

APLIKASI KAMUS BAHASA JEPANG - INDONESIA PADA PERANGKAT GENGAM MENUNAKAN J2ME

Tommy Adhi Kresna Murthi¹, Aghus Sofwan, S.T, M.T², Eko Handoyo, S.T, M.T³
Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro

Abstrak - Era globalisasi dan kemajuan negara Jepang yang sangat pesat dalam bidang industri telah membuat bahasa Jepang menjadi salah satu bahasa asing yang sangat penting untuk dipelajari selain bahasa Inggris sebagai bahasa internasional. Dengan kemajuan teknologi saat ini, dimungkinkan untuk dibuat aplikasi kamus digital yang akan memiliki berbagai kelebihan dan kemudahan dibandingkan dengan kamus konvensional.

Program dibuat dengan perangkat lunak J2ME dan berjalan pada perangkat genggam yang mendukung aplikasi Java.

Dalam pembuatan aplikasi ini terlebih dahulu dibuat daftar kata bahasa Jepang beserta artinya dan sebaliknya dalam bentuk file berekstensi .txt yang akan digunakan sebagai basis data serta dibuat huruf Kanji dengan file gambar yang memiliki format PNG. Aplikasi ini dapat berjalan pada perangkat genggam (ponsel) yang mendukung aplikasi java sehingga user akan mempunyai keleluasaan dan mobilitas dalam memanfaatkan aplikasi ini sebagai alat untuk mempermudah dalam mempelajari bahasa Jepang.

Kata kunci : kamus digital bahasa jepang, J2ME, perangkat genggam, aplikasi Java.

I. Pendahuluan

1.1 Latar Belakang

Pada era globalisasi seperti sekarang ini, bahasa sebagai salah satu alat untuk berkomunikasi merupakan sesuatu yang sangat penting. Dengan menguasai bahasa asing maka dapat meningkatkan kualitas daya saing seseorang. Oleh karena itu, perlunya pengetahuan ataupun pemahaman mengenai bahasa asing selain dari bahasa nasional telah menjadi sesuatu yang tidak dapat dihindari. Selain Bahasa Inggris yang merupakan bahasa internasional yang umum digunakan di dunia, Bahasa Jepang telah menjadi salah satu bahasa asing yang perlu untuk dipelajari dalam menyambut era globalisasi, hal ini dikarenakan oleh pertumbuhan negara Jepang yang sangat pesat di segala bidang khususnya di bidang industri.

Salah satu media dalam pembelajaran bahasa adalah kamus yang dapat digunakan untuk mencari arti dari kata – kata yang tidak diketahui. Saat ini dengan adanya kemajuan teknologi di bidang informatika dan komputer, dimungkinkan untuk dibuat sebuah kamus digital yang dapat menggantikan fungsi dari kamus konvensional dengan menyediakan kelebihan dan kemudahan dibandingkan dengan kamus konvensional. Seperti proses pencarian yang lebih mudah, dengan hanya memasukkan kata yang dicari, maka program akan mencari dan menampilkan arti dari kata tersebut secara otomatis tanpa harus mencarinya secara manual. Selain itu, karena aplikasi kamus ini akan berjalan pada perangkat genggam, maka akan memudahkan untuk dibawa sehingga dapat digunakan kapanpun dibutuhkan.

Pada tugas akhir ini akan digunakan perangkat lunak J2ME (Java 2 Micro Edition) yang merupakan subset dari J2SE (Java 2 Standard Edition) untuk membuat kamus Bahasa Jepang – Indonesia untuk dapat dijalankan dan digunakan pada perangkat genggam yang mendukung aplikasi Java.

1.2 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah untuk membuat sebuah aplikasi kamus Bahasa Jepang yang dapat dijalankan pada perangkat genggam yang mendukung aplikasi Java.

1.3 Batasan Masalah

Hal-hal yang akan dibahas pada penulisan Tugas Akhir nanti, dibatasi pada:

1. Pembuatan aplikasi kamus Bahasa Jepang dengan menggunakan perangkat lunak J2ME (Java 2 Micro Edition).
2. Perangkat genggam yang digunakan dalam implementasi dan pengujian aplikasi dibatasi hanya pada ponsel yang mendukung aplikasi Java.
3. Pencarian arti hanya dibatasi pada pencarian arti kata bukan kalimat dan dalam bentuk dasar (*futsu-kei*) serta dalam huruf romaji.

II. Pemrograman J2ME

2.1 Pengetahuan Bahasa Jepang

Dalam mempelajari bahasa Jepang tidak ada cara khusus dalam mempelajarinya karena dapat dipelajari seperti mempelajari bahasa – bahasa lainnya yang ada. Yang membedakan bahasa Jepang dengan bahasa – bahasa lainnya adalah jenis huruf yang digunakan. Bahasa Jepang mempunyai empat bentuk huruf yang harus diketahui yaitu Kanji, Hiragana, Katakana dan Romaji

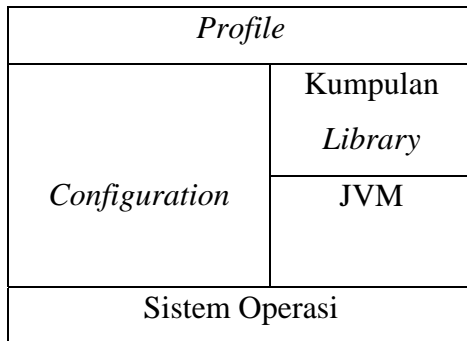
2.2 J2ME

Java 2 Micro Edition (J2ME) merupakan salah satu bagian dari teknologi Java yang dikembangkan untuk memungkinkan aplikasi – aplikasi Java bisa diimplementasikan pada peralatan *embedded system* dan *handheld* (perangkat genggam) yang tidak mampu mendukung secara penuh implementasi menggunakan J2SE.

J2ME sangat berguna untuk membangun sebuah aplikasi pada peralatan dengan jumlah memori dan kapasitas penyimpanan yang terbatas, serta *user interface*

yang terbatas seperti pada perangkat komunikasi bergerak berupa telepon genggam, *Personal Digital Assisstant* (PDA) dan sebagainya. Seperti aplikasi java umumnya yang menggunakan JVM, dalam J2ME digunakan pula *virtual machine* yang disebut *K virtual machine*. *K virtual machine* adalah *virtual machine* yang sangat kecil dalam kebutuhan memorinya. Huruf K dalam *K virtual machine* adalah singkatan dari *kilobyte*.

Dalam J2ME dibagi menjadi dua bagian yang dikenal dengan istilah *configuration* dan *profile*. Arsitektur J2ME dapat dilihat pada gambar 2.1.



Gambar 2.1. Arsitektur J2ME

Configuration mendefinisikan lingkungan kerja J2ME *runtime*. Karena setiap perangkat genggam memiliki fitur – fitur yang berbeda – beda, *configuration* dirancang untuk menyediakan *library* standar yang mengimplementasikan fitur standar dari sebuah perangkat genggam. Sedangkan *profile* menyediakan implementasi – implementasi tambahan yang sangat spesifik dari sebuah perangkat genggam.

Profil dan konfigurasi yang paling populer yang disediakan oleh Sun adalah *Mobile Information Device Profile* (MIDP) dan *Connected Limited Device Configuration* (CLDC). CLDC digunakan untuk perangkat dengan konfigurasi yang terbatas, sebagai contoh perangkat yang hanya mempunyai memori 128 hingga 512KB yang tersedia untuk aplikasi-aplikasi Java. Sehingga JVM yang disediakan sangat terbatas dan hanya mendukung sejumlah kecil kelas java tradisional.

2.3 MIDlet

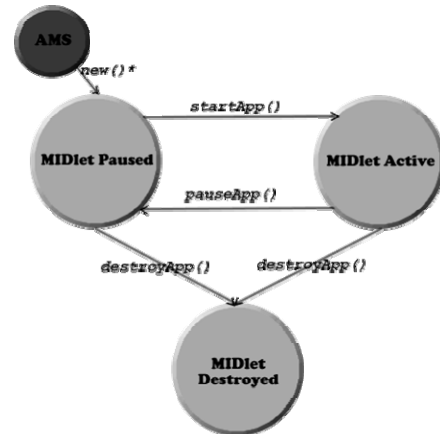
MIDlet adalah aplikasi yang ditulis untuk MIDP. Aplikasi MIDlet adalah bagian dari kelas `javax.microedition.midlet.MIDlet` yang didefinisikan pada MIDP. Kelas ini merupakan kelas utama dalam sebuah MIDlet, yaitu semua aplikasi MIDlet harus merupakan turunan dari kelas ini. Kelas ini merupakan kelas abstrak, sehingga kelas turunannya harus mengimplementasikan fungsi – fungsi abstrak di dalamnya.

2.3.1 Siklus Hidup MIDlet

Perangkat *mobile* berinteraksi dengan MIDlet menggunakan perangkat lunaknya sendiri, yang disebut *Application Management Software* (AMS). AMS merupakan perangkat lunak atau program bawaan yang ada dalam perangkat genggam tempat dimana suatu MIDlet berjalan yang mengatur manajemen aplikasi MIDlet yang

ada. AMS bertanggung jawab dalam menginisialisasi, memulai, menghentikan sementara, melanjutkan, dan memusnahkan MIDlet. Dalam melakukan manajemen MIDlet ini, suatu MIDlet dapat berada dalam tiga keadaan yang dikendalikan melalui metode kelas MIDlet, yaitu *active*, *paused* dan *destroyed*.

Jika digambarkan, perubahan status dalam daur hidup sebuah MIDlet adalah seperti yang ditunjukkan pada Gambar 2.2.



* - creates new MIDlet instance using the MIDlet's no args constructor

Gambar 2.2 Siklus hidup MIDlet

2.3.1.1 Paused

Status *paused* terjadi ketika MIDlet selesai diinisialisasi dan tidak melakukan aksi apapun. Secara garis besar status MIDlet adalah *paused* pada saat:

- Setelah MIDlet dibuat dengan konstruktor `new()`
- Dari status *active*, kemudian berhasil memanggil fungsi `MIDlet.pauseApp()`
- Dari status *active*, kemudian berhasil memanggil fungsi `MIDlet.notifyPaused()`
- Dari status *active*, namun ketika akan `start()` terjadi kesalahan berupa `exception MIDletStateChangeException`

2.3.1.2 Active

Status *active* ini terjadi ketika MIDlet sedang aktif / berjalan dengan normal, yaitu setelah memanggil fungsi `MIDlet.startApp()`

2.3.1.3 Destroyed

Status *destroyed* ini terjadi ketika MIDlet berhenti berjalan (identik dengan *exit*), sehingga seluruh sumber daya yang digunakan akan dibebaskan. Status ini terjadi ketika berhasil dilakukan pemanggilan fungsi `MIDlet.destroyApp()` atau `MIDlet.notifyDestroyed()`.

2.3.2 Eksekusi Sebuah MIDlet

Umumnya urutan eksekusi sebuah MIDlet adalah sebagai berikut:

- AMS menginisialisasi sebuah objek MIDlet, sehingga MIDlet berada pada status *Paused*.

- Ketika AMS memutuskan bahwa MIDlet sudah waktunya dijalankan, maka AMS memanggil fungsi `MIDlet.startApp()` sehingga MIDlet akan mulai aktif berfungsi.
- Ketika AMS memutuskan bahwa MIDlet harus dinonaktifkan sementara, maka AMS akan memanggil fungsi `MIDlet.pauseApp()` sehingga MIDlet akan berhenti bekerja.
- Ketika AMS memutuskan bahwa MIDlet harus dinonaktifkan selamanya, maka AMS akan memanggil fungsi `MIDlet.destroyApp()` sehingga alokasi memori yang digunakan oleh MIDlet yang bersangkutan akan dibersihkan.

2.4 Pemrograman GUI (*Graphical User Interface*) pada J2ME

Berbeda dengan sebuah komputer, perangkat genggam semacam telepon genggam, *Palm*, dan *organizer* memiliki berbagai keterbatasan, misalnya ukuran layar yang jauh lebih kecil dan masukan input yang tidak berupa *mouse*. Keterbatasan – keterbatasan ini menyebabkan perlunya ada teknik pemrograman GUI yang berbeda dengan teknik pemrograman yang umumnya digunakan pada aplikasi – aplikasi yang dijalankan pada sebuah komputer. Fungsi – fungsi untuk pemrograman GUI pada J2ME ditangani oleh MIDP.

2.4.1 Jenis Fungsi – Fungsi API pada MIDP untuk GUI

Fungsi – fungsi untuk pembuatan antarmuka berbasis *window* (GUI) yang disediakan oleh MIDP terbagi atas dua level, yaitu *highlevel* dan *lowlevel*. Semua fungsi – fungsi untuk antarmuka berbasis *window* ini ditangani oleh paket `javax.microedition.lcdui`.

Pada level pemrograman yang lebih tinggi (*highlevel*) yang merupakan level pemrograman yang dianjurkan, aplikasi MIDlet yang dibuat akan lebih portabel dengan banyak perangkat genggam yang berbeda. Kelas – kelas untuk manajemen pemrograman yang lebih tinggi akan diturunkan dari `javax.microedition.lcdui.Screen`. Aplikasi kamus bahasa Jepang – Indonesia pada perangkat genggam ini menggunakan pemrograman *high level* untuk pembuatan antarmukanya.

Pada level pemrograman yang lebih rendah (*lowlevel*), fungsionalitas yang didapatkan akan lebih spesifik ke jenis perangkat genggam yang digunakan. Hal ini memungkinkan kurangnya portabilitas MIDlet yang dibuat.

2.4.2 Model Pemrograman GUI pada MIDP

Untuk menampilkan sesuatu pada layar, misalnya tulisan, form, dan sebagainya harus mengakses *display* dari perangkat genggam yang dapat diakses dengan kelas `javax.microedition.lcdui.Display`. Objek kedua adalah *screen*, yang berbeda dengan *display*. *Screen* digunakan untuk menangani komponen – komponen GUI, misalnya *window*, form, *dialogbox*, dan menangani masukan dari pengguna misalnya dari *keypad*. Objek *screen* ini direpresentasikan dengan kelas

`javax.microedition.lcdui.Screen`. Pada setiap waktu hanya satu objek *screen* yang bisa ditampilkan oleh objek *Display*, untuk menampilkan digunakan fungsi `setCurrent()` pada objek *Display*.

Ada tiga tipe *screen* pada MIDlet, yaitu:

- *Complex Screen Object*
Tipe ini mengatur komponen – komponen GUI yang sudah tidak bisa ditambahi oleh komponen *screen* yang lain. Misalnya, *screen* yang berisi objek *ListBox* untuk meminta masukan *user*, tidak akan bisa kemudian diisi dengan objek lain.
- *Generic Screen Object*
Tipe ini mengatur komponen – komponen GUI Form, yang bisa ditambahi oleh komponen *screen* yang lain. Misalnya, *screen* yang berisi objek Form yang kemudian bisa menambahkan komponen – komponen lain pada Form tersebut.
- *Low Level Screen Object*
Tipe *screen* yang digunakan oleh subkelas `javax.microedition.lcdui.Graphics` dan `javax.microedition.lcdui.Screen`

2.4.3 Bekerja Dengan Display

Kelas *Display* merupakan kelas yang menyediakan fungsi – fungsi untuk manajemen layar pada perangkat genggam, menampilkan objek *screen*, dan menyediakan informasi tentang properti dari perangkat genggam yang digunakan. Akses ke layar dapat diacu dengan fungsi statik `getDisplay()` pada kelas *Display*

```
public static Display getDisplay(MIDlet m)
```

Penggunaan fungsi ini umumnya dilakukan di dalam fungsi `startApp()`. Setelah mendapatkan objek *Display* dengan fungsi `getDisplay()`, maka digunakan fungsi `setCurrent()` untuk menentukan objek *screen* mana yang akan ditampilkan.

```
public void setCurrent(Displayable YourScreen)
```

2.4.4 Bekerja Dengan Screen

Objek *screen* menyediakan fungsionalitas untuk interaksi antara pengguna dengan perangkat genggam. Objek *screen* mempunyai empat jenis objek turunan, yaitu *TextBox*, *Alert*, *List*, dan *Form*. Kelas `javax.microedition.lcdui.Screen` merupakan kelas abstrak dengan dua karakteristik, yaitu objek *screen* bisa memiliki *title* dan *ticker*.

2.4.4.1 TextBox

Kelas *TextBox* merupakan kelas turunan dari kelas abstrak `javax.microedition.lcdui.screen`. *TextBox* menyediakan media untuk menerima masukan teks dari *keypad* pengguna perangkat genggam. Konstruktors dari kelas *TextBox* adalah:

```
public TextBox(String title, String text, int maxSize, int constraints)
```

2.4.4.2 List

Kelas *List* (`javax.microedition.lcdui.List`) merupakan kelas turunan dari kelas abstrak

`javax.microedition.lcdui.Screen` yang menyediakan masukan pilihan (*multiple choice*) pada layar. Konstruktor dari kelas `List` ada dua, yaitu:

```
public List(String title, int listType)
public List(String title, int listType,
            String[] listElement, Image[]
            listImage)
```

2.4.5 Bekerja Dengan Form

Dengan form dimungkinkan untuk menampilkan beberapa komponen GUI seperti daftar pilihan dan masukan teks (*textbox*) dalam satu layar. Form diimplementasikan oleh kelas `javax.microedition.lcdui.Form`. Form dapat menampung komponen – komponen yang disebut item dalam satu layar. Yang termasuk dalam item adalah *textfields*, *images*, *date fields*, *gauge*, dan *choice groups*. Item – item tersebut merupakan implementasi dari kelas turunan `Item` (`javax.microedition.lcdui.Item`). Ada dua konstruktor form yang umum digunakan, yaitu:

```
public Form(String title)
public Form(String title, Item[] items)
```

2.4.6 Kelas Item

Komponen – komponen yang bisa diletakkan dalam sebuah form merupakan komponen – komponen yang memiliki kelas yang merupakan kelas turunan dari kelas abstrak `javax.microedition.lcdui.item`. Komponen – komponen tersebut adalah:

- *ChoiceGroup*
Kelas ini menyediakan komponen yang mirip dengan *list*, yaitu untuk menyediakan daftar pilihan.
- *DateField*
Kelas ini menyediakan komponen untuk memasukkan informasi tanggal dan waktu.
- *Gauge*
Kelas ini menyediakan komponen grafik horizontal yang biasanya digunakan untuk memberikan gambaran berapa persen proses berjalan.
- *Image dan ImageItem*
Kelas ini menyediakan komponen grafik untuk manipulasi gambar.
- *StringItem*
Kelas ini menyediakan komponen teks string yang tidak bisa diedit oleh pengguna.
- *TextField*
Kelas ini menyediakan komponen untuk masukan teks string oleh pengguna.

2.4.6.1 TextField

Objek `TextField` digunakan untuk meletakkan objek teks string yang bisa diubah oleh pengguna secara langsung pada form. Objek `TextField` diimplementasikan oleh kelas `TextField` (`javax.microedition.lcdui.TextField`) yang merupakan kelas turunan dari kelas abstrak `javax.microedition.lcdui.Item`. Konstruktor dari kelas `TextField` ini adalah

```
public TextField(String title, String
                text, int maxSize, int constraint)
```

2.4.6.2 Image

Objek `Image` digunakan untuk meletakkan objek gambar yang bisa bersifat *immutable* atau *mutable*. Gambar yang bersifat *immutable* artinya tidak bisa diubah – ubah oleh aplikasi, dan umumnya bersifat gambar yang diambil dari sebuah file gambar, URL, atau tempat lain. Sebaliknya, gambar yang bersifat *mutable* bisa diubah – ubah oleh aplikasi. Konstruktor dari kelas `Image` ini bersifat statik, sehingga tidak menggunakan operator `new` untuk membuat sebuah objek `Image` melainkan mengacu ke fungsi `createImage()` pada objek `Image` ini:

```
public static void createImage(String
                                name)
public static void createImage(Image
                                img)
public static void createImage(int
                                width, int height)
```

Konstruktor yang pertama dan kedua digunakan untuk membuat objek `Image` yang bersifat *immutable*. Sedangkan konstruktor ketiga digunakan untuk membuat objek `Image` yang bersifat *mutable*.

2.5 Manajemen Event

Ketika terjadi interaksi antara pengguna dengan perangkat genggam maka akan dihasilkan suatu *event*. Misalnya ketika memilih suatu menu atau memasukkan teks ke dalam objek `TextField`, maka sistem akan memproduksi sebuah *event* yang kemudian aplikasi akan diberi tahu bahwa telah terjadi suatu *event* sebagai wujud interaksi dari pengguna sehingga aplikasi MIDlet bisa melakukan sesuatu berdasarkan *event* tersebut.

2.5.1 Penanganan Level Tinggi untuk Event

Untuk bisa menangani *event* pada level tinggi (*high level*), dapat dilakukan dengan mengimplementasikan *interface* `CommandListener` yang dideklarasikan pada paket `javax.microedition.lcdui`. Ketika mengimplementasikan *interface* `CommandListener`, maka harus mengimplementasikan fungsi `commandAction()` dalam *interface* tersebut.

2.5.2 Command dan Interface CommandListener

Kelas `Command` (`javax.microedition.lcdui.Command`) merupakan kelas yang mengenkapsulasi informasi sebuah aksi. Dari informasi yang dibungkus oleh kelas ini, maka aplikasi bisa menentukan aksi apa yang dilakukan, tipenya bagaimana dan lain sebagainya. Terdapat tiga fungsi yang berkaitan dengan manajemen *event* dengan objek `Command`, yaitu:

```
public void addCommand(Command cmd)
```

Mengasosiasikan objek `GUI` yang bersangkutan dengan objek `Command cmd`.

```
public void removeCommand(Command cmd)
```

Menghapuskan asosiasi objek `GUI` yang bersangkutan dengan objek `Command cmd`.

```
public void setCommandListener
(CommandListener l)
```

Mengasosiasikan objek `GUI` yang bersangkutan dengan *interface* `CommandListener l`.

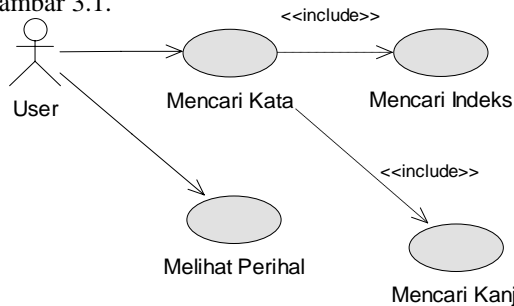
Ketiga fungsi diatas merupakan fungsi yang diturunkan oleh kelas `Displayable`. Konstruktor dari kelas `Command` adalah:

```
public Command(String label, int
commandType, int prio)
```

III. Perancangan Aplikasi Kamus Bahasa Jepang – Indonesia pada Perangkat Genggam

3.1 Use-Case Diagram

Dalam pembuatan *Use-case diagram* dari aplikasi kamus ini pertama kali ditentukan aktor yang akan menggunakan aplikasi (bersifat eksternal) dan aliran – aliran kerja (*workflow*) dalam aplikasi (bersifat internal). *Use-case diagram* dari aplikasi kamus ini ditunjukkan pada Gambar 3.1.



Gambar 3.1 *Use-case diagram* aplikasi kamus bahasa Jepang – Indonesia pada perangkat genggam.

3.2 Aliran Event

Use-case mendeskripsikan apa yang akan aplikasi kerjakan. Untuk secara nyata mengembangkan perangkat lunak, dibuat rincian – rincian yang lebih spesifik yang dituliskan sebagai aliran event. Aliran event ini digunakan untuk mendokumentasikan aliran – aliran logika dalam *use-case*. Dokumen ini akan mendeskripsikan secara rinci apa yang akan *user* lakukan dan apa yang perangkat lunak akan lakukan dalam menanggapi tindakan – tindakan pengguna padanya.

3.2.1 Aliran Normal (Primer)

Berikut ini merupakan aliran normal (primer) dari aplikasi kamus bahasa Jepang – Indonesia pada perangkat genggam:

1. *User* memasukkan kata yang hendak dicari artinya pada *TextField* “kata”.
2. Setelah memasukkan kata, *user* memilih *command* “cari” untuk memulai proses pencarian.
3. Aplikasi akan mencari indeks dari file yang berisi daftar kata dimana kata yang dicari kemungkinan berada.
4. Aplikasi akan membuka file tersebut sesuai dengan indeks yang didapatkan.
5. Aplikasi akan mencari lokasi kata di dalam daftar kata.
6. Aplikasi akan mencari karakter kanji dari kata yang dicari.
7. Aplikasi akan menampilkan arti beserta kanji dari kata yang dicari.

3.2.2 Aliran Alternatif

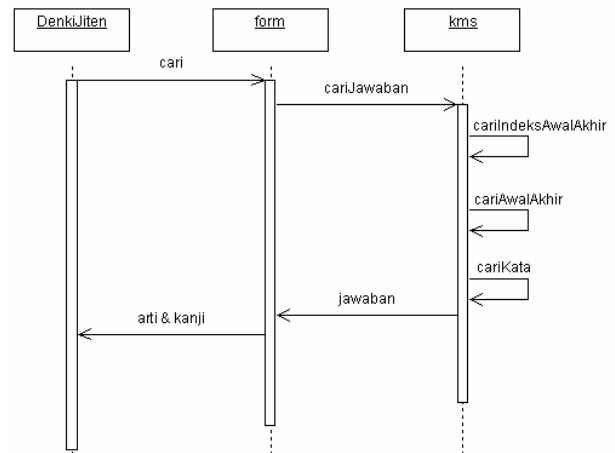
Berikut ini merupakan aliran alternatif dari aplikasi kamus bahasa Jepang – Indonesia pada perangkat genggam:

K1 : Kata yang dicari tidak ditemukan dalam daftar kata.

- Aplikasi akan menampilkan pesan bahwa kata tidak ditemukan.

3.3 Sequence Diagram

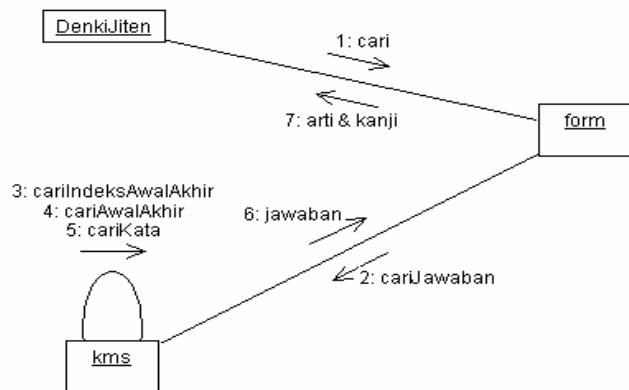
Sequence diagram memperlihatkan *event – event* yang terjadi sepanjang aliran kerja yang ada pada *use-case*, yang menyangkut objek apa yang dibutuhkan aliran, pesan mana yang dikirimkan suatu objek ke objek lainnya, serta bagaimana urutan pesan – pesan yang dikirimkan, yang berurutan sepanjang berjalannya waktu. *Sequence diagram* dari aplikasi kamus ini ditunjukkan pada Gambar 3.2.



Gambar 3.2 *Sequence diagram* aplikasi kamus bahasa Jepang – Indonesia pada perangkat genggam.

3.4 Collaboration Diagram

Collaboration diagram memperlihatkan *event – event* yang terjadi sepanjang aliran kerja yang ada pada *use-case*, yang menyangkut objek apa yang dibutuhkan aliran, pesan mana yang dikirimkan suatu objek ke objek lainnya, serta bagaimana urutan pesan – pesan yang dikirimkan, dengan berfokus pada relasi – relasi yang terjadi antara objek yang satu dengan objek – objek yang lainnya. *Collaboration diagram* dari aplikasi kamus ini ditunjukkan pada Gambar 3.3.

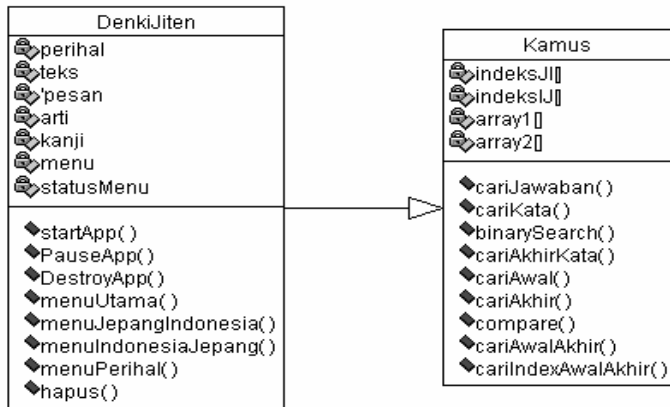


Gambar 3.3 *Collaboration diagram* aplikasi kamus bahasa Jepang – Indonesia pada perangkat genggam.

3.5 Class Diagram

Class diagram digunakan untuk menampilkan beberapa kelas yang ada dalam perangkat lunak kamus bahasa Jepang – Indonesia pada perangkat genggam serta memberikan gambaran / diagram statis tentang perangkat lunak dan relasi – relasi yang ada di dalamnya.

Dalam perangkat lunak kamus bahasa Jepang – Indonesia pada perangkat genggam ini terdapat dua kelas, yaitu kelas *DenkiJiten* dan kelas *Kamus*. *Class diagram* dari aplikasi kamus ini ditunjukkan pada Gambar 3.4.



Gambar 3.4 *Class diagram* aplikasi kamus bahasa Jepang – Indonesia pada perangkat genggam.

IV. Implementasi dan Pengujian Aplikasi Kamus Bahasa Jepang-Indonesia pada Perangkat Genggam

4.1 Tahap Implementasi

Tahap implementasi merupakan tahap dimana sistem yang telah dirancang diwujudkan dalam bentuk aplikasi. Dalam hal ini, dideskripsikan tampilan-tampilan halaman sistem dan fungsinya.

4.2 Tahap pengujian

Salah satu bagian penting pada sistem pakar adalah basis pengetahuan. Oleh karena itu, sangat penting untuk memastikan bahwa pengetahuan yang diberikan oleh sistem pakar sudah sesuai dengan pengetahuan pakar yang diperoleh dari tahap akuisisi pengetahuan. Untuk memastikan kebenaran data yang diberikan oleh sistem, maka dilakukan proses verifikasi. Proses verifikasi dimaksudkan untuk mencocokkan pengetahuan yang terdapat pada sistem pakar dengan pengetahuan pakar yang diperoleh pada tahap akuisisi pengetahuan. Setelah proses verifikasi, dilakukan pengujian setiap halaman pada aplikasi, apakah fungsinya sudah berjalan sebagaimana mestinya.

V. Penutup

5.1 Kesimpulan

1. Dengan menggunakan J2ME yang merupakan subset dari J2SE dimungkinkan untuk membuat aplikasi – aplikasi seperti kamus digital yang bisa diimplementasikan pada perangkat genggam yang

merupakan perangkat dengan jumlah memori dan kapasitas penyimpanan serta *user interface* yang terbatas dengan menggunakan profil MIDP dan konfigurasi CLDC.

2. Aplikasi kamus bahasa Jepang – Indonesia ini menggunakan file berekstensi .txt sebagai basis datanya yang dibuat dalam bentuk daftar kata beserta artinya dan aplikasi akan membandingkan kata yang dicari dengan kata yang terdapat dalam daftar kata serta kemudian akan menampilkan artinya apabila kata yang dicari ditemukan dalam daftar kata.
3. Untuk menampilkan huruf kanji dalam aplikasi ini pada menu Kamus Jepang – Indonesia digunakan file gambar yang memiliki format PNG (Portable Network Graphic) yang diletakkan dalam folder resource.
4. Aplikasi ini dapat berjalan pada perangkat genggam (ponsel) yang mendukung aplikasi java sehingga *user* akan mempunyai keleluasaan dan mobilitas dalam memanfaatkan aplikasi ini sebagai alat untuk mempermudah dalam mempelajari bahasa Jepang untuk mengetahui arti kata dari bahasa Jepang ke dalam bahasa Indonesia dan sebaliknya beserta huruf kanjinya.

5.2 Saran

1. Adanya tambahan fasilitas bagi *user* untuk dapat memasukkan kata - kata baru yang tidak dapat ditemukan dalam daftar kata yang ada sehingga *user* dapat menambah pembendaharaan kata dalam aplikasi kamus ini dengan memanfaatkan record store.
2. Mencari cara alternatif untuk menampilkan huruf kanji selain dengan menggunakan gambar sehingga dapat mengurangi besarnya ukuran file aplikasi sehingga dapat diaplikasikan pada ponsel yang tidak mempunyai memori penyimpan eksternal yang pada umumnya mempunyai ukuran memori penyimpan yang relatif kecil.
3. Menambah menu cara menulis huruf kana dengan disertai urutan cara penulisannya sehingga dapat menambah pengetahuan bagi *user* dalam cara penulisan huruf kana yang baik dan benar.
4. Menambah menu ungkapan – ungkapan (*tango*) yang sering digunakan pada situasi – situasi tertentu yang sering dijumpai dalam kehidupan sehari – hari sehingga dapat menambah pengetahuan bagi *user* dan dapat menerapkannya secara langsung dalam percakapan sehari – hari.

DAFTAR PUSTAKA

- [1]. Adimihardja, M., *Serial Tata Bahasa Jepang Ragam Kanji*, Pustaka, Bandung, 2003.
- [2]. Gakushudo, *Aksara Jepang Kanji Praktis 1*, Gakushudo, Jakarta, 1991.
- [3]. Nugroho, A., *Rational Rose untuk Pemodelan Berorientasi Objek*, Informatika, Bandung, 2005.

- [4]. Shalahuddin, S. dan Rosa A.S., *Pemrograman J2ME Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile.*, Informatika, Bandung, 2006.
- [5]. Wicaksono, A., *Dasar – Dasar Pemrograman Java 2*, PT Elex Media Komputindo, Jakarta, 2002.
- [6]. Wicaksono, A., *Pemrograman Aplikasi Wireless dengan Java*, PT Elex Media Komputindo, Jakarta, 2002.
- [7]. <http://forum.nokia.com/>
- [8]. <http://java.sun.com/j2me/>



Tommy Adhi Kresna Murthi, menyelesaikan studi Diploma III di Politeknik Negeri Semarang Jurusan Teknik Elektro. Saat ini sedang menyelesaikan studi di Jurusan Teknik Elektro Universitas Diponegoro.
Email : kamui_ichigo@yahoo.com

Menyetujui dan mengesahkan,

Pembimbing I

Pembimbing II

Aghus Sofwan, S.T, M.T.
NIP. 132 163 757

Eko Handoyo, S.T, M.T.
NIP. 132 309 142

¹ Mahasiswa Teknik Elektro

² Dosen Teknik Elektro

³ Dosen Teknik Elektro