

ANALISIS DAN PERANCANGAN APLIKASI PENGAJARAN BAHASA INGGRIS BERBASIS WEB

Ari Jubilate¹, Widya Cholil², M.Sobri³

Mahasiswa Universitas Bina Darma¹, Dosen Universitas Bina Darma^{2,3}

Jalan Jenderal Ahmad Yani No.12 Palembang

Pos-el : ai.late90@gmail.com, ai_unyu@yahoo.com

Abstract : Entering the era of globalization, English is a requirement absolute necessity for all parties to penetrate the global competition, both for students who will continue their education to higher education or for the enter the working world. Sometimes many people are able to master speaking, but weak in writing. Some people often underestimate the writing because they think writing is not so important. Writing is very important to use for writing a book and also used for writing speeches. Writing the correct is writing based on the structure. system development methods used the metod OOAD (object oriented analysis design). To help to learn English one way to create a media teaching of English that uses the internet media. Teaching Applications This is a web-based solution to implement a learning system effective and allows flexible teaching materials tersampaiannya (Whenever and wherever located) to its users.

Keywords : Computers, Internet, English

Abstrak : Memasuki era globalisasi seperti sekarang, bahasa Inggris merupakan suatu syarat mutlak bagi semua pihak untuk menembus persaingan global, baik bagi siswa yang akan melanjutkan pendidikan ke jenjang yang lebih tinggi ataupun bagi yang memasuki dunia kerja. Terkadang banyak orang mampu menguasai speaking, tapi lemah di writing. Sebagian orang sering menyepelekan writing karena mereka beranggapan writing tidak begitu penting. Writing sangat penting digunakan untuk penulisan buku dan juga digunakan untuk penulisan pidato. Writing yang benar adalah penulisan berdasarkan structure. metode pengembangan sistem yang digunakan adalah metod OOAD (object oriented analysis design). Untuk membantu belajar bahasa Inggris salah satunya dengan cara membuat suatu media pengajaran bahasa Inggris yang menggunakan media internet. Aplikasi Pengajaran berbasis web ini merupakan solusi untuk menerapkan sistem pembelajaran yang efektif dan memungkinkan tersampaiannya bahan pengajaran secara fleksibel (kapanpun dan dimanapun berada) kepada para penggunanya.

Kata Kunci : Komputer, Internet, Bahasa Inggris

1. PENDAHULUAN

1.1. Latarbelakang

Di era globalisasi seperti sekarang, bahasa Inggris merupakan salah satu bahasa

internasional yang sering digunakan. Oleh sebab itu, bahasa Inggris dianggap penting oleh banyak pihak dalam rangka mengembangkan ilmu pengetahuan, teknologi, seni, dan budaya, serta untuk membina hubungan dan mempererat pergaulan antar bangsa di dunia. Belajar bahasa

Inggris sama halnya dengan bahasa Indonesia, yang meliputi kosa kata (*vocabulary*), menulis (*writing*), membaca (*reading*) mengucapkan (*speaking*), mendengar (*listening*) dan sebagainya. Penelitian ini akan memberikan pengajaran tentang *writing* berdasarkan *tenses*, karena bagaimana mungkin kita dapat menulis dengan benar jika kita tidak memahami betul aturan menyusun kalimat dengan benar. Untuk membantu belajar bahasa Inggris salah satunya dengan cara membuat suatu media pengajaran bahasa Inggris yang menggunakan media internet. Penelitian ini dibatasi pada analisis dan perancangan aplikasi pengajaran bahasa Inggris dengan metode *object oriented analysis design (OOAD)* dan tidak sampai pada tahap implementasinya.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, maka perumusan masalahnya yaitu "Bagaimana Merancang Sebuah Aplikasi pengajaran Bahasa Inggris berbasis *web*".

1.3. Batasan Masalah

Penelitian ini dibatasi pada analisis dan perancangan aplikasi pengajaran bahasa Inggris dengan metode *object oriented analysis design (OOAD)*.

1.4. Tujuan

Adapun tujuan dari penelitian ini antara lain :

1. Menghasilkan analisa dan perancangan aplikasi pengajaran bahasa Inggris berbasis *web*
2. Membantu belajar bahasa Inggris engan memahami *structure* bahasa Inggris.
3. Memberikan media yang lain dalam mempelajari bahasa Inggris berbasis *web*

2. METODOLOGI PENELITIAN

2.1. Waktu dan Tempat Penelitian

Waktu penelitian yang dilakukan oleh penulis selama 5 (lima) bulan, dimulai dari bulan Oktober 2011 sampai bulan Februari 2012.

2.2. Metode Pengumpulan Data

Metode yang digunakan dalam mengumpulkan data untuk penelitian 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 penelitian yang dilakukan.

2.3. Metode Analisis dan Perancangan

Metode yang digunakan dalam penelitian ini adalah metode OOAD. OOAD (*object oriented analysis design*) adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan

adalah objek, yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. Pengertian “berorientasi objek” berarti bahwa kita mengorganisasi perangkat lunak sebagai kumpulan objek tertentu yang memiliki struktur data dan perilakunya. Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). 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. Sedangkan OOD adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem. (<http://dewa-hendra.blogspot.com/2010/04/i.html>).

2.3.1. *Object Oriented Analysis*

OOA mempelajari permasalahan dengan menspesifikasikannya atau mengobservasi permasalahan tersebut dengan menggunakan metode berorientasi objek. Biasanya analisa sistem dimulai dengan adanya dokumen permintaan (requirement) yang diperoleh dari semua pihak yang berkepentingan. (Misal: klien, developer, pakar, dan lain-lain). Dokumen permintaan memiliki 2 fungsi yaitu : memformulasikan kebutuhan klien dan membuat suatu daftar tugas. Analisis berorientasi obyek (OOA) melihat pada domain masalah, dengan tujuan untuk memproduksi sebuah model

konseptual informasi yang ada di daerah yang sedang dianalisis. Model analisis tidak mempertimbangkan kendala-kendala pelaksanaan apapun yang mungkin ada, seperti konkurensi, distribusi, ketekunan, atau bagaimana sistem harus dibangun. Kendala pelaksanaan ditangani selama desain berorientasi objek (OOD). Sumber-sumber untuk analisis dapat persyaratan tertulis, pernyataan, dokumen visi yang formal, wawancara dengan stakeholder atau pihak berkepentingan lain. Sebuah sistem dapat dibagi menjadi beberapa domain yang mewakili bisnis yang berbeda, teknologi, atau bidang yang diminati. Masing-masing dianalisis secara terpisah. Hasil analisis berorientasi objek adalah deskripsi dari apa sistem secara fungsional diperlukan untuk melakukan, dalam bentuk sebuah model konseptual. Itu biasanya akan disajikan sebagai seperangkat menggunakan kasus, satu atau lebih UML diagram kelas, dan sejumlah diagram interaksi. Tujuan dari analisis berorientasi objek adalah untuk mengembangkan model yang menggambarkan perangkat lunak komputer karena bekerja untuk memenuhi seperangkat persyaratan yang ditentukan pelanggan. Metode analisis berorientasi objek yang digunakan pada penelitian ini adalah metode *Unified* (Hariyanto, 2004 : 380). Adapun tahapan analisis berorientasi objek yang digunakan adalah :

1. Berpedoman pada kebutuhan pemakai sistem.

2. Mengidentifikasi skenario pemakaian atau *use case*.
3. Memilih kelas-kelas dan objek-objek menggunakan kebutuhan sebagai penuntun.
4. Mengidentifikasi atribut dan operasi untuk masing-masing kelas objek.
5. Mengidentifikasi struktur dan hirarki kelas-kelas.
6. Membangun model keterhubungan kelas dan objek.
7. Melakukan *review* model yang dihasilkan dengan skenario atau *use case*.

2.3.2. *Object Oriented Design*

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. Metode perancangan berorientasi objek yang digunakan adalah metode yang diadaptasi dari metode Coad-Yourdon (Hariyanto, 2004 : 418), yaitu :

1. Perancangan Sistem Perancangan sistem merepresentasikan arsitektur perangkat lunak, yaitu :
 - a. Arsitektur konseptual berkaitan dengan model kelas statik dan koneksi-

koneksi diantara komponen-komponen model.

b. Arsitektur modul mendeskripsikan pembagian sistem menjadi subsistem-subsistem atau modul-modul serta bagaimana modul-modul ini saling berkomunikasi (di sistem berorientasi objek adalah melalui pelewatan pesan).

c. Arsitektur kode mendefinisikan bagaimana kode program diorganisasikan menjadi *file-file* dan direktori-direktori dan dikelompokan menjadi pustaka-pustaka.

d. Arsitektur eksekusi berfokus pada aspek dinamis dari sistem dan komunikasi di antara komponen-komponen sebagai *task* dan operasi yang di eksekusi.

2. Perancangan Objek

Perancangan objek berfokus pada deskripsi objek dan interaksi-interaksinya satu dengan yang lainnya secara rinci. Spesifikasi rinci dari struktur data untuk atribut dan rancangan procedural dari semua operasi diciptakan selama perancangan objek. Ketampakan dari semua atribut kelas didefinisikan dan antarmuka antara objek-objek dielaborasi untuk mendefinisikan rincian-rincian dari model pelewatan pesan secara lengkap.

3. Perancangan Antarmuka Pemakai

Pandangan model pemakai menuntun ke proses perancangan antarmuka pemakai, menyediakan scenario yang dielaborasi secara iterasi untuk menjadi sekumpulan kelas-kelas antarmuka.

4. Perancangan Manajemen Data

Perancangan manajemen data adalah untuk mendukung *persistent objects* yang menetapkan sekumpulan kelas dan kolaborasi-kolaborasi yang memungkinkan sistem mengelola *persistent data* (yaitu *file-file* dan basis data) yang mengimplementasikan *persistent objects*. Pada umumnya manajemen data dirancang secara berlapis. Gagasan yang dimajukan adalah mengisolasi kebutuhan level rendah untuk memanipulasi struktur data dari kebutuhan level lebih tinggi untuk menangani atribut-atribut sistem. Perancangan manajemen data meliputi perancangan atribut-atribut dan operasi-operasi yang diperlukan untuk mengelola objek-objek. Coad-yourdon menyarankan pembentukan "*object-server class*" yang memberikan layanan-layanan sebagai berikut :

a. Memberitahukan ke objek untuk menyimpan dirinya sendiri.

b. Mengambil objek-objek yang disimpan untuk pengguna komponen komponen rancangan yang lain.

5. Perancangan Manajemen *Task*

Perancangan manajemen *task* menetapkan infrastruktur yang mengorganisasi subsistem-subsistem menjadi *task-task* dan kemudian mengelola kongkurensi *task*. Sistem *multitasking* memungkinkan banyak *task* berada di sistem komputer pada satu saat. Sistem operasi akan melakukan penjadwalan terhadap eksekusi *task-task* yang berada di sistem komputer. *Task-task* terpisah memisahkan perilaku-perilaku yang harus berlangsung konkuren. Perilaku konkuren dapat diimplementasikan dan dieksekusi secara *pseudoparallel* pada sistem *multiprocessing* atau ditiru secara *pseudoparallel* pada sistem *singleprocessing*. Pada era modern, sistem *multitasking* telah diperluas menjadi *multithreading*, dimana tiap *task* dapat berisi lebih dari satu jalur eksekusi yang terpisah. Perancangan untuk konkurensi *task* dan *thread* merupakan penambahan kompleksitas, namun member efisiensi dan optimasi dalam penggunaan sistem computer. Coad-Yourdon menyarankan strtegi

merancang objek-objek untuk pengelolaan *task-task* konkuren.

2.4. Tinjauan Pustaka

2.4.1. Pengertian Analisis

Pengertian Analisis (Hariyanto, 2004 : 369) merupakan bidang yang menarik, melibatkan studi interaksi antar manusia, kelompok-kelompok orang, computer dan organisasi.

Pengertian analisis dalam penelitian ini adalah untuk mengidentifikasi kekurangan yang ada pada sistem yang sebelumnya.

2.4.2. Pengertian Aplikasi

Pengertian Aplikasi (Nugroho, 2007 : 1) 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 penelitian ini adalah untuk membantu pemakai dalam melakukan proses pembelajaran bahasa Inggris khususnya dalam bidang *writing*.

2.4.3. Pengertian Website

Pengertian *Website* (Yuhefizar, 2011 : 1) adalah keseluruhan halaman-halaman web yang terdapat dari sebuah domain yang mengandung Informasi.

Pengertian *website* dalam penelitian ini adalah sebagai media/tempat pemrosesan aplikasi pengajaran bahasa Inggris dalam bidang *writing*.

2.5. Analisis dan Perancangan

2.5.1. Entity Relationship Diagram

Pengertian ERD (*entity relationship diagram*) adalah pemodelan data utama yang membantu mengorganisasikan data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas. Sistem adalah kumpulan elemen yang setiap elemen memiliki fungsi masing-masing dan secara bersama-sama mencapai tujuan dari sistem tersebut. Kebersamaan dari sistem diatas dilambangkan dengan saling berelasinya antara satu dan entitas dengan entitas yang lainnya. Entitas (*entity / entity set*), memiliki banyak istilah di dalam ilmu komputer, seperti *table (table)*, *berkas (data file)*, *penyimpanan data (data store)*, dan sebagainya.

(<http://BugiSatriaPengertianERDdanDFDdanCintosh.htm>)

Pengertian ERD dalam penelitian ini adalah sebagai gambaran *database*.

Komponen utama identifikasi untuk *entity relationship diagram* (ERD) berupa :

1. Entitas

Adalah representasi dari hampir semua informasi gabungan yang harus dipahami oleh

perangkat lunak, dengan informasi gabungan dapat diartikan sesuatu yang memiliki sejumlah sifat atau atribut yang berbeda. Objek data diwakili oleh sebuah persegi panjang yang diberi label. Objek data dihubungkan satu dengan yang lainnya, hubungan itu ditentukan oleh konteks masalah yang sedang dianalisis. Objek data dapat berupa :

- a. *Entitas eksternal* (misalnya semua yang menghasilkan informasi),
- b. Benda (berupa laporan),
- c. Tempat (misal gudang) dan sebagainya.

2. *Atribut*

Atribut menentukan properti suatu objek data dan mengambil salah satu dari tiga karakteristik yang berbeda. *Atribut* dapat digunakan untuk :

- a. Menamai sebuah contoh dari objek data,
- b. Menggambarkan contoh,
- c. Membuat referensi ke contoh yang lain pada table yang lain.

Satu atribut atau lebih didefinisikan sebagai sebuah pengidentifikasi, dimana atribut pengidentifikasi akan menjadi sebuah kunci untuk menemukan sebuah objek dari objek data.

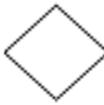
3. *Relationship*

Hubungan ditunjukkan dengan garis yang diberi label yang menghubungkan objek. Sambungan antara data dan objek dan hubungan dibangun dengan menggunakan berbagai simbol khusus yang menunjukkan kardinalitas dan modalitas.

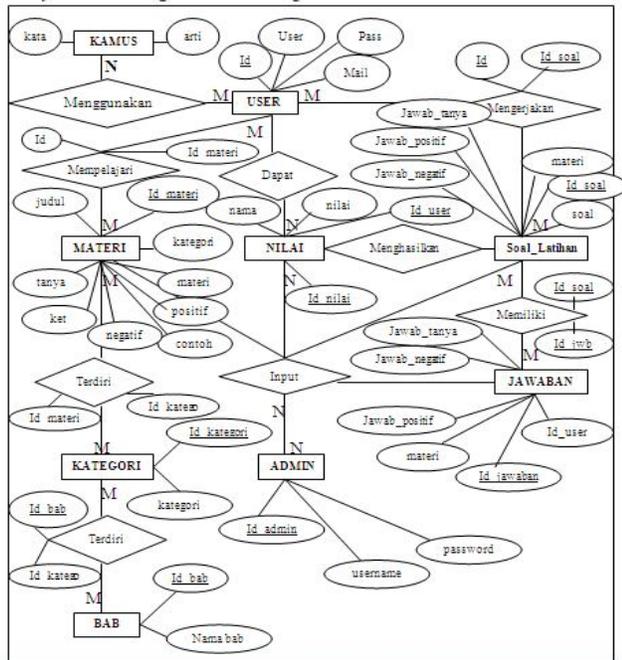
4. *Link*

Yaitu tanda garis yang digunakan untuk menghubungkan komponen-komponen ERD.

Tabel 1. *Entity Relationship Diagram(ERD)*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Entitas</i>	Adalah representasi dari hampir semua informasi gabungan yang harus dipahami oleh perangkat lunak.
2		<i>Relation</i>	Hubungan ditunjukkan dengan garis yang diberi label yang menghubungkan objek.
3		<i>Link</i>	Tanda garis yang digunakan untuk menghubungkan komponen-komponen ERD.
4		<i>Atribut</i>	Atribut menentukan properti suatu objek data dan mengambil salah satu dari tiga karakteristik yang berbeda.

Adapun rancangan proses database dalam *entity relationship diagram* (ERD) adalah sebagai berikut :



Gambar 1. Entity Relationship Diagram(ERD)

2.5.2. Unified Modelling Language (UML)

Pengertian Unified Modelling Language (UML) (Nugroho, 2005 : 16) merupakan salah satu bentuk language atau bahasa, menurut pencetusnya UML di definisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem. Perancangan sistem pada UML adalah sebagai berikut :

1. Use case diagram

Use Case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah Use Case mempresentasikan sebuah interaksi antara aktor dengan sistem. Urutan langkah-langkah yang menerangkan antara

pengguna dan sistem disebut *scenario*. Seorang / sebuah aktor adalah sebuah *entitas* manusia mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Adapun gambar simbol – simbol pada *entity relationship diagram* (ERD) adalah sebagai berikut :

Tabel 2. Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		Dependency	terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		Include	Menspesifikasikan bahwa use case sumber secara eksplisit.
5		Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
6		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2. Activity diagram

Activity diagram merupakan *state* diagram khusus yang menggambarkan berbagai

alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi.

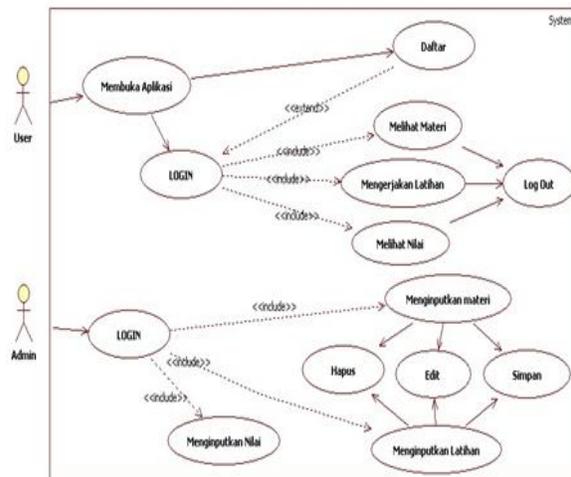
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk atau dihancurkan.
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

Tabel 3. *Activity* Diagram

2.5.2.1. Analisis Interaksi

Analisis interaksi menggambarkan aktifitas interaksi antara admin dan user untuk melihat materi, mengerjakan latihan, pendaftaran, menginputkan materi, dan menginputkan latihan. use case diagram pada penelitian ini dapat dilihat pada gambar berikut. Skenario pada *use case diagram* adalah sebagai berikut ; *user* membuka aplikasi. Jika *user* belum memiliki account, *user* terlebih dahulu mendaftarkan pada form pendaftaran. Jika *user* sudah memiliki account, *user* dapat langsung login dan dapat langsung mengakses fitur-fitur yang tersedia pada aplikasi. Seperti melihat materi, mengerjakan latihan dan melihat nilai. Dan jika *user* ingin keluar dari aplikasi, *user*

dapat menekan link logout. Untuk *admin*, sebelum *admin* menginputkan materi dan menginputkan latihan, *admin* harus login terlebih dahulu barulah *admin* bisa menghapus, menyimpan, dan mengedit inputannya tadi. Gambar *use case diagram* adalah sebagai berikut :



Gambar 2. *Use case diagram*

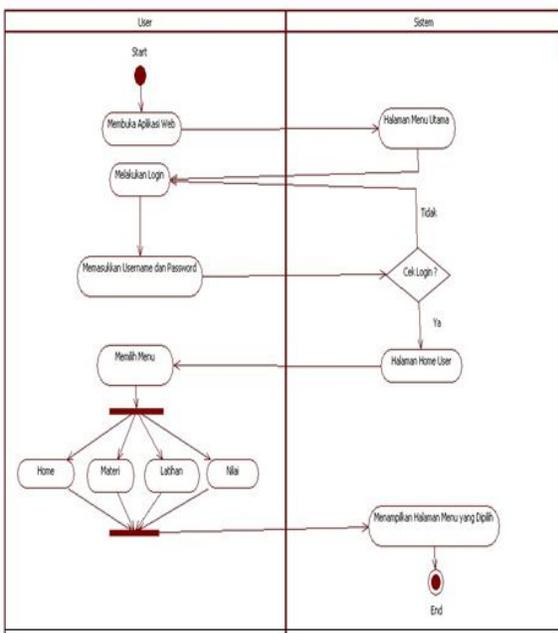
2.5.2.2. Analisis Prilaku

Diagram aktifitas menggambarkan berbagai aliran aktifitas dalam sistem yang sedang di rancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi dan bagaimana mereka berakhir. Terdapat dua *diagram activity*, yaitu ; *activity diagram user* dan *activity diagram admin*

1. *Activity diagram user*

Skenario *user* pada *Activity Diagram User* adalah sebagai berikut. *User* membuka aplikasi kemudian sistem menampilkan halaman utama. Dan *user* melakukan login dengan cara

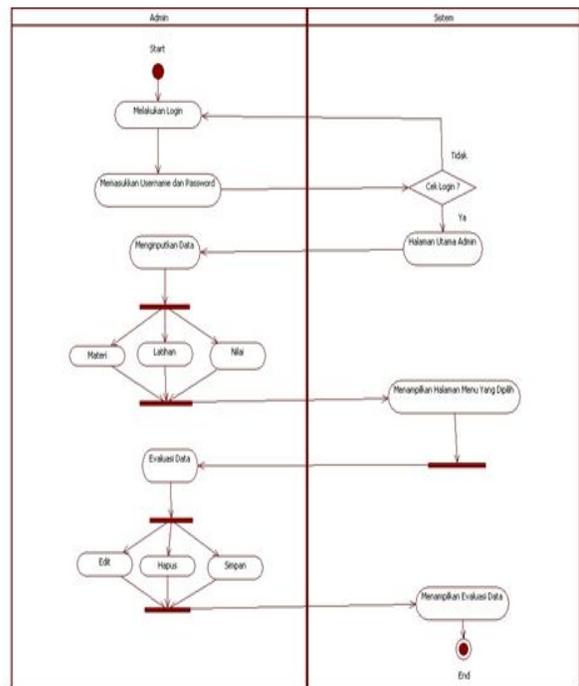
memasukkan username dan password. Kemudian sistem akan mengecek username dan password yang *user* masukan tadi. Jika terjadi kesalahan, maka sistem akan mengembalikan kehalaman utama agar *user* kembali melakukan login. Jika username dan passwordnya benar atau *valid*. Maka sistem akan menampilkan halaman home *user*. disini *user* dapat memilih beberapa menu yaitu home, materi, latihan, dan nilai. Misalnya *user* memilih menu latihan. Kemudian sistem akan menampilkan halaman menu latihan. Disitu *user* diharuskan untuk mentraslatekan kalimat bahasa Indonesia ke bahasa Inggris. Kemudian user juga merubah hasil translate kalimat tadi menjadi kalimat negatif dan interrogative. Setelah semua kalimat sudah dituliskan, maka *user* akan mengklik buttom simpan. Kemudian data yang tersimpan di *database*, kemudian di koreksi oleh *admin*. Gambar *activity diagram user* adalah sebagai berikut :



Gambar 3. Activity diagram user

2. Activity diagram admin

Skenario *admin* pada Activity Diagram *Admin* adalah sebagai berikut. *Admin* melakukan login dengan cara memasukkan username dan password. Kemudian sistem akan mengecek username dan password yang *admin* masukan tadi. Jika terjadi kesalahan, maka sistem akan mengembalikan kehalaman login agar *admin* kembali melakukan login. Jika username dan passwordnya benar atau *valid*. Maka sistem akan menampilkan halaman utama *admin*. Kemudian *admin* dapat menginputkan data ke menu materi, latihan, dan nilai. kemudian sistem akan menampilkan halaman menu yang dipilih. Kemudian *admin* akan mengevaluasi data dengan cara simpan, hapus, dan edit. Dan sistem akan menampilkan hasil evaluasi. Adapun Gambar *activity diagram user* adalah sebagai berikut :



Gambar 4. Activity diagram admin

3. Hasil dan Pembahasan

3.1. Halaman Menu Utama

Tampilan utama yang akan ditampilkan pertama kali pada saat diakses adalah menu utama. Bagi *user* yang sudah mempunyai *account*, dapat langsung *login*. Sedangkan bagi pengunjung baru, mendaftarkan dengan mengklik *daftar*.



Gambar 5. Halaman Menu Utama

3.2. Halaman Daftar

Halaman daftar ini berfungsi sebagai aturan dalam penggunaan aplikasi ini, di sini *user* terlebih dahulu harus mengisi *form* yang telah disediakan. Adapun tampilan halaman daftar seperti gambar berikut :



Gambar 6. Halaman Daftar

3.3. Halaman Home User

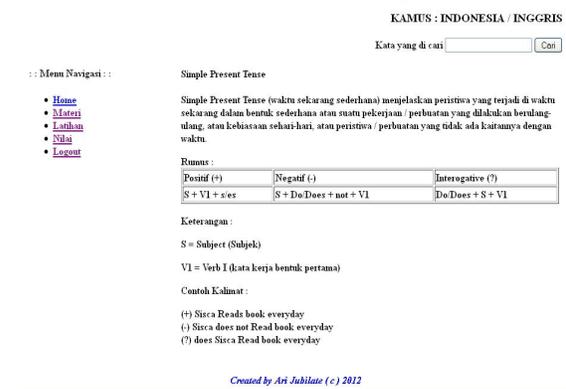
Setelah *user* mendaftarkan, kemudian login, jika *username* dan *password*nya benar maka halaman *home user* dapat digunakan. Halaman *home user* terdiri dari beberapa menu pilihan, yaitu *home*, *materi*, *latihan*, *nilai*, dan *logout*. Adapun tampilan halaman *home user* seperti gambar berikut :



Gambar 7. Halaman Home User

3.4. Halaman Isi Materi User

Halaman ini merupakan isi dari materi yang telah di pilih oleh *user*. berfungsi untuk mengetahui isi informasi mengenai *tenses* yang telah di pilih oleh *user*. format informasi dari materi tersebut adalah definisi dari *tenses*, rumus, keterangan, dan contoh kalimat.



Gambar 8. Halaman Isi Materi User

3.5. Halaman Latihan User

Halaman ini memuat beberapa pertanyaan mengenai soal-soal latihan *tenses* untuk dikerjakan dan diselesaikan oleh *user*. halaman ini berfungsi untuk mengetahui sejauh mana kemampuan *user* dalam memahami materi *tenses* yang telah di pelajari. Adapun tampilan halaman isi latihan soal seperti gambar berikut :



Gambar 9. Halaman Latihan User

3.6. Halaman Jawaban dan Kunci Jawaban

Halaman ini menampilkan jawaban *user* beserta kunci jawaban *admin*. Sehingga *user* dapat mengetahui kesalahan dari jawabannya. Adapun rancangan halaman isi latihan seperti gambar berikut :



Gambar 10. Halaman Jawaban dan Kunci Jawaban

3.7. Halaman Nilai

Halaman ini berfungsi untuk mengetahui nilai yang didapat oleh *user* ketika menjawab pertanyaan yang diberikan oleh *admin*. Adapun tampilan halaman nilai seperti gambar berikut :



Gambar 11. Halaman Nilai

3.8. Halaman Administrator

Setelah *admin* melakukan *login* dengan memasukkan *id admin* serta *password* terlebih dahulu. Pada menu *admin* ini terdapat beberapa link yaitu *input* materi dan soal latihan untuk di *update* oleh *admin*, dimana masing-masing menu memiliki *link* tersendiri yang berguna untuk *admin* menginputkan data-data yang dibutuhkan pada sistem ini. Adapun tampilan halaman *Admin* seperti gambar berikut :



Gambar 12. Halaman Admin

4. KESIMPULAN

Berdasarkan hasil pengamatan dan analisa yang telah dilakukan pada bab sebelumnya maka dapat ditarik kesimpulan, diantaranya adalah sebagai berikut :

1. Rancangan aplikasi pengajaran bahasa Inggris berbasis *web* ini bertujuan untuk memperkenalkan salah satu sistem pengajaran teknologi informasi yang lain bagi para penggunanya, dengan tidak mengabaikan sistem pengajaran yang telah ada sebelumnya.
2. Dengan adanya metode pengajaran seperti ini, diharapkan semua penggunanya dapat dengan mudah dan cepat mempelajari *tenses*.
3. Dengan adanya metode pengajaran ini penggunanya bisa belajar tanpa mengenal batas ruang dan waktu.
4. Membantu para penggunanya untuk mendapatkan materi maupun informasi lain mengenai *structure* dan khususnya *writing*.

DAFTAR RUJUKAN

Hariyanto, Bambang. (2004). *Rekayasa Sistem Berorientasi Objek*. Bandung: Informatika Bandung.

Hendra, *Konsep OOAD*.

(<http://dewahendra.blogspot.com/2010/04/i.html>) Diakses pada 14 Desember 2011

(<http://BugiSatriaPengertianERDdanDFDdanContoh.htm>).Diakses pada 15 Maret 2012

Nugroho, Bunafit. (2009). *Latihan Membuat Aplikasi Web PHP dan MySQL dengan Dreamweaver MX (6,7,2004) dan 8*. Yogyakarta: Gava Media.

Yuhefizar (2011). *Mudah Belajar Membuat Web*. Yogyakarta: Gava Media.