

APLIKASI PEMBELAJARAN *ENGLISH 16 TENSES*

Tri Septi PurnamaSari¹, Wan Yuliyanti²

Jurusan Teknik Informatika, Politeknik Negeri Tanah Laut
Jl. A Yani Km 6 Pelaihari Tanah Laut Kalimantan Selatan
Telp. (0512) 21537, Faks. (0512) 21537

E-mail: septyteknik@gmail.com¹, wanyuliyanti@gmail.com²

ABSTRAKS

Aplikasi ini merupakan suatu media pembelajaran sederhana bagi para pemula dalam menulis, baik penulisan buku, cerita pendek, pidato, artikel ataupun yang lainnya. Tujuan dibangunnya aplikasi ini adalah untuk membantu para pembelajar maupun penulis, khususnya dalam writing skill. Writing yang baik adalah penulisan yang sesuai dengan kaidah structurally dan grammatically. Metode pengembangan sistem yang digunakan dalam aplikasi ini adalah dengan menggunakan metode OOAD (object oriented analysis design). Aplikasi ini merupakan suatu media pembelajaran Bahasa Inggris yang menggunakan media internet yang berbasis web dan merupakan solusi untuk menerapkan sistem pembelajaran yang efektif dan memungkinkan tersampainya bahan pembelajaran secara fleksibel (kapanpun dan dimanapun berada) kepada para penggunanya.

Kata Kunci: aplikasi, writing skill, tenses

1. PENDAHULUAN

1.1 Latar Belakang

Bahasa merupakan alat komunikasi yang paling penting, dan suka atau tidak suka, saat ini Bahasa Inggris sudah sangat mendominasi semua aspek dalam hal komunikasi. Setiap orang bisa melihat hampir semua *electronic devices* menggunakan Bahasa Inggris di dunia *modern* saat ini yang penuh dengan tantangan dan persaingan yang ketat, sehingga setiap orang disarankan tidak hanya memiliki tingkat pendidikan yang tinggi, namun juga dituntut keterampilan khusus yang lazim disebut *skill*. Salah satu *skill* yang sangat dibutuhkan saat ini adalah Bahasa Inggris.

Secara umum, Bahasa Inggris merupakan salah satu bahasa global, maka bagi mereka yang ingin selangkah lebih maju dari pada yang lain umumnya, perlu bahkan harus menguasai Bahasa Inggris.

Banyak orang yang enggan belajar Bahasa Inggris karena merasa terlalu tua atau merasa sangat sulit untuk dapat menguasai bahasa tersebut. Bahasa Inggris dianggap sulit karena Bahasa Inggris mempunyai struktur tata bahasa yang berbeda dengan Bahasa Indonesia. Bahasa Inggris memiliki kata-kata yang terdengar mirip, tetapi maknanya benar-benar berbeda sehingga dapat menyebabkan kesalahpahaman atau membuat orang bingung. Modal utama untuk berbicara dalam Bahasa Inggris dengan tidak bergantung pada usia ataupun tingginya tingkat pendidikan seseorang, namun sebesar apa tekad seseorang untuk belajar.

Belajar Bahasa Inggris sama halnya dengan Bahasa Indonesia, yang meliputi kosa kata (*vocabulary*), menulis (*writing*), membaca (*reading*) berbicara (*speaking*), mendengar (*listening*) dan tata bahasa (*grammar*). Namun dalam belajar Bahasa Inggris, khususnya dalam membuat suatu kalimat yang baik dan benar, pembelajar perlu memperhatikan penggunaan waktunya, antara lain waktu lampau, saat ini, waktu yang akan datang, dan lain-lain yang lebih dikenal dengan istilah *tenses*.

Berdasarkan latar belakang di atas, penulis membuat suatu aplikasi pembelajaran Bahasa Inggris, khususnya bagi pembelajar yang ingin membuat suatu kalimat secara tertulis

atau disebut *writing* dan tentu saja disesuaikan dengan ketentuan *tenses* yang berlaku.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, dapat ditemukan suatu rumusan masalah, yaitu bagaimana membangun suatu aplikasi pembelajaran *writing* agar sesuai dengan *English 16 tenses*?"

2. LANDASAN TEORI

2.1. Aplikasi

Aplikasi adalah program yang ditulis untuk melaksanakan tugas khusus dari pengguna. Jenis program ini mempunyai sifat pasti tentang pemrosesan yang harus dilakukan *file* data yang harus di proses guna menyelesaikan suatu pekerjaan. Pengertian aplikasi dalam laporan ini adalah untuk membantu pemakai dalam melakukan proses pembelajaran Bahasa Inggris khususnya dalam bidang *writing*. (Nugroho, 2007)

2.2. Pembelajaran

Pembelajaran pada hakekatnya merupakan upaya membelajarkan siswa, sedangkan perancangan pembelajaran merupakan penataan upaya tersebut agar muncul perilaku belajar. Memberikan pendapatnya bahwa istilah pembelajaran digunakan untuk menunjukkan usaha pendidikan yang dilaksanakan secara sengaja, dengan tujuan di tetapkan terlebih dahulu sebelum proses dilaksanakan, serta pelaksanaannya terkendali. (Yusufhadi, 2004)

2.3. Tenses

Bahasa Inggris pada umumnya memiliki 16 *tenses*. Menurut Huri (2000), *Tenses* adalah bentuk dari sebuah kata kerja yang menunjukkan:

1. Waktu terjadinya suatu peristiwa atau perbuatan.
2. Tingkat kesempurnaan suatu kegiatan pada waktu dilakukan atau peristiwa yang terjadi (Akan dilakukan, sedang terjadi, baru saja dimulai atau telah selesai).

Secara umum, 16 *tenses* tersebut dapat dilihat dalam tabel di bawah ini.

Tabel 2.1. *Tenses* Secara Umum

Berdasarkan Situasi	Berdasarkan Waktu				
	Present	Past	Future	Past Future	
Simple	Simple Present	Simple Past	Simple Future	Past Future	
Continuous	Present Continuous	Past Continuous	Future Continuous	Past Future Continuous	
Perfect	Present Perfect	Past Perfect	Future Perfect	Past Future Perfect	
Perfect Continuous	Present Perfect Continuous	Past Perfect Continuous	Future Perfect Continuous	Past Future Perfect Continuous	

Sumber : (Novia, 2009)

2.4. Hypertext Preprocessor (PHP)

Pengertian PHP merupakan standar yang digunakan dalam dunia *website*, PHP adalah bahasa program yang berbentuk skrip yang diletakan di dalam *server web*. Pengertian PHP dalam laporan ini adalah sebagai program dalam pembuatan aplikasi pembelajaran Bahasa Inggris berbasis *web*. (Nugroho, 2008)

2.5. Xampp

Xampp adalah sebuah paket kumpulan software yang terdiri dari apache, mysql, phpmyadmin, php, Perl, Freetype2, dll. Xampp berfungsi untuk memudahkan instalasi lingkungan php, di mana biasanya lingkungan pengembangan *web* memerlukan php, apache, mysql dan phpmyadmin serta software-software yang terkait dengan pengembangan *web*. Dengan menggunakan xampp, maka tidak perlu menginstall aplikasi-aplikasi tersebut satu persatu. (Sofwan, 2011)

2.6. Hypertext Markup Language (HTML)

HTML atau yang memiliki kepanjangan *Hypertext Markup Language* adalah script dimana kita bisa menampilkan informasi dan daya kreasi kita lewat internet. HTML sendiri adalah suatu dokumen teks biasa yang mudah dimengerti dibanding bahasa pemrograman lainnya, dan karena bentuknya itu maka HTML dapat dibaca oleh platform seperti: windows, linux, macintosh. Kata "*Markup Language*" pada HTML menunjukkan fasilitas yang berupa teks tertentu dalam script HTML dimana kita bisa mengatur judul, garis, tabel, gambar, dan lain- lain dengan perintah yang telah ditentukan pada elemen HTML. (Jamal, 2009)

2.7. Unified Modelling Language (UML)

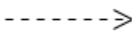
Pengertian *Unified Modelling Language* (UML) merupakan salah satu bentuk *language* atau bahasa, menurut pencetusnya UML didefinisikan sebagai bahasa *visual* untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem. Perancangan sistem pada UML adalah sebagai berikut:

2.7.1. Use case diagram

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

Tabel 2.1. Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		Use case	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2		Actor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3		Asosiasi / association	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4		Ekstensi / extend	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal arah panah mengarah pada <i>use case</i> yang ditambahkan.
5		Generalisasi / generalization	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua

			buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).
6		Menggunakan / <i>include / uses</i>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p> <ul style="list-style-type: none"> ▪ <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan. ▪ <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan. Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

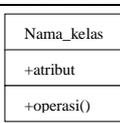
Sumber: (Shalahuddin, 2011)

2.7.2. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

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

Tabel 2.2. Simbol Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		Kelas	Kelas pada struktur sistem
2		Antarmuka/ <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.

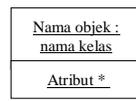
3		Asosiasi/ <i>association</i>	Relasi antar kelas dengan makna umum, asosiasi, biasanya juga disertai dengan <i>multiplicity</i> .
4		Asosiasi berarah/ <i>directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5		generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
6		Kebergantungan/ <i>dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7		Agregasi/ <i>aggregation</i>	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).

Sumber: (Shalahuddin, 2011)

2.7.3. Object Diagram

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggungjawabkan. Untuk apa mendefinisikan sebuah kelas sedangkan pada jalan sistem, objeknya tidak pernah dipakai. Hubungan *link* pada diagram objek merupakan hubungan memakai dan dipakai di mana dua buah objek akan dihubungkan oleh *link* jika ada objek yang dipakai oleh objek lainnya.

Tabel 2.3. Simbol Object Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		Objek	Objek dari kelas yang berjalan saat sistem dijalankan
2		Link	Relasi antar objek

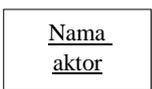
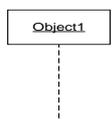
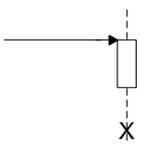
Sumber: (Shalahuddin, 2011)

2.7.4. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek, oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas diinstansiasi menjadi objek itu. Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Tabel 2.4. Simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		Aktor/ Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi

			walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2		Garis hidup/ <i>lifeline</i>	Menyatakan kehidupan suatu objek
3		Objek	Menyatakan objek yang berinteraksi pesan
4		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
5		Pesan Tipe <i>create</i>	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat
6		Pesan Tipe <i>call</i>	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7		Pesan Tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8		Pesan Tipe <i>return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9		Pesan Tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang di akhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

Sumber: (Shalahuddin, 2011)

2.7.5. Collaboration Diagram

Diagram kolaborasi menunjukkan informasi yang sama persis dengan diagram sekuensial, tetapi dalam bentuk dan tujuan yang berbeda. Sebagaimana diagram sekuensial, diagram kolaborasi digunakan untuk menampilkan aliran skenario tertentu di dalam *use case*. Jika diagram sekuensial disusun berdasarkan urutan waktu, diagram kolaborasi lebih berkonsentrasi pada hubungan antar objek-objek.

Tabel 2.5. Simbol *collaboration Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		Objek	Objek merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara <i>horizontal</i> . Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama <i>object</i> didalamnya yang diawali dengan sebuah titik koma.
2		Aktor	Aktor juga dapat berkomunikasi dengan objek, maka aktor juga dapat diurutkan sebagai kolom. Simbol aktor sama pada actor use case.
3		Pesan	<i>Message</i> digambarkan dengan anak panah yang mengarah antar objek dan diberi label urutan nomor yang mengindikasikan urutan komunikasi yang terjadi antar objek.

Sumber: (Booch, 1999)

2.7.6. State Diagram

Diagram status digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah sistem. Perubahan tersebut digambarkan dalam suatu graf berarah. Diagram status merupakan pengembangan dari diagram *Finite State Automata* dengan penambahan beberapa fitur dan konsep baru. Diagram *Finite State Automata* (FSA) ini biasanya diajarkan dalam mata kuliah *Automata*. Diagram status cocok digunakan untuk menggambarkan alur interaksi pengguna dengan sistem.

Tabel 2.6. Simbol *State Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Start (initial state)</i>	<i>Start</i> dan <i>initial state</i> adalah <i>state</i> atau keadaan awal pada saat system mulai hidup
2		<i>End (final state)</i>	<i>End</i> atau <i>final state</i> adalah <i>state</i> keadaan akhir dari daur hidup suatu sistem.
3		<i>Event</i>	<i>Event</i> adalah kegiatan yang menyebabkan berubahnya status sistem
4		<i>State</i>	<i>State</i> atau status adalah keadaan system pada waktu tertentu. <i>State</i> dapat berubah jika ada <i>event</i> tertentu yang memicu perubahan tersebut

Sumber: (Shalahuddin, 2011)

2.7.7. Activity Diagram

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

Tabel 2.7. Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		Status awal	Status awal aktivitas system, sebuah diagram aktivitas memiliki sebuah status awal

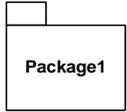
2		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3		Percabangan/ <i>decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4		Penggabungan/ <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5		Status akhir	Status akhir yang dilakukan system, sebuah diagram aktivitas memiliki sebuah status akhir
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: (Shalahuddin, 2011)

2.7.8. Component Diagram

Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem.

Tabel 2.8. Simbol *Component Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Package</i>	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen
2		Komponen	Komponen sistem
3		Kebergantungan/ <i>dependency</i>	Komponen antar komponen, arah panah mengarah pada komponen yang dipakai
4		Antarmuka/ <i>interface</i>	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen
5		<i>Link</i>	Relasi antar komponen

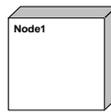
Sumber: (Shalahuddin, 2011)

2.7.9. Deployment Diagram

Diagram *deployment* atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut:

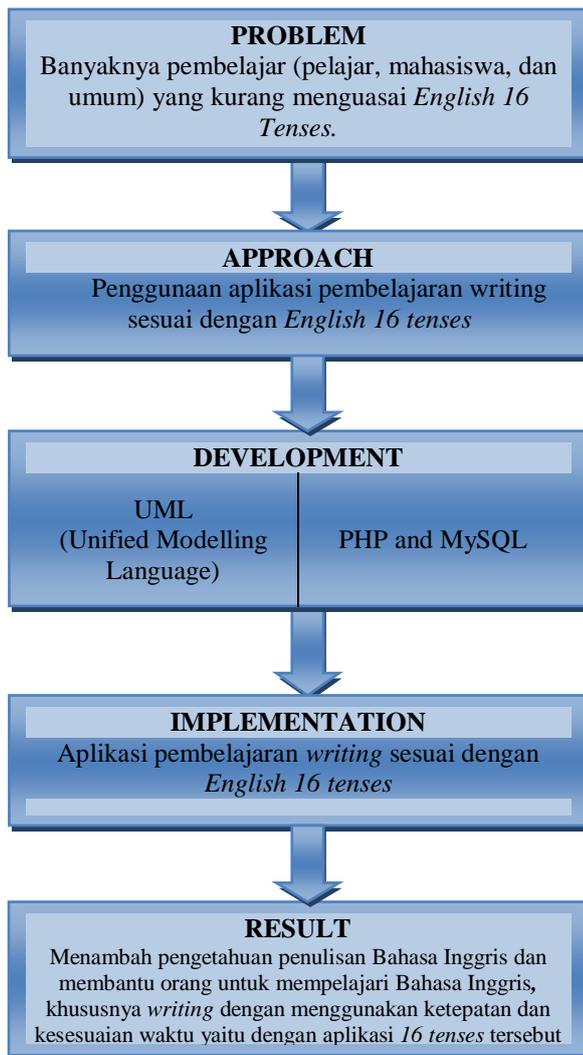
- Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*
- Sistem terdistribusi murni
- Rekayasa ulang aplikasi

Tabel 2.9. Simbol *Deployment Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Package</i>	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih <i>node</i>
2		<i>Node</i>	Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikuti sertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen
3		Kebergantungan/ <i>dependency</i>	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai
4		<i>Link</i>	Relasi antar <i>node</i>

Sumber: (Shalahuddin, 2011)

3. METODE PENELITIAN



Gambar 3.1. Kerangka Penelitian

3.1. Metode Pengumpulan Data

Metode yang digunakan dalam mengumpulkan data untuk artikel ini adalah menggunakan metode kepustakaan (*Library Research*), dalam metode ini dilakukan pencarian bahan yang mendukung dalam pendefinisian masalah melalui buku-buku dan internet yang berhubungan dengan laporan yang dilakukan.

3.2. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah metode OOAD (*object oriented analysis design*). OOA adalah metode analisis yang memeriksa *requirement* (syarat/keperluan) yang harus dipenuhi sebuah sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup perusahaan.

OOD adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem. OOD mengubah model konseptual yang dihasilkan dalam analisis berorientasi objek memperhitungkan kendala yang dipaksakan oleh arsitektur yang dipilih dan setiap non fungsional teknologi atau

lingkungan, kendala, seperti transaksi *throughput*, *response time*, *run*, waktu *platform*, lingkungan pengembangan, atau bahasa pemrograman.

3.3. Analisis

Analisa merupakan bidang yang menarik, melibatkan studi interaksi antar manusia, kelompok-kelompok orang, komputer dan organisasi. Adapun tahapan analisis berorientasi objek yang digunakan adalah sebagai berikut:

3.3.1. Analisis Kebutuhan Sistem

Dalam sistem aplikasi pembelajaran ini terdapat 2 jenis aktor yang saling berinteraksi dalam lingkungan sistem, yaitu:

- a. *User*
- b. *Admin*

Untuk masuk kedalam sistem, seorang *user* terlebih dahulu melakukan *login*. Setelah itu, *user* dapat mengakses isi dari aplikasi pengajaran Bahasa Inggris, Seperti materi dan latihan. Jika *user* belum memiliki *account* untuk *login*, *user* bisa mendaftar pada *form* daftar. *Admin* bisa melakukan *Inputan* materi dan latihan. *Admin* bisa juga menghapus isi materi dan isi latihan.

3.3.2. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, perekraya perangkat lunak (analisis) harus memahami domain informasi, tingkah laku untuk kerja, dan antarmuka (*interface*) yang diperlukan. Dalam merancang suatu perangkat lunak, perlu memperhatikan kebutuhan yang akan digunakan dalam perancangan aplikasi sebagai berikut:

a. *Hardware*

Spesifikasi kebutuhan perangkat keras dalam laporan ini adalah sebagai berikut:

- 1) Laptop Acer V-5
- 2) Processor Integerel® Celeron® CPU 877 @1.40GHz 1.40GHz
- 3) Installed memory (RAM) 2.00 GB (1.79 GB usable)
- 4) Printer IP 1200
- 5) Mouse
- 6) Flashdisk

b. *Software*

Spesifikasi kebutuhan perangkat lunak dalam laporan ini adalah sebagai berikut:

- 1) Sebagai *web browser* seperti: *Internet Explorer*, *Mozilla Firefox*, *Google Chrome*.
- 2) PHP, sebagai bahasa pemrograman untuk membuat perangkat lunak.
- 3) Xampp, sebagai *web server* dan *database*.
- 4) Microsoft Office 2013, sebagai aplikasi pengolahan data untuk penulisan laporan.
- 5) Microsoft Windows 7 ultimate, sebagai sistem operasi.

4. HASIL DAN PEMBAHASAN

4.1. Hasil

1. Halaman *login*

Halaman *login* adalah halaman untuk *user* masuk dan menggunakan aplikasi. Fungsi yang lainnya apabila *user* belum punya akun bisa daftar dengan mengklik *link sign up* seperti tampak pada gambar 4.1 dibawah ini:



Gambar 4.1. Halaman *login*

2. Halaman *Sign Up* /Pendaftaran *User* Baru

Pada halaman ini *user* dapat mendaftar dengan mengisikan *username* dan *password* seperti tampak pada gambar 4.2:



Gambar 4.2. Halaman *Sign Up*/Pendaftaran *User* Baru

3. Halaman *Home*/Beranda

Halaman *home* berisikan sekilas tentang *writing* dan *user* dapat memilih menu menu yang ada di aplikasi, seperti: *learning material* dan *exercise* seperti pada gambar 4.3 di bawah ini:



Gambar 4.3. Halaman *Sign Up*/Pendaftaran *User* Baru

4. Halaman *Learning Material*/Materi Pembelajaran

Halaman ini memuat pilihan menu *English 16 tense* yang bisa dipilih *user*.



Gambar 4.4. Halaman *Learning Material*/Materi Pembelajaran

5. Halaman Salah Satu *Learning Material*

Halaman ini berisikan materi tentang salah satu 16 tense yang bisa *user* lihat seperti tampak pada gambar 4.5:



Gambar 4.5. Halaman salah satu *Learning Material*

6. Halaman Isi Materi

Halaman ini berisi tentang penjelasan materi, rumus, dan contoh kalimat materi seperti tampak pada gambar 4.6:



Gambar 4.6. Halaman Isi Materi

7. Halaman Exercise

Halaman ini berisikan soal-soal latihan sesuai dengan English 16 tense seperti tampak pada gambar 4.7:



Gambar 4.7. Halaman Exercise

8. Halaman Kelola Inputan

Halaman ini untuk menginput-kan materi, soal latihan, dan kunci jawaban seperti tampak pada gambar 4.8:



Gambar 4.8. Halaman Kelola Inputan

5. KESIMPULAN

Aplikasi pembelajaran tentang *writing* berdasarkan *tenses* ini merupakan media yang sangat memungkinkan untuk pembelajar/*user*, terutama bagi pemula karena lebih memudahkan mereka dalam belajar Bahasa Inggris, terutama *writing* sesuai dengan aturan penggunaan 16 *tenses* yang berlaku secara umum.

DAFTAR PUSTAKA

Baehaqi, I. (2009). *A Handbook Of English Grammar*. Yogyakarta: Media Ilmu.

Booch, G. (1999). *Visual Modeling With Rational Rose 2000 And UML*.

Harmer, J. (2007). *How to Teach English*. Pearson Education Limited.

Huri, D. A. (2000). *The Understanding Grammar*.

Jamal, F. N. (2009). *Desain Web Dinamis Menggunakan Macromedia Dreamweaver 8 pada SMA Kartika 1-1 Medan*. Medan: Universitas Sumatera Utara.

Jubilate, A. (2012). *Analisis Perancangan Aplikasi Pengajaran Bahasa Inggris Berbasis Web*. Palembang: Universitas Binadarma.

Novia, W. (2009). *Essential English Grammar Complete Edition*. Wipress: Wacana Intelektual.

Nugroho, B. (2007). *Membuat Aplikasi Database SQL Server dengan Visual Basic 6.0*. Yogyakarta: Gava Media.

Nugroho, B. (2008). *Latihan Membuat Aplikasi Web PHP MySQL dengan Dreamweaver mx (6,7,2004) dan 8*. Yogyakarta: Gava Media.

Shalahuddin, R. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. MODULA.

Sofwan, A. (2011). *MySQL dan Phpmyadmin*. Retrieved from <http://ilmukomputer.org/wpcontent/uploads/2011/08/s ofwanmysqldanphpmy admin.pdf> diakses pada tanggal 30 Desember,2014

Yusufhadi, M. (2004). *Menyemai Benih Teknologi Pendidikan*. Jakarta: Prenada Media.