

III. METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Waktu : September 2013 – Februari 2015

Tempat : Laboratorium Teknik Permodelan dan Simulasi Teknik
Elektro Universitas Lampung.

Tabel 3.1. *Time Table*[illegible]

3.2. Perangkat yang Digunakan

Tabel 3.2. Alat dan Bahan

Alat dan Bahan	Jumlah	Fungsi
Komputer Spesifikasi : AMD Dual-core 1,65 Ghz, DDR3 2 GB, HDD 320 GB, VGA AMD Radeon HD6320	1 Unit	Sebagai media perancangan dan pengujian <i>software</i> .
Visual Basic 6	1 Paket	Sebagai <i>software</i> yang digunakan untuk merancang dan membangun <i>software</i>

3.3. Prosedur Kerja

Pengembangan perangkat lunak mempunyai langkah-langkah yang terstruktur agar sistem yang dihasilkan sesuai dengan permintaan. Langkah-langkah tersebut dijelaskan pada suatu metode. Salah satu metode dalam pengembangan perangkat lunak adalah model *modified waterfall*. Model ini pertama kali muncul pada tahun 1970 yang diperkenalkan oleh Winston W. Royce (Itb, 2008). *Modified waterfall* merupakan metode yang sering disebut juga dengan siklus kehidupan klasik atau model air terjun. Pembuatan aplikasi pembelajaran kanji secara interaktif ini menggunakan salah satu metode ini. Model *modified waterfall* merupakan paradigma rekayasa perangkat lunak yang paling tua dan paling banyak dipakai. Model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Model *modified waterfall* dibagi menjadi beberapa tahap dalam pengembangannya. Setiap tahap mendefinisikan suatu kegiatan yang harus dikerjakan dan merupakan bagian dari pengembangan sistem. Jika pengerjaan pada suatu tahap telah selesai, baru dapat melanjutkan ketahap berikutnya. Berikut adalah model *modified waterfall* :

Tahap-tahap pada model sekuensial linear adalah sebagai berikut :

3.3.1. Investigasi

Untuk memulai pembuatan suatu sistem yang baru, harus dimulai dari awal dengan cara mengumpulkan informasi yang selengkap-lengkapnyanya. Khususnya jika ada kelemahan-kelemahan dari sistem lama, maka sistem baru yang dibuat harus dapat menambal kelemahan sistem lama tersebut. Dari data yang dikumpulkan belum ada *software* pembelajaran kanji dengan menggunakan bahasa Indonesia dan *software* yang beredar di internet adalah aplikasi kanji yang berupa kamus seperti zkanji dan tagaini jisho. Dengan melihat *software* yang sudah ada, maka dibuat sebuah *software* kanji dari sisi pembelajaran.

Software Requirement and Specification

1. Pendahuluan

1.1 Tujuan

Dokumen SRS ini menjelaskan requirement untuk mengimplementasikan fitur-fitur pada *software* pembelajaran kanji.

1.2 Ruang lingkup perangkat lunak

Dokumen ini akan mengacu pada penggunaan perangkat lunak pembelajaran kanji

1.3 Daftar definisi dan singkatan

Dapat dilihat pada daftar Istilah

1.4 Referensi

Buku *basic kanji book* 1 dan 2

1.5 Overview SRS

Dokumen ini juga mendeskripsikan *software* pembelajaran kanji, secara global, asumsi dalam pengembangan, kebutuhan spesifik dan kebutuhan tambahan.

2. Deskripsi Umum

Software pembelajaran kanji secara interaktif ini bertujuan untuk membantu orang Indonesia yang sedang mempelajari bahasa Jepang untuk mempelajari kanji dan mengetahui penerapan dan cara menggunakan kanji tersebut. Pengguna *software* ini adalah orang Indonesia yang telah menguasai atau lulus bahasa Jepang dasar tingkat I.

2.1 Prespektif Produk

Software pembelajaran kanji adalah alat bantu *system* *rujukan* untuk mempermudah seseorang dalam mempelajari kanji. Perangkat lunak ini memiliki fungsi pembelajaran kanji, melihat daftar kanji, dan uji hafalan kanji. Perangkat lunak ini mampu memberikan contoh atau penerapan kanji dalam sebuah kalimat ataupun wacana, bila pengguna *software* ini disiplin dan mengerjakan perintah-perintah yang ada, maka user pasti akan *system* menjawab soal-soal pada *software* ini, sehingga pengguna tidak hanya hafal namun juga mengerti dan paham terhadap kanji tersebut.

2.2 Fungsi-fungsi produk

1. Mulai belajar : adalah membuka fitur pembelajaran kanji yang terdiri dari 10-15 kanji peningkatan, dan akan diberikan soal-soal untuk membantu dalam mempelajari kanji.

2. Lihat daftar kanji : adalah melihat daftar kanji yang telah dipelajari sehingga ketika kita lupa dapat mengingatnya kembali
3. Uji hafalan kanji : menguji kun yomi dan onyomi dari kanji yang telah dipelajari.

2.3 Karakteristik pengguna

Pengguna *software* ini harus sudah menguasai huruf kana yaitu katakana dan hiragana. Pengguna juga harus menguasai bahasa Jepang tingkat dasar seperti penggunaan partikel, susunan kalimat, dan perubahan bentuk kata kerja dasar.

2.4 Batasan umum

Perangkat lunak ini akan menggunakan Visual Basic sebagai *software* pengembangannya, dan hanya akan menampilkan *interface* cara menulis kanji dan hanya *system input* pada form soal dan uji hafalan.

2.5 Asumsi dan ketergantungan

Dalam pengembangan *software* pembelajaran kanji ini diasumsikan bahwa tools pengembang telah tersedia dimana tools itu adalah *software* dan hardware yang dibutuhkan. Pengguna *software* ini hanya perlu menginstal *software* ini di *system* mereka dengan menggunakan *system operasi windows*.

3. Kebutuhan Spesifik

3.1 Kebutuhan Antarmuka

Desain interface dapat dilihat pada bab 3

3.2 Kebutuhan Fungsional

DA, STD dan DFD dapat dilihat pada bab 3

3.3 Unjuk Kerja

User akan memilih tingkatan yang akan dipelajari, ketika user memilih mulai belajar maka akan ditampilkan 10-15 kanji berikut arti, cara baca, cara menulis, dan contoh kata. Setelah itu user akan diberikan soal berupa meng input cara baca dengan hiragana dan memilih jawaban dari pilihan ganda. User dapat melihat database kanji sesuai tingkatan, user juga dapat menguji hafalannya dengan menginput cara baca nya dengan hiragana dan katakana.

3.4 Batasan Perancangan

Software ini hanya memberikan informasi cara penulisan kanji, cara baca, dan arti. Untuk melatih menulis diharapkan user dapat melatih nya sendiri, karena *software* ini tidak menyediakan tools menulis kanji.

3.5 Kebutuhan lain

Perancangan database dapat dilihat di bab 3, Sistem operasi yang dibutuhkan adalah microsoft windows.

3.3.2 Analisis Kebutuhan

Salah satu tujuan dari analisis adalah untuk mengumpulkan *requirement* apa saja yang dibutuhkan untuk membangun sistem ini. Termasuk di dalamnya studi literatur dimaksudkan untuk mempelajari berbagai sumber referensi atau teori (buku dan internet) yang berkaitan dengan perancangan sistem dan teknologi yang digunakan di dalamnya. Dalam tahap ini perlu dilakukan pengumpulan informasi dari metode

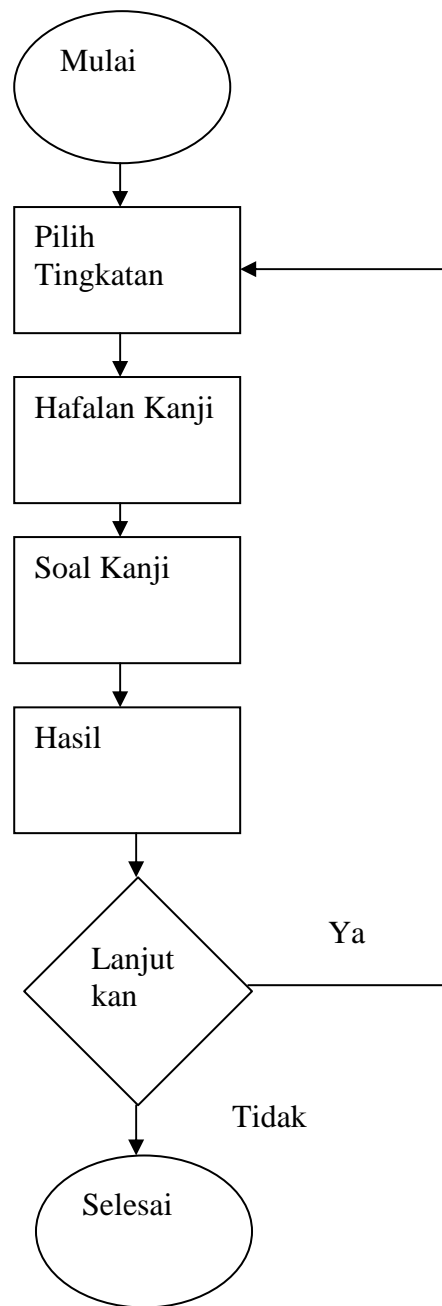
pembelajaran kanji hingga daftar-daftar kanji itu sendiri. Informasi kanji juga didapat dari *software* kamus kanji seperti Zkanji dan Tagaini jisho. Untuk bahasa Jepang sendiri referensi yang digunakan berasal dari buku Bahasa Jepang Modern dan kamus Jepang- Indonesia Goro Taniguchi.

Penelitian ini juga menggunakan beberapa buku referensi baik berupa e-book maupun berupa buku literatur untuk mempelajari metode pembelajaran kanji. Di antara sumber referensi yang akan digunakan adalah metode pembelajaran dari buku *Basic kanji book*.

Setelah menganalisa data kemudian mulai membuat Diagram alir , *Context Diagram*, *Data Flow Diagram*, *State Transition Diagram*, Kamus diagram dan *database* yang diperlukan sistem nantinya.

a. Diagram Alir

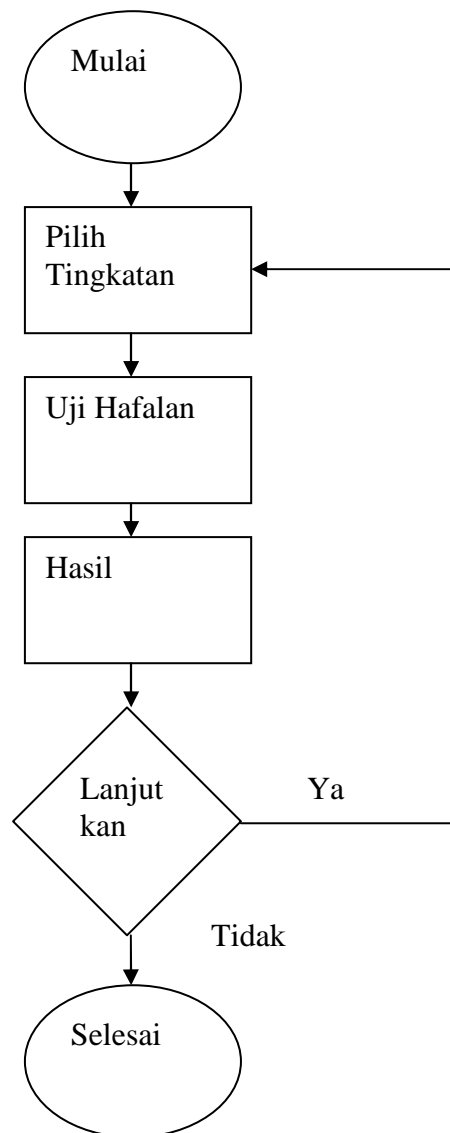
Langkah awal untuk melakukan analisis dan perancangan dari perangkat lunak ini adalah membuat diagram alir, yang merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak, beserta urutannya dengan menghubungkan masing masing langkah tersebut menggunakan tanda panah. Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut. Pada gambar berikut ini akan diuraikan diagram alir yang sudah dirancang pada gambar 16, 17, dan 18 berikut ini.



Gambar 3.2. Diagram alir menu mulai belajar

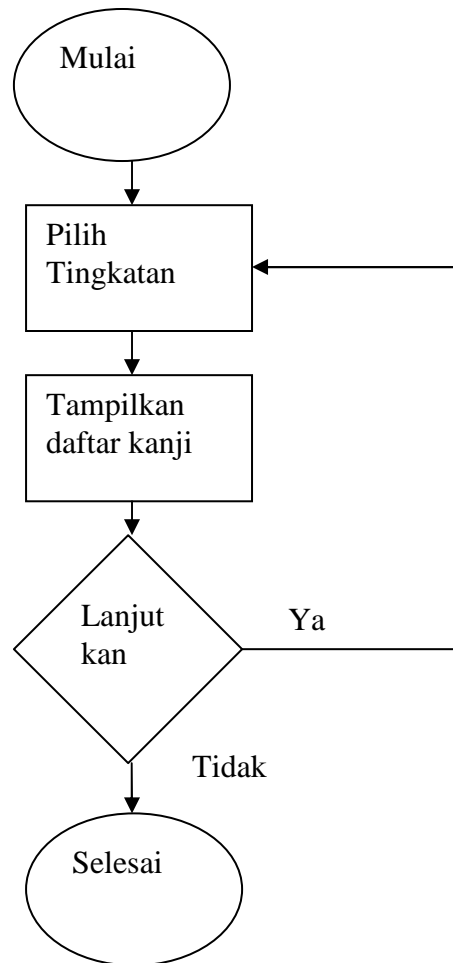
Pada gambar 3.2 digambarkan sebuah diagram alir proses menu mulai belajar. Proses pembelajaran dibagi menjadi tiga, yaitu persiapan, pelaksanaan dan penilaian. Pada proses persiapan *user* akan memilih terlebih dahulu tingkatan yang akan dipelajari, proses pelaksanaan *user* akan diberikan hafalan-hafalan kanji sesuai tingkatan yang

telah dipilih, proses penilaian *user* akan diberikan soal untuk menilai tentang pemahaman *user* pada kanji yang telah dipelajari.



Gambar 3.3. Diagram alir pada menu uji hafalan

Pada gambar 3.3 yaitu diagram alir proses uji hafalan kanji, di mana *user* akan memilih tingkatan untuk menguji hafalan kanji nya. Pada proses akan ditampilkan tabel uji hafalan setelah itu akan diberikan koreksi atau jawaban yang sesuai.



Gambar 3.4. Diagram alir menu lihat kanji

Pada gambar 3.4 dijelaskan diagram alir proses penyajian daftar kanji, dimana pada proses ini *user* akan memilih tingkatan untuk membuka daftar kanji sesuai tingkatan kanji. Pada aktivitas daftar kanji *database* akan menampilkan daftar kanji sesuai dengan tingkatan yang dipilih.

b. *Data Flow Diagram*

Langkah awal dalam perancangan sistem ini adalah pembuatan sistem *Context Diagram*. *Context Diagram* ini merupakan gambaran awal dari sistem pembelajaran kanji secara umum, yang menggambarkan sistem beserta hubungannya dengan lingkungan luar dan bagaimana sistem ini berinteraksi.

c. *State Transition Diagram*

Langkah yang diambil berikutnya dalam analisis dan perancangan adalah membuat STD (*State Transition Diagram*) untuk mengetahui *state* apa saja yang akan berjalan pada proses analisis dan perancangan ini. STD itu sendiri merupakan diagram yang memodelkan tingkah laku (*behaviour*) sistem berdasarkan pada definisi satu bagian dari keadaan sistem. STD sering dipakai untuk menggambarkan kinerja sistem.

d. *Data Dictionary*

Setelah membuat STD maka dalam rangka melakukan analisis dan perancangan, maka akan dibuat DD (*data dictionary*) untuk lebih memperjelas dalam pendeskripsian arus data yang mengalir pada DFD yang telah dirancang.

3.3.3 Perancangan

Pada tahap ini proses perancangan akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan. Proses ini berfokus pada pembuatan rancangan arsitektur perangkat lunak yang terdiri atas perancangan *database* yang diperlukan, perancangan proses dan perancangan *interface software* yang nantinya akan menghasilkan sebuah sistem secara keseluruhan.

a. Perancangan *database*

Perancangan *database* pada perangkat lunak pembelajaran kanji ini dapat dikatakan sebagai kelanjutan dari tahap analisis pada metode *modified waterfall* dengan cara mencari dan membangun segala sesuatu yang berhubungan dengan

sistem yang dibuat. Pada tahap ini pembuatan terfokus pada pembuatan daftar-daftar kanji yang berupa karakter kanji itu sendiri, arti, carabaca kun dan on nya yang telah diambil dari buku referensi sebelumnya. Pengambilan data juga dilakukan dengan melakukan pemotongan gambar coretan-coretan tahapan pembentukan karakter kanji yang terdapat pada buku referensi. Setelah proses tersebut terselesaikan, setidaknya dapat menjawab pertanyaan seperti dibawah ini :

1. Informasi apa saja input yang diperlukan dalam pembangunan database?

Informasi berupa karakter kanji, arti/maknanya, contoh kanji yang diterapkan pada kata, proses coretan kanji, cara baca kun dan on kanji. Soal-soal yang diambil dari buku referensi *basic kanji book*.

2. Bagaimana perkembangan sistem di masa mendatang?

Pada perkembangan sistem di masa depan diharapkan *software* ini akan berkembang dengan dilakukannya penambahan karakter kanji sesuai dengan jumlah kanji Jepang yang telah distandarkan oleh Pemerintah Jepang.

1. Perancangan Database Fisik

Perancangan *database* ini merupakan perancangan dari tabel-tabel yang digunakan dalam pembuatan program atau sistem nantinya.

2. Perancangan ERD (*Entity Relationship Diagram*)

Perancangan ERD adalah hal yang penting dalam merancang database dari sebuah sistem. ERD dapat mendeskripsikan hubungan antara tabel yang akan dibuat. Meski pada perangkat lunak ini hanya memiliki satu entity akan tetapi dalam

penerapannya perangkat lunak ini menggunakan kunci utama untuk mengakses tabel kanji tersebut.

b. Perancangan interface

Interface adalah sebuah media yang menghubungkan antara pengguna dengan sistem informasi yang akan dibangun. Sistem yang akan dibangun diharapkan dapat menyediakan *interface* yang mudah dipahami oleh *user* sehingga seorang user dapat langsung memahami arti dari setiap tombol-tombol yang ada pada sistem informasi. Gambar di bawah ini memperlihatkan beberapa rancangan graphical user interface (GUI) pada Perangkat lunak pembelajaran kanji secara interaktif.

1) Perancangan interface menu utama

Header <i>Software</i> Pembelajaran Kanji	
Menu Utama	<div style="border: 1px solid black; padding: 2px 10px; display: inline-block;">Pilih Tingkatan</div> <div style="border: 1px solid black; width: 20px; height: 20px; display: inline-block; vertical-align: middle; text-align: center; line-height: 20px;">↓</div>
<div style="display: flex; justify-content: space-around; margin-bottom: 20px;"> <div style="border: 1px solid black; border-radius: 10px; padding: 5px 20px;">Mulai Belajar !</div> <div style="border: 1px solid black; border-radius: 10px; padding: 5px 20px;">Uji Hafalan</div> <div style="border: 1px solid black; border-radius: 10px; padding: 5px 20px;">Daftar Kanji</div> </div> <div style="border: 1px solid black; border-radius: 10px; padding: 5px 20px; margin-left: auto;">help</div>	
wakakanji1.0 Ferdianza yudha Negara	

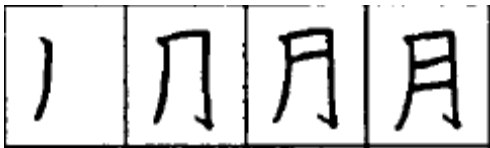
Gambar 3.5. Perancangan *interface* menu utama

Gambar 3.5 menunjukkan perancangan *interface* menu utama. Terdapat beberapa fitur yaitu, pilih tingkatan di mana user dapat memilih tingkatan, mulai belajar dimana akan ditujukan pada fitur pembelajaran, uji hafalan dimana akan ditampilkan uji hafalan pada kanji, daftar kanji dimana akan ditampilkan daftar-daftar kani. Tombol help menyajikan petunjuk penggunaan perangkat lunak beserta info tentang pembuat *software* dan versi dari *software* ini.

2) Perancangan interface hafalan kanji

Kanji	Arti	Kun yomi	On yomi
月	Bulan	つき	ゲツ、ガツ

Coretan 4



月(つき) : bulan

月曜日(げつ.よう.び) : senin

1 月(いち.がつ) : januari

1 (いっ)か月(げつ) : 1 bulan

kembali

Lanjut

Penjelasa materi :

Gambar 3.6. Perancangan *interface* hafalan kanji

Gambar 3.6 merupakan *interface* hafalan kanji di mana akan ada tabel yang terdiri atas karakter kanji, arti, kun-yomi, on-yomi. Terdapat gambar proses coretan kanji dan jumlah coretannya, dan juga beberapa contoh kata dengan menggunakan kanji tersebut. Pada interface ini juga terdapat kolom penjelasan, di mana nantinya setiap

materi kanji berupa teori, tips, dan sebagainya akan dijelaskan pada kolom penjelasan.

3) Perancangan interface soal tahap 1

Tulis Bacaan kanji di bawah ini dengan hiragana.

1. 木 :

2. 車 :

3. 月 :

4. 人口 :

5. 山田さん :

Gambar 3.7. Perancangan *interface* soal tahap 1

Gambar 3.7 adalah gambar *interface* soal tahap 1, di mana akan diberikan karakter kanji dan *user* harus mengisi cara bacanya dengan menggunakan hiragana. Terdapat tombol petunjuk yang menjelaskan tentang bagaimana cara mengerjakan soal dan bagaimana cara melakukan input dengan menggunakan hiragana maupun katakana.

4) Perancangan interface soal tahap 2

Tulis bacaan kanji dibawah ini dengan hiragana

1. きょうは3月15日です。Sekarang tanggal 15 maret

2. あの人は山川さんです。Orang itu adalah tuan yamakawa

3. 山川さんは日本人です。Tuan yamakawa adalah orang jepang

4. あの山は筑(つく)波(ば)山です。Gunung disana adalah gunung tsukuba

5. 川田さんの専(せん)門は車です。Keahlian tuan kawada adalah mobil

Gambar 3.8. Perancangan *interface* soal tahap 2

Gambar 3.8 adalah perancangan *interface* soal tahap 2, dimana akan diberikan kalimat dengan kanji yang telah dipelajari dan *user* harus mengisi cara bacanya dengan hiragana.

5) Perancangan interface soal tahap tiga

Pilihlah kanji yang tepat dari keterangan di bawah ini		
1.pohon き		
車	木	日
2.manusia ひと		
人	口	門
3.bulan つき		
山	川	月
4.mobil くるま		
田	車	口
5.gunung やま		
山	川	田
Kembali	Cek jawaban	Lanjut

Gambar 3.9. Perancangan *interface* soal tahap tiga

Gambar 3.9 adalah gambar perancangan *interface* soal tahap tiga, dimana akan diberikan cara bacanya dan artinya lalu *user* harus memilih diantara tiga kanji yang tersedia. Pada tampilan ini terdapat tombol cek jawaban, tombol ini berfungsi untuk mengecek semua jawaban yang telah di input dan akan dicocokkan dan diperlihatkan kunci jawaban dari setiap soal tersebut.

6) Perancangan *interface* daftar kanji

Daftar kanji :

Kanji	Arti	Kun yomi	On yomi
月	Bulan	つき	ゲツ、ガツ
車	Maobil	くるま	シャ
人	Manusia, orang	ひと	ジン、ニン
山	Gunung	やま	サン

Kembali

Gambar 3.10. Perancangan *interface* daftar kanji

Gambar 3.10 menunjukkan perancangan interface daftar kanji. Pada *interface* ini ditampilkan daftar kanji sesuai dengan tingkatan yang telah dipilih oleh *user*.

7) Perancangan *interface* uji hafalan

Isilah kun yomi dan on yomi dari kanji di bawah ini

Kanji	Arti	Kun yomi	On yomi
月	Bulan		
車	Maobil		
人	Manusia, orang		
山	Gunung		

Kembali

Gambar 3.11. Perancangan *interface* uji hafalan kanji

Gambar 3.11 menunjukkan perancangan *interface* uji hafalan, dimana akan ditampilkan daftar kanji dan artinya, *user* harus mengisi kun yomi dan on yomi nya untuk menguji hafalan. Pada *interface* ini tidak ada tombol cek jawaban, karena jawaban sebenarnya dapat dilihat pada daftar kanji.

3.4 Implementasi

Secara garis besar, pada tahap implementasi dilakukan dua hal untuk membangun sistem ini, yaitu :

1. Menyiapkan daftar-daftar kanji sesuai dengan tingkatan-tingkatan
2. Melakukan pengkodean. Pengkodean merupakan proses menerjemahkan desain ke dalam suatu bahasa yang bisa dimengerti oleh komputer. Pada tahap pengkodean, semua daftar kanji dibuat database yang sudah terdapat pada Microsoft Visual Basic 6 untuk memungkinkan pembuatan *user interface* program dan membuat suatu *coding* terhadap database agar dimengerti oleh computer dan dapat ditampilkan sesuai dengan perancangan *interface*. Pengkodean menggunakan bahasa VB dan menghasilkan perangkat lunak yang dapat disesuaikan dengan kebutuhan.

3.5. Pengujian dan Pemeliharaan

a. Pengujian

Strategi untuk pengujian perangkat lunak mengintegrasikan metode design test case perangkat lunak kedalam sederetan langkah yang direncanakan dengan baik,

dan hasilnya adalah konstruksi perangkat lunak yang berhasil. Semua strategi pengujian harus menggabungkan perencanaan pengujian, design test case, dan kumpulan data resultan serta evaluasi.

Pengujian juga dilakukan pada metode pembelajaran bahasa Jepang, mengingat bahwa *software* ini merupakan *software* pembelajaran, maka pengujian metode pembelajaran sangat penting,

1. Strategi Pengujian

Pada dasarnya strategi pengujian perangkat lunak memiliki karakteristik umum sebagai berikut :

1. Pengujian dimulai pada tingkat module dan kerja 'keluar' kearah integrasi dari system yang berbasis computer.
2. Teknik pengujian yang berbeda sesuai pada titik waktu yang berbeda.
3. Pengujian dilakukan oleh pengembang perangkat lunak dan suatu kelompok pengujian yang independent.
4. Pengujian dan debugging merupakan aktifitas yang berbeda, tetapi debugging harus diakomodasi pada banyak stretegi pengujian.
5. Pengujian metode pembelajaran dengan mempelajari metode pembelajaran yang sudah ada pada buku-buku kanji dan terapan pada universitas yang memiliki jurusan bahasa jepang.
6. Pengujian dilakukan dengan beberapa responden dan disertai pengisian angket untuk menguji metode pembelajaran *software*.

2. Verifikasi dan Validasi

Pengujian perangkat lunak adalah satu element dari topic yang lebih luas yang sering diacu sebagai verifikasi dan validasi.

Verifikasi mengacu pada rangkaian aktifitas yang memastikan bahwa perangkat lunak secara tepat mengimplementasikan suatu fungsi tertentu.

Validasi mengacu pada serangkaian aktifitas yang berbeda yang memastiyikan bahwa perangkat lunak yang dibangun dapat ditelusuri kepersyaratan pelanggan.

3. Debugging

Debugging terjadi sebagai akibat pengujian yang berhasil. Jika test case mengungkap kesalahan, maka debugging adalah proses yang menghasilkan penghilangan kesalahan. Debugging bukan merupakan pengujian tetapi selalu terjadi sebagai bagian akibat dari pengujian. Proses debugging dimulai dengan eksekusi terhadap suatu test case. Hasilnya dinilai, dan ditemukan kurangnya hubungan antara harapan dan sesungguhnya. Dalam banyak kasus, data yang tidak berkaitan merupakan gejala dari suatu penyebab pokok tetapi masih tersembunyi, sehingga perlu adanya koreksi kesalahan.

Proses debugging akan selalu memiliki salkah satu dari dua hasil akhir berikut :

1. Penyebab akan ditemukan, dikoreksi, dan dihilangkan.
2. Penyebab tidak akan ditemukan

dalam kasus yang terakhir, orang yang melakukan debugging mungkin mencurigai suatu penyebab, kemudian mendesain suatu test case untuk membantu

membuktikan kecurigaannya, dan bekerja untuk koreksi kesalahan dengan gaya yang iterative.

b. Pemeliharaan

Pada tahapan ini hal yang dilakukan adalah mencakup koreksi *error* yang tidak diinginkan dan tidak ditemukan pada tahap-tahap terdahulu. Seperti mengkoreksi terjemahan ke bahasa Indonesia, apaka cocok atau sesuai dalam konteks kata yang sedang diterjemahkan. Mengkoreksi kunci jawaban, dan menyediakan alamat email atau kontak untuk ikut berpartisipasi dalam menyempurnakan perangkat lunak pembelajaran kanji ini. Semua pengkoreksian akan di upload menjadi versi berikutnya. Sementara dalam penelitian selanjutnya tidak menutup kemungkinan akan dibuat mencapai 2000 kanji dalam perilisan versi kedepannya.