel 4.1 <i>Blackbox Testing</i> Proses <i>login</i>	59
Tabel 4.2 Tabel <i>Black Box Testing</i> pada Halaman Kelola Kamus.....	59
Tabel 4.3 Tabel <i>Black Box Testing</i> pada Halaman Pencarian	60
Tabel 4.4 Tabel <i>Black Box Testing</i> pada Halaman Pencarian	60
Tabel 4.5 Kuesioner Skala Likert.....	61
Tabel 4.6 Tingkat Persetujuan.....	62
Tabel 4.7 Persentase Persetujuan	62
Tabel 4.8 Penilaian Dari Responden	62
Tabel 4.9 Perhitungan Skor	63

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Contoh tahap <i>Tokenizing</i>	13
Gambar 2.2 Contoh tahap <i>Stopword Removal</i> atau <i>Filtering</i>	14
Gambar 2.3 <i>Conceptual Flowchart</i>	23
Gambar 2.4 <i>Detail Flowchart</i>	23
Gambar 2.5 Simbol aliran data dalam <i>DFD</i>	28
Gambar 2.6 Simbol proses dalam <i>DFD</i>	28
Gambar 2.7 Simbol <i>Data Store</i> dalam <i>DFD</i>	28
Gambar 2.8 Simbol terminator dalam <i>DFD</i>	29
Gambar 3.1 Waterfall Model (Sommerville, 2011).....	35
Gambar 3.2 Referensi Buku Kamus Keperawatan	37
Gambar 3.3 Contoh Isi Kamus Keperawatan	38
Gambar 3.4 <i>Flowchart</i> Sistem Lama.....	39
Gambar 3.5 <i>Flowchart</i> Sistem Baru	41
Gambar 3.6 Diagram Konteks	42
Gambar 3.7 <i>DFD</i> level 1	43
Gambar 3.8 <i>DFD</i> level 2 proses 1	43
Gambar 3.9 <i>DFD</i> level 2 proses 2	44
Gambar 3.10 <i>DFD</i> level 2 proses 3	44
Gambar 3.11 ERD (<i>Entity Relationship Diagram</i>)	45
Gambar 3.12 <i>Site Map</i> Halaman Utama untuk <i>Admin</i>	47
Gambar 3.13 <i>Site Map</i> Halaman Utama untuk Pengguna	47
Gambar 3.14 Desain <i>Interface</i> Halaman <i>Login Admin</i>	47
Gambar 3.15 Desain <i>Interface</i> Halaman <i>Dashboard Admin</i>	48
Gambar 3.16 Desain <i>Interface</i> Halaman <i>Kelola Kamus</i>	48
Gambar 3.17 Desain <i>Interface</i> Halaman <i>Pencarian Admin</i>	49
Gambar 3.18 Desain <i>Interface</i> Halaman <i>Pencarian Pengunjung</i>	49
Gambar 4.1 Implementasi Basis Data	53
Gambar 4.2 Halaman <i>Login</i>	54
Gambar 4.3 Halaman <i>Dashboard</i>	54

Gambar 4.4 Halaman Kelola Kamus	55
Gambar 4.5 Halaman Tambah Data	55
Gambar 4.6 Halaman Ubah Kamus	56
Gambar 4.7 Tampilan Hapus Data	56
Gambar 4.8 Halaman Pencarian	57
Gambar 4.9 Halaman Hasil Pencarian.....	57
Gambar 4.10 Hasil Pencarian Kata Adrenal.....	58
Gambar 4.11 Hasil Indeks Terhahap Tabel Persentasi Persetujuan	63

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kamus adalah sebuah buku berisi kata-kata dari sebuah bahasa, biasanya disusun secara alfabetis, disertai keterangan akan arti ucapannya, ejaannya, dan sebagainya. Di dalam sebuah kamus yang ideal sepatutnya diberikan juga keterangan tentang pemenggalan kata, informasi tentang asal-usul kata, informasi tentang baku dan tidaknya sebuah kata, informasi tentang kata-kata arkhais dan juga klasik, informasi tentang area penggunaan kata, informasi tentang status sebuah kata, dan berbagai informasi lainnya. (Setyawan, 2016)

Sekarang ini banyak sekali kamus yang beredar, mulai dari bentuk buku, kamus elektronik, aplikasi kamus berbasis dekstop dan web. Masing-masing kamus memiliki kelebihan dan kekurangan. Kamus dalam bentuk buku memiliki kelebihan dalam hal jumlah kosa kata dan memiliki kelemahan dalam pencarian arti kata yang memakan waktu lama. (Saputra, Renanda, Sutanto, dan Sagirani, 2012)

Saat kita menggunakan kamus berbasis *website* akan ada beberapa pengertian istilah atau kata kunci yang memiliki pengertian yang mirip dan terkait dengan istilah lainnya. Proses pencarian dan pembobotan sangat dibutuhkan untuk mempercepat menemukan kata terkait dalam suatu kamus keperawatan berbasis *website*. Untuk mempercepat dan mempermudah suatu proses pencarian kata yang relevan, dibutuhkan suatu algoritma yang dapat memaksimalkan proses pencarian dan pembobotan tersebut. Algoritma merupakan urutan langkah-langkah logis pada penyelesaian masalah yang disusun secara sistematis. Untuk mempermudah pencarian dan pembobotan kata istilah keperawatan dalam *website*, maka dirancang suatu *website* dengan menggunakan Algoritma *Term Frequency Inverse Dokumen Frequency (TF-IDF)*. Dengan adanya *website* ini, maka dalam mencari istilah keperawatan akan sangat mudah tanpa harus membawa banyak buku kamus.

Alasan itulah yang mendorong penulis memilih judul ini sebagai bahan penelitian untuk membuat **Rancang Bangun Aplikasi Kamus Keperawatan Menggunakan Algoritma Berbasis Website**. Diharapkan dengan dibuatnya projek ini dapat membantu pengguna didalam mencari dan memperoleh informasi terkait istilah-istilah keperawatan.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang diuraikan, diperoleh perumusan masalah yaitu bagaimana cara merancang dan membangun suatu *website* yang dapat membantu pengguna dalam mencari dan memperoleh informasi terkait istilah-istilah keperawatan menggunakan Algoritma *Term Frequency Inverse Dokumen Frequency (TF-IDF)*?

1.3 Batasan Masalah

1. Pembuatan program ini dengan menggunakan pemrograman *PHP* dan *database MySQL*.
2. Bahasa Indonesia sebagai bahasa dasar *website*.
3. Algoritma yang digunakan adalah algoritma *Term Frequency Inverse Dokumen Frequency (TF-IDF)*.
4. Pengguna pada *website* ini dibagi menjadi 2 yaitu *admin* dan pengunjung biasa.
5. Adapun fitur-fitur yang dimiliki *website* ini adalah sebagai berikut:
 - a) *Dashboard*
Fitur halaman utama ketika *website* diakses oleh pengguna.
 - b) *Kelola Kamus*
Fitur ini akan menampilkan daftar istilah kata yang tersedia pada kamus keperawatan, selain itu juga akan digunakan oleh *admin* untuk menambah, mengedit dan menghapus data istilah dalam kamus keperawatan berbasis *website*.

c) Pencarian

Merupakan fitur yang dapat mempermudah pencarian berdasarkan *keyword* yang di ketikkan oleh pengguna.

6. Hak Akses

a) Admin

Admin adalah orang yang dapat mengelola data kamus keperawatan pada *website* seperti menambah, mengedit dan menghapus data istilah kata yang ada. Selain itu, admin juga dapat melakukan pencarian di kamus keperawatan berbasis *website*.

b) Pengguna

Pengguna adalah orang yang dapat mengakses kamus keperawatan berbasis *website* tanpa harus *login* terlebih dahulu dan dapat melakukan pencarian kata istilah di kamus keperawatan.

1.4 Tujuan

Pembuatan kamus keperawatan berbasis *website* ini sebagai sarana dan media alternatif yang membantu pengguna didalam mencari dan memperoleh informasi terkait istilah-istilah keperawatan dengan memanfaatkan teknologi berbasis *website*.

1.5 Manfaat

Manfaat dari penelitian ini, yaitu dapat membantu pengguna dalam mencari dan memperoleh informasi terkait istilah-istilah keperawatan yang ada didaftar/list dari kamus keperawatan berbasis *website* secara cepat tanpa harus membawa banyak buku kamus.

1.6 Sistematika Penulisan

Sistematika yang digunakan di dalam penulisan Tugas Akhir (TA) ini disusun untuk memberikan gambaran umum tentang pembuatan laporan yang akan dibagi menjadi beberapa bab dan sub-bab.

Adapun pembagian bab antara lain sebagai berikut:

BAB I PENDAHULUAN

Pada bab pendahuluan ini, penulis menguraikan tentang latar belakang yang menjelaskan mengenai objek aplikasi ini dan permasalahan apa yang mungkin dapat diselesaikan dengan pembuatan aplikasi ini, perumusan masalah yang dihadapi oleh objek tersebut, batasan masalah yang akan dibahas dan diselesaikan, tujuan dan manfaat penelitian, sistematika penulisan dan jadwal kegiatan .

BAB II LANDASAN TEORI

Pada bab ini memuat tinjauan pustaka yang didalamnya terdapat beberapa contoh penelitian-penelitian yang relevan terhadap kajian penelitian yang telah dilakukan oleh beberapa pihak sebelumnya. Selain itu, bab ini juga memuat masalah tentang konsep dan prinsip dasar yang diperlukan untuk memecahkan masalah untuk merumuskan suatu hipotesis.

BAB III METODOLOGI PENELITIAN

Pada bab ini diuraikan tentang bagaimana proses desain atau perancangan aplikasi dan pengembangan yang digunakan untuk merancang aplikasi tersebut.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini berisi hasil rancangan aplikasi seperti rancangan penerapan dan tahap penggunaan dan pengujian. Bagian implementasi berupa tutorial mengenai aplikasi yang telah dibuat.

BAB V KESIMPULAN DAN SARAN

Bab ini memuat kesimpulan dan saran dari awal sampai terbentuknya aplikasi.

DAFTAR PUSTAKA

Berisi literatur yang digunakan sebagai acuan pembuatan laporan Tugas Akhir (TA).

LAMPIRAN

1.7 Jadwal Kegiatan

Tabel 1.1 Jadwal Tugas Akhir

	Bulan Maret 2019				Bulan April 2019				Bulan Mei 2019				Bulan Juni 2019				Bulan Juli 2019				Bulan November 2019				Bulan Januari 2020			
	Minggu				Minggu				Minggu				Minggu				Minggu				Minggu							
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Penyusunan Proposal	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■												
Seminar Proposal																												
Pengumpulan Data																	■	■	■	■								
Analisis dan Desain																	■	■	■	■								
Implementasi																	■	■	■	■								
Testing																	■	■	■	■								
Pembuatan Laporan Hasil																	■	■	■	■								
Seminar Hasil																					■	■	■	■				
Pembuatan Laporan Akhir																									■	■	■	■
Seminar Akhir																									■	■	■	■

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam hal ini telah diperoleh beberapa contoh penelitian-penelitian yang relevan terhadap kajian penelitian yang telah dilakukan oleh beberapa pihak sebelumnya. Oleh sebab itu untuk menjaga agar tidak ada duplikasi, maka penulis melakukan penelusuran untuk bahan perbandingan antara penelitian yang penulis lakukan dengan penelitian yang telah dilakukan oleh pihak lain. Berikut beberapa perbandingan yang menjadi dasar acuan dalam penelitian yang relevan. Berikut ini adalah *resume* dari jurnal-jurnal yang telah penulis kumpulkan:

1. Penelitian ini dilakukan oleh Harjanto, dkk (2012), dengan judul Sistem Temu Kembali Informasi pada Dokumen Teks Menggunakan Metode *Term Frequency Inverse Dokumen Frequency (TF-IDF)*. Aplikasi ini dibangun menggunakan perangkat lunak *Borland Delphi 6.0*. Sistem Temu Kembali Informasi merupakan sistem yang berfungsi untuk menemukan informasi yang relevan dengan kebutuhan pemakai. Dalam konteks ini, temu kembali informasi berkaitan dengan representasi, penyimpanan, dan akses terhadap dokumen representasi dokumen.
2. Penelitian ini dilakukan oleh Herwijayanti dkk (2018), dengan judul Klasifikasi Berita *Online* dengan menggunakan Pembobotan *TF-IDF* dan *Cosine Similarity*. Dalam klasifikasi berita *online* dengan menggunakan pembobotan *tf-idf* dan *cosine similarity* ini mendapatkan referensi penelitian sebelumnya mengenai klasifikasi berita *online* menggunakan algoritma *single pass clustering*, dimana data yang akan digunakan berasal dari *website* berita *online* yaitu *kompas.com*. Karena banyaknya berita yang dimasukkan ke dalam *website*, sehingga terkadang berita tersebut terposting tidak sesuai dengan kategorinya. *Human error* akan menjadi masalah berita yang salah posting. Selain kesalahan posting pengelompokan berita *online* juga penting untuk kenyamanan *user* untuk mencari berita sesuai dengan kategorinya. Menerapkan

klasifikasi berita *online* dengan menggunakan *tf-idf* dan *cosine similarity*, memerlukan proses *preprocessing* yaitu *tokenizing*, *stopword* dan *stemming* dapat memperkecil term sehingga mempercepat proses perhitungan pembobotan term menggunakan *tf-idf* dan mempercepat proses *cosine similarity*. Tujuannya adalah untuk mempermudah *human error* serta mengurangi terjadinya kesalahan pengkategorian.

3. Penelitian ini dilakukan oleh Ria Melita dkk (2018), dengan judul Penerapan Metode *Term Frequency Inverse Document Frequency (Tf-Idf)* dan *Cosine Similarity* pada Sistem Temu Kembali Informasi Untuk Mengetahui *Syarah Hadits* Berbasis Web (Studi Kasus: *Syarah Umdatil Ahkam*). *Hadits* merupakan sumber ajaran Islam di samping *Al-Qur'an*. Tanpa *hadits*, syari'at Islam tidak dapat dimengerti secara utuh dan tidak dapat dilaksanakan. Namun dewasa ini, tidak sedikit orang yang keliru dalam memahaminya, hal tersebut disebabkan oleh banyaknya orang yang memahami *hadits* sebatas mengandalkan teks lahiriyah saja. Salah satu hal yang dapat kita tempuh untuk mengetahui makna yang terkandung dalam *hadits* adalah dengan mempelajari *syarah hadits* guna meminimalisir kesalahan penafsiran terhadap suatu *hadits*. Sejauh ini aplikasi *syarah hadits* yang ada masih terbatas, yaitu dalam bahasa *full arab* yang tidak semua orang dapat memahaminya. Oleh karena itu, diperlukan jangkauan pemahaman dalam ilmu *hadits* dan dengan adanya sistem yang akan dibangun, maka akan memberikan solusi permasalahan tersebut, yaitu Sistem Temu Kembali Informasi yang dapat dimanfaatkan untuk mengetahui *syarah hadits*, karena dapat memberikan alternatif berupa metode *similarity* yang dapat digunakan untuk melakukan pencarian dokumen relevan dengan yang kita inginkan. Metode *similarity* yang digunakan adalah *cosine similarity* dengan pembobotan kata menggunakan metode *TF-IDF* dan menerapkan teks *preprocessing* terlebih dahulu untuk memperkecil *term* sehingga bisa mempercepat proses perhitungan *term*. Teks *preprocessing* tersebut meliputi *tokenizing*, *stopword removal* atau *filtering*, dan *stemming*.

Tabel 2.1 Tabel Perbandingan Penelitian yang Relevan

	1	2	3	4
Judul Penelitian	Sistem Temu Kembali Informasi pada Dokumen Teks Menggunakan Metode <i>Term Frequency Inverse Dokumen Frequency (TF-IDF)</i>	Klasifikasi Berita <i>Online</i> dengan menggunakan Pembobotan <i>TF-IDF</i> dan <i>Cosine Similarity</i>	Penerapan Metode <i>Term Frequency Inverse Document Frequency (Tf-Idf)</i> dan <i>Cosine Similarity</i> pada Sistem Temu Kembali Informasi Untuk Mengetahui <i>Syarah Hadits</i> Berbasis Web (Studi Kasus: <i>Syarah Umdatil Ahkam</i>)	Rancang Bangun Aplikasi Kamus Keperawatan Menggunakan Algoritma Berbasis <i>Website</i>
(Nama, Tahun)	(Syafe, Sukmawati dan Bahtiar; 2012)	(Herwijayanti, Eka, Muflikhah; 2018)	(Melita, Amrizal, Bayu, Dirjam; 2018)	(Martalia, 2019)
Pengguna	<i>User</i>	<i>User</i>	<i>User</i>	<i>Admin, Pengunjung</i>

Sambungan Tabel 2.1 Tabel Perbandingan Penelitian yang Relevan

<p>Tujuan</p>	<p>Sistem Temu Kembali Informasi merupakan sistem yang berfungsi untuk menemukan informasi yang relevan dengan kebutuhan pemakai.</p>	<p>Tujuannya adalah untuk mempermudah human <i>error</i> serta mengurangi terjadinya kesalahan pengkategorian.</p>	<p>Penelitian ini diharapkan dapat memberikan <i>output</i> berupa <i>syarah hadits</i> yang sesuai berdasarkan <i>input</i> yang dimasukkan dengan menggunakan Metode <i>Term Frequency Inverse Document Frequency (TF-IDF)</i> dan <i>Cosine Similarity</i>.</p>	<p>Pembuatan aplikasi Kamus Keperawatan berbasis <i>website</i> ini sebagai sarana dan media alternatif yang membantu pengguna didalam mencari dan memperoleh informasi terkait istilah-istilah keperawatan dengan memanfaatkan teknologi berbasis <i>website</i>.</p>
<p>Algoritma pencarian dan metodologi Perangkat Lunak</p>	<p>Algoritma <i>Term Frequency Inverse Dokumen Frequency (TF-IDF)</i></p>	<p>Algoritma <i>Term Frequency Inverse Dokumen Frequency TF-IDF</i> dan <i>Cosine Similarity</i></p>	<p>Algoritma <i>Term Frequency Inverse Dokumen Frequency TF-IDF</i> dan <i>Cosine Similarity</i></p>	<p>Algoritma <i>Term Frequency Inverse Dokumen Frequency (TF-IDF)</i></p>

Sambungan Tabel 2.1 Tabel Perbandingan Penelitian yang Relevan

Fitur	<ol style="list-style-type: none"> 1. Tokenizing Dokumen 2. <i>Stopword Removal</i> Dokumen 3. <i>Stemming</i> Dokumen 4. Pembobotan Dokumen 	<ol style="list-style-type: none"> 1. Menu Utama 2. Berita 3. Data Test 4. Klaster 	Analisa <i>Hadits</i>	<ol style="list-style-type: none"> a) <i>Dashboard</i> b) Kelola Kamus c) Pencarian
-------	--	--	-----------------------	--



2.2 Pengertian Kamus

“Secara etimologi, kata kamus berasal dari kata dalam bahasa Arab, yaitu qamus (bentuk jamaknya *qawamus*). Bahasa Arab menyerap kata kamus dari kata dalam bahasa Yunani kuno, yaitu okeanos yang berarti lautan” (Chaer, 2007).

Menurut (Waridah, Wahya, & Suzzana, 2013), kamus adalah buku acuan yang memuat kata dan ungkapan yang disusun menurut abjad beserta keterangan tentang makna, pemakaian atau terjemahannya. Kamus disusun sesuai dengan abjad dari A-Z untuk mempermudah pengguna kamus dalam mencari kata, serta kamus juga disusun secara khusus dengan arti makna kata dalam bidang tertentu yang disebut kamus istilah, fungsinya untuk kegunaan ilmiah, misalnya kamus keperawatan yang memuat arti istilah-istilah ilmiah dalam bentuk bahasa latin seperti nama penyakit, gejala-gejala, tindakan, alat-alat dan sebagainya.

2.3 Pengertian Keperawatan

Keperawatan adalah suatu bentuk pelayanan kesehatan yang bersifat profesional dalam memenuhi kebutuhan dasar manusia (biologis, psikologis, sosial dan spiritual) yang dapat ditujukan kepada individu, keluarga atau masyarakat dalam rentang sehat-sakit (Hidayat, 2009). Dalam keperawatan, dikenal juga proses keperawatan yang merupakan cara sistematis yang dilakukan oleh perawat bersama klien dalam menentukan kebutuhan asuhan keperawatan dengan melakukan pengkajian, menentukan diagnosis, merencanakan tindakan, melaksanakan tindakan serta mengevaluasi hasil asuhan yang telah diberikan dengan berfokus pada klien (Hidayat, 2009).

Menurut (Hidayat, 2009), dalam proses keperawatan terdapat beberapa tujuan khusus diantaranya:

1. Dapat mengidentifikasi berbagai kebutuhan dasar manusia yang dibutuhkan.
2. Dapat menentukan diagnosis keperawatan yang ada pada manusia setelah dilakukan identifikasi.
3. Dapat menentukan rencana tindakan yang akan dilakukan setelah diagnosis ditegakkan.
4. Dapat melaksanakan tindakan keperawatan setelah direncanakan.
5. Dapat mengetahui perkembangan pasien dari berbagai tindakan yang telah dilakukan, untuk menentukan tingkat keberhasilan.

2.4 Text Mining

Text mining merupakan proses analisis dalam data yang berupa teks dimana sumber data didapatkan dari dokumen (Feldman, 2007). Konsep *text mining* biasanya digunakan dalam klasifikasi dokumen tekstual dimana dokumen-dokumen tersebut akan diklasifikasikan sesuai dengan topik dokumen tersebut. Dengan bantuan *text mining* suatu artikel dapat diketahui jenis kategorinya melalui kata-kata yang terdapat pada artikel tersebut. Kata-kata yang dapat mewakili isi dari artikel tersebut dianalisa dan dicocokkan pada basis data kata kunci yang telah ditentukan sebelumnya. Sehingga dengan adanya *text mining* dapat membantu melakukan pengelompokan suatu dokumen dengan waktu yang singkat.

Tahapan dalam melakukan analisa pada *text mining* yaitu melakukan pengumpulan data kemudian melakukan ekstraksi terhadap fitur yang akan digunakan (Feldman, 2007). Adapun teknik yang digunakan dalam ekstraksi fitur yaitu melakukan pembersihan data mulai dari *tokenizing*, *stop words removal* dan *stemming*. Selanjutnya yaitu melakukan transform data dengan pembobotan terhadap term yang telah dibersihkan. Kemudian dilanjutkan dengan reduksi data. Tahap terakhir yaitu melakukan analisis terhadap proses klasifikasi untuk merepresentasikan hasil informasi yang ditemukan.

2.5 Processing Data

Processing bertujuan untuk mendapatkan *dataset* yang dapat diolah dengan cepat dan menghasilkan kesimpulan yang tepat. Salah satu proses *processing* data yang dapat dilakukan adalah pemilihan fitur (*feature selection*). Ada beberapa tahapan dalam pemilihan fitur, antara lain:

Tokenizing merupakan tahap pemotongan *string* input untuk memisahkan kalimat menjadi kata. *Stopword*, pada tahap ini dilakukan proses menghilangkan kata yang tidak penting dalam teks. Untuk proses ini diperlukan suatu kamus kata-kata yang menyimpan kata-kata yang bisa dihilangkan.

2.5.1 Tokenizing

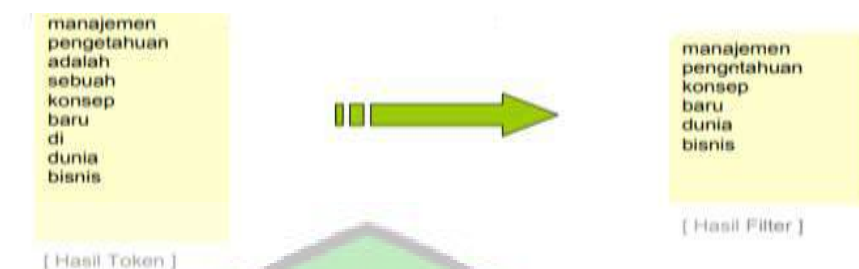
Tokenizing adalah proses memecah dokumen menjadi kumpulan kata. *Tokenization* dapat dilakukan dengan menghilangkan tanda baca dan memisahkannya per-spasi. Tahapan ini juga menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua token ke bentuk huruf kecil (*lower case*).



Gambar 2.1 Contoh tahap *Tokenizing*

2.5.2 Stopword Removal atau Filtering

Stopwords removal merupakan proses penghilangan kata tidak penting pada deskripsi melalui pengecekan kata-kata hasil parsing deskripsi apakah termasuk di dalam daftar kata tidak penting (*stoplist*) atau tidak.



Gambar 2.2 Contoh tahap *Stopword Removal* atau *Filtering*

2.6 Pembobotan *Term Frequency Inverse Document Frequency (TF-IDF)*

Term Frequency Inverse Document Frequency atau *TF-IDF* adalah suatu metode algoritma yang berguna untuk menghitung bobot setiap kata yang umum digunakan. Metode ini juga terkenal efisien, mudah dan memiliki hasil yang akurat. Metode ini akan menghitung nilai *Term Frequency (TF)* dan *Inverse Document Frequency (IDF)* pada setiap token (kata) di setiap dokumen dalam korpus. Secara sederhana, metode *TF-IDF* digunakan untuk mengetahui berapa sering suatu kata muncul di dalam dokumen.

Pada *Term Frequency (TF)*, terdapat beberapa jenis formula yang dapat digunakan:

1. *TF biner (binary TF)*, hanya memperhatikan apakah suatu kata atau *term* ada atau tidak dalam dokumen, jika ada diberi nilai satu (1), jika tidak diberi nilai nol (0).
2. *TF murni (raw TF)*, nilai *TF* diberikan berdasarkan jumlah kemunculan suatu *term* di dokumen. Contohnya, jika muncul lima (5) kali maka kata tersebut akan bernilai lima (5).
3. *TF normalisasi*, menggunakan perbandingan antara frekuensi sebuah *term* dengan nilai maksimum dari keseluruhan atau kumpulan frekuensi *term* yang ada pada suatu dokumen.

4. *TF* logaritmik, hal ini untuk menghindari dominansi dokumen yang mengandung sedikit *term* dalam *query*, namun mempunyai frekuensi yang tinggi. Nilai *TF* diformulasikan pada Persamaan (1).

$$TF = \begin{cases} 1 + \log_{10}(f_{t,d}), & f_{t,d} > 0 \\ 0, & f_{t,d} = 0 \end{cases} \quad (1)$$

Dimana :

TF = Frekuensi istilah (TF) sama dengan jumlah kali (frekuensi) suatu istilah ditemukan dalam dokumen yang diberikan.

$f_{t,d}$ = Frekuensi kemunculan *term* di dalam *document*.

Jadi, jika suatu kata atau term terdapat dalam suatu dokumen sebanyak 5 kali maka diperoleh bobot = $1 + \log(5) = 1.699$. Tetapi jika term tidak terdapat dalam dokumen tersebut, bobotnya adalah nol (0).

Lalu, yang berikutnya atau yang kedua adalah *IDF* (*Inverse Document Frequency*). Metode *IDF* merupakan sebuah perhitungan dari bagaimana *term* didistribusikan secara luas pada koleksi dokumen yang bersangkutan.

Berbeda dengan *TF* yang semakin sering frekuensi kata muncul maka nilai semakin besar, dalam *IDF*, semakin sedikit frekuensi kata muncul dalam dokumen, maka makin besar nilainya. Untuk menentukan besaran nilai *IDF*, diformulasikan pada Persamaan (2).

$$IDF = \log \left[\frac{D}{df} \right] \quad (2)$$

Dimana :

D = Jumlah total dokumen.

df = Jumlah dokumen yang mengandung kata yang di cari dalam dokumen.

Jenis formula TF yang biasa digunakan untuk perhitungan adalah TF murni (*raw TF*). Dengan demikian rumus umum untuk *Term Weighting TF-IDF* adalah penggabungan dari formula perhitungan *raw TF* dengan formula IDF dengan cara mengalikan nilai TF dengan nilai IDF . Rumus umum untuk *Term Weighting TF-IDF*, diformulasikan pada Persamaan (3).

$$w_{t,d} = tf_{t,d} \times IDF \quad (3)$$

Dimana :

- t = Kata ke- t dari kata kunci atau kata yang dicari.
- d = Dokumen ke- d .
- w = Bobot kata ke- t di dalam dokumen ke- d .
- tf = Banyak kata yang dicari di dalam dokumen.
- IDF = *Inverse Document Frequency*.

Berapapun besarnya nilai $tf_{t,d}$, apabila $D = df$, maka akan didapatkan hasil 0 (nol), dikarenakan hasil dari $\log 1$, untuk perhitungan IDF . Untuk itu dapat ditambahkan nilai 1 pada sisi IDF , sehingga perhitungan bobotnya diformulasikan pada Persamaan (4).

$$TF-IDF = w_{t,d} + 1 \quad (4)$$

Jadi, $w_{t,d}$ adalah bobot kata ke- t dari kata kunci atau kata yang dicari di dalam dokumen ke- d .

2.7 Website

World wide web sering disingkat dengan *www* atau *web* adalah suatu metode untuk menampilkan informasi di internet, baik berupa teks, gambar, suara maupun video yang interaktif dan mempunyai banyak kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen lainnya (*hypertext*) yang dapat diakses melalui sebuah *browser*. *Browser* adalah perangkat lunak

untuk mengakses halaman-halaman *web*, seperti Internet Explorer, Mozilla Firefox, Opera, Safari, dan lain-lain.

Informasi yang disajikan melalui *browser* dibangun dengan bahasa semi pemrograman HTML (*HyperText Markup Language*), dan kemudian ditingkatkan fungsinya dengan menyisipkan kode-kode bahasa pemrograman *web*, seperti PHP, ASP, JSP dan lain-lain, sehingga mampu menampilkan informasi yang lebih interaktif dan dinamis serta terhubung dengan *database*. Sir Timothy Berners-Lee adalah penemu konsep *www*, *HyperText Markup Language* (HTML), perancang skema pemberian alamat (URL), mendesain aturan-aturan di *web* yang kemudian menjadi HTTP (*HyperText Transfer Protocol*).

Website (situs *web*) adalah kumpulan dari halaman *web* yang terdapat pada satu domain, yang terdiri dari dua atau lebih halaman *web*. Faktor utama yang membuat *website* begitu cepat berkembang adalah karena penyebaran informasi melalui *website* sangat cepat dan mencakup area yang luas (dunia), tidak dibatasi oleh jarak dan waktu. (Yuhfizar, 2008)

2.8 Web Server

Web server adalah perangkat lunak *server* yang berfungsi menerima permintaan *HTTP* atau *HTTS* dari *client* yang dikenal dengan nama *web browser* dan akan mengirim kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen *HTML*. Salah satu *web server* yang banyak dipakai adalah *Aphace*. *Aphace* merupakan *web server* antara *platform* yang dapat berjalan di beberapa platform seperti *Linux* dan *Windows*.

2.9 Hypertext Markup Language (HTML)

HTML atau singkatan dari *HyperText Markup Language* adalah *script* dimana kita biasa menampilkan informasi dan daya kreasi kita lewat internet.

HTML sendiri adalah suatu dokumentasi teks biasa yang mudah dimengerti dibanding bahasa pemrograman lainnya dan karena bentuknya itu *HTML* dapat dibaca oleh berbagai *platform* seperti *Windows*, *Linux*, dan *Macintosh*.

2.10 *Personal Home Page (PHP)*

PHP adalah *script* yang berjalan pada *server side* yang ditambahkan pada *HTML*. *Script* ini akan membuat suatu aplikasi yang dapat diintegrasikan kedalam *HTML* sehingga suatu halaman *HTML* tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat *server side* membuat pengerjaan *script* tersebut dikerjakan di *server* sedangkan yang dikirimkan kepada *browser* adalah hasil proses dari *script* tersebut yang sudah berbentuk *HTML*.

PHP dibuat pada tahun 1994 oleh Rasmus Lerdof. Tetapi dikembangkan oleh orang lain dan setelah memulai tiga kali karya penulisan, akhirnya *PHP* menjadi bahasa pemrograman web. *PHP* adalah sebuah produk yang berbentuk *open source*, sehingga *source code-code* dari *PHP* dapat digunakan, diganti, diedit tanpa harus dikenakan biaya.

2.11 Penggabungan *Script PHP* dan *HTML*

Bahasa pemrograman *PHP* dapat digabungkan dengan *HTML* dengan terlebih dahulu memberikan tanda tag buka dilanjutkan tanda tanya (<?) kemudian ditutup dengan tanda tanya dilanjutkan tanda tag tutup (?>). Ada dua tipe penggabungan antara *PHP* dan *HTML* yaitu:

a.) *Embedded Script*

Embedded script adalah cara penulisan tag *PHP* disela-sela tag *HTML*. Dengan cara ini, penulisan tag *PHP* digunakan untuk mengapit bagian-bagian tertentu dalam dokumen yang memerlukan *script PHP* untuk proses di dalam *server*.

b.) *Non Embedden Script*

Non Embedded Script adalah cara penulisan tag *PHP* dibagian paling awal dan paling akhir dokumen. Dengan cara ini, penulisan tag *PHP* digunakan untuk mengawali dan mengakhiri keseluruhan bagian dalam sebuah dokumen. *Non Embedded Script* menempatkan *script HTML* sebagai bagian dari *script PHP*.

2.12 Data dan Basis Data

2.12.1 Pengertian Data

Data berasal dari kata datum yang berarti fakta yang mengandung arti, yang dihubungkan dengan kenyataan, simbol-simbol, gambar-gambar, kata-kata, huruf-huruf atau simbol-simbol yang menunjukkan ide objek, kondisi atau situasi yang lain. Manfaat data adalah sebagai suatu representasi yang dapat diingat, direkam, dan diolah menjadi informasi.

2.12.2 Basis Data

Menurut C.J. Date (2000, hal : 2) Basis data adalah sebuah sistem yang di buat untuk mengorganisasi, menyimpan dan menarik data dengan mudah. Pengertian lainnya, Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi bagi para pemakai. Penerapan *database* dalam sistem informasi disebut dengan *database system*. Sistem basis data (*database system*) adalah suatu sistem informasi yang mengintegrasikan kumpulan dari daya yang saling berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi. Pendekatan database berbeda dengan pendekatan tradisional.

Adapun tujuan dari pengolahan data yang dilakukan adalah untuk menghasilkan suatu keluaran yang nantinya dipergunakan, yaitu sebagai dasar untuk pengambilan keputusan informasi.

2.13 MySQL

MySQL adalah sebuah aplikasi *Relational Database Managemen Server (RDBMS)*. Dengan menggunakan *MySQL server*, maka data dapat diakses oleh banyak pemakai secara bersamaan. *MySQL* menggunakan bahasa *SQL (Structure Query Language)* yaitu bahasa pemrograman standar yang digunakan untuk mengakses *server database*.

Tiap *database* memiliki table-tabel, tiap tabel memiliki *field-field*. Umumnya informasi tersimpan dalam tabel-tabel yang secara logis merupakan struktur-struktur dimensi terdiri atas baris dan kolom. *Field-field* tersebut dapat berupa data seperti *int, real, char, date, time*, dan lainnya. Keunggulan *MySQL*, di antaranya:

1. Bekerja pada berbagai platform (tersedia berbagai versi untuk berbagai sistem operasi).
2. Pengaksesan database dapat dilakukan dengan mudah.
3. Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem database.
4. Mendukung record yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

MySQL dan *PHP* saling terintegrasi, maksudnya adalah pembuatan *database* dengan menggunakan sintak *PHP* dapat dibuat. Sedangkan input yang dimasukkan melalui aplikasi web yang menggunakan *script server side* seperti *PHP* dapat langsung dimasukkan ke *database MySQL* yang ada di *server* dan tentunya web tersebut berada di sebuah *web server*.

2.14 Internet

Internet adalah metode untuk menghubungkan berbagai komputer ke dalam satu jaringan komputer global, melalui protokol yang disebut *Transmission Control Protocol/ Internet Protokol (TCP/ IP)*. Protokol adalah suatu petunjuk yang menunjukkan pekerjaan yang akan pengguna lakukan dengan internet, apakah akan mengakses situs web melakukan *transfer file*, mengirim *email*, dan sebagainya.

2.15 Web Browser

Web Browser atau sering disebut dengan *Browser*, merupakan suatu *software* atau program aplikasi yang beroperasi di setiap komputer pribadi (*client*) yang meminta informasi dari *server* dan menampilkannya sehingga data-datanya dapat langsung diakses. *Web browser* adalah suatu program yang dirancang untuk mengambil informasi dari suatu *server* komputer pada jaringan internet. Informasi-informasi ini dikemas dalam *page-page*, dimana *page-page* bisa memiliki beberapa link yang menghubungkan *web page* tersebut ke sumber informasi lainnya.

2.16 XAMPP

XAMPP ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program yang mempunyai fungsi sebagai *server* yang berdiri sendiri (*localhost*) dan terdiri dari program *MySQL database*, *Apache HTTP Server* dan penerjemah ditulis dalam bahasa pemrograman *PHP* dan *Perl*. Komponen dan fungsi pada *XAMPP*, yaitu:

1. *Htdoc* adalah folder di mana Anda meletakkan *file* yang akan dijalankan, seperti *file PHP*, *HTML* dan *script* lainnya.
2. *PhpMyAdmin* adalah bagian untuk mengelola *database MySQL* yang dikomputer.

3. Untuk membukanya, membuka *browser* dan ketik alamat `http:// localhost / phpMyAdmin`, halaman *phpMyAdmin* akan muncul.
4. *Control Panel* yang berfungsi untuk mengelola layanan (*service*) *XAMPP*. Seperti *stop service* (berhenti), atau mulai (mulai).

2.17 Pengertian Dasar *Flowchart*

Flowchart adalah gambar yang menampilkan struktur bagan dari setiap halaman. Bagan aliran (*flowchart*) ini menjelaskan secara rinci langkah-langkah dari proses program. *Flowchart* dibuat dengan menggunakan simbol-simbol tertentu yang menyatakan tiap langkah program. *Flowchart* merupakan cara penyajian dari suatu algoritma (Yatini, 2010). Tujuan Membuat *Flowchart*, yaitu:

- a. Menggambarkan suatu tahapan penyelesaian masalah.
- b. Secara sederhana, terurai, rapi dan jelas.
- c. Menggunakan simbol-simbol standar.

Dalam penulisan *flowchart* dikenal dua model, yaitu sistem *flowchart* dan program *flowchart*.

1. Sistem *Flowchart*

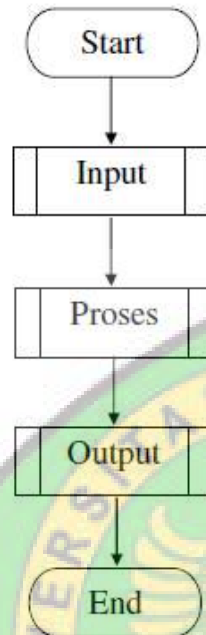
Sistem *Flowchart* merupakan bagan yang memperlihatkan urutan *prosedure* dan proses dari beberapa *file* di dalam media tertentu. Melalui *flowchart* ini terlihat jenis media penyimpanan yang dipakai dalam pengolahan data.

- a. Selain itu juga menggambarkan *file* yang dipakai sebagai *input* dan *output*.
- b. Tidak digunakan untuk menggambarkan urutan langkah untuk memecahkan masalah.
- c. Hanya untuk menggambarkan prosedur dalam sistem yang dibentuk.

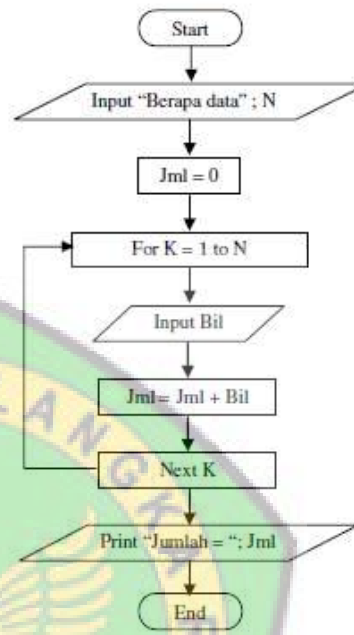
2. Program *Flowchart*

Bagan yang memperlihatkan urutan dan hubungan proses dalam suatu program. Dua jenis metode penggambaran program *flowchart* :

- Conceptual flowchart*, menggambarkan alur pemecahan masalah secara global.
- Detail flowchart*, menggambarkan alur pemecahan masalah secara rinci.



Gambar 2.3 *Conceptual Flowchart*



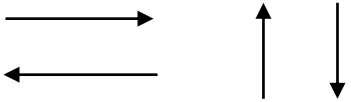
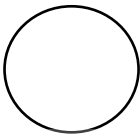
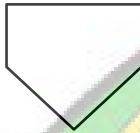


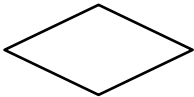
Gambar 2.4 *Detail Flowchart*

- Simbol-simbol *Flowchart*



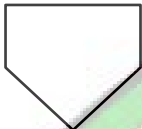
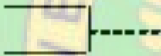

Simbol-simbol yang di pakai dalam *flowchart* dibagi menjadi 3 kelompok:

- Flow direction symbols*, digunakan untuk menghubungkan simbol satu dengan yang lain. Disebut juga *connecting line*.
- Processing symbols*, menunjukan jenis operasi pengolahan dalam suatu proses / prosedur.
- Input / Output symbols*, menunjukkan jenis peralatan yang digunakan sebagai media input atau output.

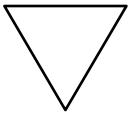
Tabel 2.2 *Flow Direction Symbols*

Simbol	Keterangan
	<p>Simbol arus / <i>flow</i>, yaitu menyatakan jalannya arus suatu proses.</p>
	<p>Simbol <i>connector</i>, berfungsi menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.</p>
	<p>Simbol <i>offline connector</i>, menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.</p>
	<p>Simbol <i>process</i>, yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.</p>
	<p>Simbol <i>manual</i>, yaitu menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.</p>
	<p>Simbol <i>decision</i>, yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya / tidak.</p>



Sambungan Tabel 2.2 *Flow Direction Symbols*

	<p>Simbol <i>predefined process</i>, yaitu menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.</p>
	<p>Simbol <i>terminal</i>, yaitu menyatakan permulaan atau akhir suatu program.</p>
	<p>Simbol <i>offline connector</i>, menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.</p>
	<p>Simbol penjelasan, digunakan untuk komentar tambahan.</p>
	<p>Simbol <i>keying operation</i>, yaitu menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i>.</p>

Tabel 2.3 *Storage Symbols*

Simbol	Keterangan
	<p>Simbol <i>offline-storage</i>, menunjukkan bahwa data dalam simbol ini akan disimpan ke dalam media tertentu.</p>

Sambungan Tabel 2.3 *Storage Symbols*

	<p>Simbol <i>manual input</i>, memasukkan data dengan menggunakan <i>online keyboard</i>.</p>
	<p>Simbol <i>input/output</i>, menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya.</p>
	<p>Simbol <i>punched card</i>, menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.</p>
	<p>Simbol <i>sub program</i>, yaitu menyatakan bagian dari program</p>
	<p>Simbol <i>disk storage</i>, menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i>.</p>
	<p>Simbol dokumen, yang akan mencetak keluaran dalam bentuk dokumen (melalui <i>printer</i>).</p>
	<p>Simbol <i>display</i>, yang akan mencetak keluaran dalam layar monitor.</p>
	<p>Simbol <i>magnetic disk</i>, menyatakan input berasal dari <i>disk magnetis</i> atau output disimpan ke <i>disk magnetis</i></p>

2.18 Data Flow Diagram (DFD)

3.18.1 Definisi Data Flow Diagram

Data flow diagram didefinisikan sebagai diagram yang mempresentasikan bagaimana informasi keluar masuk dari sistem, proses apa yang mengubah informasi tersebut dimana informasi disimpan, dan data flow diagram juga didefinisikan sebagai diagram untuk menggambarkan aliran data dalam sistem, sumber dan tujuan data, proses yang mengolah data tersebut dan tempat penyimpanan datanya. Diperkenalkan oleh Tom DeMarco (1978) dan Cris Gane dan Trish Sarson (1977).

3.18.2 Komponen DFD

Ada empat komponen yang membentuk suatu data flow diagram menurut Yourdan dan DeMarco, yaitu aliran data, proses, penyimpanan data dan sumber/tujuan data. Berikut adalah notasi yang digunakan untuk DFD.

a. Aliran Data (Data Flow)

Aliran data/ arus data atau data flow di DFD diberi simbol panah. Aliran menggambarkan perpindahan informasi dari satu bagian ke bagian lain dari sistem. Aliran data ini mengalir diantara proses (*process*), simpanan data (*data store*) dan kesatuan luar (*external entity*). Aliran data ini menunjukkan aliran data yang dapat berupa masukan untuk proses atau simpanan data dan berupa keluaran atau hasil dari suatu proses. Penghubung antar proses yang mempresentasikan informasi yang dibutuhkan proses sebagai keluaran. Aliran paket informasi dari satu bagian sistem ke bagian sistem lainnya. Umumnya mengalir antar proses, tetapi dapat juga mengalir keluar masuk dari ke file (*data store*) atau dari sumber tujuan data. Awal panah menggambarkan asal data sedangkan arah panah menggambarkan tujuan data.



Gambar 2.5 Simbol aliran data dalam DFD

b. Proses

Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Proses menggambarkan bagian dari system yang mengolah masukkan menjadi keluaran. Proses diberi nama untuk menjelaskan proses atau kegiatan apa yang sedang atau akan dilaksanakan. Pemberian nama proses dilakukan dengan menggunakan kata kerja yang membutuhkan objek. Proses di gambarkan dengan sebuah lingkaran.



Gambar 2.6 Simbol proses dalam DFD

c. Penyimpanan Data (*Data Store*)

Simbol ini digunakan untuk menggambarkan penyimpanan data. Simpanan data atau data store merupakan tempat penyimpanan data berupa *file* atau *database*.



Gambar 2.7 Simbol Data Store dalam DFD

d. Entitas Eksternal (*Terminator*)

Menggambarkan entitas yang berinteraksi dengan sistem yang berada diluar lingkup sistem (bukan yang menjalankan sistem tersebut

atau entitas yang berfungsi sebagai *producer/consumer* dari sistem (sumber atau tujuan data). Merupakan simbol yang menggambarkan entitas yang dapat berupa orang, kelompok, atau organisasi yang berhubungan dengan sistem.



Gambar 2.8 Simbol terminator dalam DFD

2.19 Entity Relationship Diagram (ERD)

Model *Entity Relationship* diperkenalkan pertama kali oleh P.P. Chen pada tahun 1976. Model ini dirancang untuk menggambarkan persepsi dari pemakai dan berisi obyek-obyek dasar yang disebut *entity* dan hubungan antar *entity-entity* tersebut yang disebut *relationship*. Pada model *ER* ini semesta data yang ada dalam dunia nyata ditransformasikan dengan memanfaatkan perangkat konseptual menjadi sebuah diagram, yaitu diagram *ER* (*Entity Relationship*).

Diagram *Entity-Relationship* melengkapi penggambaran grafik dari struktur logika. Dengan kata lain Diagram *E-R* menggambarkan arti dari aspek data seperti bagaimana *entity-entity*, atribut-atribut dan *relationship-relationship* disajikan. Sebelum membuat Diagram *E-R*, tentunya kita harus memahami betul data yang diperlukan dan ruang lingkungannya. Di dalam pembuatan diagram *E-R* perlu diperhatikan penentuan suatu konsep apakah merupakan *entity*, atribut, atau *relationship*.

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. *ERD* untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol.

Pada dasarnya ada tiga simbol yang digunakan, yaitu:

a. Entity

Entity adalah obyek yang dapat dibedakan dengan yang lain dalam dunia nyata. *Entity* dapat berupa objek secara fisik seperti orang, rumah, atau kendaraan. *Entity* dapat pula berupa objek secara konsep seperti pekerjaan, perusahaan, dan sebagainya.

Tipe *entity* merupakan sekumpulan obyek dalam dunia nyata yang mempunyai properti yang sama atau berasal dari *entity* yang sejenis. Terdapat dua tipe *Entity*, *Entity* Kuat dan *Entity* Lemah. *Entity* kuat adalah *entity* yang keberadaannya tidak tergantung pada *entity* lain, misalkan tipe *entity* pegawai atau cabang. Sedangkan *Entity* Lemah keberadaannya tergantung pada *entity* lain, misalkan tipe *entity* tanggungan, dimana keberadaannya tergantung dari pegawai.

Entitas menunjukkan objek-objek dasar yang terkait di dalam *system*. Obyek dasar dapat berupa orang, benda, atau hal yang keterangannya perlu disimpan di dalam basis data.

Untuk menggambarkan sebuah entitas digunakan aturan sebagai berikut (Sutanta, 2004):

- 1) Entitas dinyatakan dengan simbol persegi panjang.
- 2) Nama entitas dituliskan di dalam simbol persegi panjang.
- 3) Nama entitas berupa kata benda, tunggal
- 4) Nama entitas sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

b. Atribut

Atribut adalah karakteristik dari *entity* atau *relationship*, yang menyediakan penjelasan detail tentang *entity* atau *relationship* tersebut. Nilai Atribut merupakan suatu data aktual atau informasi yang disimpan pada suatu atribut di dalam suatu *entity* atau *relationship*. Atribut digambarkan dalam bentuk oval.

Atribut sering pula disebut sebagai properti (*property*), merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut berfungsi sebagai penjelas pada sebuah entitas. Untuk menggambarkan atribut digunakan aturan sebagai berikut (Sutanta, 2004):




- 1) Atribut dinyatakan dengan simbol *elips*.
- 2) Nama atribut dituliskan di dalam simbol *elips*.
- 3) Nama atribut berupa kata benda, tunggal.
- 4) Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.
- 5) Atribut dihubungkan dengan entitas yang bersesuaian dengan menggunakan sebuah garis (seyogianya menggunakan garis lurus, namun dalam kondisi yang tidak memungkinkan dapat juga tidak menggunakan garis lurus).

c. Hubungan/Relasi

Kerelasian antar entitas mendefinisikan hubungan antara dua buah entitas. Kerelasian adalah kejadian atau transaksi yang terjadi di antara dua buah entitas yang keterangannya perlu disimpan dalam basis data. kejadian atau transaksi yang tidak perlu disimpan dalam basis data (sekali pun benar-benar terjadi) bukan termasuk kerelasian. Aturan penggambaran kerelasian antar entitas adalah sebagai berikut (Sutanta, 2004).

- 1) Kerelasian dinyatakan dengan simbol belah ketupat.
- 2) Nama kerelasian dituliskan di dalam simbol belah ketupat.
- 3) Kerelasian menghubungkan dua entitas.
- 4) Nama kerelasian berupa kata kerja aktif (diawali dengan awalan me), tunggal.
- 5) Nama kerelasian sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

Tabel 2.4 Simbol-simbol *Entity Relationship Diagram (ERD)*

Komponen	Simbol
Entity (entitas)	
Atribut	
Relasi (hubungan)	

2.20 Notepad++

Notepad++ merupakan sebuah *software text editor* dan *source code editor* yang berjalan di sistem operasi *Windows*. *Notepad++* merupakan *open source* (perangkat lunak bebas) yang dapat kita miliki secara gratis. *Notepad++* dapat kita gunakan untuk melakukan *editing source code* suatu pemrograman, misalnya kode *HTML* pada pembuatan *template blog*.

2.21 Edraw Max

Edraw Max adalah *software* diagram yang dapat membantu anda membuat bagan organisasi, presentase bisnis, diagram jaringan, rencana pembangunan, peta pikiran, ilmu ilustrasi, desain *fashion*, *UML* diagram, *workflow*, struktur Program, diagram desain web, dan masih banyak lagi yang lainnya yang berhubungan dengan diagram. Dengan *software* ini anda akan lebih mudah membuat berbagai macam diagram apa saja dengan menggunakan *template*, struktur dan bentuk serta alatalat menggambar lainnya dan anda dapat mengeksponnya ke format lainnya seperti; *PDF*, *Word*, *Excel*, *file PowerPoint*, *SVG* atau *EPS*.

2.22 *Blackbox Testing*

Pengertian *black box testing* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa *fungsi* dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitam nya. Sama seperti pengujian *black box*, mengevaluasi hanya dari tampilan luarnya (*interface* nya), fungsionalitasnya tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya. (Taslim, 2013).

Pengujian *Black Box* dilakukan dengan membuat kasus yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah. Kesimpulannya adalah pada antar muka Perangkat Lunak, guna memperlihatkan bahwa fungsi-fungsi Perangkat Lunak dapat beroperasi dan menguji beberapa aspek dasar dengan memperhatikan logika internal. Keunggulan *Black Box* adalah bisa memilih *subset test* secara efektif dan efisien, dapat menemukan cacat dan memaksimalkan *testing investmen*.



BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Pengembangan Perangkat Lunak

Metodologi pengembangan yang digunakan dalam melakukan pembuatan *website* ini adalah menggunakan metode pengembangan *Waterfall*. Dimana pada metodologi *waterfall* ini terdapat beberapa tahapan yaitu:

1) *Requirements Definition* (Definisi Kebutuhan)

Langkah ini merupakan menganalisis terhadap kebutuhan yang diperlukan dalam pembuatan kamus keperawatan berbasis *website* untuk mendapatkan pilihan dan solusi fitur apa yang akan dirancang. Sehingga kebutuhan tersebut yang akan menjadi acuan sistem analisis untuk menterjemahkan kedalam bahasa pemrograman, dimana menetapkan fitur-fitur, kendala dan tujuan sistem. Pada tahap ini juga dilakukan pembuatan *flowchart* atau bisnis proses sistemnya.

2) *System and Software Design* (Desain Sistem dan Software)

Pada tahap ini dilakukan desain aplikasi yang meliputi mendesain interface atau tampilan *website* yang akan dibuat dengan menterjemahkan sesuai dengan kebutuhan pengguna ke dalam sebuah representasi aplikasi yang dapat diperkirakan demi kualitas sebelum dimulai pengkodean. Pada tahap desain ini juga dilakukan pembuatan Diagram Konteks, *Data Flow Diagram* (DFD) dan *Entity Relationship Diagram* (ERD).

3) *Implementation and Unit Testing* (Implementasi dan pengujian unit)

Tahapan inilah merupakan mengerjakan suatu sistem. Dimana desain sistem dan desain interface aplikasi yang dirancang sebelumnya diimplementasikan dengan melakukan pembangunan aplikasi yang diterjemahkan ke kode-kode dalam bahasa pemrograman *HTML*, *PHP*, *CSS*, *Javascript* dan *MySQL* sebagai perangkat lunak pembuatan databasenya. Perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Setelah pengkodean selesai maka akan dilakukan pengujian terhadap sistem yang telah dibuat tadi secara unit. Tujuan pengujian untuk

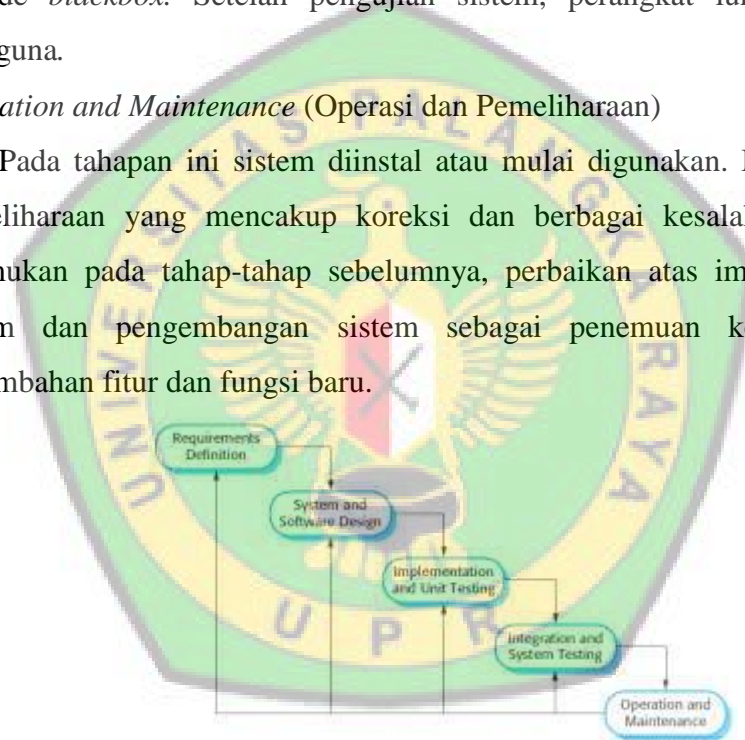
menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

4) *Integration and System Testing* (Integrasi dan Pengujian Sistem)

Dalam tahapan ini, setiap unit program akan diintegrasikan satu sama lain atau menyatukan semua unit program untuk diuji secara keseluruhan untuk mendeteksi apakah ada *bug* atau *error* didalam *website* ini agar terjamin bahwa persyaratan sistem telah dipenuhi atau sudah memenuhi spesifikasi aplikasinya. Metode pengujian sistem yang digunakan pada pembuatan web ini adalah metode *blackbox*. Setelah pengujian sistem, perangkat lunak dikirim ke pengguna.

5) *Operation and Maintenance* (Operasi dan Pemeliharaan)

Pada tahapan ini sistem diinstal atau mulai digunakan. Melakukan juga pemeliharaan yang mencakup koreksi dan berbagai kesalahan yang tidak ditemukan pada tahap-tahap sebelumnya, perbaikan atas implementasi unit sistem dan pengembangan sistem sebagai penemuan kebutuhan baru, penambahan fitur dan fungsi baru.



Gambar 3.1 Waterfall Model (Sommerville, 2011)

3.1.1 *Requirements Definition* (Definisi Kebutuhan)

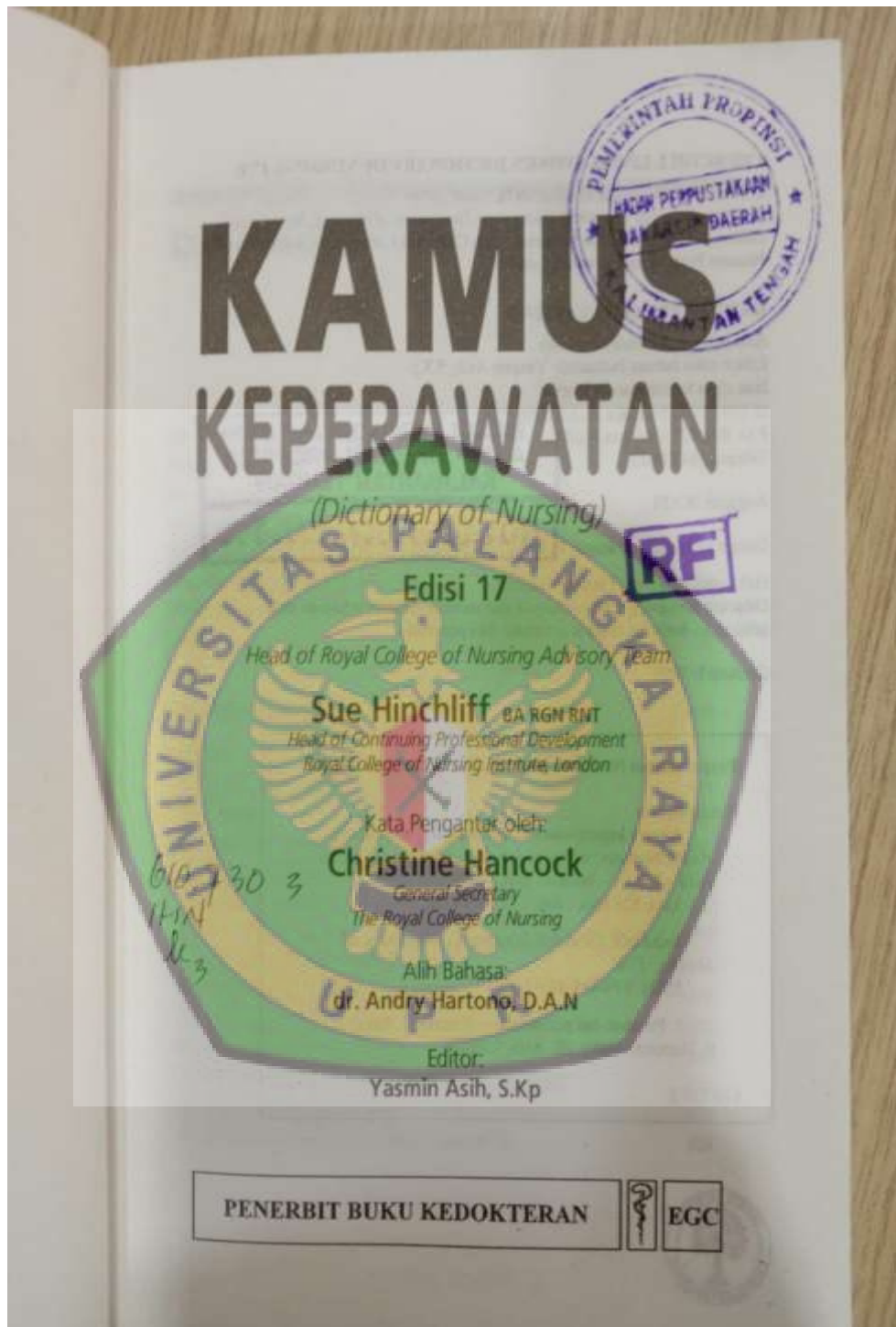
Analisa sistem penggunaan kamus keperawatan adalah untuk mengetahui bagaimana sistem yang sedang berjalan dimasyarakat dalam penggunaan kamus keperawatan dengan tujuan mengidentifikasi masalah yang terjadi sehingga dapat memberikan solusi yang diharapkan. Dari hasil analisa sistem maka dapat dideksripsikan sebagai berikut:

3.1.1.1 Alat dan Bahan

- a. *Hardware*: 1 unit laptop Asus A442U dengan spesifikasi:
1. *CPU* : Intel® Core™ i5 8250U *Processor* (6M Cache, up to 3.40 GHz)
 2. *Operating System* : *Windows 10 Home*
 3. *Memory* : 4GB / 8GB DDR4 2133MHz SDRAM
 4. *Storage* : 1TB SATA HDD 5400RPM
 5. *Display* : 14" HD (1366×768) / FHD (1920×1080)
- b. *Software* yang digunakan adalah sebagai berikut:
1. Sistem Operasi : *Microsoft Windows 10*
 2. *Mozilla Firefox*
 3. *Notepad++*
 4. *SQLyog*
 5. *XAMPP*
 6. *Edraw Max*

3.1.1.2 Metode Pengumpulan Data

Metode pengumpulan data adalah sebuah metode tentang bagaimana dalam mengumpulkan data-data yang ada. Adapun metode pengumpulan yang digunakan adalah studi pustaka dimana pengumpulan data dilakukan dengan cara mencari referensi dan hasil penelitian yang berkaitan dengan permasalahan yang diambil. Pada pembuatan kamus keperawatan berbasis website ini penulis menggunakan referensi kamus keperawatan edisi 17.



Gambar 3.2 Referensi Buku Kamus Keperawatan

Beberapa contoh isi dari kamus keperawatan tersebut:



Gambar 3.3 Contoh Isi Kamus Keperawatan

3.1.1.3 Sasaran Pengguna

Sasaran pengguna dari *website* kamus keperawatan yang telah dirancang adalah untuk semua pengguna agar dapat mempermudah mereka mencari dan memperoleh informasi terkait istilah-istilah keperawatan yang ada didaftar/*list* dari kamus keperawatan berbasis *website* secara cepat tanpa harus membawa banyak buku kamus.

3.1.1.4 Analisis Sistem Lama

Analisa sistem lama pada penggunaan kamus keperawatan di masyarakat, yaitu:

1. Pengguna harus membeli atau meminjam buku kamus keperawatan.
2. Pengguna membuka buku kamus keperawatan dan mencari kata yang diinginkan.
3. Setelah kata yang diinginkan ada maka pengguna tersebut baru mendapatkan arti istilah.



Gambar 3.4 *Flowchart* Sistem Lama

3.1.1.5 Kesimpulan Analisis Sistem Lama

Dari hasil analisis ditemukannya beberapa kelemahan alur penggunaan buku kamus keperawatan yaitu pengguna diharuskan memiliki atau setidaknya meminjam buku kamus keperawatan. Selain itu, pengguna membutuhkan ketelitian dalam mencari kata yang diinginkan. Maka dari itu

direkomendasikan sistem baru yang dapat menjadi media alternatif kamus keperawatan berbasis *website*.

3.1.1.6 Rekomendasi Sistem Baru

Dari kesimpulan diatas maka dibuatkanlah kamus keperawatan berbasis *website*. *Website* ini memiliki beberapa fitur diantaranya sebagai berikut:

a) *Dashboard*

Fitur halaman utama dapat diakses oleh siapa saja tanpa harus *login* terlebih dahulu dan didalamnya terdapat Admin, Kelola kamus, dan pencarian.

b) Kelola Kamus

Fitur ini akan menampilkan daftar istilah kata yang tersedia di kamus keperawatan. Selain itu, *admin* dapat menambah, mengedit dan menghapus data istilah dalam aplikasi kamus keperawatan berbasis *website*.

c) Pencarian

Merupakan fitur yang dapat mempermudah pencarian berdasarkan *keyword* yang di masukkan oleh pengguna.

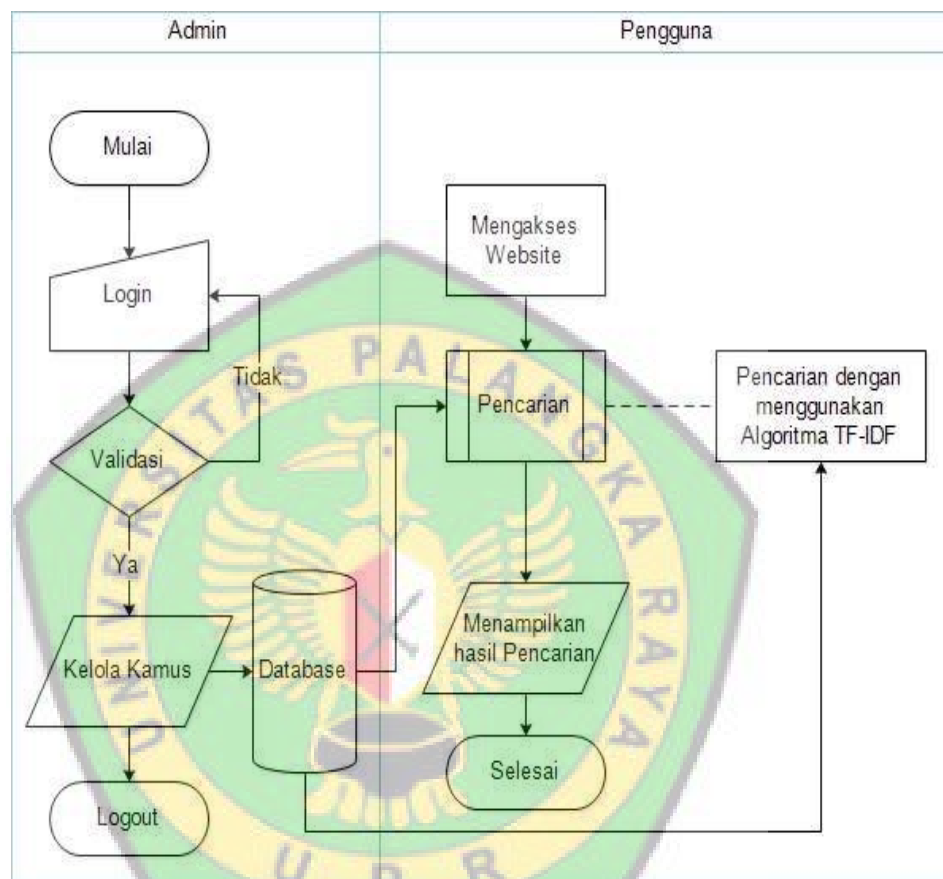
3.1.1.7 Bisnis Proses Sistem Baru

Bisnis Proses Sistem Baru sebagai berikut:

1. *Admin* mengakses *website* dengan *url admin*.
2. *Admin login*.
3. *Website* melakukan validasi akun, jika *error* maka kembali ke langkah kedua, jika tidak maka *admin* dapat melanjutkan ke langkah berikutnya.
4. *Admin* mengelola kamus.
5. *Admin* dapat melakukan pencarian.
6. Pengguna mengakses *website* kamus keperawatan.
7. Pengguna memasukkan kata kunci pencarian pada fitur pencarian.

8. Sistem akan menampilkan istilah yang sesuai dengan kata kunci yang di masukkan.

9. *Admin logout.*



Gambar 3.5 Flowchart Sistem Baru

3.1.1.8 Kesimpulan Bisnis Proses Sistem Baru

1. Membantu pengguna dalam mencari istilah keperawatan tanpa harus membeli atau meminjam buku kamus keperawatan terlebih dahulu.
2. Dengan adanya *Website Kamus Keperawatan* ini dapat membantu pengguna didalam mencari dan memperoleh informasi terkait istilah-istilah keperawatan dengan memanfaatkan teknologi berbasis *website*.

3.1.2 System Dan Software Design (Desain Sistem Dan Software)

Sistem arsitektur yang digunakan dalam proses pembuatan kamus keperawatan berbasis *website* ini adalah dengan sebuah bahasa yang digunakan untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak yang akan dijelaskan sebagai berikut.

3.1.2.1 Diagram Konteks

Pada diagram konteks ini terlihat proses bisnis yang terjadi dan menunjukkan entitas yang akan memberikan dan menerima informasi ke sistem.

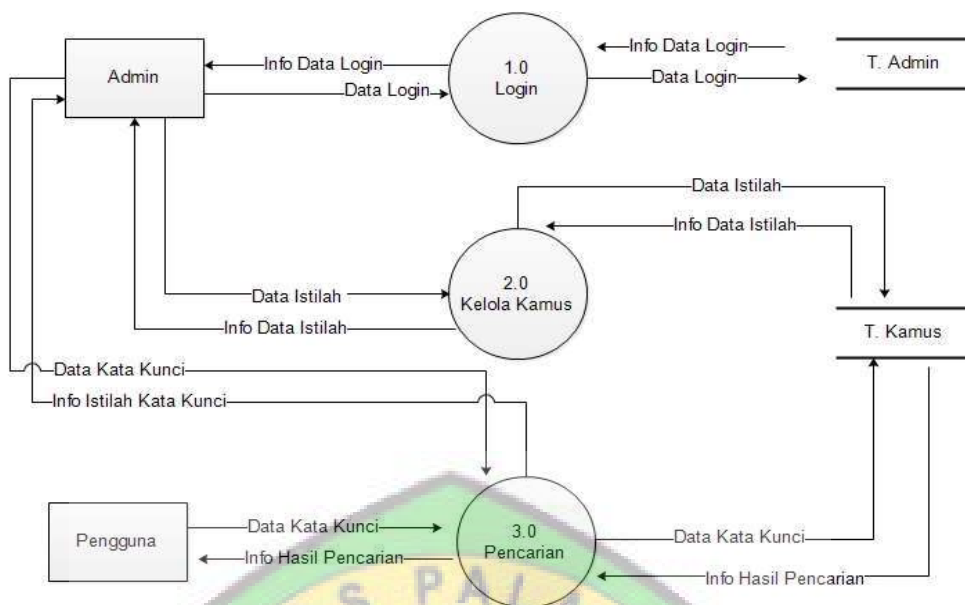


Gambar 3.6 Diagram Konteks

Pada gambar 3.6 terdapat Diagram Konteks, dimana dalam diagram ini terdapat 2 pihak terkait yang digambarkan secara umum. Kesatuan luar yang terlibat secara langsung adalah *admin* dan pengguna. *Admin* berfungsi sebagai pengelola *website* kamus keperawatan, sedangkan pengguna sebagai pihak yang dapat mengakses dan menggunakan fitur pencarian yang terdapat pada *website*.

3.1.2.2 DFD level 1

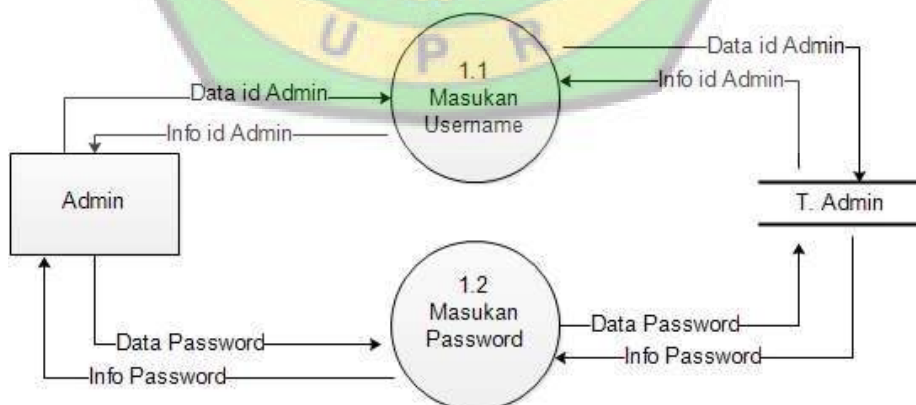
Pada *DFD* level 1 merupakan penjabaran dari diagram konteks level 0, sekaligus menunjukkan bagaimana informasi berpindah dari satu proses ke proses yang lainnya. Pada *DFD* level 1 untuk *website* kamus keperawatan ini terdapat 3 proses, yaitu sebagai berikut:



Gambar 3.7 DFD level 1

Gambar 3.7 merupakan aliran proses *admin* dan pengguna. Sebelum mengelola sistem, *admin* akan melakukan *login* terlebih dahulu. Setelah itu, baru *admin* dapat melakukan proses kelola. Admin melakukan kelola kamus dan dapat melakukan pencarian. Sedangkan untuk pengguna hanya dapat melakukan pencarian.

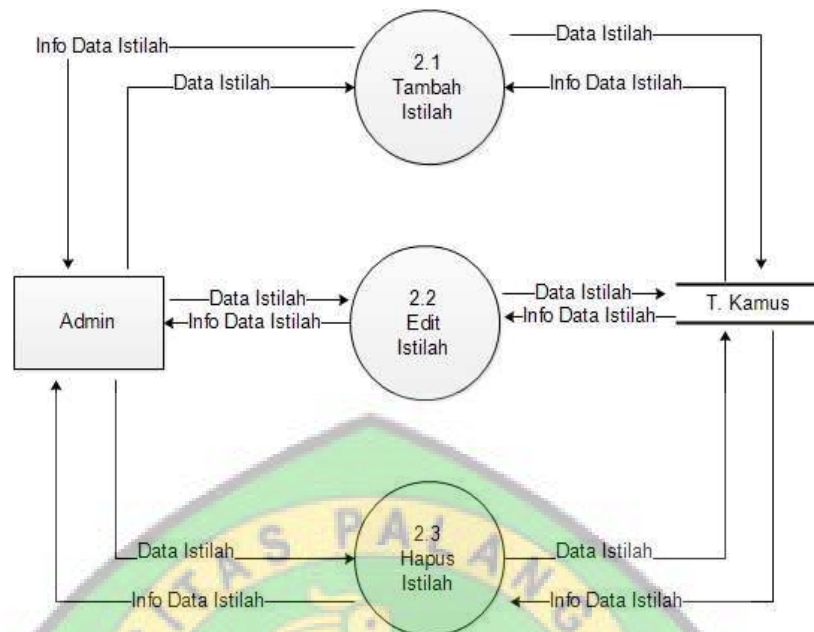
3.1.2.3 DFD Level 2 Proses 1 (Proses Login)



Gambar 3.8 DFD level 2 proses 1

Pada gambar 3.8 diatas menggambarkan tentang proses masuk kedalam *website* oleh *admin* dengan memasukkan *username* dan *password*. *Username* yang digunakan oleh *admin* adalah id admin.

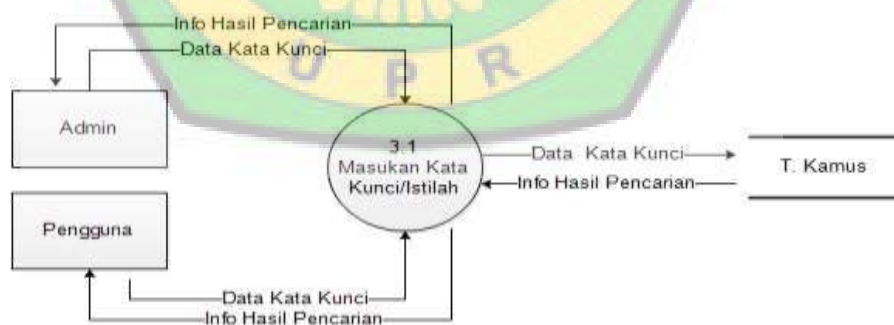
3.1.2.4 DFD Level 2 Proses 2 (Proses Kelola Kamus)



Gambar 3.9 DFD level 2 proses 2

Pada gambar 3.9 diatas dapat menjelaskan bahwa alur kelola kamus yaitu tambah istilah, edit istilah dan hapus istilah. Dimana proses kelola kamus ini dilakukan oleh *admin*.

3.1.2.5 DFD Level 2 Proses 3 (Proses Pencarian)

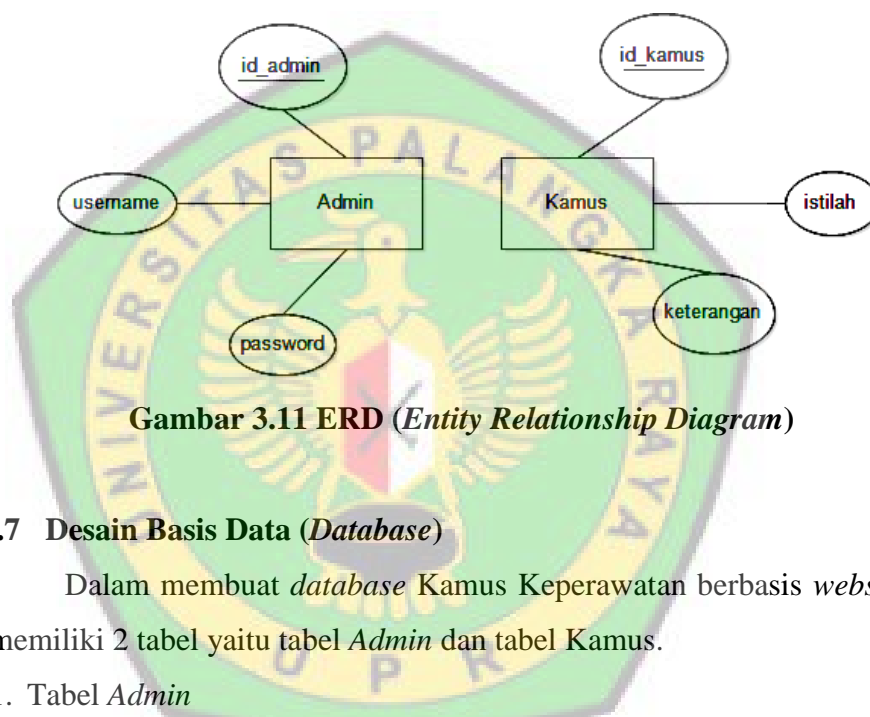


Gambar 3.10 DFD level 2 proses 3

Pada gambar 3.10 diatas dijelaskan proses pencarian yang dilakukan oleh *admin* dan pengguna yaitu untuk menemukan arti istilah dari kata kunci yang di masukan pada fitur pencarian.

3.1.2.6 ERD (*Entity Relationship Diagram*)

Entity-Relationship Diagram (ERD) adalah suatu pemodelan dari basis data relasional yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut *entity* dan hubungan yang dimilikinya disebut *relationship*. Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan *entity* lainnya. Berikut gambar *ERD* dari sistem ini:



Gambar 3.11 ERD (*Entity Relationship Diagram*)

3.1.2.7 Desain Basis Data (*Database*)

Dalam membuat *database* Kamus Keperawatan berbasis *website* ini memiliki 2 tabel yaitu tabel *Admin* dan tabel *Kamus*.

1. Tabel *Admin*

Tabel 3.1 Tabel *Admin*

Field	Panjang Field	Tipe Field
id_admin (PK)	11	<i>Int</i>
username	50	<i>varchar</i>
password	100	<i>varchar</i>

Pada tabel 3.1 merupakan tabel *Admin*, *id_admin* (*Primary key*) bertipe *int* dengan panjang data 11, *username* bertipe data *varchar* dengan panjang data 50, dan *password* bertipe data *varchar* dengan panjang data 100.

2. Tabel Kamus

Tabel 3.2 Tabel Kamus

Field	Panjang Field	Tipe Field
id_kamus (PK)	11	<i>Int</i>
istilah	50	<i>Varchar</i>
Keterangan		<i>Text</i>

Pada tabel 3.2 merupakan tabel Kamus, id_kamus (*Primary key*) bertipe *int* dengan panjang data 11, istilah bertipe data *varchar* dengan panjang data 50, dan keterangan bertipe data *text*.

3.1.2.8 Desain Struktur Navigasi

Perancangan *user interface* diperlukan pada aplikasi *website* ini dengan tujuan untuk mempermudah pengguna dalam mengoperasikannya. Dengan adanya *user interface*, baik pengguna awam, maupun yang sudah berpengalaman dapat mengoperasikan aplikasi *website* ini tanpa adanya kesulitan.

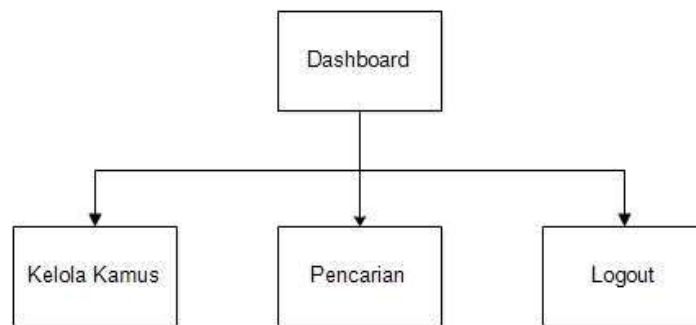
Dalam merancang *sitemap* (peta situs), seperti kerangka aplikasi *website* yang menggambarkan denah dari aplikasi *website*, memuat informasi mengenai setiap halaman yang ada di dalam aplikasi *website*. Peta Situs Kamus Keperawatan berbasis *Website*:

a) Admin:

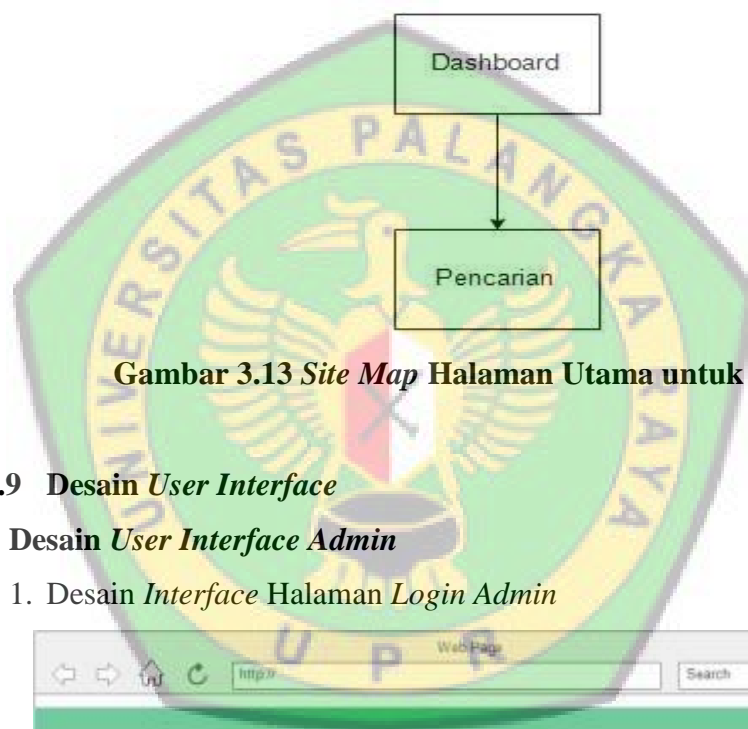
1. *Dashboard*
2. *Kelola Kamus*
3. *Pencarian*
4. *Logout*

b) Pengguna:

1. *Dashboard*
2. *Pencarian*



Gambar 3.12 Site Map Halaman Utama untuk Admin



Gambar 3.13 Site Map Halaman Utama untuk Pengguna

3.1.2.9 Desain User Interface

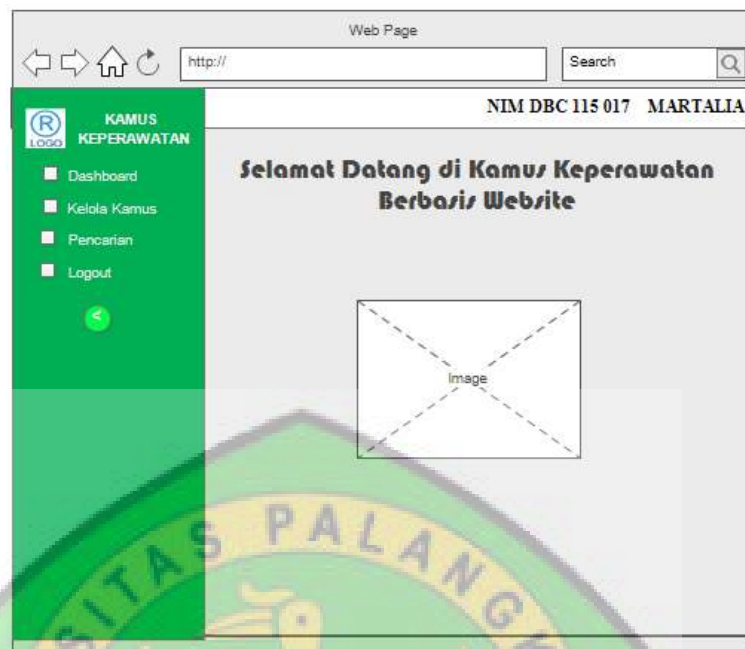
a. Desain User Interface Admin

1. Desain Interface Halaman Login Admin

The screenshot shows a web browser window with a green background. The main content is a white login form titled "Administrator Kamus Keperawatan". At the top of the form is a logo consisting of a registered trademark symbol (®) above the word "LOGO". Below the title, there are two input fields: "Masukkan Username" and "Masukkan Password". At the bottom of the form is a blue button labeled "Login".

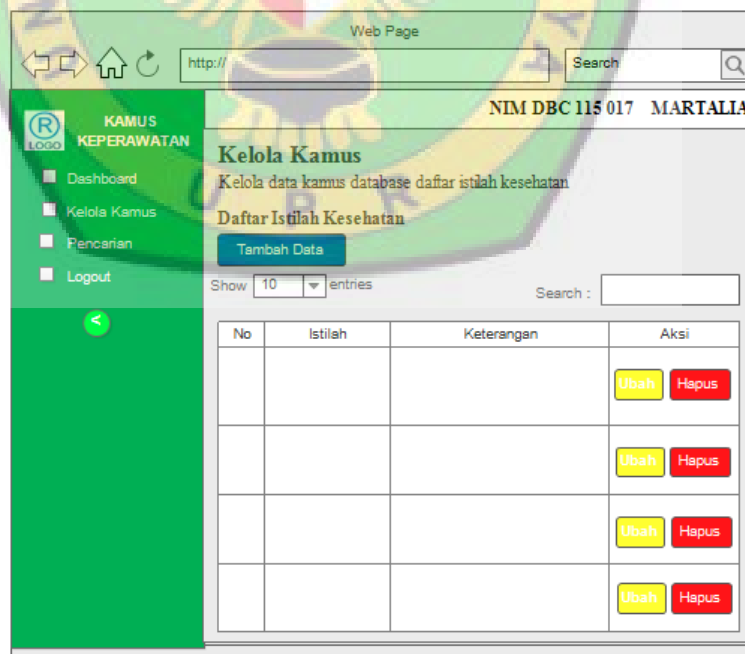
Gambar 3.14 Desain Interface Halaman Login Admin

2. Desain *Interface* Halaman *Dashboard Admin*



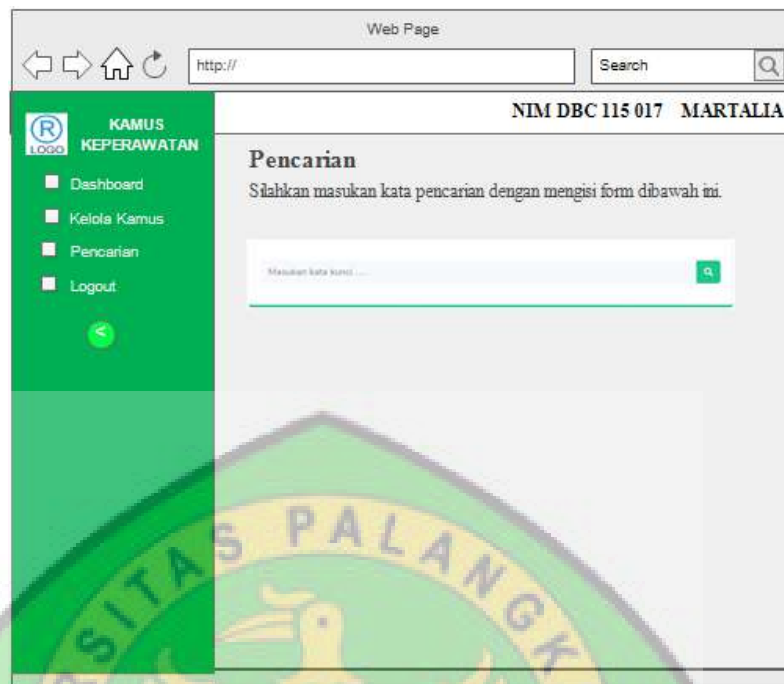
Gambar 3.15 Desain *Interface* Halaman *Dashboard Admin*

3. Desain *Interface* Halaman Kelola Kamus



Gambar 3.16 Desain *Interface* Halaman Kelola Kamus

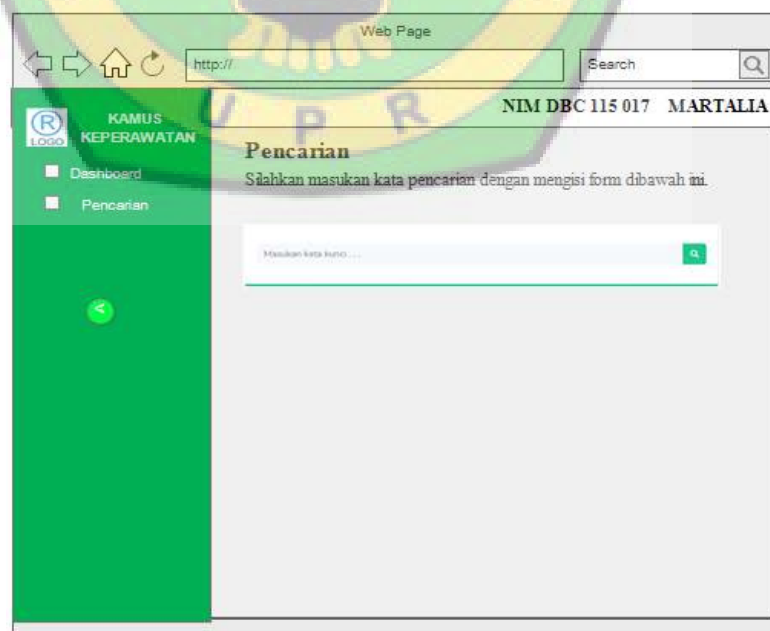
4. Desain *Interface* Halaman Pencarian *Admin*



Gambar 3.17 Desain *Interface* Halaman Pencarian *Admin*

b. Desain *User Interface* Pengunjung

1. Desain *Interface* Halaman Pencarian Pengunjung



Gambar 3.18 Desain *Interface* Halaman Pencarian Pengunjung

3.2 Perhitungan Manual Algoritma *Term Frequency Inverse Document Frequency (TF-IDF)*

a. Menghitung *Term Frequency (tf)*

Term frequency (tf) merupakan frekuensi kemunculan term (Q) pada dokumen (D).

Contoh:

Terdapat kalimat:

Adenosine diphosphate (ADP).

Catatan: tiap kalimat dianggap sebagai dokumen.

1. Tentukan nilai tf !

Jawaban:

Jadi dokumen tiap kalimat ditandai: Adenosine diphosphate (ADP).

Tabel 3.3 Tabel Nilai tf

Q	D1	df	D/df	IDF	IDF+1
adenosine	1	1	1	0	1
diphosphate	1	1	1	0	1
(adp)	1	1	1	0	1

b. Menghitung *document frequency (df)*

Document frequency (df) adalah banyaknya dokumen dimana suatu term (t) muncul.

Contoh: Dari soal yang sama pada menghitung tf , tentukan nilai df !

Jawaban: rumus $df = D1 + D2$ dst

Tabel 3.4 Tabel Nilai df

Q	D1	df	D/df	IDF	IDF+1
adenosine	1	1	1	0	1
diphosphate	1	1	1	0	1
(adp)	1	1	1	0	1

c. Menghitung *document frequency (TF)*

TF logaritmik, hal ini untuk menghindari dominansi dokumen yang mengandung sedikit term dalam *query*, namun mempunyai frekuensi yang tinggi.

Tabel 3.5 Tabel Nilai *D/df*

Q	D1	df	D/df	IDF	IDF+1
adenosine	1	1	1	0	1
diphosphate	1	1	1	0	1
(adp)	1	1	1	0	1



$$1 + \log(1) = 1$$

d. Menghitung *IDF (Inverse Document Frequency)*

Tabel 3.6 Tabel Nilai *IDF*

Q	D1	df	D/df	IDF	IDF+1
adenosine	1	1	1	0	1
diphosphate	1	1	1	0	1
(adp)	1	1	1	0	1



$$\log(1) = 0$$

e. Menghitung *TF-IDF*

Jenis formula *TF* yang biasa digunakan untuk perhitungan adalah *TF* murni (*raw TF*). Dengan demikian rumus umum untuk *Term Weighting TF-IDF* adalah penggabungan dari formula perhitungan *raw TF* dengan formula *IDF* dengan cara mengalikan nilai *TF* dengan nilai *IDF*. Sehingga perhitungannya sebagai berikut :

$$\begin{aligned}
 TF-IDF &= df \times IDF \\
 &= 1 \times 0 \\
 &= 0
 \end{aligned}$$

F. Menghitung $TF-IDF + 1$

Berapapun besarnya nilai tf_{ij} , apabila $D = df_j$, maka akan didapatkan hasil 0 (nol), dikarenakan hasil dari $\log 1$, untuk perhitungan IDF . Untuk itu dapat ditambahkan nilai 1 pada sisi IDF , sehingga perhitungan bobotnya menjadi sebagai berikut :

$$\begin{aligned} TF-IDF + 1 &= (df \times IDF) + 1 \\ &= (1 \times 0) + 1 \\ &= 1 \end{aligned}$$

Tabel 3.7 Lengkap Perhitungan Manual Algoritma

Term Frequency Inverse Document

Q	D1	df	D/df	IDF	IDF+1
adenosine	1	1	1	0	1
diphosphate	1	1	1	0	1
(adp)	1	1	1	0	1
Score Pembobotan					3

BAB V PENUTUP

5.1 Kesimpulan

Rancang Bangun Aplikasi Kamus Keperawatan Menggunakan Algoritma Berbasis *Website* adalah kamus keperawatan ini dibuat dengan menggunakan bahasa pemrograman *PHP* dan *MYSQL*. Metodologi pengembangan menggunakan *waterfall*, dengan tahapan *analysis and design* yaitu tahap analisis kebutuhan *software*, *hardware* dan pengumpulan data yaitu berdasarkan studi pustaka dimana pengumpulan data dilakukan dengan cara mencari referensi dan hasil penelitian yang berkaitan dengan permasalahan yang diambil. Tahapan *Requitments Definition* melalui penggambaran *flowchart*. *System and Software Design* desain digambarkan melalui *DFD* (*Data Flow Diagram*), *ERD* (*Entity Relationship Diagram*), *database*, dan desain *interface*. *Implementation* merupakan tahap untuk menampilkan hasil dari desain yang sudah dibuat. *Testing* merupakan implementasi dari desain dibuat menggunakan *PHPMYAdmin*, *testing* menggunakan *black box*. Berdasarkan dari hasil pengujian *black box* semua fitur dalam sistem telah berjalan dengan baik serta dari hasil perhitungan kuisioner dengan skala likert memperoleh hasil 88% responden setuju jika kamus keperawatan dapat digunakan dengan mudah dan membantu pengguna.

Algoritma *Term Frequency Inverse Document Frequency (Tf-Idf)* telah berhasil diterapkan dalam sistem dengan baik dimana sistem dapat memberikan hasil pembobotan kata dari kata kunci yang diinputkan. Sistem juga dapat menghasilkan output berupa hasil pencarian kamus sesuai dengan kata kunci yang diinputkan, dengan melalui 2 tahapan teks *processing* yaitu *tokenizing* dan *stopword removal* atau *filtering*.

5.2 Saran

Adapun saran untuk kamus keperawatan berbasis *website* ini, yaitu: Kamus keperawatan ini diharapkan dapat menampilkan rekomendasi berdasarkan keterangan istilah yang terkait dengan hasil pencarian.



DAFTAR PUSTAKA

- Chaer, Abdul. 2007. *Linguistik Umum* cetakan ketiga. Jakarta: Rineka Cipta.
- Date, C.J. 2000. *Date On Database*. New York : Apress
- DeMarco, Tom. 1978. *Structured Analysis and System Specifications*. New York : Prentice-Hall
- Edhy, Sutanta. 2004. *Sistem Basis Data*. Yogyakarta : Graha Ilmu
- Feldman, R., & Sanger, J. 2007. *The Text Mining Handbook*. New York: Cambridge University Press.
- Gane, Chris and Sarson, Trish. (1977). *Structured Systems Analysis: Tools and Techniques*. Dallas: McDonnell Douglas Systems Integration Company.
- Harjanto, Dhony Syafe. dkk. 2012. Sistem Temu Kembali Informasi pada Dokumen Teks Menggunakan Metode *Term Frequency Inverse Dokumen Frequency (TF-IDF)*. *Jurnal Sains dan Matematika*. Semarang: Fakultas
- Herwijayanti, Bening. dkk. 2018. Klasifikasi Berita *Online* dengan menggunakan Pembobotan *TF-IDF* dan *Cosine Similarity*. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*. Malang: Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
- Hidayat, A. 2009. *Metode Penelitian Keperawatan dan Teknik. Analisis Data*. Jakarta: Salemba Medika.
- Hinchliff, S. (1999). *Kamus keperawatan edisi 17*. Jakarta : EGC
- Melita, Ria. dkk. 2018. Penerapan Metode *Term Frequency Inverse Document Frequency (Tf-Idf)* Dan *Cosine Similarity* Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web. *Jurnal Jurnal Teknik Informatika*. Jakarta: Program Studi Teknik Informatika, Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah.
- P. P. Chen. 1976. *Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned*. Baton Rouge: Louisiana State University.

Saputra, Renanda Cahaya. dkk. 2012. Rancang Bangun Aplikasi Kamus Percakapan Bahasa Arab Berbasis *Mobile* Menggunakan Teknologi J2ME. Jurnal SI. Surabaya: Jurusan Sistem Informasi Sekolah Tinggi Manajemen Informatika dan Komputer Surabaya.

Setyawan, Aan. 2016. *Pengertian dan jenis-jenis kamus* (Online: <https://belajarbahasa.id/artikel/dokumen/>, diakses 17 Juni 2019)

Sommerville, Ian. 2011. *Software Engineering 9ed*. New York : Pearson

Taslim, Ahmad. 2013. *Blackbox* (Online : <http://www.academia.edu/5574402/Blackbox>, diakses 27 Mei 2019)

Wahyu, Suzana, Ernawati Waridah. 2013. Kamus Bahasa Indonesia Untuk Pelajar, Mahasiswa, & Umum, Jakarta: Kawah Media.

Yatini B, Indra. 2010. Pemrograman Terstruktur. Informatika, Bandung.

Yourdon, Edward. 1989. Modern Structured Analysis. Prentice-Hall International Inc., Englewood Cliffs: New Jersey

Yuhefizar. 2008. 10 jam Mengenal Internet Teknologi dan Aplikasinya. Penerbit PT Elex Media Komputindo: Jakarta.

