

SKRIPSI

**DESAIN APLIKASI KAMUS BAHASA INGGRIS PADA PONSEL
MENGUNAKAN JAVA (J2ME)**



Disusun Oleh :

NURIL FATIH AWALUDDIN 04.12.639

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
MARET 2009**

LEMBAR PERSETUJUAN

DESAIN APLIKASI KAMUS BAHASA INGGRIS PADA
PONSEL MENGGUNAKAN JAVA (J2ME)

SKRIPSI

*Disusun dan Diajukan sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Komputer Dan Informatika Strata Satu (S-1)*

Disusun Oleh :

NURIL FATIH AWALUDDIN

NIM : 04.12.639

Diperiksa dan Disetujui

Dosen Pembimbing I



Ir. TH. Mimien Mustikawati, MT.

NIP Y. 130000352

Dosen Pembimbing II



M. Ashar, ST., MT.

NIP P. 1030500408



Mengetahui

Ketua Jurusan Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT.

NIP Y. 1039500274

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2009

**DESAIN APLIKASI KAMUS BAHASA INGGRIS PADA PONSEL
MENGUNAKAN JAVA (J2ME)**

Nuril Fatih Awaluddin

04.12.639

**Konsentrasi Teknik Komputer & Informatika, Jurusan Teknik Elektro S-1
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang
Jln. Raya Karanglo Km 2, Tasik Madu – Malang
Email : Factich_99@yahoo.com**

**Dosen Pembimbing : I. Ir. TH. Mimien Mustikawati, MT.
II. M. Ashar ST., MT.**

Abstrak

Kata Kunci: Kamus, Komunikasi Mobile, Java, Internet, GPRS, Database.

Saat ini pengguna ponsel tidak hanya terbatas sebagai alat komunikasi bergerak (*mobile phone*) sebagai pengganti telepon rumah semata, tetapi sudah berkembang ke arah penggunaan internet pada ponsel yang didukung oleh perkembangan teknologi *GPRS*. Selain itu perkembangan teknologi Java terutama teknologi *j2me* juga turut melengkapi perkembangan ponsel, sehingga kita dapat membuat dan menjalankan aplikasi tersebut pada ponsel. Oleh karena itu dibuat suatu aplikasi kamus bahasa Inggris pada ponsel yang berbasis pemrograman Java yang berkomunikasi dengan *server* menggunakan jaringan *GPRS*. Sehingga kita dapat mencari arti kata bahasa Inggris dan mendapatkan terjemahannya dalam bahasa Indonesia serta sebaliknya dimana saja dan kapan saja tanpa mengurangi kinerja ponsel.

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul :

“DESAIN APLIKASI KAMUS BAHASA INGGRIS PADA PONSEL MENGUNAKAN JAVA (J2ME)”

Skripsi ini merupakan persyaratan kelulusan Studi di Jurusan Teknik Elektro S-1 Konsentrasi Teknik Komputer dan Informatika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan penyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Bapak Prof. Dr. Eng. Ir. Abraham Lomi, MSEE. selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. H. Sidik Noertjahjono, MT. selaku Dekan Fakultas Teknik Industri serta Ketua Panitia Ujian Skripsi Fakultas Teknik Industri.
3. Bapak Ir. F. Yudi Limpraptono, MT. selaku Ketua Jurusan Teknik Elektro S-1.
4. Ibu Ir. TH. Mimien Mustikawati, MT. selaku Dosen Pembimbing I.
5. Bapak M. Ashar, ST., MT. selaku Dosen Pembimbing II serta Kepala Laboratorium Pemrograman Komputer dan Multimedia.
6. Ayah dan Ibu serta saudara-saudara kami yang telah memberikan do'a restu dan dorongan semangat.

7. Semua pihak yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penyusun telah berusaha semaksimal mungkin serta penulis menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Oleh karena itu penyusun mengharapkan saran dan kritik yang bersifat membangun dari para pembaca sekalian demi kesempurnaan laporan skripsi ini.

Harapan penyusun semoga laporan skripsi ini dapat memberikan manfaat bagi perkembangan ilmu pengetahuan bagi semua kalangan dan pembaca.

Malang, Maret 2009

Penyusun

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAKSI	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan.....	2
1.4. Batasan Masalah.....	3
1.5. Metode Penelitian.....	3
1.6. Sistematika Penulisan.....	4
BAB II TINJAUAN	
2.1. Kamus	5
2.1.1. Macam – Macam Kamus	5
2.1.1.1. Berdasarkan Penggunaan Bahasa	5
2.1.1.2. Berdasarkan Isi	6
2.1.1.3. Kamus Istimewa	7
2.1.2. Kosakata	9
2.2. Java	9

2.2.1. J2ME	11
2.2.1.1. <i>Configuration</i>	12
2.2.1.2. <i>Profile</i>	13
2.2.1.3. <i>CLDC</i>	13
2.2.1.4. <i>Generic Connection Framework</i>	13
2.2.1.5. <i>MIDP</i>	14
2.2.1.6. <i>MIDlet</i>	15
2.3. <i>Web Server</i>	16
2.4. <i>JSP (Java Server Pages)</i>	18
2.5. <i>WAP (Wireless Application Protocol)</i>	18
2.6. <i>GPRS (General Packet Radio Sistem)</i>	21

BAB III DESAIN DAN ANALISA SISTEM

3.1. Spesifikasi Sistem.....	25
3.1.1. Spesifikasi <i>Software</i>	26
3.1.2. Spesifikasi <i>Hardware</i>	28
3.2. Desain Keseluruhan Sistem.....	28
3.2.1. Desain Komunikasi <i>HTTP</i>	29
3.2.2. Desain <i>Content</i>	34
3.2.3. Desain <i>Server</i>	35
3.2.4. Desain <i>DataBase</i>	36
3.2.4.1. Tabel t_a_ind sampai tabel t_z_ind	36
3.2.4.2. Tabel tabel t_a_ing sampai t_z_ing	36
3.2.4.3. Tabel t_ket_ind	37

3.2.4.4. Tabel t_ket_ing	37
--------------------------------	----

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Aplikasi Kamus Bahasa Inggris	39
4.1.1. <i>Content</i> Menu Utama	39
4.1.2. <i>Content</i> Menu English – Indonesia	40
4.1.3. <i>Content</i> Menu Indonesia – English	41
4.1.4. <i>Content</i> Menu Help	42
4.1.5. <i>Content</i> Menu About	42
4.2. Implementasi Aplikasi Kamus Bahasa Inggris	43
4.2.1. Pengujian Koneksi <i>HTTP</i>	43
4.2.2. Pengujian Kata Tunggal	44
4.2.2.1. Pengujian Penerjemahan Inggris – Indonesia ..	44
4.2.2.2. Pengujian Penerjemahan Indonesia – Inggris ..	45
4.2.3. Pengujian Kata Jamak	45
4.2.3.1. Pengujian Penerjemahan Inggris – Indonesia ...	46
4.2.3.2. Pengujian Penerjemahan Inggris – Indonesia ...	47

BAB V PENUTUP

5.1. Kesimpulan	49
5.2. Saran	49

DAFTAR PUSTAKA	50
-----------------------------	----

LAMPIRAN-LAMPIRAN

DAFTAR TABEL

2.1. <i>Platform Java</i>	10
3.1. <i>ResponseCode</i> Koneksi <i>HTTP</i>	33
4.1. Pengujian penerjemahan Inggris – Indonesia untuk kata tunggal	44
4.2. Pengujian penerjemahan Indonesia – Inggris untuk kata tunggal.....	45
4.3. Pengujian penerjemahan Inggris – Indonesia untuk kata jamak	46
4.4. Pengujian penerjemahan Indonesia – Inggris untuk kata jamak	47

DAFTAR GAMBAR

2.1. Platform Java.....	10
2.2. Arsitektur J2ME	11
2.3. Hirarki Koneksi GCF (<i>Generic Connection Framework</i>)	14
2.4. Siklus Hidup <i>MIDlet</i>	16
2.5. Layer WAP	19
2.6. Arsitektur WAP	20
2.7. Arsitektur <i>GPRS</i>	23
3.1. Desain Keseluruhan Sistem	28
3.2. Analisa Komunikasi <i>HTTP</i> secara <i>Hardware</i>	29
3.3. <i>Flowchart</i> Komunikasi <i>HTTP</i>	32
3.4. Desain <i>Content</i> aplikasi kamus bahasa Inggris	35
3.5. <i>Flowchart</i> JSP	35
3.6. Desain tabel t_a_ind sampai t_z_ind pada <i>DataBase</i> db_kamus.....	36
3.7. Desain tabel t_a_ing sampai t_z_ing pada <i>DataBase</i> db_kamus.....	37
3.8. Desain tabel t_ket_ind pada <i>DataBase</i> db_kamus	37
3.9. Desain tabel t_ket_ing pada <i>DataBase</i> db_kamus	38
4.1. <i>Content</i> Menu Utama	39
4.2. <i>Content</i> menu English – Indonesia	40
4.3. Tampilan bila arti tidak ditemukan	40
4.4. <i>Content</i> menu Indonesia – English.....	41
4.5. Tampilan bila arti tidak ditemukan	41

4.6. <i>Content</i> menu Help	42
4.7. <i>Content</i> menu About.....	42
4.8. Tampilan <i>Error in Connection</i>	43

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kemajuan teknologi sekarang ini, setiap jam, menit, bahkan hitungan detik terus berjalan. Seperti yang kita ketahui teknologi tersebut banyak menggunakan bahasa asing terutama bahasa Inggris sebagai bahasa internasional. Dengan hal tersebut kita dituntut untuk menguasai bahasa Inggris. Untuk menguasai bahasa Inggris kita membutuhkan sebuah media untuk mengetahui arti dari bahasa Inggris.

Pada awalnya kita diperkenalkan buku kosakata yang biasa disebut kamus asing (Inggris), kamus ini dianggap kurang praktis dalam penggunaannya. Kemudian diciptakan sebuah alat penerjemah (*translator*) yang bisa kita pakai untuk menerjemahkan dua bahasa yakni bahasa Indonesia ke bahasa Inggris dan sebaliknya. Penggunaan alat ini memang terbilang praktis, tetapi masih dirasa kurang memuaskan yang dikarenakan harganya yang relatif mahal. Dengan perkembangan teknologi komunikasi yang sangatlah pesat, terutama teknologi *mobile communication* yang tidak terelakkan lagi, dimana yang kita tahu ponsel saat ini tidak hanya sebagai alat komunikasi saja, namun juga sebagai alat pendukung kerja kita sehari – hari.

Dengan fiturnya yang semakin komplit, saat ini terdapat aplikasi kamus yang dapat dijalankan di ponsel, sehingga dapat diakses kapanpun dan dimanapun. Namun dari aplikasi ini juga memiliki kekurangan diantaranya kosakata yang kurang lengkap karena para pembuat aplikasi harus mempertimbangkan

besarnya *DataBase* kosakata dengan kinerja ponsel. Semakin besar *DataBase* kosakata, maka kinerja ponsel semakin berkurang.

Teknologi J2ME saat ini banyak di manfaatkan untuk membangun aplikasi pada *mobile device*, karena dukungan dari *vendor* telepon selular yang kuat menyebabkan perkembangan teknologi inipun semakin pesat. Dengan menggunakan teknologi J2ME dan teknologi *internet* kita dapat merancang suatu aplikasi kamus bahasa Inggris tanpa mengurangi performa ponsel, yakni dengan memanfaatkan teknologi internet sebagai media untuk menyimpan kosakata dalam jumlah yang besar. Sehingga sangat praktis dan berguna bagi masyarakat terutama para siswa sekolah.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas maka timbul suatu perumusan masalah yang diajukan yaitu:

Bagaimana mendesain aplikasi kamus bahasa Inggris pada ponsel menggunakan J2ME tanpa mengurangi kinerja ponsel.

1.3. Tujuan

Adapun tujuan yang ingin dicapai dalam pelaksanaan skripsi ini adalah:

Membuat aplikasi kamus bahasa Inggris pada ponsel menggunakan J2ME tanpa mengurangi kinerja ponsel.

1.4. Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan maka pembahasan dibatasi pada hal-hal sebagai berikut:

1. Dalam merancang sistem hanya pada sistem aplikasi user.
2. Tidak membahas teknologi pengamanan.
3. Tidak membahas teknologi *GPRS* secara luas.
4. Tidak membahas masalah tarif dari komunikasi *GPRS*.

1.5. Metode Penelitian

Metode yang digunakan dalam penyusunan skripsi ini adalah :

1. Studi literatur, yaitu tinjauan pustaka untuk mempelajari teori – teori yang terkait dan berhubungan dengan permasalahan melalui media referensi buku, dan literatur dari *internet*.
2. Perancangan
Merancang dan mengembangkan desain kamus bahasa Inggris yang telah banyak beredar di Indonesia.
3. Pembuatan Program
Pembuatan aplikasi kamus bahasa Inggris dengan bahasa pemrograman java.
4. Implementasi & Pengujian
Merancang dan mengembangkan desain kamus bahasa Inggris berdasarkan data – data yang diperoleh.

1.6. Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan skripsi ini, maka peneliti menyajikan secara sistematis sebagai berikut:

Bab I : PENDAHULUAN

Berisi Latar Belakang, Perumusan Masalah, Tujuan, Pembatasan Permasalahan, Metodologi Penelitian dan Sistematika Penulisan.

Bab II : TINJAUAN PUSTAKA

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan pembahasan yang dilakukan, yang di dalamnya memuat teori – teori tentang kamus bahasa Inggris dan J2ME.

Bab III : DESAIN & ANALISA SISTEM

Bab ini membahas desain dan analisis dari aplikasi kamus bahasa Inggris pada *hardware* dan *software*.

Bab IV : IMPLEMENTASI DAN PENGUJIAN SISTEM

Berisi tentang Implementasi dari hasil desain aplikasi pada *emulator* dan pengujian pada aplikasi.

Bab V : PENUTUP

Berisi tentang kesimpulan yang dapat diambil berdasarkan hasil uraian pada bab – bab sebelumnya dan saran mengenai hasil yang telah diperoleh.

BAB II

TINJAUAN PUSTAKA

2.1. KAMUS

Kamus adalah sejenis buku rujukan yang menerangkan makna kata-kata. Ia berfungsi untuk membantu seseorang mengenal perkataan baru. Selain menerangkan maksud kata, kamus juga mungkin mempunyai pedoman sebutan, asal-usul (*etimologi*) sesuatu perkataan dan juga contoh penggunaan bagi sesuatu perkataan. Untuk memperjelas kadang kala terdapat juga ilustrasi di dalam kamus.

Kata kamus diserap dari bahasa Arab *qamus*, dengan bentuk jamaknya *qawamis*. Kata Arab itu sendiri berasal dari kata Yunani *okeanos* yang berarti 'lautan'. Sejarah kata itu jelas memperlihatkan makna dasar yang terkandung dalam kata kamus, yaitu wadah pengetahuan, khususnya pengetahuan bahasa, yang tidak terhingga dalam dan luasnya. Dewasa ini kamus merupakan khazanah yang memuat perbendaharaan kata suatu bahasa, yang secara ideal tidak terbatas jumlahnya. Setiap kebudayaan besar di dunia bangga akan kamus bahasanya. Dalam kenyataannya kamus itu tidak hanya menjadi lambang kebanggaan suatu bangsa, tetapi juga mempunyai fungsi.

2.1.1. Macam – Macam Kamus

2.1.1.1. Berdasarkan Penggunaan Bahasa

Kamus bisa ditulis dalam satu atau lebih dari satu bahasa. Dengan itu kamus bisa dibagi menjadi beberapa jenis yaitu:

- **Kamus Ekabahasa**

Kamus ini hanya menggunakan satu bahasa. Kata – kata yang dijelaskan dan penjelasannya adalah terdiri daripada bahasa yang sama. Kamus ini mempunyai perbezaan yang jelas dengan kamus dwibahasa kerana penyusunan dibuat berasaskan pembuktian data korpus. Ini bermaksud definisi makna ke atas kata – kata adalah berdasarkan makna yang diberikan dalam contoh kalimat yang mengandungi kata-kata berhubung. Contoh bagi kamus ekabahasa ialah *Kamus Besar Bahasa Indonesia* (di Indonesia) dan *Kamus Dewan* di (Malaysia).

- **Kamus Dwibahasa**

Kamus ini menggunakan dua bahasa, yakni kata masukan daripada bahasa yang dikamuskan diberi padanan atau pemerian takrifnya dengan menggunakan bahasa yang lain. Contohnya: *Kamus Inggris-Indonesia*, *Kamus Dwibahasa Oxford Fajar (Inggris-Melayu; Melayu-Inggris)*.

- **Kamus Aneka Bahasa**

Kamus ini sekurang – kurangnya menggunakan tiga bahasa atau lebih. Misalnya kata bahasa Melayu, bahasa Inggris, dan bahasa Mandarin secara serentak. Contoh bagi kamus aneka bahasa ialah *Kamus Melayu-Cina-Inggris Pelangi* susunan Yuen Boon Chan pada tahun 2004.

2.1.1.2. Berdasarkan Isi

Kamus bisa muncul dalam berbagai isi. Ini adalah karena kamus diterbitkan dengan tujuan memenuhi keperluan golongan tertentu. Contohnya, golongan pelajar sekolah memerlukan kamus berukuran kecil untuk memudahkan

mereka membawa kamus ke sekolah. Secara umumnya kamus dapat dibagi kepada 3 jenis ukuran:

- **Kamus Mini**

Pada zaman sekarang sebenarnya susah untuk menjumpai kamus ini. Ia juga dikenali sebagai kamus saku karena ia dapat disimpan dalam saku. Tebalnya kurang daripada 2 cm.

- **Kamus Kecil**

Kamus berukuran kecil yang biasa dijumpai, kamus ini merupakan kamus yang mudah dibawa. Kamus Dwibahasa Oxford Fajar (Inggris-Melayu; Melayu-Inggris).

- **Kamus Besar**

Kamus ini memuatkan segala *leksikal* yang terdapat dalam satu bahasa. Setiap perkataannya dijelaskan maksud secara lengkap. Biasanya ukurannya besar dan tidak sesuai untuk dibawa ke sana sini. Contohnya : Kamus Besar Bahasa Indonesia.

2.1.1.3. Kamus Istimewa

Kamus istimewa merujuk kepada kamus yang mempunyai fungsi yang khusus.

Contohnya:

- **Kamus Istilah**

Kamus ini berisi istilah – istilah khusus dalam bidang tertentu. Fungsinya adalah untuk kegunaan ilmiah. Contohnya ialah Kamus Istilah Fiqh.

- **Kamus Etimologi**

Kamus yang menerangkan asal usul sesuatu perkataan dan maksud asalnya.

- **Kamus Tesaurus (perkataan searti)**

Kamus yang menerangkan maksud sesuatu perkataan dengan memberikan kata – kata searti (sinonim) dan dapat juga kata-kata yang berlawanan arti (antonim). Kamus ini adalah untuk membantu para penulis untuk meragamkan penggunaan diksi. Contohnya, Tesaurus Bahasa Indonesia.

- **Kamus Peribahasa/Simpulan Bahasa**

Kamus yang menerangkan maksud sesuatu peribahasa/simpulan bahasa. Selain daripada digunakan sebagai rujukan, kamus ini juga sesuai untuk dibaca dengan tujuan keindahan.

- **Kamus Kata Nama Khas**

Kamus yang hanya menyimpan kata nama khas seperti nama tempat, nama tokoh, dan juga nama bagi institusi. Tujuannya adalah untuk menyediakan rujukan bagi nama – nama ini.

- **Kamus Terjemahan**

Kamus yang menyedia kata searti bahasa asing untuk satu bahasa sasaran. Kegunaannya adalah untuk membantu para penerjemah.

- **Kamus Kolokasi**

Kamus yang menerangkan tentang padanan kata, contohnya kata 'terdiri' yang selalu berpadanan dengan 'dari' atau 'atas'.

2.1.2. Kosakata

Kosakata (*vocabulary*) adalah himpunan kata yang diketahui oleh seseorang atau entitas lain, atau merupakan bagian dari suatu bahasa tertentu. Kosakata seseorang didefinisikan sebagai himpunan semua kata-kata yang dimengerti oleh orang tersebut atau semua kata – kata yang kemungkinan akan digunakan oleh orang tersebut untuk menyusun kalimat baru. Kekayaan kosakata seseorang secara umum dianggap merupakan gambaran dari intelegensi atau tingkat pendidikannya. Karenanya banyak ujian standar, seperti SAT, yang memberikan pertanyaan yang menguji kosakata.

Penambahan kosakata seseorang secara umum dianggap merupakan bagian penting, baik dari proses pembelajaran suatu bahasa ataupun pengembangan kemampuan seseorang dalam suatu bahasa yang sudah dikuasai. Murid sekolah sering diajarkan kata-kata baru sebagai bagian dari mata pelajaran tertentu dan banyak pula orang dewasa yang menganggap pembentukan kosakata sebagai suatu kegiatan yang menarik dan mendidik.

2.2. Java

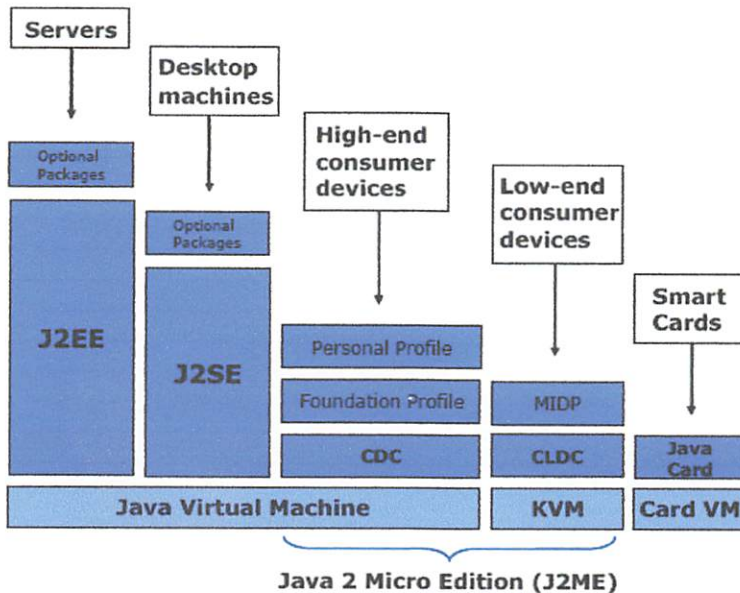
Java dibuat pada tahun 1991 oleh *James Gosling*. Pada awalnya diberi nama *Oak*, dimana untuk menghormati pohon yang ada di luar jendela *Gosling*. Kemudian namanya diubah ke Java karena telah ada sebuah bahasa yang diberi nama *Oak*. Motivasi sesungguhnya dari Java adalah kebutuhan akan sebuah bahasa yang bisa digunakan pada berbagai *platform* yang bisa dimasukkan ke dalam berbagai produk elektronik seperti pemanggang roti dan lemari es. Salah

satu dari proyek pertama yang dikembangkan menggunakan Java sebuah *remote control* yang diberi nama *Star 7*.

Pada saat yang sama, *World Wide Web* dan *internet* berkembang sangat cepat. *Gosling* menyadari bahwa Java dapat digunakan untuk pemrograman *internet*. Dengan keluarnya versi 1.2, *platform* Java telah dipilah – pilah menjadi beberapa edisi : *The Standard Edition (J2SE)*, *Enterprise Edition (J2EE)*, *Mobile Edition (J2ME)*, dan *JavaCard API*.

Tabel 2.1. Platform Java

J2SE – Java 2 Platform, Standard Edition	Aplikasi Desktop
J2EE – Java 2 Platform, Enterprise Edition	Aplikasi enterprise dengan fokus pada pengembangan sisi webserver, termasuk servlet, JSP, EJB, dan XML
J2ME – Java 2 Platform, Micro Edition	Perangkat Mobile
JavaCard	Smart Cards



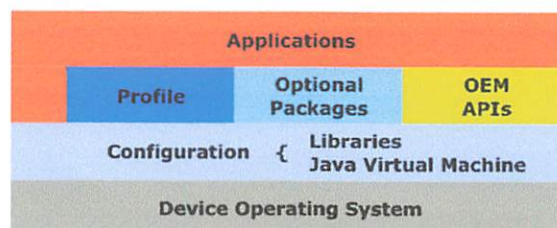
Gambar 2.1. Platform Java

2.2.1. J2ME (*JAVA2 MICRO EDITION*)

J2ME adalah satu set spesifikasi dan teknologi yang fokus kepada perangkat konsumen. Perangkat ini memiliki jumlah memori yang terbatas, menghabiskan sedikit daya dari baterai, layar yang kecil dan bandwidth jaringan yang rendah.

Dengan perkembangan perangkat *mobile* konsumen dari telepon, PDA, kotak permainan ke peralatan-peralatan rumah, Java menyediakan suatu lingkungan yang *portable* untuk mengembangkan dan menjalankan aplikasi pada perangkat ini. Program J2ME, seperti semua program Java diterjemahkan oleh *Virtual Machine*. Program-program tersebut *dikompile* ke dalam *bytecode* dan diterjemahkan dengan *Java Virtual Machine* (JVM). Ini berarti bahwa program-program tersebut tidak berhubungan langsung dengan perangkat.

J2ME menyediakan suatu *interface* yang sesuai dengan perangkat. Aplikasi-aplikasi tersebut tidak harus *dikompile* ulang supaya mampu dijalankan pada mesin yang berbeda. Inti dari J2ME terletak pada *configuration* dan *profile-profile*. Suatu *configuration* menggambarkan lingkungan *runtime* dasar dari suatu sistem J2ME. Ia menggambarkan *core library*, *virtual machine*, fitur keamanan dan jaringan.



Gambar 2.2. Arsitektur J2ME

Sebuah *profile* memberikan *library* tambahan untuk suatu kelas tertentu pada sebuah perangkat. *profile-profile* menyediakan *User Interface* (UI) API, *Persistence*, *Messaging Library*, dan sebagainya. Satu set *library* tambahan atau *package* tambahan menyediakan kemampuan program tambahan. Pemasukan *package* ini ke dalam perangkat J2ME dapat berubah-ubah karena tergantung pada kemampuan sebuah perangkat. Sebagai contoh, beberapa perangkat MIDP tidak memiliki *Bluetooth built-in*, sehingga *Bluetooth API* tidak disediakan dalam perangkat ini.

2.2.1.1. Configuration

Suatu *configuration* menggambarkan fitur minimal dari lingkungan lengkap Java *Runtime*. Untuk menjamin kemampuan *portabilitas* dan *interoperabilitas* optimal diantara berbagai macam perangkat yang dibatasi sumber dayanya (memory, prosesor, daya, koneksi yang dibatasi), *Configuration* tidak menggambarkan fitur tambahan. Suatu *Configuration* J2ME menggambarkan suatu komplemen yang minimum dari teknologi Java merupakan tugas *profile – profile* untuk menggambarkan tambahan *library* untuk suatu kategori perangkat tertentu.

Configuration menggambarkan:

- Subset bahasa pemrograman Java
- Kemampuan *Java Virtual Machine* (JVM)
- *Core platform libraries*
- Fitur keamanan dan jaringan

2.2.1.2.Profile

Suatu *Profile* menggambarkan set-set tambahan dari API dan fitur untuk pasar tertentu, kategori perangkat atau industri. Sementara *Configuration* menggambarkan *library* dasar, *profile-profile* menggambarkan *library* yang penting untuk membuat aplikasi-aplikasi efektif. *Library* ini memasukkan *user interface*, jaringan dan penyimpanan API.

2.2.1.3.CLDC (Connected Limited Device Configuration)

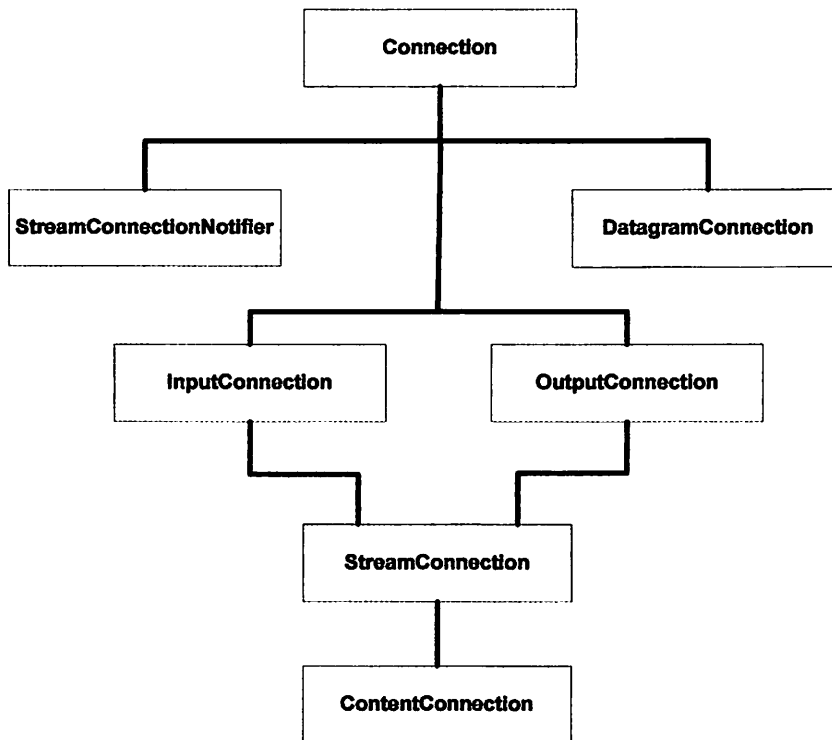
Connected Limited Device Configuration (CLDC) menggambarkan dan menunjuk pada area berikut ini:

- Fitur Bahasa Java dan *Virtual Machine* (VM)
- *Library* dasar (java.lang.*,java.util.*)
- *Input/Output* (java.io.*)
- Keamanan
- Jaringan
- *Internationalization*

2.2.1.4.Generic Connection Framework (GCF)

The *Generic Connection Framework* menyediakan API dasar untuk koneksi dalam CLDC. *Framework* ini menyediakan suatu pondasi umum untuk koneksi seperti *HTTP*, *Socket*, dan *Datagrams*. GCF menyediakan suatu set API yang umum dan biasa yang memisahkan semua jenis koneksi. Perlu dicatat bahwa tidak semua jenis koneksi dibutuhkan untuk diterapkan oleh perangkat MIDP. Hirarki *interface* yang dapat diperluas dari GCF membuat proses penyamarataan

menjadi mungkin. Jenis koneksi baru mungkin bisa ditambahkan ke dalam *framework* ini dengan memperluas hirarki ini.



Gambar 2.3. Hirarki Koneksi GCF (*Generic Connection Framework*)

2.2.1.5. *MIDP (Mobile Information Device Profile)*

The Mobile Information Device Profile (MIDP) menggambarkan model aplikasi, UI API, penyimpanan dan jaringan yang kuat, permainan dan media API, kebijakan keamanan, penyebaran aplikasi dan ketentuan *over-the-air*.

Spesifikasi *MIDP* menggambarkan suatu perangkat *MIDP* yang memiliki karakteristik – karakteristik seperti di bawah ini sebagai batas minimum:

- Tampilan:
 - Ukuran Layar: 96x54
 - kedalaman tampilan: 1-bit
 - Ketajaman pixel: sekitar 1:1

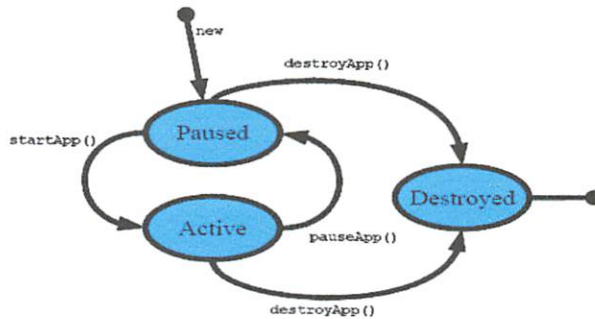
- **Masukan:**
 Satu atau lebih mekanisme *user-input*: satu *keyboard*, dua *keyboard*, atau *touch screen*.
- **Memory:**
 256 Kilobytes of *non-volatile memory* untuk implementasi *MIDP*
 8 Kilobytes of *non-volatile memory* untuk aplikasi
 128 Kilobytes of *volatile memory* untuk *Java runtime* (e.g., the *Java heap*)
- **Jaringan:**
 dua jalur, *wireless*, *bandwidth* terbatas
- **Sound:**
 Kemampuan untuk memainkan nada – nada

2.2.1.6.MIDlet

Suatu aplikasi *MIDP* disebut *MIDlet*. Perangkat *application management software* (AMS) berinteraksi langsung dengan *MIDlet* dengan *method MIDlet create, start, pause, dan destroy*. *MIDlet* adalah bagian dari *package javax.microedition.midlet*. Sebuah *MIDlet* harus di-*extend* dengan *class MIDlet*. Dan dapat meminta parameter dari AMS seperti dirumuskan dalam *Java Application Descriptor (JAD)*. Suatu *MIDlet* tidak harus memiliki (dan memang harus tidak mempunyai) sebuah *method public static void main(String[] argv)*. *Method* tersebut tidak akan dikenal lagi oleh AMS sebagai titik awal sebuah program.

Siklus MIDlet

Kehidupan *MIDlet* dimulai ketika di-instantiate oleh AMS. *MIDlet* pada awalnya masuk status “*Paused*” setelah perintah baru dibuat. AMS memanggil *constructor public* tanpa argumen dari *MIDlet*. Jika sebuah *exception* terjadi dalam *constructor*, *MIDlet* memasuki status “*Destroyed*” dan membuangnya segera. *MIDlet* masuk ke dalam status “*Active*” atas pemanggilan *method startUp()* oleh AMS. *MIDlet* masuk ke dalam status “*Destroyed*” ketika AMS memanggil *method destroyApp()*. Status ini juga kembali diakses ketika *method notifyDestroyed()* kembali dengan sukses kepada aplikasi. Dengan catatan bahwa *MIDlet* hanya bisa memasuki status “*Destroyed*” sekali dalam masa hidupnya.



Gambar 2.4. Siklus hidup MIDlet

2.3. Web Server

Server HTTP Apache Tomcat atau Server Web/WWW Apache Tomcat adalah server web yang dapat dijalankan di banyak sistem operasi (*Unix, BSD, Linux, Microsoft Windows dan Novell Netware* serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP. Apache Tomcat memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi,

otentikasi berbasis *DataBase* dan lain-lain. *Apache Tomcat* juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan *server* menjadi mudah. *Apache Tomcat* merupakan perangkat lunak terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang – pengembang dibawah naungan *Apache Software Foundation*.

Pada awal mulanya, *Apache* merupakan perangkat lunak terbuka yang menjadi alternatif dari *server web Netscape* (sekarang dikenal sebagai Sun Java Sistem Web Server). Sejak April 1996 *Apache* menjadi *server web* terpopuler di *internet*. Pada Mei 1999, *Apache* digunakan di 57% dari semua *web server* di dunia. Pada November 2005 persentase ini naik menjadi 71%. (sumber: Netcraft Web Server Survey, November 2005). Asal mula nama *Apache* berasal ketika sebuah *server web* populer yang dikembangkan pada awal 1995 yang bernama NCSA *HTTPd* 1.3 memiliki sejumlah perubahan besar terhadap kode sumbernya (*patch*). Karena banyaknya *patch* pada perangkat lunak tersebut sehingga disebut sebuah server yang memiliki banyak *patch* ("a patchy" server). Tetapi pada halaman FAQ situs web resminya, disebutkan bahwa "*Apache*" dipilih untuk menghormati suku asli *Indian Amerika Apache* (Indé), yang dikenal karena keahlian dan strategi perangnya. Versi 2 dari *Apache* ditulis dari awal tanpa mengandung kode sumber dari NCSA.

2.4. JSP (*Java Server Pages*)

JSP merupakan teknologi yang didasarkan pada bahasa Java, yang digunakan untuk membentuk halaman – halaman web yang bersifat dinamis. Dimana teknologi ini dikembangkan oleh *Sun Microsystems*.

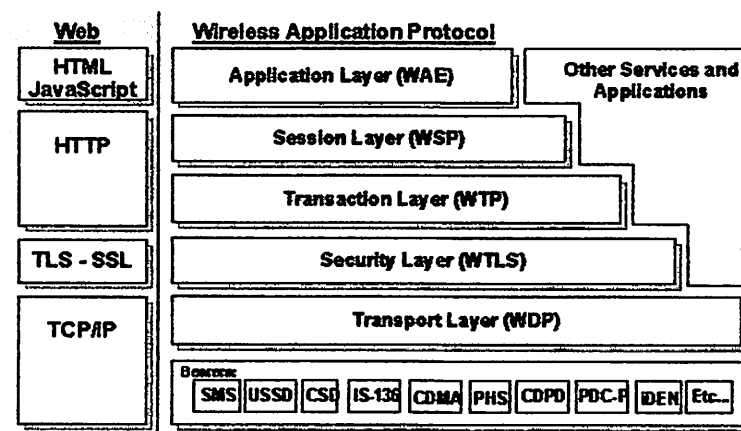
Berbeda dengan *applet*, suatu fitur pada bahasa Java yang memungkinkan pengembang membuat aplikasi web yang dieksekusi pada sisi *client*, JSP menggunakan pendekatan pemrosesan di sisi *server*. Pada model ini, *SourceCode JSP* dijalankan pada *Web Server*. Salah satu keuntungan pada model ini adalah memungkinkan untuk membuat aplikasi yang *independent* terhadap keberadaan sistem Java di sisi *client*.

Pada dasarnya pemakai yang ingin mengakses halaman web mula – mula mengirimkan *request* melalui protokol *HTTP* dalam bentuk JSP (berekstensi.jsp), setelah itu permintaan akan disampaikan pada *web server*. Kemudian *web server* mengambil dokumen JSP dan mengirimkan ke *JSP Servlet Engine*, bagian inilah yang melakukan pemrosesan kode – kode JSP (termasuk di dalamnya proses pengompilasian) dan membentuk kode *HTML*. Berikutnya, kode *HTML* ini disampaikan oleh *web server* ke *client* yang memintanya, kode *HTML* selanjutnya diproses oleh *browser* sehingga pemakai bisa memperoleh informasi dari halaman web yang dikendakinya.

2.5. WAP (*Wireless Application Protocol*)

WAP merupakan standar internasional terbuka untuk aplikasi yang menggunakan komunikasi *wireless*. Tujuan utamanya untuk membangun aplikasi yang dapat mengakses *internet* dari telepon genggam atau PDA. Ada beberapa

versi WAP antara lain WAP 1.2.1 dan 2.0. WAP 1.2.1 hanya dapat menampilkan laman sederhana saja dibandingkan dengan WAP 2.0 yang mendukung bahasa *XHTML* dan gambar. WAP dibuat pertama kali sebagai protokol komunikasi bergerak yang tidak bergantung pada sistem tertentu. WAP dirancang sebagai bagian dari sistem di masa depan sama halnya dengan *Bluetooth* dan *GPRS*. WAP merupakan protokol komunikasi bergerak yang terdiri dari beberapa *layer* dan dapat dijalankan pada sistem jaringan yang berbeda.



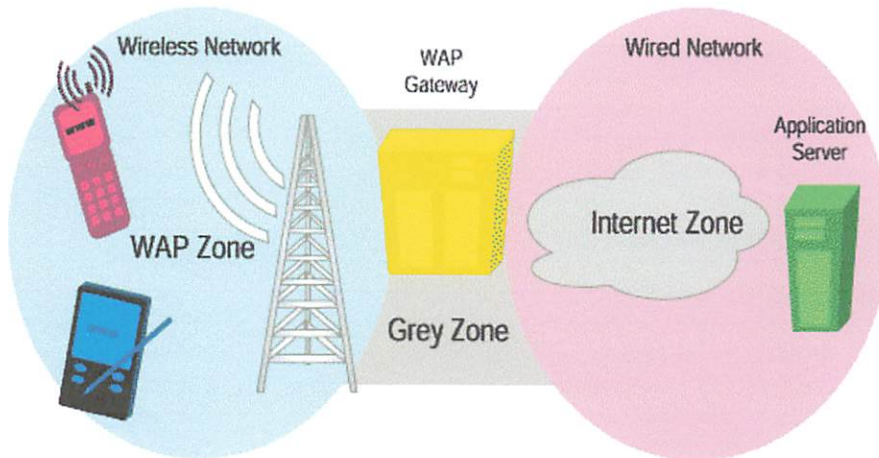
Gambar 2.5. Layer WAP

WAP sendiri terdiri dari 5 layer, yaitu :

1. *Wireless Application Environment* (WAE),
2. *Wireless Session Protocol* (WSP),
3. *Wireless Transport Protocol* (WTP),
4. *Wireless Data Protocol* (WDP),
5. *Wireless Transport Layer Security* (WTLS).

Dari kelima layer tersebut yang terlihat hanyalah WAE, yang menampilkan isi web ke layar. WAE terdiri atas *Wireless Markup Language* (WML) sebuah variant *HTML* yang biasa digunakan untuk menampilkan isi *web*

di monitor. WML terdiri dari teks dan *hyperlink*, tetapi bukan grafik. Ketika halaman WAP terlihat di mini *browser* pada ponsel, bagian lain dari WAP menjalankan bagian peranan penting di belakang layar. *Wireless Session Protocol* membuka dan menutup koneksi ke situs WAP. *Wireless Transport Protocol* membuat paket data berjalan sesuai dengan arahnya. Koneksi *wireless* kurang dapat diandalkan dibandingkan dengan koneksi kabel. Sehingga untuk menjaga agar data yang anda kirim dan terima berjalan dengan benar sangatlah penting. *Wireless Layer Security*, sebuah bagian dari *Secure Socket layer* yang biasanya digunakan untuk keamanan di web, mengompres dan mengenkripsi data yang dikirim dari piranti wireless.



Gambar 2.6. Arsitektur WAP

Saat kita terkoneksi ke jaringan *wireless* dan meminta akses ke sebuah situs *web* yang mendukung WAP, ponsel akan mengirimkan permintaan tersebut ke gelombang radio ke sel terdekat dan langsung dirutekan ke *internet* melalui sebuah *server gateway*. *Server gateway* tersebut menerjemahkan permintaan ke format standar *HTTP* dan meneruskannya ke situs *web*. Jika situs tersebut

merespon, maka ia akan mengirimkan dokumen *HTML* ke *server gateway* kemudian dikoneksikan ke WML dan merutekannya ke antena terdekat. Antena tersebut akan mengirim melalui gelombang radio ke piranti WAP dan akhirnya *micro browser* akan menampilkan halaman tersebut.

2.6. ***GPRS (General Packet Radio System)***

GPRS merupakan sistem transmisi berbasis paket untuk *GSM* yang menggunakan prinsip '*tunnelling*'. Ia menawarkan laju data yang lebih tinggi, laju datanya secara kasar sampai 160 Kbps dibandingkan dengan 9,6 Kbps yang dapat disediakan oleh rangkaian tersakelar *GSM*. Kanal – kanal radio ganda dapat dialokasikan bagi seorang pengguna dan kanal yang sama dapat pula digunakan secara berbagi ('*sharing*') di antara beberapa pengguna sehingga menjadi sangat efisien.

Dari segi biaya, pentarifan diharapkan hanya mengacu pada volume penggunaan. Penggunanya ditarik biaya dalam kaitannya dengan banyaknya *Byte* yang dikirim atau diterima, tanpa memperdulikan panggilan, dengan demikian dimungkinkan *GPRS* akan menjadi lebih cenderung dipilih oleh pelanggan untuk mengaksesnya daripada layanan-layanan *IP*. *GPRS* merupakan teknologi baru yang memungkinkan para operator jaringan komunikasi bergerak menawarkan layanan data dengan laju bit yang lebih tinggi dengan tarif rendah, sehingga membuat layanan data menjadi menarik bagi pasar massal. Para operator jaringan komunikasi bergerak di luar negeri kini melihat *GPRS* sebagai kunci untuk mengembangkan pasar komunikasi bergerak menjadi pesaing baru di lahan yang pernah menjadi milik jaringan kabel, yakni layanan *internet*. Kondisi ini

dimungkinkan karena ledakan penggunaan *internet* melalui jaringan kabel (telepon) dapat pula dilakukan melalui jaringan bergerak. Sebagai gambaran kecil, layanan bergerak yang kini menjadi sukses di pasar (bagi operator di manca negara) misalnya adalah, laporan cuaca, pemesanan makanan, berita olah raga sampai ke informasi seperti berita – berita penting harian.

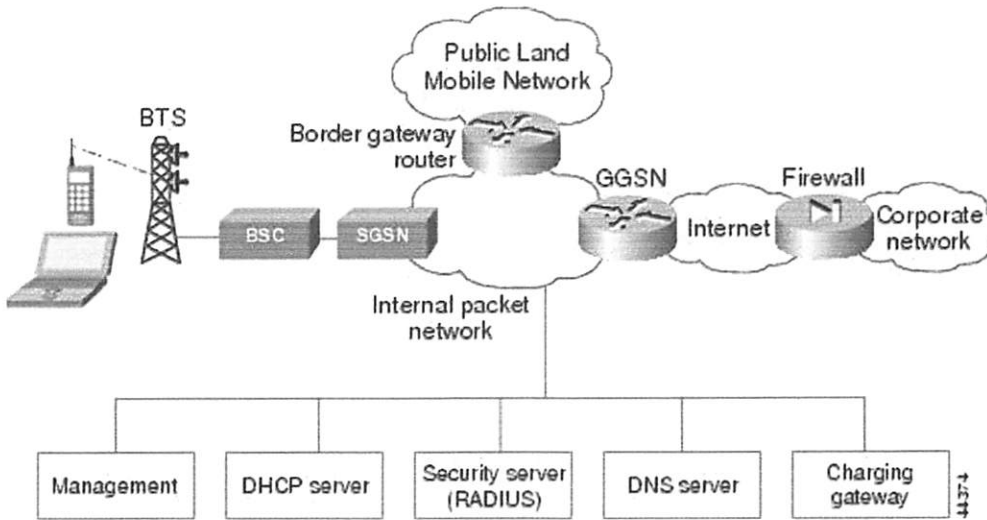
Dalam teorinya *GPRS* menjanjikan kecepatan mulai dari 56 Kbps sampai 115 Kbps, sehingga memungkinkan akses *internet*, pengiriman data multimedia ke komputer, *notebook* dan *handheld computer*. Namun, dalam implementasinya, hal tersebut sangat tergantung faktor – faktor sebagai berikut:

- Konfigurasi dan alokasi *time slot* pada level BTS
- *Software* yang dipergunakan
- Dukungan fitur dan aplikasi ponsel yang digunakan

Ini menjelaskan mengapa pada saat-saat tertentu dan di lokasi tertentu akses *GPRS* terasa lambat, bahkan lebih lambat dari akses CSD yang memiliki kecepatan 9,6 Kbps. Komponen-komponen utama jaringan *GPRS* adalah :

- GGSN (*Gateway GPRS Support Node*): gerbang penghubung jaringan *GPRS* ke jaringan *internet*. Fungsi dari komponen ini adalah sebagai interface ke PDN (Public Data Network), *information routing, network screening, user screening, address mapping*.
- SGSN (*Serving GPRS Support Node*): gerbang penghubung jaringan BSS/BTS ke jaringan *GPRS*. Komponen ini berfungsi untuk mengantarkan paket data ke MS, *update* pelanggan ke HLR, registrasi pelanggan baru.

- PCU : komponen di level BSS yang menghubungkan terminal ke jaringan GPRS



Gambar 2.7. Arsitektur GPRS

SGSN bertugas : 1. Mengirim paket ke *Mobile Station* (MS) dalam satu area 2. Mengirim sejumlah pertanyaan ke HLR untuk memperoleh *profile* data pelanggan GPRS (*management mobility*) 3. Mendeteksi MS-GPRS yang baru dalam suatu area servis yang menjadi tanggung jawabnya (*location management*) 4. SGSN dihubungkan ke BSS pada GSM dengan koneksi *Frame Relay* melalui PCU (*Packet Control Unit*) di dalam BSC

GGSN bertugas : 1. Sebagai *interface* ke jaringan IP external seperti : *public internet* atau *mobile service provider* 2. Meng-*update* informasi *routing* dari PDU (*Protokol Data Units*) ke SGSN.

GPRS menggunakan sistem komunikasi *packet switch* sebagai cara untuk mentransmisikan datanya. *Packet switch* adalah sebuah sistem di mana data yang akan ditransmisikan dibagi menjadi bagian-bagian kecil (paket) lalu

ditransmisikan dan diubah kembali menjadi data semula. Sistem ini dapat mentransmisikan ribuan bahkan jutaan paket per detik. Transmisi dilakukan melalui PLMN (*Public Land Mobile Network*) dengan menggunakan *IP backbone*. Karena memungkinkan untuk pemakaian kanal transmisi secara bersamaan oleh pengguna lain maka biaya akses *GPRS*, secara teori lebih murah daripada biaya akses *CSD*.

GPRS didesain untuk menyediakan layanan *transfer packet* data pada jaringan *GSM* dengan kecepatan yang lebih baik dari *GSM*. Kecepatan yang lebih baik ini didapat dengan menggunakan *coding scheme* (*CS*) yang berbeda dari *GSM*.

BAB III

DESAIN DAN ANALISA SISTEM

Dalam bab ini akan dibahas desain dan analisa sistem. Analisa akan dilakukan pada setiap blok desain, Desain disini yang dibuat adalah desain dari aplikasi *mobile* dalam melakukan pengaksesan kamus bahasa Inggris.

Server meliputi *database*, dan JSP penerima *request* dari *client*, sedangkan *client* meliputi *content* kamus bahasa Inggris, analisa masing-masing blok desain, secara garis besar terdapat dua bagian analisa yaitu : Analisa secara *hardware*, Analisa secara *software*.

Pada analisa secara *hardware* meliputi cara kerja komunikasi dari handphone hingga ke *server* secara *hardware* pada setiap blok desain, sedangkan analisa secara *software* meliputi algoritma & tahapan program yang digunakan pada setiap blok desain.

3.1. Spesifikasi Sistem

Sistem kamus bahasa Inggris yang dibangun melakukan komunikasi antara hardware (ponsel) dimana di dalam ponsel diinstall aplikasi kamus bahasa Inggris yang telah didesain, dari aplikasi itulah *user* melakukan komunikasi dengan *web server*, melakukan berbagai kegiatan sesuai dengan *content* yang ada, di *server* juga terdapat aplikasi untuk menerima paket data dari *user*, mengolah dan membandingkan dengan *DataBase* lalu mengirimkan kembali pada *user*. Maka adapun spesifikasi *hardware* dan *software* pada sistem kamus bahasa Inggris sebagai berikut:

3.1.1. Spesifikasi *Software*

Dalam melakukan desain aplikasi kamus bahasa Inggris dibutuhkan beberapa *software* pendukung yang mampu melengkapi dan mendukung kinerja aplikasi kamus bahasa Inggris yang di desain baik itu di sisi *server* ataupun di sisi *client*, adapun spesifikasi software di sisi *server* dan *client* sebagai berikut: *Server side* (*Web Server*), dan *Client side* (*Development application & Testing application*).

- ***Server Side***

- Web Server Apache Tomcat*

Untuk mewujudkan koneksi *online* maka digunakan sebuah *domain* dan *hosting* yang berfungsi sebagai *web server*, dimana di dalam *web server* ini nantinya disimpan *DataBase* serta aplikasi *JSP* untuk komunikasi dengan *user*, *server* ini menggunakan sistem operasi *unix* adapun spesifikasi *hosting* yang dimiliki adalah sebagai berikut :

- *Web Server* : *Apache Tomcat 5.5*
- *Mysql Version* : 4
- *JSP Version* : 2.0
- *Support* : CGI, SSI, Perl

Web server inilah yang nantinya diakses oleh aplikasi kamus bahasa Inggris, *server* ini diprogram untuk bisa menerima, memproses dan membandingkan dengan *database* dan perintah yang ada sesuai dengan permintaan yang di kirim oleh *user*, dan nantinya hasil dari proses dikirimkan lagi pada *user* kamus bahasa Inggris.

- ***Client Side***

Development Application

Dalam mendesain kamus bahasa Inggris dibutuhkan software pembangun aplikasi yang terdiri dari :

- ***JDK (Java Development Kit)***

Lingkungan pembangun java yang di gunakan adalah J2SDK 1.6 dimana selain terdapat *tool* untuk *develop* aplikasi Java juga terdapat *Java Runtime Environment* yaitu sebuah lingkungan untuk menjalankan aplikasi Java.

- ***JWTK (Java Wireless Tool Kit)***

JWTK merupakan sebuah lingkungan khusus dari Java untuk membangun aplikasi *wireless* dalam hal ini aplikasi yang berjalan pada *device* yang memiliki portabilitas tinggi. JWTK juga menyediakan lingkungan *emulator*, dan berbasiskan pada *CLDC* dan *MIDP*. Pada dasarnya JWTK adalah program yang meniru kerja ponsel yang mendukung MIDP atau yang biasa disebut *emulator*. Sehingga, belum tentu MIDlet yang berjalan pada *emulator* juga berjalan pada ponsel sebenarnya, karena juga bergantung pada kemampuan dan kapasitas ponsel yang digunakan.

Testing Application

Setelah dilakukan pembangunan *software* maka dibutuhkan *emulator* untuk testing aplikasi sebelum kita testing pada *hardware* yang sesungguhnya ini dilakukan agar jika terjadi *error* program tidak sampai mengganggu kinerja *hardware*, dan kesalahan bisa di tangani lebih awal.

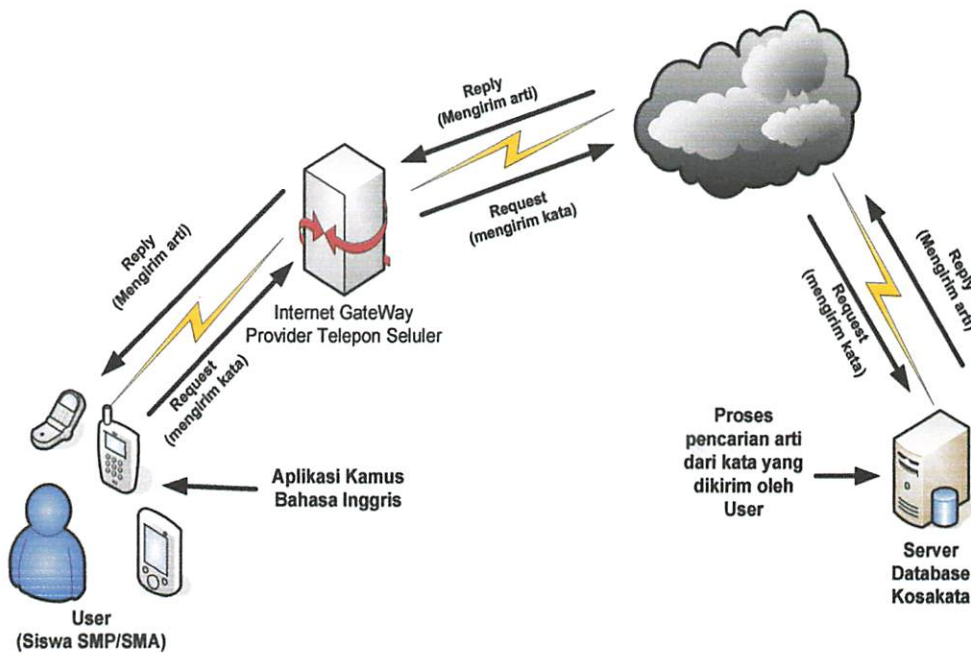
3.1.2. Spesifikasi *Hardware*

Setelah aplikasi yang dibuat selesai maka nantinya dilakukan implementasi pada *hardware* yang sesungguhnya, *hardware* yang di pakai adalah ponsel dimana pada ponsel tersebut kita *install* aplikasi berekstensi **.jar**, adapun spesifikasi untuk lingkungan Java pada ponsel sebagai berikut:

- CLDC Version : 1.1
- MIDP Version : 2.0

3.2. Desain Keseluruhan Sistem

Perancangan desain aplikasi kamus bahasa Inggris agar dapat dilakukan secara sistematis dan terstruktur maka perlu dibuat blok sistem yang menjelaskan dari sistem yang dirancang. Secara garis besar perancangan desain sistem ditunjukkan pada gambar berikut :



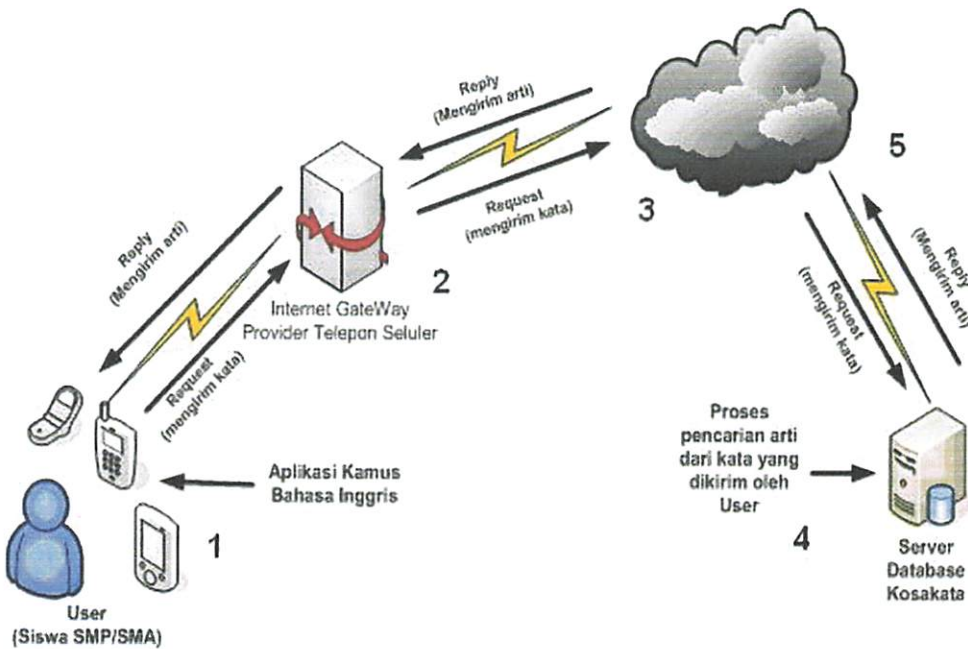
Gambar 3.1. Desain keseluruhan sistem

3.2.1. Desain Komunikasi *HTTP*

Web server membuka *port* 80 untuk berkomunikasi dengan dunia luar, maka dalam melakukan komunikasi dengan *web server* aplikasi *user* memanfaatkan *port* tersebut, dimana *request* dilakukan dengan memetakan *URL* yang telah ditanam pada *script* Java dan melakukan *request* dengan metode *GET*.

- Analisa *Hardware*

Analisa Hardware ini membahas aliran koneksi secara *hardware* dimana aliran data melewati beberapa proses sebelum diterima oleh *server*, adapun aliran sebagai berikut:



Gambar 3.2. Analisa komunikasi *HTTP* secara *hardware*

Keterangan :

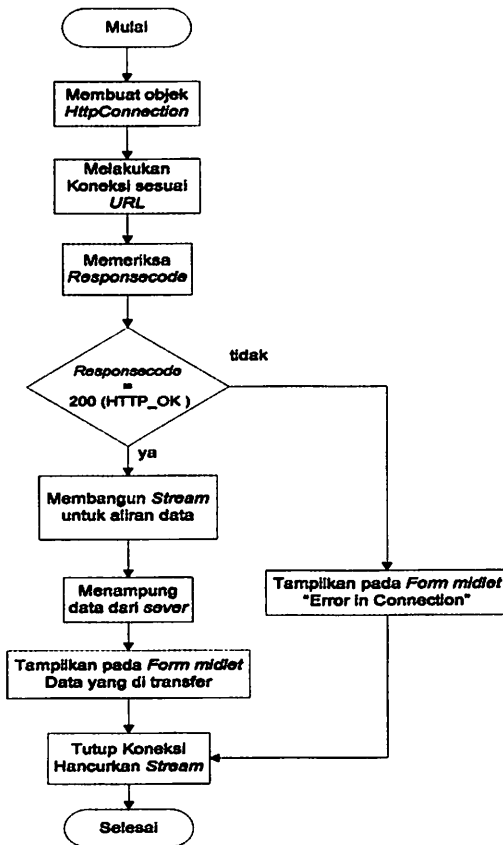
1. *User* dengan ponsel yang mendukung J2ME diinstall aplikasi kamus bahasa Inggris, kemudian melakukan *request* ke *server*, dengan

mengakses *port HTTP* 80 dalam komunikasi dengan *server*. Teknologi transfer data yang digunakan pada provider *GSM* adalah *GPRS*, awalnya *request* dari user adalah berupa sinyal *GPRS*.

2. Pada *provider GSM request* dari *user* yang berupa sinyal *GPRS* akan di terjemahkan menjadi *request* dengan format *HTTP* yang nantinya berkomunikasi dengan *internet* melalui *protocol wireless* (*WAP*).
3. *Request* dari user yang telah di terjemahkan oleh *provider* ke dalam format *HTTP* lalu di alamatkan sesuai dengan *URL* yang telah dikodekan di dalam aplikasi *user*, dalam hal ini user melakukan *request* ke sebuah *web server* yang telah di persiapkan di salah satu *hosting* yang telah dibuat.
4. *Server HTTP* membuka port 80 untuk berkomunikasi dengan pihak lain, dalam kondisi normal maka *server* bersikap *listening*, menunggu paket yang masuk, setelah paket dari *user* diterima oleh *server* maka *server* akan memberikan respon berupa koneksi oke (*HTTP_OK 200*), setelah aplikasi menerima respon tersebut maka aplikasi *user* pada Java membentuk hubungan khusus dalam hal ini *Stream*, untuk aliran data khusus antara *user* dan *server*.
5. Lalu *server* melakukan *reply* kepada *user* melalui *route internet* untuk dikembalikan pada alamat pengirim dalam hal ini *provider GSM*, masih lewat hubungan *Stream* yang diciptakan Java, lalu dari *provider GSM* selanjutnya di terjemahkan kembali dalam bentuk sinyal *GPRS* dan di terima kembali oleh *user*, setelah *user* menerima *reply* dari *server* maka hubungan stream tadi di hancurkan oleh *DestroyApp* pada *packet* Java.

- **Analisa Software**

Komunikasi dengan web *server* yang dilakukan oleh aplikasi user dalam hal ini lebih mengacu pada pembangunan *software* J2ME, dalam melakukan komunikasi dengan *server*, algoritma yang digunakan adalah dengan melemparkan paket ke sebuah *server* sesuai dengan *URL* aplikasi, setelah itu jika *server* itu ada maka *server* tersebut akan memberikan nilai kembalian berupa *responsecode* jika *responsecode* = 200 itu berarti koneksi bisa dilakukan, tetapi jika bukan 200 maka koneksi tidak bisa dilakukan, dalam melakukan koneksi tersebut program aplikasi memanfaatkan API dari java yang support dalam melakukan *HttpConnection*, dimana koneksi akan diciptakan dari melalui objek untuk mengakses API komunikasi, API inilah yang nantinya menangani komunikasi *HTTP* terlihat pada *flowchart* berikut:



Gambar 3.3. *Flowchart komunikasi HTTP*

Tahapan yang dilakukan dalam melakukan komunikasi dengan web server *HTTP* sebagai berikut:

0. Inisialisasi Sistem.
1. Membangun objek dari kelas *HTTPConnection*, dimana kelas ini berfungsi untuk mengatur koneksi *HTTP*.
2. Membangun koneksi ke suatu alamat sesuai dengan *URL (Uniform Locator Resource)*.
3. Dalam melakukan komunikasi dengan *server*, maka sebelum melakukan komunikasi sesungguhnya ada yang dinamakan pengecekan untuk mengecek apakah alamat yang di tuju sesuai *URL* benar-benar ada selain

itu juga untuk mengecek kondisi jaringan, biasanya *server* memberikan respon berupa kode untuk menggambarkan kondisi yang ada di *server* adapun beberapa *responscode* dari *server* sebagai berikut :

Tabel 3.1. ResponseCode Koneksi HTTP

Code	Response
200	<u>HTTP OK</u>
203	<u>HTTP NOT AUTHORITATIVE</u>
204	<u>HTTP NO CONTENT</u>
305	<u>HTTP USE PROXY</u>
400	<u>HTTP BAD REQUEST</u>
403	<u>HTTP FORBIDDEN</u>
404	<u>HTTP NOT FOUND</u>
405	<u>HTTP BAD METHOD</u>
406	<u>HTTP NOT ACCEPTABLE</u>
408	<u>HTTP CLIENT TIMEOUT</u>
409	<u>HTTP CONFLICT</u>
410	<u>HTTP GONE</u>
413	<u>HTTP ENTITY TOO LARGE</u>
414	<u>HTTP REQ TOO LONG</u>
415	<u>HTTP UNSUPPORTED TYPE</u>
416	<u>HTTP UNSUPPORTED RANGE</u>
417	<u>HTTP EXPECT FAILED</u>
500	<u>HTTP INTERNAL ERROR</u>
501	<u>HTTP NOT IMPLEMENTED</u>
502	<u>HTTP BAD GATEWAY</u>
503	<u>HTTP UNAVAILABLE</u>
504	<u>HTTP GATEWAY TIMEOUT</u>

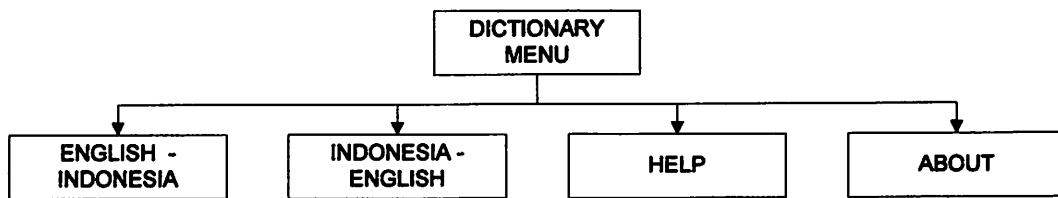
4. Jika *ResponsCode* yang di kirim oleh *server* bukan 200 maka bisa dikatakan koneksi yang terjadi tidak bisa di lanjutkan karena bermasalah, kita tidak perlu mengetahui apa masalahnya yang jelas jika *responsCode* bukan 200 maka koneksi putus dan java membangkitkan *error message* pada form berupa “Error in Connection”, sebaliknya jika *responsCode* yang di kirim oleh *server* adalah 200 maka koneksi bisa dilakukan dan aplikasi Java *user* bisa berkomunikasi dengan *web server*.

5. Setelah *server* mengirim *responsCode* dan kode itu bernilai 200 maka Java membuat suatu stream dari kelas *InputStreamReader* yang berfungsi untuk melakukan aliran data khusus koneksi ini saja.
6. Setelah *Stream* terbentuk maka *server* menerima *request* dari user dan memprosesnya, setelah *server* memproses maka *server* mengirimkan hasil dari proses, yang mana kiriman dari *server* tersebut ditampung dalam *Buffer* yang terdapat dalam *InputStreamReader*.
7. Pemeriksaan terhadap kiriman *server*, apakah kiriman *server* berisi data atau tidak.
8. Jika hasil dari *server* kosong atau tidak berisi apa-apa bisa saja akses terhadap file JSP tujuan user tidak ada dan pada saat itu akan di bangkitkan *error message* oleh Java, sedangkan jika kiriman dari *server* berisi data maka ditampilkan pada *form MIDlet*.
9. Setelah *request* dari *user* di terima dan proses pengiriman hasil proses *server* telah di dapatkan oleh aplikasi *user* maka hubungan untuk satu koneksi ini di tutup dan *Stream* otomatis di hancurkan.

3.2.2. Desain Content

Dalam sebuah aplikasi kamus ada hal yang harus di perhatikan yaitu *content*, jika kita berbicara *content* tentunya harus sesuai dengan kebutuhan aplikasi jangan sampai terdapat *content* yang tidak berguna, oleh karena itu dibuat *content* aplikasi kamus bahasa Inggris yang dapat memenuhi segala kegiatan *user*, dimana dengan *content – content* inilah aplikasi kamus menjadi lebih terorganisir, karena setiap kebutuhan telah didesain sedemikian rupa sehingga memudahkan

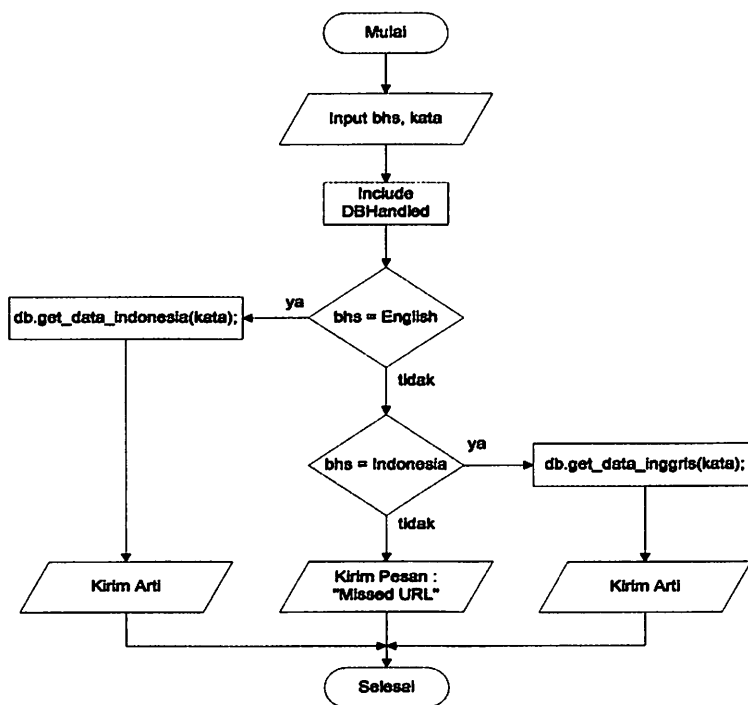
user dalam melakukan pengaksesan kamus bahasa Inggris secara *mobile*, adapun rancangan *content* dalam aplikasi kamus bahasa Inggris adalah sebagai berikut :



Gambar 3.4. Desain *content* aplikasi kamus bahasa Inggris

3.2.3. Desain Server

Dalam Melakukan komunikasi dengan *server*, aplikasi kamus bahasa Inggris melakukan *request* pada *server* kemudian *server* memberikan *reply* pada aplikasi kamus bahasa Inggris, *request* yang dilakukan oleh aplikasi kamus bahasa Inggris diolah terlebih dahulu oleh *script* JSP dan dilakukan koneksi ke *DataBase*, adapun *flowchart* JSP di sisi *server* sebagai berikut :




Gambar 3.5. Flowchart JSP

3.2.4. Desain *DataBase*

Dalam mengolah *request* dari aplikasi kamus bahasa Inggris, *server* melakukan komunikasi dengan *DataBase* untuk mengambil dan menyimpan data, dari desain *content* yang ada maka dibutuhkan *DataBase* untuk memenuhi kebutuhan dari desain *content* tersebut, *DataBase* yang digunakan adalah *db_kamus* yang di dalamnya terdapat tabel – tabel kosakata, berikut desain tabel yang digunakan :

3.2.4.1. Tabel *t_a_ind* sampai tabel *t_z_ind*

Tabel *t_a_ind* sampai tabel *t_z_ind* merupakan tabel yang digunakan untuk menyimpan kata – kata yang digunakan dalam kamus, dimana pada tabel ini terdapat 4 *feild* pada tabel ini, yaitu : *feild* ID, KATA, ARTI, KET. Dimana pada *feild* ID berisi id setiap *record*, *feild* KATA berisikan kata – kata Indonesia, *feild* ARTI berisikan kata – kata Inggris, *feild* KET berisikan kode dari keterangan yang nantinya akan direlasikan dengan *t_ket_ing*. Berikut merupakan desain dari tabel *t_a_ind* sampai tabel *t_z_ind* :


Name	Type	Length	Decimals	Allow Null	
▶ ID	varchar	5	0	<input type="checkbox"/>	
KATA	varchar	25	0	<input type="checkbox"/>	
ARTI	varchar	100	0	<input type="checkbox"/>	
KET	varchar	2	0	<input type="checkbox"/>	

Gambar 3.6. Desain tabel *t_a_ind* sampai *t_z_ind* pada *DataBase* *db_kamus*

3.2.4.2. Tabel tabel *t_a_ing* sampai *t_z_ing*

Tabel tabel *t_a_ing* sampai *t_z_ing* merupakan tabel yang digunakan untuk menyimpan kata – kata yang digunakan dalam kamus, dimana pada tabel ini


terdapat 4 *field* pada tabel ini, yaitu : *field* ID, KATA, ARTI, KET. Dimana pada *field* ID berisi id setiap *record*, *field* KATA berisikan kata – kata Inggris, *field* ARTI berisikan kata – kata Indonesia, *field* KET berisikan kode dari keterangan yang nantinya akan direlasikan dengan t_ket_ind. Berikut merupakan desain dari tabel t_a_ing sampai tabel t_z_ing :

Name	Type	Length	Decimals	Allow Null	
▶ ID	varchar	5	0	<input type="checkbox"/>	
KATA	varchar	25	0	<input type="checkbox"/>	
ARTI	varchar	100	0	<input type="checkbox"/>	
KET	varchar	2	0	<input type="checkbox"/>	

Gambar 3.7. Desain tabel t_a_ing sampai t_z_ing pada *DataBase* db_kamus

3.2.4.3. Tabel t_ket_ind

Tabel t_ket_ind merupakan tabel yang digunakan untuk menyimpan keterangan dari setiap kata – kata Indonesia. Pada tabel ini terdapat 2 *field* yaitu : *field* KET, KETERANGAN. Dimana *field* KET berisikan kode dari keterangan yang nantinya akan direlasikan dengan t_a_ing sampai t_z_ing, *field* KETERANGAN berisikan keterangan dari kata Indonesia yang berfungsi sebagai arti. Berikut merupakan desain dari tabel t_ket_ind :


Name	Type	Length	Decimals	Allow Null	
▶ KET	varchar	5	0	<input type="checkbox"/>	
KETERANGAN	varchar	25	0	<input type="checkbox"/>	

Gambar 3.8. Desain tabel t_ket_ind pada *DataBase* db_kamus

3.2.4.1. Tabel t_ket_ing

Tabel t_ket_ing merupakan tabel yang digunakan untuk menyimpan keterangan dari setiap kata – kata Indonesia. Pada tabel ini terdapat 2 *field* yaitu :

field KET, KETERANGAN. Dimana *field* KET berisikan kode dari keterangan yang nantinya akan direlasikan dengan t_a_ind sampai t_z_ind, *field* KETERANGAN berisikan keterangan dari kata Inggris yang berfungsi sebagai arti. Berikut merupakan desain dari tabel t_ket_ing :

Name	Type	Length	Decimals	Allow Null	
▶ KET	varchar	5	0	<input type="checkbox"/>	
KETERANGAN	varchar	25	0	<input type="checkbox"/>	

Gambar 3.9. Desain tabel t_ket_ing pada *DataBase* db_kamus

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Aplikasi Kamus Bahasa Inggris

Sebagai cara untuk mengetahui kerja dari sistem yang dirancang dan dibuat, maka dilakukan implementasi dan pengujian sistem. Sebelum aplikasi diinstall pada *hardware*, maka terlebih dahulu melakukan implementasi dengan menggunakan NetBeans IDE, karena di dalam NetBeans IDE telah mencakup *compiler* atau *builder*, dan *debugger internal*. Selain itu dalam NetBeans juga menyertakan paket *web server apache tomcat*, dimana *server* ini dapat diakses secara langsung dari dalam NetBeans baik pada saat perancangan maupun proses *deployment*.

4.1.1. Content Menu Utama

Pada *content* Menu Utama terdapat beberapa menu yang dapat diakses oleh *user*, diantaranya menu English–Indonesia, menu Indonesia–English, menu Help, dan menu About. Berikut tampilan *content* menu utama dari aplikasi kamus bahasa Inggris :



Gambar 4.1. Content Menu Utama

4.1.2. Content Menu English – Indonesia

Proses penerjemahan dari English – Indonesia dilakukan untuk menerjemahkan kata dari bahasa Inggris sebagai *input* dan bahasa Indonesia sebagai *output*. Berikut tampilan dari proses penerjemahan English – Indonesia :



Gambar 4.2. Content menu English – Indonesia

Apabila arti yang dicari tidak ditemukan, maka akan tampil pesan “*Arti Tidak Ditemukan*”. Berikut tampilan apabila arti yang dicari tidak ditemukan :



Gambar 4.3. Tampilan bila arti tidak ditemukan

4.1.3. *Content* Menu Indonesia – English

Proses penerjemahan dari Indonesia – English dilakukan untuk menerjemahkan dari bahasa Indonesia sebagai *input* dan bahasa Inggris sebagai *output*. Berikut tampilan dari proses penerjemahan Indonesia – English :



Gambar 4.4. *Content* menu Indonesia – English

Apabila arti yang dicari tidak ditemukan, maka akan tampil pesan “*Arti Tidak Ditemukan*”. Berikut tampilan apabila arti tidak ditemukan :



Gambar 4.5. Tampilan bila arti tidak ditemukan

4.1.4. *Content* Menu Help

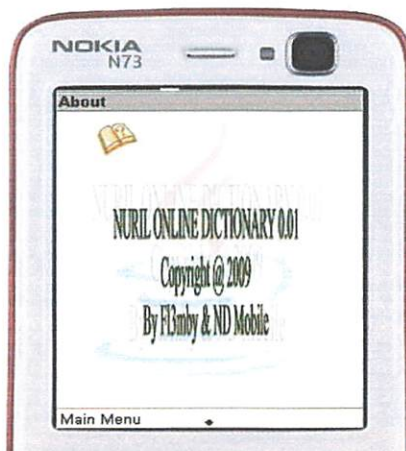
Menu help merupakan menu yang berisikan bantuan bagi *user* pengguna aplikasi kamus bahasa Inggris dalam menggunakan aplikasi tersebut. Berikut tampilan menu help dalam aplikasi kamus bahasa Inggris :



Gambar 4.6. *Content* menu Help

4.1.5. *Content* Menu About

Menu about merupakan menu yang menginformasikan tentang pembuat aplikasi kamus bahasa Inggris. Berikut tampilan dari *content* menu About :



Gambar 4.7. *Content* menu About

4.2. Pengujian Aplikasi Kamus Bahasa Inggris

Pengujian aplikasi kamus bahasa Inggris merupakan proses pengujian terhadap aplikasi yang sudah dibuat. Beberapa pengujian yang dilakukan diantaranya pengujian koneksi *HTTP*, pengujian penterjemahan arti Inggris – Indonesia untuk kata tunggal dan jamak, pengujian penterjemahan Indonesia – Inggris untuk kata tunggal dan jamak.

4.2.1. Koneksi *HTTP*

Pengujian koneksi *HTTP* dilakukan dengan menggunakan ponsel yang telah di install aplikasi kamus Bahasa Inggris, yang mana nantinya melakukan *request* ke *web server*, apabila komunikasi dengan server berhasil maka proses penerjemahan dapat dilakukan, namun bila koneksi gagal maka akan memunculkan pesan “*Error In Connection*”. Berikut tampilan apabila terjadi “*Error In Connection*” :



Gambar 4.8. Tampilan *Error In Connection*

4.2.2. Pengujian Kata Tunggal

Pengujian ini dilakukan dengan cara langsung mengambil arti pada *DataBase* sesuai dengan kata yang dikirimkan oleh *user* dan kemudian menampilkannya pada aplikasi *user* tanpa merubah bentuknya.

4.2.2.1. Pengujian Penerjemahan Inggris – Indonesia

Pengujian penerjemahan ini dilakukan dengan cara menerjemahkan kata dasar dari bahasa Inggris sebagai *input* dan bahasa Indonesia sebagai *output* sebanyak beberapa kali. Berikut hasil pengujian penerjemahan Inggris – Indonesia untuk kata tunggal :

Tabel 4.1. Pengujian penerjemahan Inggris – Indonesia untuk kata tunggal

NO. Pengujian	Kata Inggris	Arti Indonesia	Keterangan
1	Eat	Makan	Sukses
2	Dirve	“Arti Tidak Ditemukan”	Gagal
3	Fail	Gagal	Sukses
4	Park	Taman	Sukses
5	Flower	Bunga	Sukses

Keterangan :

- “Sukses” dapat dikatakan arti yang *direquest* oleh *user* dapat dipenuhi dan dapat ditampilkan dalam aplikasi *user*.
- “Gagal” dapat dikatakan arti yang *direquest* oleh *user* tidak dapat dipenuhi dan kegagalan ini dapat terjadi karena “kata yang dikirimkan *user* tidak terdapat dalam *DataBase*” serta “kegagalan komunikasi dengan *server*”.

4.2.2.2. Pengujian Penerjemahan Indonesia – Inggris

Pengujian penerjemahan ini dilakukan dengan cara menerjemahkan kata dasar dari bahasa Indonesia sebagai *input* dan bahasa Inggris sebagai *output* sebanyak beberapa kali Berikut hasil pengujian penerjemahan Indonesia – Inggris untuk kata tunggal :

Tabel 4.2. Tabel pengujian penerjemahan Indonesia – Inggris untuk kata tunggal

NO. Pengujian	Kata Indonesia	Arti Inggris	Keterangan
1	Air	Water	Sukses
2	Kabel	Cable	Sukses
3	Gading	Tusk	Sukses
4	Sejarah	History	Sukses
5	Nanas	Pinnapple	Sukses

Keterangan :

- “Sukses” dapat dikatakan arti yang *request* oleh *user* dapat dipenuhi dan dapat ditampilkan dalam aplikasi *user*.
- “Gagal” dapat dikatakan arti yang *request* oleh *user* tidak dapat dipenuhi dan kegagalan ini dapat terjadi karena “kata yang dikirimkan *user* tidak terdapat dalam *DataBase*” serta “kegagalan komunikasi dengan *server*”.

4.2.3. Pengujian Kata Jamak

Pada aplikasi ini penerjemahan kata jamak tidak menggunakan keterangan dibelakang kata dasar, tetapi dengan melakukan pengulangan kata dasar pada kata dasar dengan menggunakan tanda “-” sebagai tanda pemisah untuk kata Indonesia jamak, dan memberikan imbuhan “s” dan “es” untuk kata Inggris jamak.

Pengujian ini dilakukan dengan tujuan untuk menghindari pengisian kata pada *DataBase* secara berulang.

4.2.3.1. Pengujian Penerjemahan Inggris – Indonesia

Proses yang dilakukan pada pengujian penerjemahan ini yakni dengan melakukan *substring* pada kata yang di *inputkan* oleh *user*, apabila huruf terakhir mengandung huruf “s” atau “es” di akhir kata maka dicari kata tersebut tanpa akhiran “s” dan “es”, jika kata tersebut ada dalam *DataBase* maka arti yang ditampilkan pada *user* diulang sebanyak dua kali dan dipisahkan dengan menggunakan tanda “-”. Dengan kata lain untuk mendapatkan arti kata jamak dalam bahasa Indonesia hanya dengan memanfaatkan kata dasarnya saja. Dalam hal ini tidak semua kata dapat dijadikan kata jamak, tetapi terbatas hanya pada kata benda saja. Berikut hasil pengujian penerjemahan Inggris – Indonesia untuk kata jamak :

Tabel 4.3. Tabel pengujian penerjemahan Inggris – Indonesia untuk kata jamak

NO. Pengujian	Kata Inggris	Arti Indonesia	Keterangan
1	Hands	Tangan – Tangan	Sukses
2	Flowers	Bunga – Bunga	Sukses
3	Girls	Gadis – Gadis	Sukses
4	Grassses	“Arti Tidak Ditemukan”	Gagal
5	Germes	Benih – Benih	Sukses

Keterangan :

- “Sukses” dapat dikatakan arti yang *request* oleh *user* dapat dipenuhi dan dapat ditampilkan dalam aplikasi *user*.
- “Gagal” dapat dikatakan arti yang *request* oleh *user* tidak dapat dipenuhi dan kegagalan ini dapat terjadi karena “kata yang dikirimkan

user tidak terdapat dalam *DataBase*” serta “kegagalan komunikasi dengan *server*”.

4.2.3.2. Pengujian Penerjemahan Indonesia – Inggris

Proses yang dilakukan pada pengujian penerjemahan ini yakni dengan melakukan *substring* pada kata yang di *inputkan* oleh *user*, apabila huruf terakhir mengandung tanda “-” maka kata tersebut dibandingkan (antara kata sebelum dengan sesudah tanda “-”), apabila hasil perbandingannya sama maka proses pencarian dapat dilakukan. Jika kata tersebut ditemukan maka dilakukan proses *substring* kembali terhadap arti, apabila huruf terakhir adalah “s” maka ditambahkan huruf “es” diakhir arti, manus apabila huruf terakhir pada arti adalah selain huruf “e” maka ditambahkan huruf “s” diakhir arti. Dengan kata lain untuk mendapatkan arti kata jamak dalam bahasa Inggris hanya dengan memanfaatkan kata dasarnya saja. Dalam hal ini tidak semua kata dapat dijadikan kata jamak, tetapi terbatas hanya pada kata benda saja. Berikut hasil pengujian penerjemahan Indonesia – Inggris untuk kata jamak :

Tabel 4.4. Tabel pengujian penerjemahan Indonesia – Inggris untuk kata jamak

NO. Pengujian	Kata Indonesia	Arti Inggris	Keterangan
1	Kaca-Kaca	Glasses	Sukses
2	Zat-Zat	Substances	Sukses
3	Jaket-Jaket	Jackets	Sukses
4	Akara-Akar	“Arti Tidak Ditemukan”	Gagal
5	Foto-Foto	Photos	Sukses

Keterangan :

- “Sukses” dapat dikatakan arti yang *direquest* oleh *user* dapat dipenuhi dan dapat ditampilkan dalam aplikasi *user*.
- “Gagal” dapat dikatakan arti yang *direquest* oleh *user* tidak dapat dipenuhi dan kegagalan ini dapat terjadi karena “kata yang dikirimkan *user* tidak terdapat dalam *DataBase*” serta “kegagalan komunikasi dengan *server*”.

BAB V

PENUTUP

5.1. Kesimpulan

Dari analisa aplikasi kamus bahasa inggris dapat diambil beberapa kesimpulan, diantaranya adalah:

1. Jumlah kata yang dapat diartikan dalam sekali proses terbatas, hanya satu kata saja.
2. Pada aplikasi ini pembedaan kata jamak dan kata tunggal pada bahasa Indonesia yakni dengan menggunakan tanda “-”, sedangkan pada bahasa Inggris pembedaanya dengan melihat imbuhan di belakang kata dasar.

5.2. Saran

Dari sistem aplikasi kamus bahasa Inggris ini memiliki beberapa kelemahan, sehingga dibutuhkan saran – saran yang kemudian akan digunakan untuk proses pengembangan aplikasi kamus bahasa Inggris ini.

1. Untuk menghindari kegagalan akses terhadap *database* yang disebabkan *server* yang dituju mati, maka perlu dibuat suatu *server* lain sebagai alternatif.
2. Untuk melengkapi fitur dari aplikasi kamus bahasa Inggris ini, dapat dikembangkan dengan melakukan penerjemahan perkalimat.
3. Agar mempermudah *admin* dalam memasukkan kata dalam *DataBase* perlu dibuat suatu aplikasi sendiri.

Daftar Pustaka

- [1] **Shalahudin, M.** *Pemrograman J2ME*. Bandung : Penerbit Informatika Bandung. 2006.
- [2] **Aditya Hartanto, Antonius.** *Pemrograman Mobile Java dengan MIDP 2.0*. Yogyakarta : Penerbit Andi Yogyakarta. 2004.
- [3] **J.E.N.I. (Jardiknas Education Network Indonesia).** *Pengembangan Perangkat dan Aplikasi Mobile*. 31 Mei 2006.
- [4] **J.E.N.I. (Jardiknas Education Network Indonesia).** *optimisasi aplikasi mobil*. 27 September 2006.
- [5] **Riyanto, Sigit.** *Arsitektur J2ME*. Surabaya, 21 April 2006.
- [6] **Wikipedia**, 2008, Kamus, [URL: http://id.wikipedia.org/Kamus](http://id.wikipedia.org/Kamus)
- [7] *Connecting to a MySQL Database - NetBeans IDE 6.5 Tutorial using SQL Editor*, 3 januari 2009, [URL: http://dev.mysql.com/doc/refman/5.0/en/installing-cs](http://dev.mysql.com/doc/refman/5.0/en/installing-cs)
- [8] **Hartati Wijono, Sri.** *Pemrograman Java Servlet dan JSP dengan NetBeans*. Yogyakarta : Penerbit Andi Yogyakarta. 2006.
- [9] **Hartati Wijono, Sri.** *Pemrograman GUI Swing Java dengan NetBeans*. Yogyakarta : Penerbit Andi Yogyakarta. 2006.
- [10] **Kadir, Abdul.** *Dasar Pemrograman Web Dinamis dengan JSP (Java Server Pages)*, Inssbruck : Penerbit Andi Yogyakarta. 2004.

LAMPIRAN



**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama Mahasiswa : Nuril Fatih Awaluddin
NIM : 04.12.639
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika
Judul Skripsi : Desain Aplikasi Kamus Bahasa Inggris Pada Ponsel
Menggunakan Java (J2ME).

Dipertahankan dihadapan tim penguji skripsi jenjang Strata Satu (S-1) pada :

Hari : Sabtu
Tanggal : 21 Maret 2009
Dengan Nilai : 84,245 (A) *Bef*

PANITIA UJIAN SKRIPSI



KETUA

Ir. H. Sidik Noertjahjono, MT.
NIP.Y. 1028700163

SEKRETARIS

Ir. F. Yudi Limpraptono, MT.
NIP.Y. 1039500274

ANGGOTA PENGUJI

Dr. Eng. Aryuanto S, ST., MT.
NIP. 1030800417

PENGUJI II

Sandy Nataly Mantja, SKom.



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 5 Februari 2009

Nomor : ITN-686/I.TA/2/09
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr./i. Ir. TH. Mimien Mustikawati, MT.
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama : NURIL FATIHA
Nim : 0412639
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya
kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai
tanggal :

03 Februari 2009 s/d 03 Juli 2009

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan
terima kasih



Ketua Jurusan
Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT.
Nip. Y. 1039500274

Tembusan Kepada Yth :

1. Mahasiswa Yang Bersangkutan
2. Arsip



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 5 Februari 2009

Nomor : ITN-686/I.TA/2/09
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr./i. **MOCH. ASHAR, ST, MT**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama : NURIL FATIHA
Nim : 0412639
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik **Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya
kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai
tanggal :

03 Februari 2009 s/d 03 Juli 2009

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan
terima kasih



Ketua Jurusan
Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT
Nip. Y. 1039500274

Tembusan Kepada Yth :









1. Mahasiswa Yang Bersangkutan
2. Arsip

Form. S 4a



FORMULIR BIMBINGAN SKRIPSI

Nama : NURIL FATIH AWALUDDIN
Nim : 04.12.639
Masa Bimbingan : 03 FEBRUARI 2009 s/d 03 AGUSTUS 2009
Judul Skripsi : DESAIN APLIKASI KAMUS BAHASA INGGRIS
PADA PONSEL MENGGUNAKAN JAVA (J2ME)

D.	Tanggal	Uraian	Paraf Pembimbing
.	10-02-09	Bab I + II revisi	
.	15-02-09	Bab I + II OK	
.	17-02-09	Bab III + IV revisi	
.	17-02-09	Bab III + IV OK	
.	18-02-09	Bab V + Abstrak revisi	
.	20-02-09	Bab V + Abstrak OK	
.	04-03-09	Seminar	
.	19-03-09	kompre	
.			
.			

Malang,
Dosen Pembimbing



Ir. TH. Mimin Mustikawati, MT.
NIP : Y.130000352



FORMULIR BIMBINGAN SKRIPSI

Nama : **NURIL FATIH AWALUDDIN**
 Nim : **04.12.639**
 Masa Bimbingan : **03 FEBRUARI 2009 s/d 03 AGUSTUS 2009**
 Judul Skripsi : **DESAIN APLIKASI KAMUS BAHASA INGGRIS
 PADA PONSEL MENGGUNAKAN JAVA (J2ME)**

No.	Tanggal	Uraian	Paraf Pembimbing
	16-02-09	Bab I, II (Revisi)	A
	17-02-09	Bab III Desain system.	A
	17-02-09	Blok system	A
	18-02-09	Metode kerangka kamus.	A
	18-02-09	Bab IV. pelynsia ayp -	A
	28-02-09	Desain form janda	A
	29-03-09	Screen Querin pd kt Jand	A
	29-03-09	Makalah Seminar krl.	A
	19-03-09	Kompre	A

Malang,
Dosen Pembimbing

M. Ashari ST, MT
NIP : Y.1030500408



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi


Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : NURIL FATIHA
NIM : 0412639
Perbaikan meliputi :

1. Bab IV dituliskan menggunakan netbinder IDE.
2. Cara no 3 perlu dibuat program aplikasi jika menyimpulkan data (lihat ST)
3. Pada pengujian bab IV. (A.2.4) keterangan subser diberi arti akan lebih baik jika ditambahkan :

Kt. lns	arti lnsris	ketangan
lupa	file & temuan	gagal

Malang, 21 MARET 2006


(PADDY MADY)



Formulir Perbaikan Skripsi

Dalam pelaksanaan ujian skripsi jenjang Strata satu (S-1) Jurusan Teknik Elektro konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk Mahasiswa :

Nama : NURIL FATIH AWALUDDIN
NIM : 04.12.639
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika
Masa Bimbingan : 03 Februari 2009 s/d 03 Agustus 2009
Judul Skripsi : Desain Aplikasi Kamus Bahasa Inggris Pada Ponsel Menggunakan Java (J2ME)

Tanggal	Uraian	Paraf
Penguji I 21 Maret 2009	Pengujian : Keterangan proses "sukses" ditambah dengan proses "gagal"	
	Kesimpulan : no.3 tidak perlu	
Penguji II 21 Maret 2009	Bab IV : dituliskan menggunakan NetBeans IDE	
	Pengujian : Keterangan proses "sukses" diberi arti, akan lebih baik dtambahkan proses "gagal"	
	Saran : no.3 perlu dibuat program aplikasi untuk menginputkan data (bukan SI)	

Mengetahui,

Dosen Pembimbing I

Ir. TH. Mimien Mustikawati, MT.
NIP.Y. 130000352

Dosen Pembimbing II

M. Ashar, ST., MT.
NIP.P. 1030500408

Dosen Penguji,

Penguji I

Dr. Eng. Aryuanto, ST., MT.
NIP. 1030800417

Penguji II

Sandy Nataly Mantja, SKom.

Online Dictionary

```
package Dictionary.nuril;

import java.io.IOException;
import java.io.InputStream;
import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import org.netbeans.microedition.lcdui.SplashScreen;

public class OnlineDictionary extends MIDlet implements CommandListener,
ItemCommandListener {
    private boolean midletPaused = false;
    private boolean Indonesia_English = false;
    private StringItem data_indonesia;
    private Form Help_form;
    private ImageItem imageItem2;
    private Form About_form;
    private ImageItem About_imageItem;
    private SplashScreen splashScreen;
    private List MenuUtama_list;
    private Form Indonesia_English_form;
    private ImageItem imageItem;
    private Spacer spacer1;
    private Spacer spacer;
    private Spacer spacer2;
    private StringItem OK_stringItem;
    private TextField Indonesia_Text_textField;
    private TextField English_textField;
    private Spacer spacer4;
    private Spacer spacer3;
    private Form English_Indonesia_form;
    private Spacer spacer8;
    private Spacer spacer7;
    private TextField Indonesia_EI_textField;
    private Spacer spacer9;
    private ImageItem imageItem1;
    private StringItem stringItem;
    private TextField English_EI_textField;
    private Spacer spacer6;
    private Spacer spacer5;
    private Command okCommand;
    private Command Menu_Utama_Command;
    private Command cancelCommand;
    private Command screenCommand;
    private Image ImgHlp;
    private Font font;
    private Image About_Menu_image;
    private Image Logo_Image;
    private Image About_image;
    private Ticker Splash_ticker;
    private Image Kamus_Image;
    private Image Help_image;
    private Image Exit_image;
    private Image English_Indonesia_Image;
    private Image Indonesia_English_Image;
```

```

public OnlineDictionary() {
}

private void ambil_Data_English() {
    String bahasa = "Indonesia";
    String kata = Indonesia_Text_textField.getString();
    English_textField.setString(kata);
    HandledKoneksi konn = new HandledKoneksi(1, bahasa, kata);
    konn.start();
    switchDisplayable(null, Indonesia_English_form );
}

private void ambil_Data_indonesia() {
    Indonesia_EI_textField.setString("halo");
    String bahasa = "English";
    String kata = English_EI_textField.getString();

    HandledKoneksi konn = new HandledKoneksi(2, bahasa, kata);
    konn.start();
    switchDisplayable(null, English_Indonesia_form);
}

class HandledKoneksi extends Thread{
    public final int Indonesia = 1;
    public final int English = 2;

    String bahasa;
    String kata;
    int status;
    private String data;
    HandledKoneksi(int status, String bahasa, String kata){
        this.bahasa = bahasa;
        this.kata = kata;
        this.status = status;
    }
    public void run(){
        if(status == 1){
            English_textField.setString(get_resource(bahasa, kata));
        }
        else if(status == 2){
            Indonesia_EI_textField.setString(get_resource(bahasa,
            kata));
        }
    }
}

private String get_resource(String bahasa, String kata ){
    String Translate;
    HttpURLConnection conn = null;
    InputStream IS = null;

    StringBuffer buff = new StringBuffer();
    try {
        conn = (HttpURLConnection)
            Connector.open(URL,Connector.READ_WRITE);
        conn.setRequestMethod(HttpURLConnection.GET);

        int status = conn.getResponseCode();
        if(status==HttpURLConnection.HTTP_OK){
            IS = conn.openInputStream();
            long data = conn.getLength();

```



```

        int huruf;
        String hasil = "";
        while((huruf=IS.read())!=-1){
            hasil+=(char)huruf;
        }
        IS.close();
        conn.close();
        Translate = hasil;
    }
    else{
        Translate = "Missing URL";
    }
} catch (IOException ex) {
    Translate = "error in conection";
}
return Translate;
}

private void reset(){
    English_textField.setString("");
    English_EI_textField.setString("");
    Indonesia_Text_textField.setString("");
    Indonesia_EI_textField.setString("");
}

private void initialize() {
    About_form = new Form("About", new Item[] { getAbout_imageItem()
    });
    About_form.addCommand(getMenu_Utama_Command());
    About_form.setCommandListener(this);
    MenuUtama_list = new List("DICTIONARY MENU", Choice.IMPLICIT);
    MenuUtama_list.append("    English - Indonesia",
        getKamus_Image());
    MenuUtama_list.append("    Indonesia - English",
        getKamus_Image());
    MenuUtama_list.append("    Help", getHelp_image());
    MenuUtama_list.append("    About", getAbout_image());
    MenuUtama_list.append("    Exit", getExit_image());
    MenuUtama_list.addCommand(getOkCommand());
    MenuUtama_list.setCommandListener(this);
    MenuUtama_list.setSelectedFlags(new boolean[] { true, false,
        false, true, false });
    MenuUtama_list.setFont(0, getFont());
    MenuUtama_list.setFont(1, getFont());
    MenuUtama_list.setFont(2, getFont());
    MenuUtama_list.setFont(3, getFont());
    MenuUtama_list.setFont(4, getFont());
    OK_stringItem = new StringItem("", "Translate", Item.BUTTON);
    OK_stringItem.addCommand(getOkCommand());
    OK_stringItem.setItemCommandListener(this);
    OK_stringItem.setDefaultCommand(getOkCommand());
    OK_stringItem.setFont(getFont());
    Indonesia_Text_textField = new TextField("Indonesia (64)", null,
        64, TextField.ANY);
    Indonesia_Text_textField.setLayout(ImageItem.LAYOUT_LEFT |
        ImageItem.LAYOUT_NEWLINE_BEFORE |
        ImageItem.LAYOUT_NEWLINE_AFTER);
    English_textField = new TextField("English", "..", 255,
        TextField.ANY);
    Indonesia_English_form = new Form("", new Item[] {
        getImageItem(), getSpacer(), getSpacer1(), getSpacer2(),

```

```

        Indonesia_Text_textField, OK_stringItem, getSpacer3(),
        getSpacer4(), English_textField });
Indonesia_English_form.addCommand(getMenu_Utama_Command());
Indonesia_English_form.setCommandListener(this);
spacer7 = new Spacer(16, 1);
Indonesia_EI_textField = new TextField("Indonesia", null, 255,
        TextField.ANY);
stringItem = new StringItem("", "Translate", Item.BUTTON);
stringItem.addCommand(getOkCommand());
stringItem.setItemCommandListener(this);
stringItem.setDefaultCommand(getOkCommand());
stringItem.setFont(getFont());
English_EI_textField = new TextField
        ("English", null, 64, TextField.ANY);
English_EI_textField.setLayout(ImageItem.LAYOUT_LEFT);
English_Indonesia_form = new Form("", new Item[] {
        getImageItem1(), getSpacer5(), getSpacer6(),
        English_EI_textField, stringItem, spacer7, getSpacer8(),
        Indonesia_EI_textField, getSpacer9() });
English_Indonesia_form.addCommand(getMenu_Utama_Command());
English_Indonesia_form.setCommandListener(this);
}

public void startMIDlet() {
    switchDisplayable(null, getSplashScreen());
}

public void resumeMIDlet() {
}

public void switchDisplayable(Alert alert, Displayable nextDisplayable)
{
    Display display = getDisplay();
    if (alert == null) {
        display.setCurrent(nextDisplayable);
    } else {
        display.setCurrent(alert, nextDisplayable);
    }
}

public void commandAction(Command command, Displayable displayable) {
    if (displayable == About_form) {
        if (command == Menu_Utama_Command) {
            switchDisplayable(null, MenuUtama_list);
        }
    } else if (displayable == English_Indonesia_form) {
        if (command == Menu_Utama_Command) {
            reset();
            switchDisplayable(null, MenuUtama_list);
        }
    } else if (displayable == Help_form) {
        if (command == Menu_Utama_Command) {
            switchDisplayable(null, MenuUtama_list);
        }
    } else if (displayable == Indonesia_English_form) {
        if (command == Menu_Utama_Command) {
            reset();
            Indonesia_English = true;
            switchDisplayable(null, MenuUtama_list);
        }
    } else if (displayable == MenuUtama_list) {
        if (command == List.SELECT_COMMAND) {

```

```

        MenuUtama_listAction();
    } else if (command == okCommand) {
        MenuUtama_listAction();
    }
} else if (displayable == splashScreen) {
    if (command == SplashScreen.DISMISS_COMMAND) {
        switchDisplayable(null, MenuUtama_list);
    }
}
}

public SplashScreen getSplashScreen() {
    if (splashScreen == null) {
        splashScreen = new SplashScreen(getDisplay());
        splashScreen.setTitle("ONLINE DICTIONARY");
        splashScreen.setTicker(getSplash_ticker());
        splashScreen.setCommandListener(this);
        splashScreen.setFullScreenMode(true);
        splashScreen.setImage(getLogo_Image());
        splashScreen.setText("Nuril Online Dictionary");
        splashScreen.setTextFont(getFont());
        splashScreen.setTimeout(4000);
        splashScreen.setAllowTimeoutInterrupt(false);
    }
    return splashScreen;
}

public Image getLogo_Image() {
    if (Logo_Image == null) {
        try {
            Logo_Image = Image.createImage("/Image/Logo/kamus.png");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
    }
    return Logo_Image;
}

public Font getFont() {
    if (font == null) {
        font = Font.getFont(Font.FACE_PROPORTIONAL, Font.STYLE_BOLD,
            Font.SIZE_LARGE);
    }
    return font;
}

public void MenuUtama_listAction() {
    String __selectedString = MenuUtama_list.getString
        (MenuUtama_list.getSelectedIndex());
    if (__selectedString != null) {
        if (__selectedString.equals(" English - Indonesia")) {
            switchDisplayable(null, English_Indonesia_form);
        } else if (__selectedString.equals(" Indonesia - English")) {
            switchDisplayable(null, Indonesia_English_form);
        } else if (__selectedString.equals(" Help")) {
            switchDisplayable(null, getHelp_form());
        } else if (__selectedString.equals(" About")) {
            switchDisplayable(null, About_form);
        } else if (__selectedString.equals(" Exit")) {
            exitMIDlet();
        }
    }
}

```

```

    }
}

public Image getKamus_Image() {
    if (Kamus_Image == null) {
        try {
            Kamus_Image = Image.createImage("/kamus_2_PNG.PNG");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
    }
    return Kamus_Image;
}

public Image getHelp_image() {
    if (Help_image == null) {
        try {
            Help_image = Image.createImage("/Help_2_PNG_2.PNG");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
    }
    return Help_image;
}

public Image getAbout_image() {
    if (About_image == null) {
        try {
            About_image = Image.createImage("/about_2_2.png");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
    }
    return About_image;
}

public Ticker getSplash_ticker() {
    if (Splash_ticker == null) {
        Splash_ticker = new Ticker("ITN MALANG");
    }
    return Splash_ticker;
}

public Image getExit_image() {
    if (Exit_image == null) {
        try {
            Exit_image = Image.createImage("/Exit_PNG_2_2.png");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
    }
    return Exit_image;
}

public Image getIndonesia_English_Image() {
    if (Indonesia_English_Image == null) {
        try {
            Indonesia_English_Image = Image.createImage("/IE.PNG");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
    }
}

```

```

        return Indonesia_English_Image;
    }

    public Image getEnglish_Indonesia_Image() {
        if (English_Indonesia_Image == null) {
            try {
                English_Indonesia_Image = Image.createImage("/EI.PNG");
            } catch (java.io.IOException e) {
                e.printStackTrace();
            }
        }
        return English_Indonesia_Image;
    }

    public Command getOkCommand() {
        if (okCommand == null) {
            okCommand = new Command("OK", Command.OK, 0);
        }
        return okCommand;
    }

    public Command getMenu_Utama_Command() {
        if (Menu_Utama_Command == null) {
            Menu_Utama_Command = new Command("Main Menu", Command.EXIT,
0);
        }
        return Menu_Utama_Command;
    }

    public ImageItem getImageItem1() {
        if (imageItem1 == null) {
            imageItem1 = new ImageItem("", getEnglish_Indonesia_Image(),
                ImageItem.LAYOUT_CENTER |
                ImageItem.LAYOUT_NEWLINE_BEFORE |
                ImageItem.LAYOUT_NEWLINE_AFTER, "<Missing Image>");
        }
        return imageItem1;
    }

    public ImageItem getImageItem() {
        if (imageItem == null) {
            imageItem = new ImageItem("", getIndonesia_English_Image(),
                ImageItem.LAYOUT_CENTER |
                ImageItem.LAYOUT_NEWLINE_BEFORE |
                ImageItem.LAYOUT_NEWLINE_AFTER, "<Missing Image>",
                Item.PLAIN);
        }
        return imageItem;
    }

    public Spacer getSpacer() {
        if (spacer == null) {
            spacer = new Spacer(16, 1);
        }
        return spacer;
    }

    public Spacer getSpacer1() {
        if (spacer1 == null) {
            spacer1 = new Spacer(16, 1);
        }
    }

```

```

    }
    return spacer1;
}

public Spacer getSpacer2() {
    if (spacer2 == null) {
        spacer2 = new Spacer(16, 1);
    }
    return spacer2;
}

public void commandAction(Command command, Item item) {
    if (item == OK_stringItem) {
        if (command == okCommand) {
            ambil_Data_English();
        }
    } else if (item == stringItem) {
        if (command == okCommand) {
            ambil_Data_indonesia();
        }
    }
}

public Command getCancelCommand() {
    if (cancelCommand == null) {
        cancelCommand = new Command("Cancel", Command.CANCEL, 0);
    }
    return cancelCommand;
}

public Command getScreenCommand() {
    if (screenCommand == null) {
        screenCommand = new Command("Screen", Command.SCREEN, 0);
    }
    return screenCommand;
}

public Spacer getSpacer3() {
    if (spacer3 == null) {
        spacer3 = new Spacer(16, 1);
    }
    return spacer3;
}

public Spacer getSpacer4() {
    if (spacer4 == null) {
        spacer4 = new Spacer(16, 1);
    }
    return spacer4;
}

public Spacer getSpacer5() {
    if (spacer5 == null) {
        spacer5 = new Spacer(16, 1);
    }
    return spacer5;
}

public Spacer getSpacer6() {
    if (spacer6 == null) {
        spacer6 = new Spacer(16, 1);
    }
}

```

```

    }
    return spacer6;
}

public Spacer getSpacer8() {
    if (spacer8 == null) {
        spacer8 = new Spacer(16, 1);
    }
    return spacer8;
}

public Spacer getSpacer9() {
    if (spacer9 == null) {
        spacer9 = new Spacer(16, 1);
    }
    return spacer9;
}

public Form getHelp_form() {
    if (Help_form == null) {
        Help_form = new Form("", new Item[] { getImageItem2() });
        Help_form.addCommand(getMenu_Utama_Command());
        Help_form.setCommandListener(this);
    }
    return Help_form;
}

public ImageItem getImageItem2() {
    if (imageItem2 == null) {
        imageItem2 = new ImageItem("Help", getImgHlp(),
            ImageItem.LAYOUT_CENTER, "<Missing Image>");
    }
    return imageItem2;
}

public Image getImgHlp() {
    if (ImgHlp == null) {
        try {
            ImgHlp = Image.createImage("/Help.png");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
    }
    return ImgHlp;
}

public Image getAbout_Menu_image() {
    if (About_Menu_image == null) {
        try {
            About_Menu_image = Image.createImageAbout.png");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
    }
    return About_Menu_image;
}

public ImageItem getAbout_imageItem() {
    if (About_imageItem == null) {
        About_imageItem = new ImageItem("", getAbout_Menu_image(),
ImageItem.LAYOUT_CENTER, "<Missing Image>");
    }
}

```

```

    }
    return About_imageItem;
}

public Display getDisplay () {
    return Display.getDisplay(this);
}

public void exitMIDlet() {
    switchDisplayable (null, null);
    destroyApp(true);
    notifyDestroyed();
}

public void startApp() {
    if (midletPaused) {
        resumeMIDlet ();
    } else {
        initialize ();
        startMIDlet ();
    }
    midletPaused = false;
}

public void pauseApp() {
    midletPaused = true;
}

public void destroyApp(boolean unconditional) {
}

}

```

Index.JSP

```

%@page import="com.kamus.nuril.DBHandled" %

<%
try{
    String eng = "English";
    String ind = "Indonesia";

    String bahasa = request.getParameter("bhs");
    String kata = request.getParameter("kata");
    if(bahasa.equals(ind)){
        out.print(db.get_data_inggris(kata));
    }
    else if(bahasa.equals(eng)){
        out.print(db.get_data_indonesia(kata));
    }
}
catch(NullPointerException e){
    out.print("Missed URL");
}

%>

```


DBHandled

```
package com.kamus.nuril;

public class DBHandled {
    private String DB_URL = "*****";
    private String user = "*****";
    private String password = "*****";
    public DBHandled() {
    }

    public String get_data_inggris(String kata){
        String hasil = kata;
        String huruf_awal = kata.substring(0,1);
        String kode_huruf = huruf_awal.toUpperCase();
        String tabel_DB = "";

        if(kode_huruf.equals("A")){
            tabel_DB = "t_a_ind";
        }else if(kode_huruf.equals("B")){
            tabel_DB = "t_b_ind";
        }else if(kode_huruf.equals("C")){
            tabel_DB = "t_c_ind";
        }else if(kode_huruf.equals("D")){
            tabel_DB = "t_d_ind";
        }else if(kode_huruf.equals("E")){
            tabel_DB = "t_e_ind";
        }else if(kode_huruf.equals("F")){
            tabel_DB = "t_f_ind";
        }else if(kode_huruf.equals("G")){
            tabel_DB = "t_g_ind";
        }else if(kode_huruf.equals("H")){
            tabel_DB = "t_h_ind";
        }else if(kode_huruf.equals("I")){
            tabel_DB = "t_i_ind";
        }else if(kode_huruf.equals("J")){
            tabel_DB = "t_j_ind";
        }else if(kode_huruf.equals("K")){
            tabel_DB = "t_k_ind";
        }else if(kode_huruf.equals("L")){
            tabel_DB = "t_l_ind";
        }else if(kode_huruf.equals("M")){
            tabel_DB = "t_m_ind";
        }else if(kode_huruf.equals("N")){
            tabel_DB = "t_n_ind";
        }else if(kode_huruf.equals("O")){
            tabel_DB = "t_o_ind";
        }else if(kode_huruf.equals("P")){
            tabel_DB = "t_p_ind";
        }else if(kode_huruf.equals("Q")){
            tabel_DB = "t_q_ind";
        }else if(kode_huruf.equals("R")){
            tabel_DB = "t_r_ind";
        }else if(kode_huruf.equals("S")){
            tabel_DB = "t_s_ind";
        }else if(kode_huruf.equals("T")){
            tabel_DB = "t_t_ind";
        }else if(kode_huruf.equals("U")){
            tabel_DB = "t_u_ind";
        }else if(kode_huruf.equals("V")){
            tabel_DB = "t_v_ind";
        }else if(kode_huruf.equals("W")){
            tabel_DB = "t_w_ind";
        }
    }
}
```

```

        tabel_DB = "t_w_ind";
    }else if(kode_huruf.equals("X")){
        tabel_DB = "t_x_ind";
    }else if(kode_huruf.equals("Y")){
        tabel_DB = "t_y_ind";
    }else if(kode_huruf.equals("Z")){
        tabel_DB = "t_z_ind";
    }

    if(kata.contains("-")){
        String temp [] = kata.split("-");
        if(temp.length==2){
            if(temp[0].equals(temp[1])){
                hasil = get_data_english_from_indonesia(temp[0],
                    tabel_DB);
                return hasil;
            }
        }
    }

    if(!tabel.equals("")){
        String SQL = "SELECT "+tabel_DB+".ARTI, " +
            "t_ket_ing.KETERANGAN FROM "+tabel_DB+" " +
            "Inner Join t_ket_ing ON "+tabel_DB+".KET = " +
            "t_ket_ing.KET " +
            "WHERE "+tabel_DB+".KATA = '"+kata+"'";

        try{
            String data[][] = ambilData(SQL);
            hasil = data[0][0]+" (" +data[0][1]+")";
        }catch(Exception e){
            hasil = "arti tidak ditemukan";
        }
    }
    return hasil;
}

public String get_data_english_from_indonesia(String kata, String
tabel_DB){
    String hasil = "";
    if(!tabel.equals("")){
        String SQL = "SELECT " +
            tabel +".ARTI, " +
            tabel +".KET, " +
            "t_ket_ing.KETERANGAN " +
            "FROM " +
            tabel_DB + " " +
            "Inner Join t_ket_ing ON "+tabel_DB+".KET = " +
            "t_ket_ing.KET " +
            "WHERE " +
            tabel +".KATA = '"+kata+"' ";

        try{
            String data[][] = ambilData(SQL);
            String arti      = data[0][0];
            String ket       = data[0][1];
            String keterangan = data[0][2];

            if(ket.equals("02")){
                if(arti.substring(arti.length()-1,
arti.length()).equals("s")){
                    arti = arti;

```

```

        }
        else{
            arti+="s";
        }
    }
    hasil = arti + " (" +keterangan+" )";

    }catch(Exception e){
        hasil = "arti tidak ditemukan";
    }
}
return hasil;
}

public String get_data_indonesia(String kata){
    String hasil = kata;
    String huruf_awal = kata.substring(0,1);
    String kode_huruf = huruf_awal.toUpperCase();
    String tabel_DB = "";

    if(kode_huruf.equals("A")){
        tabel_DB = "t_a_ing";
    }else if(kode_huruf.equals("B")){
        tabel_DB = "t_b_ing";
    }else if(kode_huruf.equals("C")){
        tabel_DB = "t_c_ing";
    }else if(kode_huruf.equals("D")){
        tabel_DB = "t_d_ing";
    }else if(kode_huruf.equals("E")){
        tabel_DB = "t_e_ing";
    }else if(kode_huruf.equals("F")){
        tabel_DB = "t_f_ing";
    }else if(kode_huruf.equals("G")){
        tabel_DB = "t_g_ing";
    }else if(kode_huruf.equals("H")){
        tabel_DB = "t_h_ing";
    }else if(kode_huruf.equals("I")){
        tabel_DB = "t_i_ing";
    }else if(kode_huruf.equals("J")){
        tabel_DB = "t_j_ing";
    }else if(kode_huruf.equals("K")){
        tabel_DB = "t_k_ing";
    }else if(kode_huruf.equals("L")){
        tabel_DB = "t_l_ing";
    }else if(kode_huruf.equals("M")){
        tabel_DB = "t_m_ing";
    }else if(kode_huruf.equals("N")){
        tabel_DB = "t_n_ing";
    }else if(kode_huruf.equals("O")){
        tabel_DB = "t_o_ing";
    }else if(kode_huruf.equals("P")){
        tabel_DB = "t_p_ing";
    }else if(kode_huruf.equals("Q")){
        tabel_DB = "t_q_ing";
    }else if(kode_huruf.equals("R")){
        tabel_DB = "t_r_ing";
    }else if(kode_huruf.equals("S")){
        tabel_DB = "t_s_ing";
    }else if(kode_huruf.equals("T")){
        tabel_DB = "t_t_ing";
    }else if(kode_huruf.equals("U")){

```

```

        tabel_DB = "t_u_ing";
    }else if(kode_huruf.equals("V")){
        tabel_DB = "t_v_ing";
    }else if(kode_huruf.equals("W")){
        tabel_DB = "t_w_ing";
    }else if(kode_huruf.equals("X")){
        tabel_DB = "t_x_ing";
    }else if(kode_huruf.equals("Y")){
        tabel_DB = "t_y_ing";
    }else if(kode_huruf.equals("Z")){
        tabel_DB = "t_z_ing";
    }
}

if(!tabel.equals("")){
    try{
        String huruf_akhir = kata.substring(kata.length()-
            1,kata.length()).toUpperCase();
        String huruf_akhir_min_2 = kata.substring(kata.length()-
            3,kata.length()-2).toUpperCase();
        String huruf_akhir_min_1 = kata.substring(kata.length()-
            2,kata.length()-1).toUpperCase();

        if(huruf_akhir.equals("S")){
            if(!tabel.equals("")){
                if(huruf_akhir_min_2.equals("S")){
                    kata = kata.substring(0, kata.length());
                    hasil =
                        get_jamak_indonesia_from_english(kata
                            , tabel_DB);
                }
                else if(!huruf_akhir_min_1.equals("S")){
                    kata = kata.substring(0, kata.length());
                    hasil =
                        get_jamak_indonesia_from_english(kata
                            , tabel_DB);
                }
            }
        }
        else{
            hasil = get_indonesia_from_english(kata, tabel_DB);
        }
    }catch(Exception e){
        hasil = get_indonesia_from_english(kata, tabel_DB);
    }
}
return hasil;
}

private String get_indonesia_from_english(String kata, String tabel_DB){
    String hasil = "";
    String SQL = "SELECT "+tabel_DB+".ARTI, " +
        "t_ket_ind.KETERANGAN FROM "+tabel+" " +
        "Inner Join t_ket_ind ON "+tabel+" .KET = " +
        "t_ket_ind.KET " +
        "WHERE "+tabel+" .KATA = '"+kata+"'";

    try{
        String data[][] = ambilData(SQL);
        hasil = data[0][0]+" (" +data[0][1]+")";
    }catch(Exception e){
        hasil = "arti tidak ditemukan";
    }
    return hasil;
}

```



```

        else
            if((metacr.getColumnTypeName(h).equals("VARCHAR"))){
                try{
                    data[barisdata][h-1] = datacr.getString(h);
                }
                catch(java.lang.NullPointerException e){
                    data[barisdata][h-1] = "null";
                }
            }
        }
        barisdata++;
    }
    datacr.close();
    stat.close();
    con.close();
} catch (Exception e) {
    data = null;
}
return data;
}

```

```

private int update_database(String Query){
    int hasil = 0;
    try{
        int barisdata = 0;
        int baris = 0;
        Class.forName(DB_Driver);
        java.sql.Statement stat = con.createStatement();
        hasil = stat.executeUpdate(Query);
        stat.close();
        con.close();
    }
    catch (ClassNotFoundException ex) {
        hasil = 0;
    }
    catch (java.sql.SQLException e){
        hasil = 0;
    }
    return hasil;
}

```

```

private String Waktu(java.util.Date d){
    int hari = d.getDate();
    int bulan = d.getMonth()+1;
    int tahun = d.getYear()+1900;
    return (""+hari + "-" + bulan+"-"+ tahun);
}
}

```