

# Makalah Seminar Tugas Akhir Pemanfaatan Teknologi BREW Dengan Membangun Aplikasi Kamus Bahasa

SURYADY  
NIM : L2F398340  
Jurusan Teknik Elektro  
Fakultas Teknik Universitas Diponegoro Semarang

## ABSTRAK

Saat ini ada banyak platform teknologi yang mendukung majunya teknologi selular dengan segala kelebihan dan kekurangannya. Salah satu platform yang sudah lama dikenal adalah Java dengan J2ME-nya (Java 2 Micro Edition). J2ME merupakan platform yang mendukung perangkat keras yang menjalankan aplikasinya melalui applet Java.

Belum lama ini telah hadir teknologi BREW (Binary Runtime Environment for Wireless) sebagai teknologi yang menyaingi J2ME dan memiliki konsep yang hampir sama dengan J2ME, dimana BREW mendukung penuh sistem CDMA. Hadir sebagai Software Developer Kit, BREW menyediakan library-library yang dapat dimanfaatkan untuk membangun aplikasi perangkat mobile yang tangguh.

Pada tugas akhir ini dibangun suatu aplikasi yang berguna untuk memanfaatkan teknologi BREW. Selain itu dengan aplikasi ini dicoba untuk mengeksplorasi fitur-fitur yang disediakan oleh teknologi BREW.

## I. PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi telekomunikasi telah meningkat pesat seiring dengan bertambahnya kuantitas dan kualitas layanan yang diharapkan oleh para pengguna jalur telekomunikasi. Munculnya teknologi CDMA merupakan salah satu jawaban yang muncul berdasarkan tantangan yang ada. Dengan kemampuannya untuk melakukan manajemen komunikasi dalam suatu jaringan, maka timbul inspirasi untuk menciptakan fasilitas pendukung yang mampu memaksimalkan kinerjanya.

BREW (*Binary Runtime Environment for Wireless*) hadir sebagai salah satu fitur teknologi yang melengkapi dan mendukung teknologi CDMA. BREW merupakan suatu platform yang memungkinkan untuk membangun suatu aplikasi di sisi *client*. Sisi *client* di sini bisa berbentuk telepon genggam maupun PDA yang beroperasi pada saluran CDMA. Mengingat terbatasnya *resource* yang ada di sisi *client*, maka BREW didisain untuk dapat mengatur dengan seksama *resource* tersebut.

Berdasarkan pada kondisi ini, maka pada tugas akhir ini akan mencoba mengenalkan teknologi BREW disertai penggunaannya pada perangkat *mobile*. Untuk penggunaannya, diterapkan pada pembuatan aplikasi Kamus Inggris Indonesia. Pembangunan suatu sistem aplikasi dengan menggunakan teknologi BREW yang dapat berjalan pada piranti *mobile* memungkinkan

pengguna untuk mengakses dan menjalankan aplikasi di sisi *client*.

### 1.2 Tujuan Pembuatan Tugas Akhir

Tujuan pembuatan tugas akhir ini adalah untuk membangun aplikasi kamus bahasa Inggris Indonesia dengan memanfaatkan dan mengimplementasikan teknologi BREW.

Sasaran lain adalah bahwa aplikasi ini diharapkan mampu membantu pengguna untuk mencari padanan kata satu kata berbahasa Inggris ke bahasa Indonesia.

### 1.3 Batasan Masalah

Dari permasalahan-permasalahan di atas, maka batasan dalam Tugas Akhir ini adalah:

1. Aplikasi dibangun menggunakan BREW dengan output berupa grafis sederhana dan text.
2. Aplikasi hanya menggunakan database lokal, sehingga aplikasi tidak perlu mengakses server.
3. Kamus yang dibangun merupakan kamus Inggris Indonesia. Untuk membuktikan bahwa aplikasi dapat berjalan cukup digunakan emulator ponsel yang berjalan di PC yang merupakan tiruan dari kondisi ponsel sebenarnya.

## II. BREW SEBAGAI PLATFORM APLIKASI

### 2.1 Gambaran Umum Teknologi BREW

*Binary Runtime Environment for Wireless* (BREW), adalah teknologi baru yang dibangun oleh

Qualcomm. Qualcomm sendiri adalah perusahaan yang bergerak di bidang telekomunikasi nirkabel yang telah mendunia dengan menciptakan teknologi CDMA. Pada mulanya BREW diciptakan untuk keperluan internal perusahaan dalam membangun perangkat lunak yang mendukung teknologi CDMA, namun pada akhirnya Qualcomm melepas BREW sebagai *platform* teknologi yang dapat digunakan oleh khalayak umum.

Saat ini khalayak umum dapat menggunakan *library* program yang disediakan teknologi BREW untuk membangun *applets* (aplikasi), dimana *applets* yang telah dibangun dapat dijalankan pada pesawat *mobile* yang dapat menjalankan BREW.

BREW bukan hanya teknologi perangkat lunak. Bersama dengan SDK (*Software Development Kit*) dan spesifikasi perangkat kerasnya, Qualcomm juga membuat metode pemasaran khusus untuk perangkat lunak berbasis teknologi BREW yang dibangun oleh *developer*. Hal ini merupakan metode khusus yang dirancang oleh Qualcomm agar *developer* memperoleh keuntungan yang besar dari perangkat lunak yang dibangun.

### 2.1.1 Perangkat Keras

Sebelum membangun *applets*, sebaiknya ditentukan dulu pesawat *mobile* tipe dan merk apa yang akan digunakan untuk menjalankan *applets*. Hal ini disebabkan karena adanya kemungkinan *applets* yang dapat berjalan di satu pesawat *mobile* tidak dapat berjalan dengan baik di pesawat lain. Untuk itu sebelum membangun *applets* sebaiknya diteliti dulu spesifikasi pesawat yang dijadikan media untuk menjalankan *applets*. Untuk mengetahui spesifikasi pesawat *mobile* yang dimaksud dan memperoleh *emulator*-nya, maka *developer* dapat bergabung menjadi anggota *Qualcomm's BREW Developer Extranet* lalu *download* file informasi dan *emulator*-nya.

Pada awalnya ada dua jenis standar perangkat *mobile* yang mendukung teknologi BREW. Kedua standar tersebut adalah perangkat yang *low end* dan *high end* yang diwakili oleh pesawat Kyocera QCP3035 (*low end*) dan Sharp Z-800 (*high end*). Untuk selanjutnya standar ini terus dikembangkan sehingga kinerja dan fungsionalitas teknologi BREW akan semakin meningkat.

### 2.1.2 Software Development Kit (SDK)

BREW SDK adalah *library* yang dapat digunakan programmer untuk membangun aplikasi berbasis BREW menggunakan bahasa C dan C++. Pada SDK sudah ada modul-modul yang dapat digunakan untuk fungsi-fungsi yang berhubungan dengan grafik, *networking*, suara, tampilan teks,

GUIs dan banyak lagi lainnya. BREW SDK telah menyediakan *tools* yang dapat digunakan untuk membantu membangun *applets*. *Tool* yang paling sering digunakan adalah *emulator*. Berikut ini adalah penjelasan dari beberapa *tool* yang disediakan BREW SDK.

#### BREW Emulator

BREW *emulator* adalah program yang dapat digunakan untuk menguji *applets*, sebelum *applets* di-*install* pada perangkat *mobile*. *Emulator* dijalankan menggunakan system operasi Windows. Penggunaan *emulator* dapat memudahkan *developer* untuk melakukan debuing dan pengetesan kode. *Emulator* juga dapat diubah-ubah konfigurasinya untuk mensimulasikan penggunaan alokasi memori, resolusi *screen* dan ketajaman gambar.

#### BREW Device Configurator

*Device Configurator* adalah *tool* yang digunakan untuk menyeting konfigurasi *emulator* agar sesuai dengan perangkat *mobile* tempat implementasi. Konfigurasi meliputi ukuran layar, jumlah warna, alokasi memori, dan atribut lainnya, kemudian digunakan *emulator* untuk mensimulasikan pengoperasian perangkat *mobile*.

#### BREW MIF Editor

File MIF adalah file yang digunakan BREW untuk mengidentifikasi *applet*. Pada file ini terdapat gambar *icon* yang nantinya ditampilkan pada layar perangkat *mobile*. Pada file ini juga terdapat informasi mengenai jenis aplikasi dan sistem keamanan aplikasi serta siapa saja yang berhak menjalankan.

#### BREW Resource Editor

BREW *Resource Editor* adalah *tool* yang digunakan untuk membuat file *resource*. File *resource* adalah file yang menyimpan semua gambar, strings, komponen GUI dan data-data lain yang diperlukan untuk menjalankan aplikasi. Fungsi utama dari file *resource* adalah untuk memudahkan pengorganisasian data aplikasi.

#### BREW Documentation

BREW *Documentation* adalah dokumen yang menyertai BREW SDK yang berisikan cara-cara pengoperasian *tool* yang ada. Dokumentasi meliputi segala sesuatu yang berhubungan dengan pembangunan aplikasi berbasis BREW.

#### BREW Examples

BREW *Examples* adalah kumpulan aplikasi yang telah dibangun menggunakan BREW SDK. Pada aplikasi-aplikasi ini dapat dilihat implementasi fungsi-fungsi yang dimiliki BREW SDK beserta kode aplikasi yang telah didokumentasi dengan baik.

#### BREW SDK Libraries

BREW *SDL Libraries* adalah kumpulan *library* yang diperlukan untuk menguji dan menjalankan *applet*. *Library-library* ini didisain

untuk diintegrasikan dengan Microsoft's Visual Studio dan secara khusus dijalankan pada C dan C++.

### 2.1.3 Elemen BREW

Agar *applet* BREW dapat dijalankan, harus ada elemen-elemen lain yang berfungsi sebagai sumber data bagi *applet* tersebut. Berikut adalah penjelasan mengenai elemen pada BREW.

#### The BID File

*Class ID* adalah nomor unik untuk mengidentifikasi *applet*. Pada BREW, programmer dapat membuat *applet* yang dapat memanfaatkan fungsi-fungsi yang dimiliki *applet* lain, pengenalan *applet* lain ini dilakukan melalui *class ID* yang dimilikinya. *Class ID* suatu *applet* disimpan pada file BID, dimana file BID sendiri merupakan file *header*.

#### The Applet File

File *applet* adalah file eksekusi yang berisi kode *applet* yang sudah dikompilasi. File untuk *emulator* berbeda dengan perangkat *mobile*. Untuk emulator, file berupa file biner yang merupakan Windows DLL, sedangkan untuk aplikasi perangkat *mobile* filenya berekstensi *.mod*.

#### The Resource File

File ini sifatnya opsional, boleh ada boleh tidak. Isinya adalah data-data seperti gambar, teks, tampilan elemen GUI, yang akan ditampilkan pada *applet*.

#### The MIF File

Semua *applet* harus memiliki file MIF. File MIF mengidentifikasi *applet* dan menyimpan *icon*-nya, *title*, informasi *copyright* dan data-data lain yang berhubungan dengan *applet*. *Class ID* juga diset pada file ini.

### 2.1.4 Distribusi Model

Hal yang membedakan BREW dengan teknologi sejenis lainnya, seperti J2ME, adalah "*Distribution Model*". *Distribution model* adalah sistem pendistribusian yang dibangun Qualcomm untuk memasarkan *applets* yang dibangun oleh *developers* ke konsumen. Pemasaran *applets* berbasis BREW dilakukan secara terpusat melalui Qualcomm menggunakan *central billing* dan *distribution model*.

Setiap *applets* yang ingin dipasarkan oleh *developer* harus diuji dulu oleh Qualcomm untuk memastikan bahwa *applet* benar-benar menggunakan standar BREW, bebas dari *error* dan siap untuk dipasarkan. Setelah dinyatakan lolos uji, maka Qualcomm mengeluarkan sertifikasi bahwa *applets* tersebut siap dipasarkan. *Applets* yang sudah disertifikasi tersebut ditawarkan oleh Qualcomm ke perusahaan lain (*distributor*) yang siap untuk menjual *applets* ke *end user*.

### 2.1.5 Tools Untuk Membangun Applets BREW

Ada beberapa *tools* yang dapat digunakan untuk mempermudah membangun *applets* berbasis BREW, yakni:

1. Microsoft Windows 2000 atau XP  
Saat ini BREW *development tools* hanya dapat berjalan pada system operasi Windows 2000 dan Windows XP. Namun untuk penggunaannya di masa depan Qualcomm mengembangkan juga *development tool* yang dapat berjalan di system operasi Linux dan MacOS.
2. Microsoft Visual C++  
Pada saat ini *compilers* yang dapat digunakan dengan BREW dan *emulator*-nya adalah Visual C++ 6 dan Visual C++ .NET. Pada tugas akhir ini yang digunakan adalah Visual C++ 6, dengan pertimbangan bahwa *tools* ini sangat fleksibel penggunaannya. Namun untuk keperluan implementasi *applets* pada perangkat *mobile*, diperlukan *tools compiler* lain, yaitu ARM BREW Builder.
3. Jasc Paint Shop Pro  
Ini adalah *tool* yang dapat digunakan untuk menggambar grafik bitmap guna keperluan perangkat *mobile*.

### 2.2 Perbandingan BREW Dengan Teknologi Lain (Java)

Pada teknologi aplikasi nirkabel, terdapat perdebatan yang hangat antara kelebihan BREW dan *Java 2 Micro Edition* (J2ME). Selama ini perbedaan antara kedua *platform* tidak terlalu jelas, sehingga muncul persaingan hampir pada semua lini aplikasi perangkat *mobile*. Persaingan muncul semenjak teknologi BREW muncul di khalayak umum. Di Amerika Serikat, perusahaan distributor sistem CDMA *Verizon Wireless* mensupport penggunaan BREW, sedangkan distributor sistem CDMA lain seperti *SprintPCS* mendukung penggunaan J2ME. Di Korea Selatan penggunaan J2ME berada di belakang BREW. Di Eropa, BREW hanya digunakan untuk membuat *splash*, sedangkan J2ME digunakan pada semua lini.

#### 2.2.1 Pengenalan Java

Ketika kita sering mendengar tentang "*mobile Java*", maka yang dimaksud adalah *Java 2 Micro Edition* (J2ME). Seperti halnya pemrograman Java, J2ME adalah bahasa interpreter, sehingga tetap membutuhkan *virtual machine* untuk menjalankan aplikasi. Secara teknis program yang telah ditulis menggunakan kode J2ME akan di-*compile* menjadi *bytecodes*, *bytecodes* ini kemudian dieksekusi oleh *Java Virtual Machine*. Salah satu kerugian menggunakan Java adalah bahwa *virtual machine* telah menambah satu *layer* antara kode program dengan perangkat keras. *Virtual machine* berfungsi untuk

mengubah kode yang kita tulis dari *bytecode* menjadi instruksi yang siap dieksekusi perangkat keras.

Ketika kita menyebut tentang “*wireless Java*”, maka yang dimaksud adalah J2ME yang menggunakan *Connected Limited Device Configuration (CLDC)* and *Mobile Information Device Profile (MIDP)*. CLDC dan MIDP mendefinisikan sekelompok paket dan servis untuk peralatan nirkabel. Berdasarkan MIDP yang ada sekarang, maka aplikasi Java yang sifatnya nirkabel mengacu pada penggunaan MIDlets dan *applets*.

Salah satu contoh perubahan adalah pada *Abstract Windowing Toolkit* untuk disain GUI. Bahkan pada saat ini *toolkit* ini telah diganti dengan *toolkit* lain, yakni LCDUI yang berfungsi untuk menghasilkan tampilan yang sesuai untuk layar ukuran kecil. Kemampuan dari *toolkit* ini antara lain kemampuan untuk menampilkan kontrol menu, *progress bar*, dan tombol yang sesuai untuk sebagian besar aplikasi. Namun bagaimanapun juga untuk fleksibilitas penempatan dan susunan, BREW lebih fleksibel daripada J2ME.

Setelah kode selesai ditulis dan dikompilasi, *developer* dapat memeriksa dan membuat MIDlet untuk selanjutnya dijalankan pada *emulator* atau diuji pada perangkat *mobile* yang sesungguhnya. Pada proses pemeriksaan dan pembuatan MIDlet dapat diketahui apakah MIDlet yang dibuat dapat disupport oleh MIDP.

Setiap perusahaan pembuat perangkat *mobile* kebanyakan menyertakan *emulator*, bersama dengan perangkat *mobile* yang dihasilkannya. Namun tingkat akurasi *emulator* terhadap perangkat tersebut tidaklah sama. Pada banyak kasus, dukungan terhadap resolusi layar atau transparansi piksel, kemungkinan memiliki perbedaan yang cukup banyak antara *emulator* dengan perangkat *mobile*-nya. Seperti halnya pada BREW, performa suatu perangkat *mobile* dalam menjalankan aplikasi tidak selamanya dapat diukur dengan *emulator*.

Untuk mengeksekusi MIDlet di perangkat keras, JAR harus disimpan di perangkat kemudian dijalankan. Untuk menyimpan file JAR, masing-masing perusahaan penghasil perangkat *mobile* menyediakan fasilitasnya sendiri-sendiri. Untuk menguji kode program pada perangkat yang sebenarnya, Java tidak memerlukan MIF atau klas ID seperti halnya pada BREW. Sehingga proses pengujian pada Java dirasa lebih mudah daripada BREW.

### 2.2.2 Implementasi Java Pada BREW

Karena BREW adalah kode hasil kompilasi yang siap dijalankan oleh perangkat keras, maka sangat mungkin untuk membuat *virtual machine* menggunakan BREW. Saat ini

ada dua perusahaan yang membangun Java *virtual machine* yang ditulis menggunakan BREW yakni IBM's J9 Virtual Machine yang dibuat oleh OTI sebagai *emulator* dan Insignia's sebagai implementasi MIDP.



Gambar 2.6 Karl Hornell's MIDP Man yang berjalan di sistem BREW.

Dengan membuat aplikasi yang dapat dijalankan pada sistem BREW, maka *developer* memiliki keuntungan dalam hal pemasarannya, karena MIDlets yang sudah dibuat dapat dijual melalui Qualcomm dengan *central billing* dan *distribution model*-nya. Contoh aplikasi dapat dilihat pada gambar 2.6. Pada beberapa sisi tetap ada kendala untuk penerapan J2ME di BREW, karena banyak perangkat *mobile* berbasis BREW tidak memenuhi standar *interface* J2ME, dan beberapa perangkat *mobile* kelas *low-end* tidak memiliki memori yang cukup untuk menjalankan *virtual machine*. Meskipun demikian, dapat disimpulkan bahwa BREW telah menjadi solusi aplikasi *mobile* yang atraktif karena kemampuannya untuk menjalankan aplikasi J2ME dengan *virtual machine* yang dibangun menggunakan teknologi BREW.

## III. PERANCANGAN APLIKASI

### 3.1 Kebutuhan Pemakai

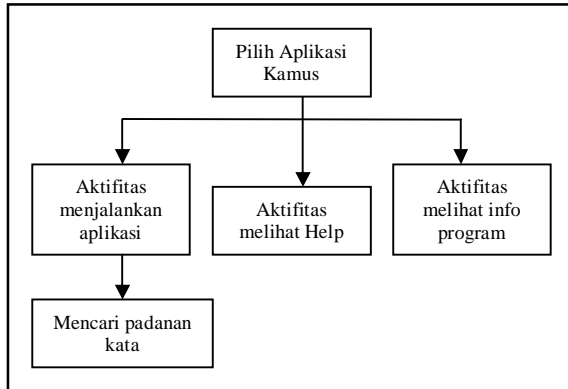
Pencarian padanan kata ini diharapkan dapat dilakukan dengan cepat menggunakan perangkat yang biasa digunakan, yakni perangkat *mobile*. Untuk mendukung kebutuhan ini ada beberapa parameter yang perlu diperhatikan, yakni:

1. Kemudahan dalam pengoperasian
2. Adanya instruksi pengoperasian yang jelas

### 3.2 Diagram Alir Aplikasi

Sebagai gambaran umum mengenai program aplikasi, maka dibuat diagram alir yang menggambarkan jalannya aplikasi. Pertama-tama dibuat suatu rancangan mengenai alur program secara umum, seperti yang terlihat pada gambar 3.1. Rancangan umum tersebut memiliki beberapa poin-poin untuk pengembangan. Dari setiap poin

ini dibuat disain terperinci mengenai alir program. Berikut ini adalah gambar rancangan program secara umum:

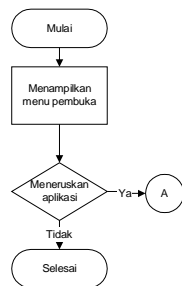


Gambar 3.1 Diagram alir aplikasi secara umum

Untuk rancangan pada gambar 3.1, maka dirancang lima menu yang diharapkan dapat memenuhi parameter-parameter yang sudah ditentukan dengan sebaik-baiknya. Adapun parameter-parameter tersebut adalah:

1. Menu pembuka

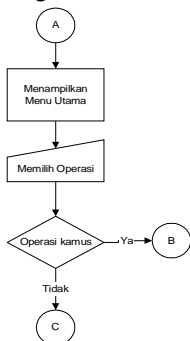
Menu pembuka berfungsi untuk menunjukkan bahwa program kamus akan dijalankan. Diagram alirnya dapat dilihat pada gambar 3.2.



Gambar 3.2 Diagram alir untuk menjalankan aplikasi

2. Menu Utama

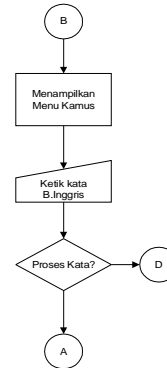
Menu Utama digunakan untuk memilih apakah akan menuju Menu Kamus, Menu Help atau Menu About. Untuk diagram alirnya seperti yang terlihat pada gambar 3.3.



Gambar 3.3 Diagram alir untuk memilih operasi

3. Menu Kamus

Merupakan menu untuk mengoperasikan proses pencarian arti kata Bahasa Indonesia dari Bahasa Inggris. Diagram alirnya dapat dilihat pada gambar 3.4 dan gambar 3.5.



Gambar 3.4 Diagram alir untuk operasi kamus



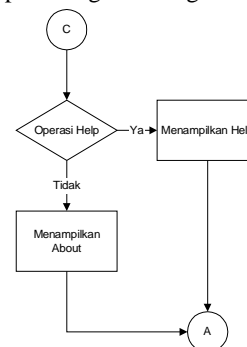
Gambar 3.5 Diagram alir untuk menampilkan arti kata Indonesia

4. Menu Help

Adalah menu yang digunakan untuk menunjukkan langkah-langkah dalam menjalankan aplikasi.

5. Menu About

Adalah menu yang digunakan untuk menampilkan info pembuat. Seperti yang terlihat pada diagram alir gambar 3.6.



Gambar 3.6 Diagram operasi help atau about

### 3.3 Tampilan Aplikasi

Ketika *user* masuk ke aplikasi, *user* akan memilih satu diantara tiga menu, yaitu menu kamus, menu help dan menu about. Dengan memilih menu help dan menu about, *user* membuka form help dan form about yang berisi informasi teks sederhana. Jika yang dipilih adalah menu kamus, maka *user* akan mendapatkan form kamus yang terdiri dari satu teks box untuk entri kata dan dua tombol untuk memroses kata atau keluar ke menu utama.

### 3.4 Elemen Aplikasi

Ada beberapa elemen yang mendukung aplikasi ini, yang paling utama adalah file *applet* beserta file pendukungnya dan file database yang menyimpan data kamus bahasa. Untuk lebih jelasnya, berikut ini adalah daftar file-nya:

1. Kamus.dll  
Merupakan file utama, yang berupa file biner yang dapat secara langsung dieksekusi oleh perangkat keras.
2. Kamus.mif  
Menyimpan logo untuk aplikasi kamus bahasa beserta *class* ID aplikasi.
3. Kamus.bid  
Menyimpan data mengenai *class* ID yang dimiliki oleh aplikasi
4. Kamus.bri  
File ini tidak digunakan oleh file *applet* dalam menjalankan aplikasi, namun berfungsi untuk digunakan untuk mengubah konfigurasi file resource.
5. Kamus.bar  
Merupakan file *resource* yang berisi data gambar, teks dan *resource-resource* lain yang digunakan oleh aplikasi.
6. Database kamus  
Database kamus berupa file-file teks. Kata-kata dalam Bahasa Inggris dikelompokkan berdasarkan huruf depannya. Dari setiap group ini kemudian akan dibuat file teksnya masing-masing. Sehingga nantinya ada 29 file teks *database*.
7. About file  
Adalah file teks yang berisi tentang data pembuat yang akan ditampilkan pada menu *about*.
8. Help file  
Merupakan file teks yang berisi teks instruksi untuk menjalankan aplikasi.

## IV. IMPLEMENTASI DAN UJI COBA APLIKASI

Implementasi yang dilakukan meliputi implementasi pembuatan *applet* aplikasi untuk *user*, implementasi pembuatan database,

implementasi pembuatan *.mif* file, implementasi pembuatan *.bid* file, pembuatan *.bri* file, serta implementasi pembuatan *.bar* file. Bahasa pemrograman yang digunakan untuk membuat *applet* adalah Visual C++ 6.0, sedangkan untuk file lainnya menggunakan *tool* yang disediakan oleh BREW SDK.

### 4.1 Implementasi Pembuatan Aplikasi

Aplikasi untuk *user* dibuat sedemikian rupa agar mudah dan nyaman dijalankan oleh *user*. Pada aplikasi ini digunakan beberapa *library-library* yang sudah disediakan oleh BREW SDK.

#### 4.1.1 Pembuatan Applet Aplikasi

Pada applet ini digunakan beberapa interface yang dimiliki oleh BREW SDK, antara lain IShell, IApplet, IDisplay, IMenuCtl, ITextCtl, IImage, IStatic, IFile, IFileMgr. Ada beberapa modul utama yang ada pada program aplikasi yang dibangun. Diantaranya modul untuk menampilkan menu utama, modul untuk event handler, modul menu help, modul menu about, modul menu input, modul untuk menampilkan arti kata Indonesia.

Ketika *user* menjalankan aplikasi kamus, aplikasi akan menampilkan menu utama. Untuk itu dibuat fungsi untuk membangun dan menampilkan menu tersebut, yaitu `Kamus_BuildMainMenu(myapp_t * pApp)`. Seperti yang terlihat pada gambar 4.1 berikut ini.

```
void Kamus_BuildMainMenu(myapp_t * pApp)
{
    CtlAddItem ai;

    SETAERECT(&(pApp->m_rFrame), 0,
pApp->rImageInfo.cy, pApp->di.cxScreen,
pApp->di.cyScreen - pApp->rImageInfo.cy);

    ISHELL_CreateInstance(pApp-
>a.m_pIShell, AECLSID_MENUCTL, (void
**) &pApp->pIMenu);
    IMENUCTL_SetTitle(pApp->pIMenu,
Kamus_RES_FILE, STR_MAINMENU, NULL);
    IMENUCTL_SetRect(pApp->pIMenu,
&pApp->m_rFrame);

    //Add in our menu items
    ai.pText = NULL;
    ai.pImage = NULL;
    ai.pszResImage = ai.pszResText =
Kamus_RES_FILE;
    ai.wFont = AEE_FONT_NORMAL;
    ai.dwData = 0;

    . . . . .
    . . . . .
    . . . . .

    pApp->m_eAppState =
APP_STATE_MAIN;
}MacroName:="TemplateProject.NewMacros.Sh
owCreate"
```

Gambar 4.1 Potongan *script* Menu Utama.

Setelah memasuki menu utama, *user* dapat memilih untuk melakukan operasi konversi bahasa, melihat menu help atau melihat info pembuat. Jika yang ingin dilihat adalah info pembuat dan *help*, maka dijalankan fungsi `Kamus_BuildPrompt(...)` dengan potongan program seperti pada gambar 4.2 berikut:

```
void Kamus_BuildPrompt(myapp_t * pApp,
uint16 wIDTitle, uint16 wIDText, char*
pFileName, EAppStateType eAppState)
{
    . . . . .
    default :
        if (Kamus_OpenFile(pApp,pFileName))
        {
            ISTATIC_SetProperties(pApp-
>pIStatic,
                ST_ASCII|ST_NOSCROLL); int
                StrLen=0;
                boolean isFound=FALSE, isEnd=FALSE;
                . . . . .
        }
}
```

Gambar 4.2 Potongan program help dan info pembuat

Jika yang dipilih adalah operasi konversi bahasa atau kamus, maka akan ditampilkan menu input. Untuk keperluan ini dibuat fungsi `Kamus_BuildInput(myapp_t * pApp)`. Seperti pada gambar 4.3 berikut.

```
void Kamus_BuildInput(myapp_t * pApp)
{
    CtlAddItem ai;

    ISHELL_CreateInstance(pApp-
>a.m_pIShell, AEECLSID_SOFTKEYCTL, (void
**) &pApp->pISoft);
    ai.pText = NULL;
    ai.pImage = NULL;
    ai.pszResImage = ai.pszResText =
Kamus_RES_FILE;
    ai.wFont = AEE_FONT_NORMAL;
    ai.dwData = 0;
    . . . . .
    . . . . .
    . . . . .

    ITEXTCTL_SetTitle(pApp->pIText,
Kamus_RES_FILE, STR_INPUT, NULL);
    ITEXTCTL_SetProperties(pApp->pIText,
TP_MULTILINE|TP_FRAME);
    ITEXTCTL_SetMaxSize(pApp-
>pIText, 20);

    ITEXTCTL_Redraw(pApp->pIText);
    ITEXTCTL_SetActive(pApp-
>pIText, TRUE);
    IMENUCTL_SetActive(pApp-
>pISoft, TRUE);

    pApp->m_eAppState = APP_STATE_INPUT;
}
```

Gambar 4.3 Potongan program menu input.

Setelah kata dalam bahasa Inggris diketikkan, maka kata tersebut dapat diproses agar

diperoleh artinya dalam Bahasa Indonesia. Untuk memprosesnya ada beberapa fungsi yang harus dijalankan. Pertama-tama dijalankan fungsi untuk mencari lokasi kata Bahasa Inggris pada database, untuk itu dijalankan fungsi `Kamus_GetFileName(myapp_t* pApp, uint16 wFileName)`. Seperti pada potongan program gambar 4.4 berikut.

```
void Kamus_GetFileName(myapp_t*
pApp, uint16 wFileName)
{
    pApp->pIFileName =
(char*) MALLOC(10*sizeof(char));

    if (STRBEGINS("a", (char*)
pApp->pTextInput))
        STRCPY(pApp-
>pIFileName, (char*)"A");
    else
    if (STRBEGINS("b", (char*)pApp-
>pTextInput))
        STRCPY(pApp-
>pIFileName, (char*)"B");
    . . . . .
    . . . . .
    . . . . .
}
```

Gambar 4.4 Potongan program mencari lokasi kata pada database

Setelah lokasi kata Bahasa Inggris ditemukan, selanjutnya dijalankan fungsi yang sama dengan fungsi untuk menampilkan help, yaitu `Kamus_BuildPrompt(...)`. Namun di dalam fungsi tersebut tag yang dijalankan berbeda dengan tag yang dijalankan untuk menampilkan *help* atau *about*. Tag lain yang dijalankan tersebut adalah menampilkan terjemahan dalam bahasa Indonesia, seperti yang terlihat pada potongan program gambar 4.5.

```

void Kamus_BuildPrompt(myapp_t *
pApp, uint16 wIDTitle, uint16
wIDText, char* pFileName,
EAppStateType eAppState)
{
    . . . .
    switch(wIDTitle)
    {
        case STR_RESULT:
            //Terjemahkan ke Indonesia
            if
            (Kamus_OpenFile(pApp,pFileName))
            {
                if(WSTRLEN(pApp-
>pTextInput)>0)
                {
                    pTextBuf = (AECHAR*)
MALLOC(25 * sizeof(AECHAR));
                    pTextBufC = (char*)
MALLOC(25 * sizeof(char));

                    WSTRCPY(pTextBuf,(AECHAR*)BOUNDRS)
;
                    WSTRCAT(pTextBuf,pApp-
>pTextInput);

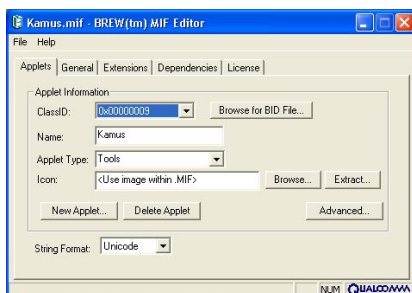
                    WSTRCAT(pTextBuf,(AECHAR*)EQUAL);
                    WSTRCPY(pApp-
>pTextInput,pTextBuf);
                    StrLen = WSTRLEN(pApp-
>pTextInput);
                    do
                    . . . .
                }
            }
    }
}

```

Gambar 4.5 Potongan Program terjemahan bahasa Indonesia

#### 4.1.2 Pembuatan file .mif

File .mif digunakan sebagai dasar untuk menampilkan *icon* aplikasi di menu pilihan aplikasi perangkat *mobile*. Untuk membuat file .mif, digunakan *tool* BREW SDK yang bernama BREW\_MIF.exe seperti yang terlihat pada gambar 4.6.

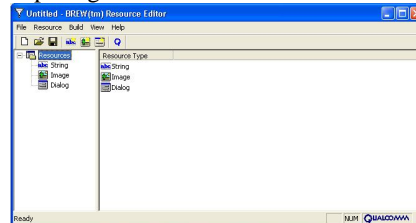


Gambar 4.6 Tool untuk membuat dan mengedit file .mif

#### 4.1.3 Pembuatan file .bri dan file .bar

File .bri dan file .bar adalah file *resource*. File .bri merupakan file teks untuk melakukan konfigurasi, sedangkan file .bar merupakan file biner dari file.bri. Ketika kita mengkonfigurasi *resource*, maka yang diubah adalah file .bri. File

.bri ini kemudian diubah menjadi file .bar agar dapat diakses oleh *applet* aplikasi kamus. Untuk mengonfigurasi file .bri, dijalankan *tool* BREW yaitu BREW\_RES.exe, tampilannya sebagaimana terlihat pada gambar 4.7 berikut:



Gambar 4.7 Tool untuk membuat dan mengedit file .bri

#### 4.1.4 Pembuatan file database

File database yang digunakan pada aplikasi ini hanyalah file teks biasa. Terdapat 26 file teks yang diberi nama menurut abjad alfabetik. Misalnya kata “axe” disimpan dalam file Aeng2ind.dic, kata “zip” disimpan dalam file Zeng2ind.dic. Sedangkan cara penulisan kata Bahasa Inggris beserta artinya dilakukan sebagaimana diperlihatkan pada gambar 4.8.

```

B|baa=suara biri2|babble=meniru
suara bayi,membuka
rahasia|babe=bayi|babel=bising|babt
ismal=berkenaan dgn
pembaptisan|baptize=mempermandikan|b
abyhood=masa
bayi|bachelor=bujang|backbone=tulang
punggung/blk,kekuatan,kemantapan|bac
ker=penyokong,org yg
bertaruh|backfire=meletus tdk pd
tempatny|backlog=pesanan yg blm
terpenuhi,terblk
pekerjaannya|backslide=kembali lagi
ke jln sesat

```

Gambar 4.8 Contoh file database

Setiap kata Bahasa Inggris disertai langsung dengan arti dalam Bahasa Indonesia, dipisahkan menggunakan tanda “=”. Sedangkan untuk memisahkan kata Bahasa Inggris dengan kata Bahasa Inggris lainnya digunakan tanda “|”.

#### 4.2 Uji Coba Aplikasi

Uji coba dilakukan untuk memastikan apakah *applet* dapat dijalankan dengan baik atau tidak. Perangkat lunak yang digunakan adalah *tool* dari BREW, yaitu *emulator* versi 2.1.0.20. Sedangkan perangkat *mobile* yang diemulasikan adalah LGE\_CX-300L. Untuk perangkat kerasnya digunakan komputer Pentium 4 dan sistem operasi Windows XP.

Langkah-langkah pengujiannya diawali dengan menjalankan *emulator* dan didapatkan hasil sebagaimana pada gambar 4.9 berikut ini:





Gambar 4.9 Emulator untuk perangkat tipe LGE\_CX-300L

Setelah menjalankan program *emulator* kemudian memilih *applet* kamus dengan menekan kursor hingga *icon applet* kamus ditampilkan, dilanjutkan dengan menekan OK, dengan cara menekan tombol Enter pada keyboard. Dengan ditekannya tombol OK, maka menu utama ditampilkan sebagaimana Gambar 4.10.




Gambar 4.10 Menu Utama applet

Untuk menjalankan operasi pencarian arti kata Inggris tekan tombol OK, maka akan didapatkan hasil sebagaimana pada gambar 4.11. Pada halaman input ini *user* dapat mengetikkan kata Bahasa Inggris yang ingin dicari artinya dalam Bahasa Indonesia.



Gambar 4.11 Menu Input applet

Setelah memasukkan kata Bahasa Inggris, misalnya saja “axe” proses dilanjutkan dengan menekan tombol OK. Kursor kemudian menyorot kursor yang ada dibawah halaman. Dengan mengarahkan kursor pada tanda , dan diikuti dengan menekan OK, maka ditampilkan arti kata “axe” dalam bahasa indonesia seperti gambar 4.12.



Gambar 4.12 Menu hasil operasi

Untuk menampilkan menu help, dari menu utama seperti pada gambar 4.10, arahkan kursor menuju tombol Help, lalu tekan tombol OK, maka didapatkan hasil sebagaimana yang diperlihatkan pada gambar 4.13 berikut ini.



Gambar 4.13 Menu Help

Sedangkan untuk menampilkan Menu About, langkah yang dilakukan diawali dari Menu Utama dengan memilih tombol About diikuti menekan OK. Hasilnya adalah seperti gambar 4.14 berikut ini.



Gambar 4.14 Menu About

### 4.3 Evaluasi Uji Coba

Dengan langkah-langkah uji coba di atas, dapat dilakukan evaluasi sebagaimana berikut:

1. Program dapat dijalankan dengan baik dan diperoleh hasil yang sesuai dengan tujuan pembuatan tugas akhir.
2. Untuk menjalankan aplikasi tidak diperlukan langkah-langkah yang rumit, hanya prosedur sesuai dengan disain aplikasi yang sudah dibuat.
3. Response aplikasi tidak terlalu cepat, dikarenakan setting memori emulator diset sesuai dengan perangkat *mobile* sesungguhnya.

Pada uji coba untuk melakukan pencarian arti kata Bahasa Inggris dapat dievaluasi hasil sebagaimana berikut:

1. Aplikasi dapat menunjukkan padanan kata Bahasa Inggris ke Bahasa Indonesia.
2. Untuk menunjukkan “help” dan “about” dapat dilakukan dengan mudah melalui Menu Utama yang sudah disediakan. Langkah-langkah dalam menjalankan aplikasi sudah sesuai dengan diagram alir yang dirancang.

## V. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Mengenai pembuatan aplikasai *applet* menggunakan teknologi BREW ini didapat kesimpulan sebagaimana berikut:

1. Aplikasi Kamus Bahasa Inggris Indonesia dapat digunakan untuk membantu pengguna untuk mencari padanan kata satu kata berbahasa Inggris ke Bahasa Indonesia.
2. Aplikasi dapat dibangun menggunakan teknologi BREW secara mudah dengan system operasi Windows dan BREW SDK.
3. Library BREW SDK dengan kelengkapan *interface*-nya dapat digunakan secara mudah untuk membangun aplikasi.

### 5.2 Saran

Pada program aplikasi dapat dikembangkan aplikasi kamus sebagaimana berikut:

1. Aplikasi untuk konversi Bahasa Indonesia ke Bahasa Inggris.
2. Aplikasi kamus dari Bahasa Indonesia ke bahasa-bahasa lain atau sebaliknya.
3. Untuk kelengkapan bisa dibangun modul cara mengucapkan suatu kata.
4. Database yang ada saat ini disimpan di pesawat *mobile*, namun untuk pengembangan *database* disimpan di sisi *server*. Sehingga jika aplikasi dijalankan, program akan melakukan akses *http* ke server.

Untuk pengembangan di masa depan dapat dibuat aplikasi yang lebih kompleks, misalnya adalah aplikasi untuk mengecek jaringan, aplikasi permainan yang menampilkan grafik yang

mendetail dan lain sebagainya. Dengan membuat aplikasi yang kompleks ini maka *interface-interface* yang dimiliki oleh BREW SDK dapat dieksplorasi lebih lanjut.

## DAFTAR PUSTAKA

1. Barbagallo, Ralph, *Wireless Games Development in C/C++ with BREW*, Wordware Publishing, 2003.
2. Dreamtech Software Team, *WAP, Bluetooth and 3G Programming : Cracking the Code*, Hungry Minds Incorporated, New York, 2002.
3. Echolas, John M, *Kamus Inggris-Indonesia*, PT. Gramedia Indonesia.
4. Haykin S, *Digital Communication*, Addison Wesley, 1988.
5. Lee, Jhong Sam, *CDMA Systems Engineering Handbook*, Artech House, ISBN 0-89006-990-5.
6. Lee, William C.Y, *Mobile Cellular Telecommunication System*, McGraw-Hill Book Co, New York, 1990.
7. Qualcomm Team, *BREW API reference*, Qualcomm Incorporated, San Diego, 2001.
8. Qualcomm Team, *BREW Overview*, Qualcomm Incorporated, San Diego, 2002.
9. Qualcomm Team, *BREW User Interface Design Guidelines*, Qualcomm Incorporated, San Diego, 2002.
10. Qualcomm Team, *Application Developer's TRUE BREW Process Overview*, Qualcomm Incorporated, San Diego, 2002.
11. Qualcomm Team, *BREW Guidelines on setting Extention Depedency in the MIF*, Qualcomm Incorporated, San Diego, 2002.
12. Qualcomm Team, *BREW Fact Sheet*, Qualcomm Incorporated, San Diego, 2002.
13. Qualcomm Team, *BREW Distribution System*, Qualcomm Incorporated, San Diego, 2002.
14. Qualcomm Team, *BREW and J2ME*, Qualcomm Incorporated, San Diego, 2002.
15. Qualcomm Team, *Application Developer's TRUE BREW Test Guide: Requirement and Test Cases*, Qualcomm Incorporated, San Diego, 2002.

Mengesahkan,

Pembimbing I

Pembimbing II

Ir. Tejo Sukmadi, MT.  
NIP. 131 764 876

Agung Budi Prasetyo, ST, MIT.  
NIP. 132 137 932