



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### TINJAUAN PUSTAKA

Penelitian ini bertujuan untuk membuat sebuah *knowledge management system* dengan menggunakan *Content Management System* (CMS) berupa MediaWiki yang diharapkan dapat membantu dalam pengelolaan dan penyebaran informasi ataupun *knowledge* terkait dengan Universitas Multimedia Nusantara. Pada bab ini akan menjelaskan *knowledge management* menurut berbagai pengertian, *journal*, buku dan sumber lainnya serta beberapa penelitian terdahulu.

#### 2.1. Platform MediaWiki

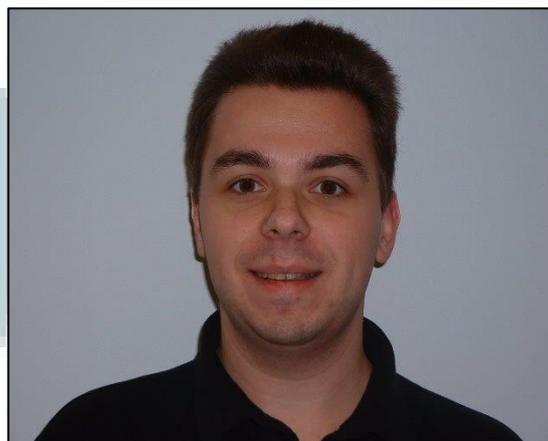
Menurut situs ([mediawiki.org](http://mediawiki.org), 2017), platform MediaWiki merupakan sebuah *content management system* (CMS) untuk pembuatan *knowledge management system* (KMS) dengan perangkat lunak berbasis server gratis, berlisensi di bawah GNU *General Public License* (GPL) serta perangkat lunak yang kuat dan terukur dan implementasi wiki kaya fitur yang menggunakan PHP untuk memproses dan menampilkan data yang tersimpan dalam database, seperti MySQL. Logo yang ada pada gambar 2.1 menyimbolkan bagaimana bahasa markahnya menggunakan kurung siku ganda ("[[", "]]") untuk menggabungkan artikel-artikel. MediaWiki mampu untuk bekerjasama dengan perangkat lunak lainnya untuk meningkatkan unjuk kerja dan kemampuannya antara lain dengan *cached*, sebuah sistem *squid cache* dan TeX sebuah *math rendering*. Dengan menggunakan extension dan hook system, pengguna dapat menambahkan fitur-fitur yang mereka buat untuk digunakan di MediaWiki. Halaman menggunakan

format *wikitext* MediaWiki, sehingga akan lebih mudah untuk mengedit konten-konten yang ada. Dibawah ini gambar 2.1 merupakan logo dari mediawiki.



**Gambar 2.1 Logo MediaWiki**

Sejarah MediaWiki ditulis untuk Wikipedia pertama kali oleh seorang mahasiswa bernama Magnus Manske (gambar 2.2) yang berkebangsaan Jerman dengan *portrait* gambar 2.2. Perangkat lunak ini pertama kali menggunakan UseModWiki berbasis Perl (dinamakan "Fase I" oleh komunitas Wikipedia), dan kemudian digantikan dengan PHP ("Fase II") pada 25 Januari 2002.



**Gambar 2.2 Magnus Manske**

Pada pertengahan 2002, perangkat lunak ini dikembangkan lebih lanjut menjadi "Fase III". Sejak pertengahan 2003, perangkat lunak yang tadinya tidak bernama kemudian dinamakan "MediaWiki" dan sejak itu versi-versi terbaru dirilis secara teratur. Sudah banyak situs web Indonesia yang berbasis MediaWiki contohnya yaitu Crayonpedia, MyQWiki dan Wiki Linux Indonesia.

## 2.2. *Wikitext*

Menurut situs ([mediawiki.org](http://mediawiki.org), 2017) halaman penulisan di MediaWiki menggunakan format *wikitext* dimana digunakan karena lebih memudahkan dalam penulisan serta pengeditan konten yang ada. *Wikitext* merupakan sebuah *script* sederhana yang diterjemahkan kedalam bahasa HTML web oleh perangkat lunak sebelum mengirimkan ke setiap *browser*. Didalam *wikitext* pengguna dapat mengkolaborasikan dengan bahasa HTML serta bisa menambahkan format CSS didalamnya serta dapat juga menambahkan gambar, tabel dan struktur lainnya. Selain itu, ada *template* yang tersedia yang lebih memudahkan dalam mengerjakan suatu proyek.

### 2.2.1. Metode Pengeditan halaman Wiki

Metode umum dalam mengedit menggunakan *wikitext* cukup sederhana bagi pengguna yang ingin dan tertarik untuk mengedit suatu artikel ataupun halaman tertentu, yakni dengan melakukan *sign-in user* secara gratis kemudian pilih bagian "*edit halaman*" lalu *user* dilanjutkan ke jendela *edit* yang menyerupai *word processor*. Didalamnya *user* dapat mengubah kata-kata, menambahkan gambar atau tabel serta memperbaiki

tata letak dihalaman tersebut. Selain itu, Metode pembuatan halaman baru juga sederhana, ada ringkasan metode yang digunakan untuk memulai halaman baru di bagian “*Help Pages*”. Serta pengguna dapat melihat hasil edit dan memeriksanya pada bagian “*Show Preview*” dan jika halaman tersebut telah selesai maka lanjutkan dengan *save* halaman.

### 2.2.2. Format Penulisan

Menurut situs (*community.wikia.com* dan *meta.wikimedia.org*, 2017), Didalam *wikitext* pengguna bisa melakukan penambahan foto, tabel, *style* dalam penulisan, *link* serta berbagai perubahan dalam penampilannya.

Berikut tabel 2.1 yang merupakan format *text* yang digunakan untuk melakukan *style* dalam penulisan

**Tabel 2.1 Format Wikitext**

Deskripsi	Input	Output
Italic teks	“ <i>italic</i> ”	<i>Italic</i>
Bold teks	“ <b>bold</b> ”	<b>Bold</b>
Bold dan italic teks	“ <b><i>bold &amp; italic</i></b> ”	<b><i>Bold &amp; italic</i></b>
<i>Internal link</i>	[[Help:Contents]] [[Help:Contents[Displayed text]]]	<a href="#">Help:Contents</a> <a href="#">Displayed Text</a>
<i>Redirect ke halaman lain</i>	#REDIRECT [[Help:Contents]]	 <a href="#">Help:Contents</a>
<i>External Link</i>	<a href="http://www.example.org">http://www.example.org</a>	<a href="http://www.example.org">http://www.example.org</a>
<i>Headings</i>	==Level 2== ===Level 3=== ====Level 4==== =====Level 5===== =====Level 6=====	Level 2 Level 3 Level 4 Level 5 Level 6
<i>Bulleted List</i>	*One *Two	<ul style="list-style-type: none"> <li>• One</li> <li>• Two</li> </ul>
<i>Number List</i>	#One #Two	<ol style="list-style-type: none"> <li>1. One</li> <li>2. Two</li> </ol>

Selain itu juga ada format untuk melakukan *edit* pada foto. Pada tabel 2.2 ini adalah teks format untuk melakukan perubahan pada format gambar.

**Tabel 2.2 Format *Input Image***

Deskripsi	<i>Input</i>
<i>Image</i>	[[File:Mediawiki.png]]
<i>Size Image</i>	[[File:Mediawiki.png 100px]]
<i>Thumbnail image</i>	[[File:Mediawiki.png thumb Caption Text]]
<i>Aligned Image</i>	[[File:Mediawiki.png thumb 100px left]] [[File:Mediawiki.png thumb 100px center]] [[File:Mediawiki.png thumb 100px right]]
<i>Linked Image</i>	[[File:Mediawiki.png link=Help:Contents]]

Berikut tabel 2.3 yaitu format untuk membuat sebuah tabel menggunakan *wikitext*

**Tabel 2.3 Format Tabel *Wikitext***

Deskripsi	<i>Input</i>
Mulai untuk membuat tabel	{
Mulai untuk baris tabel	-
<i>Table Caption</i>	+
<i>Table Header</i>	! atau !!
<i>Table Cell</i>	atau
Menutup sebuah tabel	}

U  
M  
M  
N

### 2.3. 7 Layer Architecture Knowledge Management System

Dalam membuat sebuah *Knowledge Management System* (KMS), maka dibutuhkan rancangan arsitektur dari KMS tersebut. Pada pembuatan *knowledge management system* berbasis wiki di Universitas Multimedia Nusantara menggunakan *blueprints* berupa *7 layer architecture knowledge management system*, Menurut (Suprianto, 2014) berikut adalah definisi dari ke-7 layer arsitektur yang digunakan:

#### 1 *Interface Layer*

*Interface Layer* merupakan lapisan paling atas dari *knowledge management system* (KMS), dimana pada lapisan ini pengguna akan berinteraksi dengan KMS. Maka pada tampilan KMS haruslah *user friendly* dan menarik, agar menarik minat pengguna untuk menggunakan KMS.

#### 2 *Access and Authentication Layer*

*Access and Authentication Layer* merupakan layer yang mengatur keamanan sistem KMS dan pengaturan hak akses kedalam sistem KMS.

#### 3 *Collaborative Intelligence and Filtering Layer*

*Collaborative Intelligence and Filtering Layer* merupakan layer dimana proses penyimpanan, penamaan, penambahan metadata, dan penyaringan dilakukan. Serta pada layer ini terdapat fungsi-fungsi yang sangat membantu pengguna dalam menggunakan KMS, seperti proses pencarian dan rekomendasi. Sehingga memudahkan pengguna untuk dapat mencari dan mendapatkan informasi dengan mudah.

#### 4 *Application Layer*

*Application Layer* merupakan layer yang mengintegrasikan aplikasi-aplikasi yang digunakan KMS. Aplikasi yang umum digunakan di KMS berupa forum diskusi antar *user*, *chatting*, *email*, manajemen dokumen dan sebagainya.

#### 5 *Transport Layer*

*Transport Layer* merupakan tempat dimana proses transfer atau pengiriman data terjadi antara satu pengguna ke pengguna lainnya. Biasanya pada layer ini menggunakan protokoler TCP/IP.

#### 6 *Middleware and Layer Integration Layer*

*Middleware and Layer Integration Layer* digunakan untuk menyatukan atau mengintegrasikan berbagai jaringan di organisasi yang berbeda *platform*. Dengan tujuan agar seluruh jaringan yang ada dapat menggunakan sistem secara bersamaan.

#### 7 *Repositories Layer*

*Repositories Layer* merupakan inti dari KMS ini, karena pada layer ini digunakan sebagai tempat penyimpanan data-data dari *knowledge management system*.

### 2.4. *Data Flow Diagram*

Menurut buku yang berjudul “*System Analysis & Design 5th Edition*” (Dennis , Wixom, & Roth, 2012) mengatakan bahwa pengertian dari *data flow diagram* adalah teknik yang menggambarkan sebuah bisnis proses dan data yang

lewat terjadi di sebuah sistem. Didalam sebuah *data flow diagram* terdapat beberapa elemen simbol didalamnya, teknik *modeling* simbol menggunakan gaya dari Gane dan Sarson sebagai berikut:

1. *Process*

Suatu Proses adalah suatu aktifitas atau fungsi yang dilakukan untuk beberapa spesifik alasan bisnis. Proses bisa secara manual atau terkomputerisasi. Setiap proses seharusnya diberi nama dimulai dengan kata kerja dan diakhiri dengan kata benda. Nama harus disingkat, namun harus mengandung informasi yang cukup sehingga pembaca dapat dengan mudah memahami apa yang dilakukan. Secara umum, setiap proses hanya melakukan satu aktifitas. Setiap proses harus memiliki setidaknya satu aliran data masukan dan setidaknya satu data keluar.

2. *Data Flow*

Aliran data adalah satu bagian data atau koleksi logis dari beberapa bagian informasi. Setiap aliran data harus diberi nama dengan kata benda. Deskripsi dari aliran data mencantumkan secara persis elemen data apa yang ada didalam aliran.

3. *Data Store*

*Data store* adalah kumpulan data yang disimpan dalam beberapa cara (yang mana ditentukan kemudian ketika membuat model fisik). Setiap penyimpanan data diberi data dengan kata benda dan diberi nomor identifikasi dan deskripsi.

#### 4. *External Entity*

*External Entity* adalah orang, organisasi, unit organisasi atau sistem yang berada diluar sistem, tetap berinteraksi dengannya (misalnya, pelanggan, organisasi pemerintah, sistem akuntansi). *External entity* biasanya sesuai dengan aktor utama yang diidentifikasi dalam *usecase*. *External entity* menyediakan data ke sistem atau menerima data dari sistem dan berfungsi untuk menetapkan batasan sistem. Setiap entitas eksternal memiliki nama dan deskripsi. Orang yang menggunakan informasi dari sistem untuk melakukan proses lain atau siapa yang memutuskan informasi apa yang masuk ke dalam sistem didokumentasikan sebagai *External entity* (misalkan, manajer, staf)

Didalam *data flow diagram* terdiri dari beberapa level diagram didalamnya, berikut adalah level diagram beserta penjelasan menurut Allan Dennis:

##### 1. *Context Diagram*

Diagram pertama dalam setiap proses bisnis adalah diagram konteks.

Diagram konteks menunjukkan keseluruhan sistem dalam konteks dengan lingkungannya. Semua proses model pasti memiliki satu diagram konteks.

##### 2. *Level 0 Diagram*

Diagram berikutnya disebut dengan diagram level 0, diagram tingkat 0 ini menunjukkan semua proses pada tingkat pertama (yaitu, proses nomor 1 hingga 3), penyimpanan data, *external entity*, dan *data flow* ada didalam diagram tersebut. Semua model proses hanya memiliki satu diagram level 0.

### 3. *Level 1 Diagram*

Diagram level 0 hanya menunjukkan bagaimana proses tingkat utama dalam sistem berinteraksi. Setiap proses diagram 0 dapat didekomposisi menjadi DFD yang lebih eksplisit, yang disebut dengan diagram level 1. Diagram level 1 menunjukkan cara kerja dalam sistem secara lebih detail.

### 4. *Level 2 Diagram*

Diagram level 2 menunjukkan proses yang lebih mendetail pada diagram level 1.

### 5. *Alternative Data Flows*

Misalkan dalam suatu proses menghasilkan dua arus data yang berbeda dalam situasi yang berbeda. Bagaimana kita menunjukkan jalur alternatif ini di DFD? Jawabannya adalah dengan menunjukkan aliran data dan menggunakan deskripsi proses untuk menjelaskan bahwa mereka adalah alternatif.

## **2.5. *Entity Relationship Diagram***

Menurut Allan Dennis (2012), *Entity Relationship Diagram* merupakan gambar yang menunjukkan informasi itu dibuat, disimpan, dan digunakan oleh sistem bisnis. Seorang analis dapat membaca ERD untuk menemukan setiap bagian informasi dalam suatu sistem dan bagaimana mereka diorganisasikan dan terkait satu sama lain. Pada ERD, jenis informasi serupa terdaftar bersama-sama dan ditempatkan di dalam kotak yang disebut entitas. Garis ditarik antara entitas ke mewakili hubungan antar data, dan simbol khusus ditambahkan ke diagram

untuk mengomunikasikan aturan bisnis tingkat tinggi yang perlu didukung oleh sistem. ERD menyiratkan tidak ada perintah, meskipun entitas yang terkait satu sama lain biasanya ditempatkan berdekatan.

Dalam diagram ERD ada tiga elemen dasar dalam bahasa pemodelan data (entitas, atribut, dan hubungan), yang masing-masing diwakili oleh simbol grafik yang berbeda. Ada banyak set simbol yang berbeda yang dapat digunakan pada ERD. Tidak ada yang mengatur simbol mendominasi penggunaan industri, dan tidak ada yang lebih baik dari yang lain. Berikut adalah penjelasan masing-masing elemen yang ada.

a. *Entity*

Entitas adalah bagian dasar dari model data. Dalam entitas bisa berupa seseorang, tempat, acara atau hal tentang data mana yang dikumpulkan (misalnya, karyawan, produk, atau pesanan). Sebuah entitas digambarkan oleh sebuah persegi panjang, dan digambarkan oleh dengan huruf kapital. Semua entitas punya nama, deskripsi singkat menjelaskan apa itu.

b. *Attribute*

Atribut adalah beberapa jenis informasi yang ditangkap tentang suatu entitas. Misalnya, nama belakang, alamat rumah, dan alamat e-mail adalah semua atribut pelanggan. Mudah untuk datang dengan ratusan atribut untuk suatu entitas (mis. Pelanggan memiliki warna mata, hobi favorit), tetapi hanya mereka yang benar-benar akan digunakan oleh proses bisnis harus dimasukkan dalam model

c. *Relationship*

Hubungan adalah asosiasi antar entitas, dan mereka ditampilkan oleh garis yang menghubungkan entitas bersama. Setiap hubungan memiliki entitas induk dan entitas anak, orang tua menjadi entitas pertama dalam hubungan, dan anak menjadi kedua.

## 2.6. Metodologi

*System Developing Life Cycle (SDLC)* merupakan tahapan-tahapan atau pola pengerjaan yang dilakukan untuk membangun sebuah sistem informasi ataupun perangkat lunak. SDLC terdiri dari tahap-tahap : rencana (*planning*), analisa (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*). Didalam SDLC memiliki beberapa model-model yang digunakan untuk tahap pengembangan dan membangun sistem salah satunya adalah model Rapid Application Development (RAD). Rapid Application Development Merupakan model pengembangan system yang melakukan beberapa penyesuaian terhadap SDLC pada beberapa bagian sehingga lebih cepat untuk sampai ke tangan pengguna system. metodologi ini biasanya mensyaratkan beberapa teknik dan alat-alat khusus agar proses bisa cepat. RAD ini menekankan pada siklus pembangunan sistem yang pendek, singkat, dan tepat. RAD ini menggunakan metode iteratif (berulang) dalam mengembangkan sistem dimana working model (model kerja) sistem dikerjakan diawal tahap pengembangan dengan tujuan menetapkan kebutuhan (*requirement*) pengguna. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus

hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi. RAD sama-sama berusaha memenuhi syarat-syarat bisnis yang berubah secara cepat dengan mempersingkat waktu dalam pengembangan sistem dan meningkatkan kualitas sistem.

Menurut jurnal yang ditulis oleh Ade Setiawan, dkk yang berjudul “Rapid Application Development” RAD memiliki banyak unsur-unsur yang membuat sebuah metodologi yang unik, diantaranya adalah *prototyping, iterative development, time boxing, team members, management approach*, dan *RAD tools* (Setiawan, 2011). Selain itu RAD juga memiliki berbagai keuntungan, diantara lain :

1. Sangat berguna dilakukan pada kondisi user tidak memahami kebutuhan apa saja yang digunakan pada proses pengembangan perangkat lunak.
2. RAD mengikuti tahap pengembangan system seperti umumnya, tetapi mempunyai kemampuan untuk menggunakan kembali komponen yang ada sehingga pengembangan tidak perlu membuat dari awal dan waktu akan lebih singkat. Selain itu penggunaan biaya akan lebih rendah.

## **2.7. Journal dan Penelitian Terdahulu**

Berikut tabel 2.4 adalah Jurnal – jurnal dan penelitian yang menjadikan referensi dalam pembuatan *knowledge management system* dan laporan.

**Tabel 2.4 Penelitian Terdahulu**

No	Judul Penelitian	Penulis	Nama Jurnal	Deskripsi
1	<p><b>Penerapan Knowledge Management System (KMS) Berbasis Web Studi Kasus bagian Teknisi dan Jaringan Fakultas Ilmu Komputer Universitas Sriwijaya</b></p>	<p><b>Winda Kurnia Sari, Ken Ditha Tania</b></p>	<p><i>Jurnal Sistem Informasi (JSI), VOL. 6, NO. 2, Oktober 2014,</i></p>	<p>Penelitian ini bertujuan untuk membangun sebuah <i>knowledge management system</i> di perguruan tinggi di Universitas Sriwijaya dan masih dalam jangkauan fakultas ilmu komputer. <i>Knowledge management system</i> yang dibangun berbasis web dengan orientasi <i>clients/server</i> yang dibangun menggunakan metode <i>10-step knowledge management roadma</i>. <i>Knowledge Management System</i> ini berhasil dibangun sebagai penerapan KMS untuk komponen teknologi dan berhasil memfasilitasi proses pengelolaan pengetahuan dan membutuhkan tim <i>Knowledge Management</i> untuk sebagai admin untuk melakukan pengupdatetan data dan maintenance. Dari Jurnal ini didapat bahwa KMS yang dibangun memfasilitasi proses pengelolaan dan dibutuhkannya tim untuk melakukan update informasi dan maintenance data.</p>
2	<p><b>Prototype Knowledge Management System</b></p>	<p><b>Salman El Farisi dan Dana</b></p>	<p><i>Journal of Information Systems, Volume 8,</i></p>	<p>Dalam penelitian ini meneliti bahwa proses belajar yang dilakukan industri perangkat lunak</p>

<p><b>Berbasis Wiki Untuk Proyek Rekayasa Perangkat Lunak</b></p>	<p><b>Indra Sensuse</b></p>	<p><i>Issue 1, April 2012</i></p>	<p>di Indonesia masih sangat lambat. Maka dibutuhkan adanya proses <i>knowledge management</i> yang baik agar <i>knowledge</i> dan pengalaman yang dimiliki oleh perusahaan dapat dikelola dengan baik dan dimanfaatkan untuk meningkatkan inovasi. Di penelitian ini penulis membuat sebuah prototipe <i>knowledge management system</i> yang berbasis wiki untuk memenuhi kebutuhan <i>knowledge management</i>. Dan hasilnya penulis berhasil membuat prototipe <i>knowledge management</i> yang digunakan untuk menyimpan seluruh <i>knowledge</i> tentang penyelesaian masalah yang muncul.</p> <p>Dari Jurnal ini didapat bahwa <i>knowledge</i> dan pengalaman yang dimiliki oleh sebuah institusi harus dapat dikelola dengan baik dan dimanfaatkan untuk di update yang diharapkan dapat meningkatkan inovasi. Jadi dengan adanya KM bisa membuat sebuah institusi bisa menyelesaikan masalah yang muncul sebab masih adanya <i>knowledge</i> yang dikelola dalam menghadapi masalah yang ada.</p>
---	-----------------------------	-----------------------------------	---

3	<p><i>Design and Implement a knowledge management system to support web-based learning in higher education</i></p>	<p><b>Jinyue Peng, Dongxing Jiang, Xinyu Zhang</b></p>	<p><i>Procedia Computer Science 22 ( 2013 ) 95 – 103</i></p>	<p>Penelitian ini Menerapkan sistem manajemen pengetahuan (KMS-THU) untuk mendukung layanan pengetahuan untuk Tsinghua Web Sekolah (THU-WS), yang merupakan platform pembelajaran berbasis web Tsinghua. KMS-THU berfokus pada pengetahuan Manajemen oleh orang-orang dan juga abstrak pengetahuan khusus untuk kursus. Dengan adanya layanan <i>cloud campus</i> dan beragam <i>mobile client</i>, jadi bisa belajar di mana-mana dan untuk mengoptimalkan pengalaman belajar bagi siswanya. Selain menggambarkan desain layanan pengetahuan dan kerangka kerja KMS untuk pembelajaran berbasis web, paper ini Sebagainya rincian teknis untuk implementasi KMS. Dari jurnal ini didapat bahwa penyimpanan data harus selalu online agar data bisa di sinkronisasikan dan dipastikan selalu <i>update</i>. Selain itu user juga diberikan level hak akses login untuk melakukan pembaharuan data agar bisa di kontrol bahwa siapa yang melakukan pengupdatean data pada KMS.</p>
---	--	--	--	---