Jurnal Computech & Bisnis, Vol. 2, No. 1, Juni 2008, 33-44
ISSN 1978-9629

SISTEM PENERJEMAH KATA BAHASA INGGRIS KE BAHASA INDONESIA MEDIA *HANDPHONE* BERBASIS *MOBILE APLICATION*

Rohmat Nur Ibrahim
STMIK Mardira Indonesia, Bandung

Abstract

Mobile phones (cell phones) are now used not only for conversation but also to play games and so on. The presence Java2 Micro Edition (J2ME) is expected to answer those needs while making more multifunctional mobile phones.

Currently the latest generation of mobile phones has the ability to apply the J2ME program, which is a Java program that is specifically designed for mobile devices. Java enabled mobile applications meet the needs of its users through a variety of exciting programs.

Because it can be done offline phones not only can be used as a communication tool but it can also be used as tool translator English words into Indonesian.

Keywords: *Phones, J2ME, Java Program, Translator*

Abstrak

Telepon seluler (ponsel) sekarang digunakan tidak hanya untuk percakapan tapi juga untuk bermain games dan sebagainya. Hadirnya Java2 Micro Edition (J2ME) diharapkan dapat menjawab kebutuhan tersebut sekaligus membuat ponsel lebih multifungsi.

Saat ini ponsel generasi terbaru memiliki kemampuan mengaplikasikan program J2ME, yaitu program Java yang khusus dirancang untuk peranti bergerak. Aplikasi Java memungkinkan ponsel memenuhi kebutuhan penggunanya melalui berbagai program menarik.

Karena dapat dilakukan secara *offline* ponsel tidak hanya dapat digunakan sebagai alat komunikasi tapi dapat juga digunakan sebagai alat penerjemah kata bahasa Inggris ke bahasa Indonesia.

Kata Kunci: Ponsel, J2ME, Program Java, Penerjemah

PENDAHULUAN

" You can find Java technology all over. It's embedded in mobile phones, PDAs, and pagers; it's inside video games, TVs, and web sites. It's even in cars and on the planet Mars. But Java technology is not just for fun. It also means business. It's in business applications and tools of all kinds – used by giant enterprises and small businesses alike. But wherever you find it, Java technology gives you a great digital experience. "

Kutipan di atas diambil dari official web site Java : http://www.java.com, *Sun Microsystem*. Kalimat-kalimat tersebut memang tidak berlebihan. Selama hampir satu dasawarsa terakhir ini, Java telah memainkan peran penting dalam segala aspek teknologi IT dan Telekomunikasi. Berawal dari tahun 1995 dari suatu projek tersebut bertujuan membuat bahasa yang bersifat multiplatform untuk chip-chip device mikro-elektronika. Dengan mengadaptasi dan meningkatkan performansi C++ serta menerapkan konsep *Object Oriented*, bahasa tersebut ternyata sesuai dengan pemograman dalam lingkungan jaringan maupun internet. Telepon seluler (ponsel) sekarang digunakan tidak hanya untuk percakapan tapi juga untuk bermain *games* dan sebagainya. Tentunya aplikasi-aplikasi tersebut memerlukan program yang lebih canggih. Hadirnya *Java2 Micro Edition* (J2ME) diharapkan dapat menjawab kebutuhan tersebut sekaligus membuat *ponsel* lebih multifungsi. Penyebaran Java sebelumnya banyak menghasilkan program menarik di komputer. Sekarang aplikasi Java tidak lagi hanya ditemukan di komputer, tapi juga di berbagai peranti bergerak seperti telepon seluler.

Saat ini *ponsel* generasi terbaru memiliki kemampuan mengaplikasikan program J2ME, yaitu program Java yang khusus dirancang untuk peranti bergerak. Aplikasi Java memungkinkan *ponsel* memenuhi kebutuhan penggunanya melalui berbagai program menarik. *Trend* menunjukkan *ponsel* sekarang dipakai pula untuk sarana hiburan. Karenanya sebagai langkah awal *vendor ponsel* berkonsentrasi membangun *games ponsel* berbasis Java yang tidak kalah hebat dengan *games* komputer. *Chatting* dengan program *messenger* pun tidak lagi hanya milik pengguna komputer. Dengan basis Java, kita dapat melakukan *chatting* dengan sesama pengguna *ponsel* atau dengan *chatter* yang sedang online di komputer melalui *ponsel*. Di masa depan aplikasi Java juga akan banyak digunakan untuk aplikasi bisnis, misalnya penjualan secara *online*.

Salah satu kelebihan utama aplikasi Java adalah sifatnya yang *multiplatform*. Dampaknya semua program yang dibuat dengan basis Java dapat digunakan di berbagai merek *ponsel*. Pemakai juga dapat menambahkan *games* sendiri di *ponsel*. Selain itu, Java membuat sistem keamanan di *ponsel* meningkat daripada sebelumnya. Biaya aplikasi juga akan lebih murah karena aplikasi bisa dilakukan secara *offline*. Karena dapat dilakukan secara *offline ponsel* tidak hanya dapat digunakan sebagai alat komunikasi tapi dapat juga digunakan sebagai alat penerjemah. Karena sifatnya yang *multiplatform*, banyak aplikasi Java yang dapat di-*download* gratis dari internet. Misalnya program untuk *games* baru. Seiring dengan waktu, diharapkan makin banyak pengembang program yang menciptakan aplikasi menarik untuk pengguna *ponsel*. Agar mampu menciptakan lingkungan pemrograman, *ponsel* berkemampuan Java harus dilengkapi memori rata-rata sebesar 128 KB. Jadi, perangkat kerasnya memang berbeda dengan *ponsel* biasa. *Nokia* akan meluncurkan tiga seri *ponsel*

berkemampuan Java dalam waktu dekat. *Siemens* telah mengeluarkan dua seri *ponsel* Java yaitu SL45i dan M50 sedangkan *Sony Ericsson* juga akan memasarkan *ponsel* Java. Program Java di *ponsel* berbeda dengan yang dipakai di komputer, tapi keduanya diciptakan oleh perusahaan teknologi informasi *Sun Microsystems*. Seperti dikatakan Adrianus Budiarjo, sebagai konsultan teknis *Sun Microsystems* Indonesia, tren di *ponsel* adalah pemakaian J2ME sedangkan jenis di *browser* atau komputer adalah *Java script*.
Berbeda dengan komputer, Kendro menilai pengembangan di *ponsel* memiliki karakteristik sendiri. Misalnya orang terbiasa mendapatkan secara gratis dari komputer atau internet, sedangkan di *ponsel* sudah terbiasa membayar sehingga menarik minat pengembang aplikasi.

Dengan fasilitas media *handphone* yang mendukung bahasa pemograman J2ME *(java 2 micro edition)tersebut* Penulis akan mencoba membuat untuk memudahkan pencarian kata terjemahan Bahasa Inggris ke dalam Bahasa Indonesia Sekilas saja kenapa penulis menggunakan media *handphone* tersebut, sebagian orang berpendapat di zaman modernisasi sekarang ini fasilitas *handphone* bukan barang asing lagi dan dapat dengan mudah untuk dibawa-bawa selain sebagai alat untuk komunikasi antar teman, kerabat, atau orang lain di muka bumi ini. BREW *(Binary Runtime Environment for Wireless)* mungkin sudah lama anda kenal. Kata BREW sering dijumpai di fitur-fitur *ponsel* CDMA. Mungkin anda sendiri belum begitu faham apa manfaat dari fitur ini. Secara sederhana, BREW adalah aplikasi yang memungkinkan kita menikmati yang disediakan pengembang lewat fasilitas download yang disediakan operator.Lebih jauh lagi BREW sering disebut sebagai saingan *Java2 Mobile Edition (J2ME).* Tapi banyak kalangan yang menilai bahwa keduanya memiliki karakteristik berbeda. Misalnya, BREW saat ini hanya dikembangkan di *ponsel-ponsel* CDMA yang menggunakan *chipset*, sedang J2ME sudah dikembangkan baik di GSM ataupun CDMA. J2ME dan BREW saat ini menjadi teknologi yang dibutuhkan oleh para pengembang untuk menciptakan model baru untuk akses *online* yang memungkinkan aplikasi tersebut didownload *via web* dan dimainkan pada kondisi *offline*.

J2ME adalah teknologi Java yang dikustomisasi untuk konsumen kecil dengan *prosesor, memory*, layar dan kemampuan input yang terbatas. Arsitektur J2ME dilandasi oleh kategori dan keluarga perangkat-perangkat tersebut memiliki struktur yang sudah ditentukan. Di Indonesia, teknologi J2ME

ini sudah mulai menjadi bahan mata kuliah di lembaga-lembaga pendidikan IT. Selain itu, *training* (pelatihan) aplikasi J2ME ini juga sudah sering digelar. Untuk menjalankan aplikasi J2ME, di *ponsel-ponsel* yang mendukungnya terdapat *Java Virtual Machine* (JVM). JVM ini berjalan di atas sistem operasi perangkat dan sudah diatur untuk sistem operasi yang spesifik. Ukuran dan kompleksitas JVM tergantung dari konfigurasi utama J2ME yang didukung. Sedangkan BREW berjalan pada *level firmware (chipset CDMA)* dan khususnya untuk aplikasi nirkabel yang dapat didownload dan dieksekusi di *ponsel*. Sama seperti Java, BREW membutuhkan *runtime environment*, semacam mesin *virtual*.
Berikut beberapa perbandingan antara BREW dan J2ME:

	Konsumsi Daya	Protokol Komunikasi	Model *Ponsel*
BREW	Kecil, dapat mengurangi konsumsi	Seluruh protokol TCP/IP	Dapat diinstal oleh

	daya pada saat tidak digunakan		seluruh model mulai entry level hingga high end
J2ME	Relatif besar, sebab mesin virtualnya tetap beroperasi saat statusnya tidak aktif.	Hanya HTTP	Hanya mid – high end

Table 1 perbandingan antara BREW dan J2ME

Singkatnya BREW tampak lebih unggul dibanding J2ME. Tapi sifat-sifat Java yang menganut sistem sumber terbuka *(open source)* membuat J2ME lebih fleksibel dan tentu ini juga membuatnya menjadi pilihan banyak pengembang hingga saat ini. BREW sangat identik dengan teknologi CDMA. Di Indonesia, dari 5 operator CDMA baru *Mobile-8* yang sudah merilis layanan berbasis BREW. Bekerja sama dengan *Jastis* dan *Qualcomm*, hadirlah layanan yang disebut *b-live*. Saat ini layanan *b-life* terdapat di *ponsel bundling* ZTE C300 dan ZTE C330, keduanya dibandrol di bawah 500 ribu. Membuktikan bahwa layanan BREW dapat diminati oleh *ponsel-ponsel low end* hingga *high end*.

Dari beberapa layanan yang ditawarkan memang tidak gratis, untuk sekedar menjajal anda cukup membayar Rp. 1000, untuk satu hari pemakaian. Bila berminat silahkan berlangganan. Konten yang cukup menarik adalah berita terkini, meski tidak dilengkapi WAP anda tetap tidak ketinggalan informasi. Penulis akan menggunakan bahasa pemograman Java yang sebagian fasilitas dari media *handphone* telah dipergunakan. Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform* / sistem operasi komputer, sesuai dengan prinsip *write once, run every where*. Dengan kelebihan ini pemrogram cukup menulis sebuah program java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / *bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java berjalankan dengan baik.

PEMBAHASAN

Karakteristik Utama Pemrograman Berorientasi Objek (PBO)

Object itu secara mudahnya dapat dikatakan terdiri dari *property* dan *method*. Konsep *object oriented* memiliki karakteristik utama yaitu :

1. *Enkapsulasi*

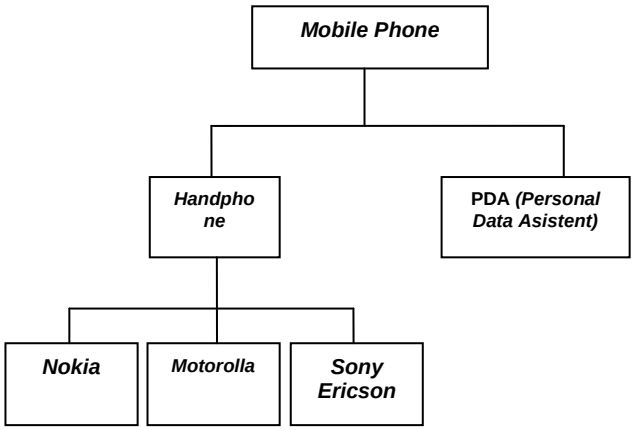
Enkapsulasi adalah suatu mekanisme untuk menyembunyikan atau memproteksi suatu proses dari kemungkinan interferensi atau penyalahgunaan dari luar sistem sekaligus menyederhanakan penggunaan sistem itu sendiri. Akses ke internal sistem diatur sedemikian rupa melalui seperangkat *interface*. Contoh kasus yang penulis teliti *handphone*, pada sistem pemindahan *touch pad* pada *handpone* dengan menampilkan ke layar *display handphone*, maka pengguna *handphone* tidak perlu tahu detail dari bagaimana proses *text* yang terdapat pada *touch pad handphone* dapat tampil ke *display handphone* itu dilakukan oleh mesin, cukup tahu bagaimana menggunakan *touch pad handphone* itu. *Touch pad* pada *handphone* itu merupakan *interface* (antar muka) *handphone*.

Dengan kata lain : *Enkapsulasi*, sebuah prinsip yang digunakan ketika membangun struktur program secara keseluruhan yang mana setiap komponen dari program dibungkus, pembungkusan *property* dan operasi dalam satu *event*. Secara simple dapat kita katakan bahwa kita

menyembunyikan keruwetan kode-kode program dalam suatu *object*.

2. Pewarisan *(Inheritance)*

Sebagai manusia kita sebenarnya terbiasa untuk melihat objek yang berada disekitar kita tersusun secara hierarki berdasarkan *class*-nya masing-masing. Dari sini kemudian timbul suatu konsep tentang pewarisan yang merupakan suatu proses dimana suatu *class* diturunkan dari *class* lainnya sehingga ia mendapatkkan ciri atau sifat dari *class* tersebut. Perhatikan contoh hirarki berikut ini:



Gambar 1 Hirarki *Class Inheritance Mobile Phone*

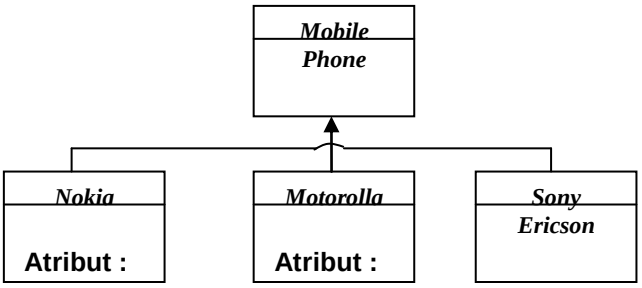
Dari hirarki diatas dapat dilihat bahwa, semakin kebawah, *class* akan semakin bersifat *spesifik*. *Class handphone* memiliki seluruh sifat yang dimiliki oleh *Mobile Phone*, demikian halnya juga *Nokia*, *Motorolla* dan *Sony Ericson* memiliki seluruh sifat yang diturunkan dari class *handphone*. Dengan konsep ini, karakteristik yang dimiliki oleh *class mobile phone* cukup didefinisikan dalam *class mobile phone* saja. *Class handphone* tidak perlu mendefinisikan ulang apa yang telah dimiliki oleh *class mobile phone*, karena sebagai *class* turunannya, ia akan mendapatkan karakteristik dari *class mobile phone* secara otomatis. Demikian juga dengan *class Nokia*, *Motorolla* dan *Sony Ericson*, hanya perlu mendefinisikan karakteristik yang spesifik dimiliki oleh class-nya masing-masing. Dengan memanfaatkan konsep pewarisan ini dalam pemrograman, maka

hanya perlu mendefinisikan karakteristik yang lebih umum akan didapatkan dari *class* darimana ia diturunkan.

Dengan kata lain : *Inheritance* (pewarisan), merupakan sarana untuk menghilangkan penulisan ulang terhadap kode yang dapat digunakan berulang kali yang didasarkan pada hubungan relasional hirarki.

3. Polymorphism

Polymorphism berasal dari bahasa Yunani yang berarti banyak bentuk. Dalam Pemograman Berorientasi Objek, konsep ini memungkinkan digunakannya suatu *interface* yang sama untuk memerintah objek agar melakukan aksi atau tindakan yang mungkin secara prinsip sama namun secara proses berbeda. Dalam konsep yang lebih umum sering kali *polymorphism* disebut dalam istilah satu *interface* banyak aksi.



Gambar 2 Class *Polymorphism* Mobile Phone

Pada contoh diatas *class* dasar adalah *class* bentuk yang memiliki atribut berupa CLDC dan operasi hitung Series CLDC, *class* tersebut dapat diturunkan kedalam berbagai macam *class* bentuk seperti *Nokia, Motorolla, Sony Ericson. Class Nokia, Motorolla, Sony Ericson* memiliki atribut “Series CLDC” dari hasil penurunan *class* bentuk, akan tetapi operasi Series CLDC pada masing

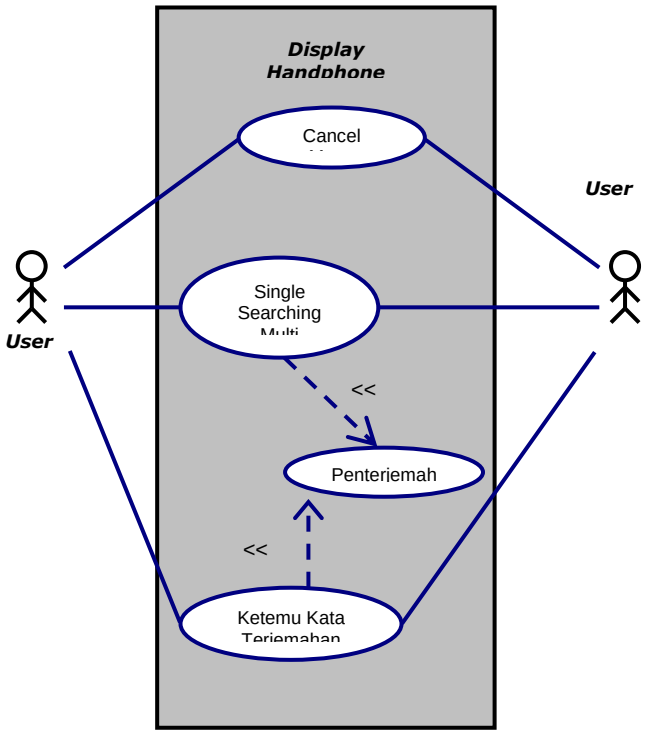
masing *class* akan berbeda-beda, inilah yang disebut sebagai *polymorphism*.

Dengan kata lain : *Polymorphism*, suatu kondisi dimana dua object atau lebih mempunyai antarmuka yang identik namun mempunyai perilaku berbeda.

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* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/ sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. Pada gambar 2. dapat dilihat *use case diagram* teknik *mobile application* yang menjadi aktor pada gambar 2. adalah *user*. *User* tersebut melakukan aktivitas-aktivitas terhadap system yang berjalan. Misalkan pada gambar 3.4. aktornya adalah *user*. Aktor ini melakukan aktivitas-aktivitas terhadap system, yaitu mencari kata terjemahan. Aktor yang lainnya tidak ada hanya *user* sendiri yang melakukan aktivitas-aktivitas tersebut.



Gambar 3 Use Case Diagram

4. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/ properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/ fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti

containment, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1.1 *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.

2.1 *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.

3.1 *Public*, dapat dipanggil oleh siapa saja.

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class.* Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.

Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.

a. Hubungan Antar *Class*

Ada beberapa uraian dalam hubungan antar *class,* diantaranya adalah :

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Generalisasi, yaitu hubungan antar elemen yang lebih general dengan elemen yang lebih spesifik yang masih memiliki sifat induknya hanya saja memiliki informasi yang lain.
5. Ketergantungan *(dependency)*, yaitu hubungan antar elemen independent dan elemen dependen. Perubahan pada elemen independent akan mengakibatkan perubahan pula pada elemen yang dependen terhadapnya.

Dalam melakukan kata terjemahan, ada beberapa kelas yang terlibat, diantaranya adalah :

1. Kelas *User*, yaitu sebuah kelas yang mengalami agregasi, sehingga memiliki dua kelas, yaitu kelas penerima dan kelas pengirim. Kedua kelas tersebut saling berasosiasi.
2. Kelas menu utama, yaitu suatu kelas yang mempunyai hubungan asosiasi dengan kelas pengirim dan kelas penerima.
3. Kelas terjemahan, yaitu suatu kelas yang mempunyai hubungan asosiasi dengan kelas pengirim dan penerima.

Kemudian kita menentukan masing-masing atribut dan metoda dari kelas yang telah diidentifikasi, diantaranya :

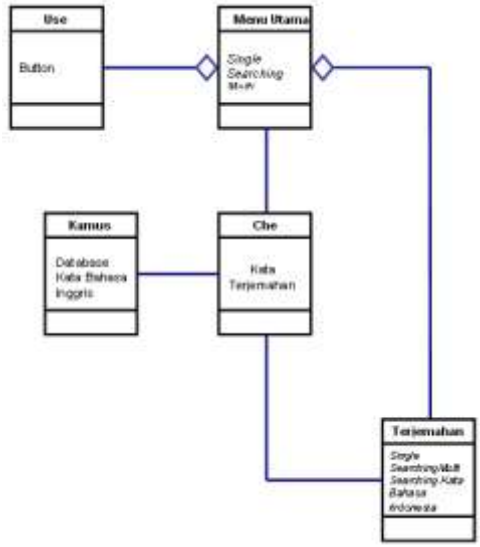
1. Kelas *User*
   a. Identifikasi atribut *user*
      a.1. Memilih input kata Bahasa Inggris *single searching*
      a.2. Memilih input kata Bahasa Inggris *multi searching*
   b. Identifikasi metode
      b.1. Kelas user tidak memiliki metode
2. Kelas menu utama
   a. Identifikasi atribut *single searcing* atau *multi searching*
      a.1. Input kata Bahasa Inggris
   b. Identifikasi metode

b.1. Cari *(browse)*

b.2. Proses kata Bahasa Indonesia

3. Kelas terjemahan
   a. Identifikasi atribut terjemahan
      a.1. Menampilkan output kata Bahasa Indonesia *single searching*
      a.2. Menampilkan output kata Bahasa Indonesia *multi searching*
   b. Identifikasi metode
      b.1. Proses
      b.2. *close*

Setelah mengidentifikasi kelas, atribut dan metode, kita implementasikan ke *class diagram* yang masih sederhana pada gambar 4.



Gambar 4 *Class diagram*

b. Activity Diagram

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

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

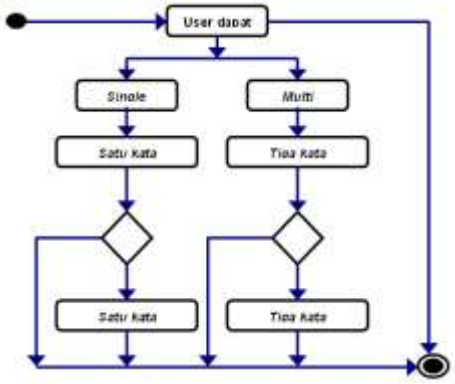
Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Pada gambar 5 akan dijelaskan mengenai file (kata yang akan diterjemahkan) yang diimplementasikan ke dalam *activity diagram.*

Deskripsi file (kata yang akan diterjemahkan) sebelum diimplementasikan ke dalam *activity diagram* adalah sebagai berikut :

1. User atau pengguna *handphone* akan diberi pilihan *single searching* (hanya satu kata Bahasa Inggris) atau *multi searching* (tiga kata Bahasa Inggris) untuk menterjemahkan kata yang diinputkan.
2. Bila user memilih *single searching* user akan mendapatkan form input satu kolom dalam *display* (layar) *handphone* kata terjemahan Bahasa Inggris.
3. Kemudian bila user memilih *multi searching* user akan mendapatkan form input tiga kolom sekaligus

dalam *display* (layar) *handphone* kata terjemahan Bahasa Inggris.

Setelah user menginputkan kata Bahasa Inggris ke dalam kolom yang tersedia user akan mendapatkan terjemahan kata dalam bentuk Bahasa Indonesia.



Gambar 5 *Activity Diagram*

c. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline vertikal*.

Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada gambar 6 akan dijelaskan mengenai sebuah proses yang terdapat tiga objek, yaitu :

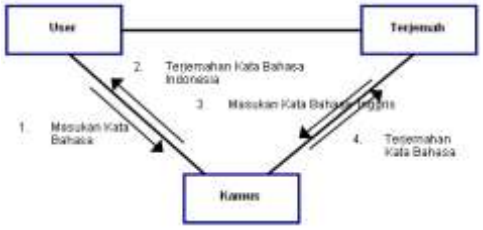
1. Objek User
2. Objek Terjemah
3. Objek Kamus.

Ketiga objek tersebut saling berinteraksi. Pertama, objek *user* memasukan kata ke dalam terjemah. Kemudian objek terjemah memberikan jawaban yang diinginkan oleh objek *user* . Begitu juga dengan objek kamus. Objek ini merupakan kamus Bahasa Inggris, dan oleh objek terjemah diberikan kata terjemahan yang diinginkan oleh objek *user*. Antara objek *user* dan objek terjemah akan saling memberitahukan kata yang diinginkannya masing-masing.

Gambar 6 *Sequence Diagram*

d. Collaboration Diagram

Collaboration diagram memperlihatkan kolaborasi dinamika antar objek tanpa memperlihatkan aspek waktu. *Collaboration diagram* juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*.



PERANCANGAN SISTEM

Setelah melakukan tahap analisis sistem maka dilakukan tahap perancangan sistem, dalam perancangan sistem akan dijelaskan mengenai perkembangan sistem dalam perangkat bantu

penterjemahan Bahasa Inggris ke dalam Bahasa Indonesia.

Tujuan dari rancangan sistem adalah untuk mempermudah pemakai sistem dalam memberikan gambaran yang jelas mengenai rancang bangun yang lengkap dari sistem perangkat bantu penterjemahan Bahasa Inggris ke dalam Bahasa Indonesia, yang selanjutnya digunakan untuk membuat program komputer melalui media *handphone*.

Untuk mencapai tujuan di atas pada rancangan sistem perangkat bantu penterjemahan Bahasa Inggris ke dalam Bahasa Indonesia, maka rancangan sistem yang dibuat harus mencapai sasaran sebagai berikut :

1. Rancangan sistem yang dibuat dapat membantu para pengguna atau bagi siapapun yang tertarik.
2. Rancangan sistem yang dibuat harus disesuaikan dengan kebutuhan.
3. Rancangan sistem yang dibuat harus lebih mempercepat pengerjaan yang sedang dikerjakan.

1. Class Diagram

Class diagram dapat merupakan implementasi dari sebuah *interface*, yaitu *class abstrak* yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class.* Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*. Seperti telah dijelaskan sebelumnya, bahwa *class* memiliki tiga area pokok, yaitu :

1. Nama
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1.1 *Private (-)*, tidak dapat dipanggil dari luar *class* yang bersangkutan.

2.1 *Protected (#)*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya

3.1 *Public (+)*, dapat dipanggil oleh siapa saja

Seperti telah dijelaskan sebelumnya kita tentukan dahulu langkah-langkah di bawah ini sebelum diimplementasikan ke *class diagram.* Ada 3 (tiga) *class*, dimana dari ketiga *class* ini, merupakan agregasi dari *class user*, dan dua *class* merupakan asosiasi dari *class user* ke *class* menu utama. Sisanya *class* terjemahan. Untuk lebih jelasnya dijelaskan di bawah ini :

1. Terdapat 3 (tiga) kelas yaitu :
   a. Kelas *User*
   b. Kelas Menu Utama
   c. Kelas Terjemahan
2. Kelas *User*
   a. Identifikasi atribut *user*
      i. #Memilih input kata Bahasa Inggris *single searching* : string
      ii. #Memilih input kata Bahasa Inggris *multi searching* : string
   b. Identifikasi metode *user*
      i. Kelas *user* tidak memiliki metode
3. Kelas menu utama
   a. Identifikasi atribut *single searcing* atau *multi searching*
      i. #Input kata Bahasa Inggris : string
   b. Identifikasi metode menu utama
      i. #B*rowse*()
      ii. #Proses kata Bahasa Indonesia : string
4. Kelas terjemahan
   a. Identifikasi atribut terjemahan
      i. #Menampilkan output kata Bahasa Indonesia *single searching* : string
      ii. #Menampilkan output kata Bahasa Indonesia *multi searching* : string
   b. Identifikasi metode terjemahan
      i. #Proses()
      ii. +*close*()

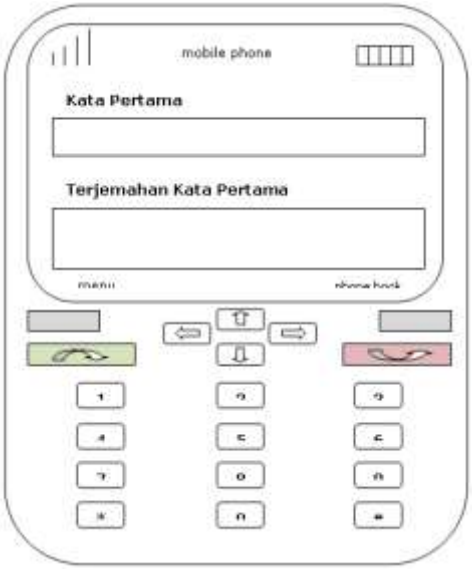
5. Identifikasi relasi (hubungan) pada gambar 1. *class diagram* lengkap

Tabel 2 Identifikasi Relasi *Class Diagram*

Identifikasi Relasi *Class Diagram*

Kelas	Kelas Relasi	Nama Asosiasi	Nama Agregasi
	User		Penterjemah kata Bahasa Inggris
User	*public and private key*	Mencari kata terjemahan Bahasa Indonesia	
User	Menu utama	Single searching	
User	Menu utama	Multi searching	

a. Interface Menu Utama *Single Searching*

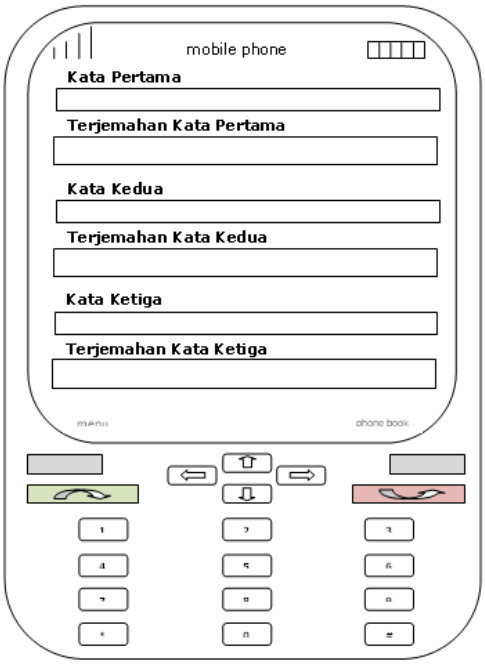


Gambar 8 Interface Menu Utama *Single Searching*

Keterangan dari *interface* menu utama *single searching*, adalah sebagai berikut :

1. Tombol menu dapat berfungsi sebagai awal mula pilihan pencarian menggunakan *single searching* atau *multi searching* dan terdapat pula tombol *about.*
2. Gambar 4.2 menjelaskan *Interface* menu utama dengan menggunakan *single searching.*
3. Tombol *search* berfungsi untuk mencari dan menterjemahkan kata yang telah diinputkan dengan file yang berektensi text (*.text).
4. Tombol *about* yang terdapat dalam tombol menu berisi informasi pembuat program *mobile application*.

b. Interface Menu Utama *Multi Searching*



Keterangan dari *interface* menu utama *multi searching*, adalah sebagai berikut :

1. Tombol menu dapat berfungsi sebagai awal mula pilihan pencarian menggunakan *single searching* atau *multi searching* dan terdapat pula tombol *about.*
2. Gambar 4.3 menjelaskan *Interface* menu utama dengan menggunakan *multi searching.*
3. Tombol *search* berfungsi untuk mencari dan menterjemahkan kata yang telah diinputkan dengan file yang berektensi text (*.text).
4. Tombol *about* yang terdapat dalam tombol menu berisi informasi pembuat program *mobile application*.

KESIMPULAN

Setelah penulis melakukan pembuatan penerjemahan Bahasa Inggris ke ke dalam Bahasa Indonesia, maka penulis akan mengemukakan beberapa kesimpulan diantaranya, sebagai berikut :

1. Penerjemah Bahasa Inggris ke dalam Bahasa Indonesia di dunia pendidikan dapat membantu dan menambah pengetahuan tentang Bahasa Inggris.
2. Penerjemah Bahasa Inggris ke dalam Bahasa Indonesia media *handphone* dapat membantu pencarian dengan cepat.
3. Hasil dari penyelesaian penerjemahan ini, menghasilkan suatu kata yang dimasukkan di dalam media *handphone* kata Bahasa Inggris dan akan menghasilkan suatu kata dalam Bahasa Indonesia.
4. Penerjemahan Bahasa Inggris ke dalam Bahasa Indonesia di media *handphone* akan menambah software yang belum ada.

DAFTAR PUSTAKA

Guojie, Jackwin Li. Build Your Stock With J2ME. ibm.com

Muchow, John W. 2002. Core J2ME Technology & MIDP. Sun Microsystem

M. Shalahuddin. 2006. Pemrograman J2ME

Rosa A.S. 2006. Pemrograman J2ME

Topley, Kim.2002. J2ME in a Nutshell.

O'Reilly White, James dan David Hemphill. 2002. Java 2 Micro Edition, Java in Small Things. Manning Publication Co. Greenwich

http://forum.nokia.com

http://java.sun.com/j2me

http://students.if.itb.ac.id/~if11037/index.html

http://students.if.itb.ac.id/~if11068/index.html

http://udinrosa.t35.com

http://www.java2s.com/Code/Java/J2ME/CatalogJ2ME.htm

http://www.pdfoo.com/pdf-42/java-p6.html