

**PERANGKAT LUNAK PENGUCAPAN TEKS SMS
PADA TELEPON SELULER**

Anggi Erlangga¹, Alex Wijaya², Wydyanto³

Mahasiswa Universitas Bina Darma¹, Dosen Universitas Bina Darma^{2,3}

Jalan Jenderal Ahmad Yani No.12 Palembang

Pos-el:erlangga.gembrel21@yahoo.co.id¹,@lexw1jaya@yahoo.com²,

widiwidyanto@yahoo.com³

Abstract

Communication technologies is also growing very rapidly as more and more demand for people to technology. One of these emerging technologies is currently Communicatin Global System for Mobile (GSM), as well as service Short Message Service (SMS) as a cellular telephone services are the most popular and most sought after today, because its use is relatively easy and cost is very cheap. With a system that can turn SMS text to speech people do not need to read but enough to hear it. Text conversion system of SMS (Short Message Service) to the speech useful for users who are busy, in order to more easily get information from a text SMS (Short Message Service) without having to open the SMS (Short Message Service) on mobile phones and can still perform activity other. With the problems described above, the authors are interested in taking the title "Pronunciation Software on Mobile SMS Text".

Keywords: Software, pronunciation, Text SMS, Cell Phones

Teknologi komunikasi juga berkembang sangat pesat seiring dengan semakin banyaknya permintaan masyarakat akan teknologi tersebut. Salah satunya teknologi yang berkembang saat ini adalah Global System for Mobile Communicatin (GSM), demikian pula dengan layanan Short Message Service (SMS) sebagai salah satu layanan telephone seluler yang paling populer serta paling diminati saat ini, karena penggunaannya yang relative mudah serta biayanya yang sangat murah. Dengan adanya sistem yang dapat mengubah teks SMS ke ucapan orang tidak perlu membaca namun cukup mendengarnya. Sistem pengubahan teks SMS (Short Message Service) ke ucapan bermanfaat bagi pengguna yang sibuk, agar lebih mudah mendapat informasi dari suatu teks SMS (Short Message Service) tanpa harus membuka SMS (Short Message Service) tersebut pada telepon seluler dan masih bisa melakukan aktivitas yang lain. Dengan permasalahan diatas maka penulis tertarik mengambil judul "Perangkat Lunak Pengucapan Teks SMS pada Telepon Seluler".

1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi telah merambah dan dirasakan oleh masyarakat di semua bidang. Salah satu teknologi yang mengalami perkembangan yang sangat pesat adalah dibidang komputer khususnya pada teknologi informasi. Dengan teknologi informasi dalam pengiriman dan mendapatkan suatu data dengan cepat dan tepat.

Teknologi komunikasi juga berkembang sangat pesat seiring dengan semakin banyaknya permintaan masyarakat akan teknologi tersebut. Salah satunya teknologi yang berkembang saat ini adalah *Global System for Mobile Communication (GSM)* atau yang lebih dikenal dengan telepon seluler digital. Telepon seluler merupakan salah satu media yang banyak digunakan oleh masyarakat sekarang ini.

Demikian pula dengan layanan *Short Message Service (SMS)* sebagai salah satu layanan *telephone* seluler yang paling populer serta paling diminati saat ini, karena penggunaannya yang relatif mudah serta biayanya yang sangat murah. Menurut Febrian

(2007:374) *Short Message Service (SMS)* adalah satu fasilitas dari teknologi *Global System for Mobile Communication (GSM)* yang memungkinkan mengirim dan menerima pesan-pesan singkat berupa *text* dengan kapasitas maksimal 160 karakter dari *mobile station (MS)*.

Saat ini SMS (*Short Message Service*) sebagai sarana penyampaian pesan sudah digunakan masyarakat dari *handphone* termurah sampai *handphone* termahal dan juga disebabkan karena biayanya murah. Banyak operator seluler menawarkan gratis SMS (*Short Message Service*) setelah kita melakukan sejumlah SMS (*Short Message Service*), dengan promo tersebut banyak perusahaan yang memanfaatkannya untuk fasilitas promosi melalui SMS (*Short Message Service*). Pesan SMS (*Short Message Service*) yang berupa teks membutuhkan waktu untuk membacanya sehingga sistem yang dapat mengubah teks SMS (*Short Message Service*) menjadi ucapan dibutuhkan. Dengan adanya sistem yang dapat mengubah teks SMS ke ucapan orang tidak perlu membaca namun cukup mendengarnya. Sistem pengubahan teks SMS ke ucapan bermanfaat bagi pengguna yang sibuk,

agar lebih mudah mendapat informasi dari suatu teks SMS (*Short Message Service*) tanpa harus membuka SMS (*Short Message Service*) tersebut pada telepon seluler dan masih bisa melakukan aktivitas yang lain. Dengan pembahasan diatas maka penulis tertarik mengambil judul “**Perangkat Lunak Pengucapan Teks SMS pada Telepon Seluler**”.

1.2 Perumusan Masalah

Adapun yang menjadi rumusan permasalahan dalam penulisan skripsi ini adalah ”Bagaimana Membangun Perangkat Lunak Pengucapan Teks SMS pada Telepon Seluler”.

1.3. Batasan Penelitian

Agar pembahasan lebih terarah, maka penulis membatasi permasalahan hanya pada membangun perangkat lunak pengucapan teks SMS pada telepon seluler berbasis android dengan menggunakan *library text to speech android*.

1.4 Tujuan Penelitian

Tujuan yang hendak dicapai penulis dengan penyusunan skripsi adalah untuk membangun perangkat lunak pengucapan teks SMS pada telepon seluler agar dapat bermanfaat bagi pengguna telepon seluler yang sibuk.

2. METODELOGI PENELITIAN

2.1 Metode Pengumpulan Data

Dalam melakukan penelitian untuk mendapatkan data dan informasi, maka metode yang digunakan dalam proses pengumpulan data dilakukan sebagai berikut :

1. Metode Observasi

Dalam hal ini yang akan dilakukan adalah melihat serta mempelajari permasalahan yang ada dilapangan yang erat kaitannya dengan objek yang diteliti, objek yang diteliti yaitu Perangkat Lunak Pengucapan Teks SMS pada Telepon Seluler.

2. Studi Pustaka

Metode yang dilakukan adalah dengan cara mencari bahan yang mendukung dalam pendefinisian masalah melalui buku-buku, *internet*, yang erat kaitannya dengan objek permasalahan.

2.2 Metode Pengembangan

Perangkat Lunak

Metode yang digunakan untuk pengembangan perangkat lunak adalah metode *sekunsial linier* yang disebut siklus kehidupan klasik atau waterfall model (model air terjun), yang dimulai pada tingkat danb kemajuan sistem pada

seluruh analisis, desain, kode, pengujian dan pemeliharaan Pressman (2002:36).

1. Rekayasa dan pemodelan sistem
Rekayasa dan pemodelan sistem menyangkut pengumpulan kebutuhan pada tingkat sistem serta informasi yang mencakup pengumpulan kebutuhan atau pengumpulan data.
2. Analisis kebutuhan perangkat lunak
Tahap analisis adalah tahap pengumpulan kebutuhan-kebutuhan dari semua elemen sistem perangkat lunak yang akan dibangun. Pada tahap ini dibentuk.
3. Perancangan Perangkat Lunak
Membuat Perangkat Lunak Pengucapan Teks SMS pada Telepon Seluler dengan menggunakan metode perancangan model fungsional yaitu dengan dengan tabel-tabel yang diperlukan, mendesain rancangan dengan UML desain tampilan *input* dan *output*.
4. Pembuatan Kode
Mengimplementasikan hasil perancangan Perangkat Lunak Pengucapan Teks SMS pada Telepon Seluler Berbasis Android.
5. Pengujian Perangkat Lunak
Pengujian Perangkat Lunak Pengucapan Teks SMS pada

Telepon Seluler diperlukan untuk mengetahui apakah sistem baru yang dibuat telah sesuai dengan spesifikasi kebutuhan yang diinginkan.

6. Pemeliharaan
Pemeliharaan merupakan bagian akhir dari pengembangan Perangkat Lunak Pengucapan Teks SMS pada Telepon Seluler dan dilakukan setelah perangkat lunak digunakan.

2.3 Landasan Teori

2.3.1 Perangkat Lunak

memungkinkan program memanipulasi informasi secara proporsional. Kristanto (2004:1).

Dari dua pendapat diatas maka dapat disimpulkan bahwa perangkat lunak adalah suatu instruksi (program komputer) yang dibuat oleh sebuah perusahaan komputer untuk mengerjakan tugas tertentu. Contoh *Microsoft Word*.

2.3.2 Short Message Service (SMS)

Short Message Service (SMS) adalah pesan singkat berupa teks yang dikirim dan diterima antar sesama pengguna telepon, pada awalnya pesan ini digunakan antar telpon genggam, namun dengan berkembangnya teknologi, pesan

tersebut bisa dilakukan melalui komputer ataupun telpon rumah. Dengan short message service Menurut Pressman (2002:10), perangkat lunak adalah perintah (program komputer) yang bila dieksekusi memberikan fungsi dan unjuk kerja seperti yang diinginkan atau yang mengatur struktur data memungkinkan program memanipulasi informasi secara professional dan mengatur dokumen yang menggambarkan operasi kegunaan program. Perangkat lunak merupakan perangkat yang dapat dilihat oleh mata, namun tidak dapat diraba. Perangkat lunak juga dapat digunakan untuk menunjukkan sebuah program yang dipakai oleh sistem komputer. Komputer memerlukan perangkat lunak untuk beroperasi dan membutuhkan program untuk membuat komponen-komponen bekerja bersama-sama dengan baik.

Karakteristik perangkat lunak menurut Pressman (2002 : 4) meliputi :

1. Merupakan produk yang unik
2. Perangkat lunak tidak akan pernah rusak karena selalu diperbaiki
3. Perangkat lunak pada umumnya dibangun sesuai keinginan, jadi tidak dibentuk dari komponen yang sudah ada.
4. Fleksibel, sehingga mudah di modifikasi.

5. Dihubungkan dengan sistem komputer.

Perangkat lunak bantu adalah alat bantu untuk mempermudah suatu penyelesaian masalah yang timbul seperti komputer. Dimana komputer merupakan suatu alat bantu dalam mempermudah suatu pekerjaan dengan cepat serta dapat memberikan informasi dalam permasalahan yang timbul.

Perangkat lunak adalah kecanggihan teknologi dari perangkat keras yang akan berfungsi apabila diberikan tugas-tugas dasar tertentu yang diperlukan semua pemakai suatu komputer. Dan membantu mempermudah pengolahan sumber daya fisik, berfungsi sebagai *interface* antara *user*, *software* yang memproses sebuah data tersebut.

Perangkat Lunak adalah instruksi (program komputer) yang diketikan dijalankan menyediakan fungsi dan tampilan yang diinginkan, struktur data yang memberikan kesempatan program untuk memanipulasi informasi dan dokumen yang mendeskripsikan operasi dan penggunaan program, mengatur struktur data (SMS), pengguna hp GSM dapat mengirim dan menerima message singkat yang biasanya sampai 160 karakter. Text dapat berupa kata atau nomor atau kombinasi alphanumeric.

Febrian (2007:374). SMS adalah singkatan dari *Short Message Service* atau layanan pesan singkat. Awalnya SMS berfungsi untuk memberikan layanan pengiriman pesan teks singkat antarperangkat *mobile phone* (telepon genggam/telepon bergerak). SMS sebetulnya hanya layanan tambahan terhadap dua layanan utama (layanan *voice* dan *switched data*) dalam sistem jaringan komunikasi GSM. GSM, singkatan dari *Global System for Mobile Communications* adalah Perkumpulan Penyedia Perangkat Komunikasi Eropa yang menyediakan standarisasi perangkat telepon genggam/telepon bergerak di Eropa. SMS yang muncul di Eropa awal tahun 1991, sangat populer di Asia, Australia dan Eropa. Zakaria dan Widiadhi (2006:2).

Dari dua pendapat diatas maka dapat disimpulkan bahwa *Short Message Service (SMS)* adalah fasilitas dari *Global System for Mobile Communications (GSM)* untuk mengirim dan menerima pesan dalam bentuk *text* ke dan dari sebuah *handphone*.

2.3.4 Audio

Audio adalah bunyi yang dapat didengar oleh manusia. sistem-sistem yang berkaitan dengan proses perekaman dan transmisi yaitu sistem

pengambilan/penangkapan suara, sambungan transmisi pembawa bunyi. Sudarmo (2006:28).

Audio adalah dalam sistem komunikasi bercirikan video, sinyal elektrik digunakan untuk membawa unsur bunyi. Istilah ini juga biasa digunakan untuk menerangkan sistem-sistem yang berkaitan dengan proses perekaman dan transmisi yaitu sistem pengambilan/penangkapan suara, sambungan transmisi pembawa bunyi, amplifier dan lainnya. Febrian (2007:46).

Dari dua pendapat diatas audio dapat disimpulkan sistem komunikasi bercirikan video, sinyal elektrik digunakan untuk membawa unsur bunyi.

2.4 Java

Menurut Shalahuddin (2008:1), *java* adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan *Java* tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*.

2.4.1 Sebagian Fitur dari Java

2.4.2 Java Virtual Machine (JVM)

Menurut Shalahuddin (2008: 10), *JVM* adalah sebuah mesin imajiner

(*maya*) yang bekerja dengan menyerupai aplikasi pada sebuah mesin nyata. *JVM* menyediakan spesifikasi *hardware* dan *platform* dimana kompilasi kode *Java* terjadi. Spesifikasi inilah yang membuat aplikasi berbasis *Java* menjadi bebas dari *platform* manapun karena proses kompilasi diselesaikan oleh *JVM*.

Aplikasi program *Java* diciptakan dengan *file* teks berekstensi *.Java*. Program ini dikompilasi menghasilkan satu berkas *bytecode* berekstensi *.class* atau lebih. *Bytecode* adalah serangkaian instruksi serupa instruksi kode mesin. Perbedaannya adalah kode mesin harus dijalankan pada sistem komputer dimana kompilasi ditujukan, sementara *bytecode* berjalan pada *Java interpreter* yang tersedia di semua *platform* sistem komputer dan sistem operasi.

2.4.3 Garbage Collection

Banyak bahasa pemrograman lain yang mengizinkan seorang programmer mengalokasikan memori pada saat dijalankan. Program *Java* melakukan *garbage collection* yang berarti program tidak perlu menghapus sendiri objek-objek yang tidak digunakan lagi.

2.4.4 Code Security

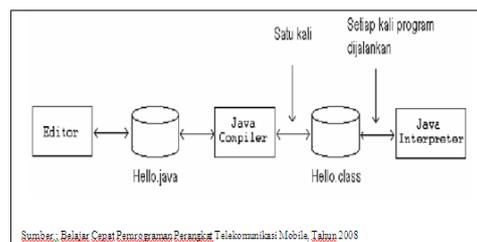
Code Security terimplementasi pada *Java* melalui penggunaan *Java Runtime Environment(JRE)*. *Java*

menggunakan model pengamanan 3 lapis untuk melindungi sistem dari *untrusted Java Code*.

1. Pertama, *class-loader* menangani pemuatan kelas *Java* ke runtime *interpreter*. Proses ini menyediakan pengamanan dengan memisahkan kelas-kelas yang berasal dari *local disk* dengan kelas-kelas yang diambil dari jaringan. Hal ini membatasi aplikasi Trojan karena kelas-kelas yang berasal dari *local disk* yang dimuat terlebih dahulu.
2. Kedua, *bytecode verifier* membaca *bytecode* sebelum dijalankan dan menjamin *bytecode* memenuhi aturan-aturan dasar bahasa *Java*.
3. Ketiga, manajemen keamanan menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumber daya seperti sistem *file*, *port* jaringan, proses eksternal dan sistem *windowing*.

2.4.5 Fase-fase pemrograman Java

Gambar di bawah ini menjelaskan aliran proses kompilasi dan eksekusi sebuah program *Java* :



Gambar 2.1 Fase dari sebuah Program Java

Keterangan :

1. *Editor* merupakan aplikasi untuk membuat atau mengedit aplikasi *java*.
2. *Hello.java* merupakan *file java* yang sudah mempunyai kode-kode *java* dan di simpan dengan extensi *java*.
3. *Java compiler* merupakan merubah *file java* yang extensi *java* menjadi *file* yang extensi *class*.
4. *Hello.class* merupakan hasil *file java compiler* yang asalnya berupa *file extensi java*.
5. *Java interpreter* merupakan menterjemahkan baris per baris kode ke dalam bahasa mesin dan menampilkan hasilnya.

2.5 Android

2.5.1 Pengertian Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan platform yang terbuka bagi para pengembang untuk menciptakan aplikasi baru. Android merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembangan untuk melakukan pengembangan sesuai dengan yang

diharapkan. Pada saat perilisan perdana android, 5 November 2007, android bersama *open handset alliance* menyatakan mendukung pengembangan *open source* pada perangkat lunak *mobile*. Di lain pihak, google merilis kode-kode android dibawah lisensi *apache*, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler. Safaat (2011:1).

2.5.2 Android SDK (*Software Development Kit*)

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk memulai pengembangan aplikasi pada *platform* Android menggunakan bahasa pemrograman *Java*. Android merupakan subset perangkat lunak untuk ponsel yang meliputi, sistem operasi, *middleware* dan aplikasi yang di *release* oleh *google*. Saat ini disediakan android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi platform android menggunakan bahasa pemrograman *java*. Sebagai platform aplikasi-netral, android memberi anda kesempatan untuk membuat aplikasi yang kita butuhkan yang buka merupakan aplikasi bawaan *handphone*. Safaat (2011:5).

2.5.3 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan salah satu bentuk language atau bahasa, menurut pencetusnya UML di definisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem. Nugroho (2004:16).

a. Use Case Diagram

Use case diagram bersifat statis, diagram ini memperlihatkan himpunan *use case* dan aktor-aktor suatu jenis khusus dari kelas. Terutama sangat penting untuk mengorganisasikan dan memodelkan perilaku dari suatu sistem.

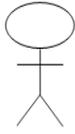
2.5.4 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan salah satu bentuk language atau bahasa, menurut pencetusnya UML di definisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem. Nugroho (2004:16).

a. Use Case Diagram

Use case diagram bersifat statis, diagram ini memperlihatkan himpunan *use case* dan aktor-aktor suatu jenis khusus dari kelas. Terutama sangat penting untuk mengorganisasikan dan memodelkan perilaku dari suatu sistem.

Tabel 2.1 Simbol Use Case

No.	Simbol	Keterangan
1.	Aktor 	Merupakan kesatuan <i>eksternal</i> yang berinteraksi dengan sistem.
2.	<i>Use Case</i> 	Rangkaian uraian sekelompok yang saling terkait dan membentuk sistem.
3.	<i>Generalization</i> 	Menggambarkan hubungan khusus atau interaksi dalam objek.

b. Activity Diagram

Activity diagram bersifat dinamis, diagram state ini memperlihatkan state-state pada sistem, diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kalaborasi dan pemodelan sistem-sistem yang reaktif.

Tabel 2.2 Simbol Activity Diagram

No.	Simbol	Keterangan
1.		<i>Start state</i> adalah sebuah kondisi awal sebuah <i>object</i> sebelum ada perubahan keadaan. <i>Start state</i> digambarkan dengan sebuah lingkaran solid.
2.		<i>End state</i> adalah menggambarkan ketika objek berhenti memberi respon terhadap sebuah event. <i>End state</i> digambarkan dengan lingkaran solid di dalam sebuah lingkaran kosong.
3.		<i>State</i> atau <i>activities</i> menggambarkan kondisi sebuah entitas, dan digambarkan dengan segiempat yang pinggirnya.
4.	 	<i>Fork</i> atau percabangan merupakan pemisah beberapa aliran konkuren dari suatu aliran tunggal. penggabungan beberapa aliran konkuren dalam aliran tunggal.
6.		<i>Decision</i> merupakan suatu logika aliran konkuren yang mempunyai dua cabang aliran konkuren.

Sumber: Nugroho, *Rational Rose Untuk Pemodelan Berorientasi Objek*, 2004.

2.6 Analisis Dan Perancangan

2.6.1 Rekayasa dan Pemodelan Sistem

Adapun objek yang diteliti adalah membahas permasalahan perangkat lunak pengucapan teks sms pada telepon seluler. Diharapkan dengan perangkat lunak pengucapan teks SMS pada telepon seluler dapat membantu pengguna telepon seluler dalam mendapat informasi dari suatu teks SMS (*Short Message Service*) tanpa harus membuka SMS (*Short Message Service*) tersebut pada telepon seluler dan masih bisa melakukan aktivitas yang lain.

Kebutuhan *hardware* dan *software* yang digunakan dalam perangkat lunak

pengucapan teks sms pada telepon seluler yang digunakan meliputi alat serta bahan–bahan penunjang lainnya.

1. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan adalah komputer dengan spesifikasi minimal sebagai berikut :

1. *Processor Intel Core 2 Duo*
2. *RAM 1 GB*
3. *Hardisk 80 GB,*
4. *Monitor SVGA Color*
5. *CDRW Room 52 x*
6. *Printer*
7. *Mouse*
8. *Keyboard*

2. Perangkat Lunak (*Software*)

1. *Microsoft Windows XP* atau *Sesuai denagn kebutuhan*
2. *Java (jdk-6u22-windows-i586)*
3. *Android SDK (software Development Kit)*
4. *Eclipse IDE For java Developers*
5. *Microsoft Word XP*
6. *Microsoft Power Point Xp*

2.6.2 Analisis Kebutuhan Perangkat Lunak

Pada tahap analisis ini menjelaskan bagaimana merancang dan membangun perangkat lunak pengucapan teks sms pada telepon seluler berbasis android. Saat ini SMS

(*Short Message Service*) sebagai sarana penyampaian pesan sudah digunakan masyarakat dari *handphone* termurah sampai *handphone* termahal dan juga disebabkan karena biayanya murah. Banyak operator seluler menawarkan gratis SMS (*Short Message Service*) setelah kita melakukan sejumlah SMS (*Short Message Service*), dengan promo tersebut banyak perusahaan yang memanfaatkannya untuk fasilitas promosi melalui SMS (*Short Message Service*).

Permasalahan yang ada pada saat ini Pesan SMS (*Short Message Service*) yang berupa teks membutuhkan waktu untuk membacanya sehingga sistem yang dapat mengubah teks SMS (*Short Message Service*) menjadi ucapan dibutuhkan. Dengan adanya sistem yang dapat mengubah teks SMS ke ucapan orang tidak perlu membaca namun cukup mendengarnya. Sistem perubahan teks SMS ke ucapan bermanfaat bagi pengguna yang sibuk, agar lebih mudah mendapat informasi dari suatu teks SMS (*Short Message Service*) tanpa harus membuka SMS (*Short Message Service*) tersebut pada telepon seluler dan masih bisa melakukan aktivitas yang lain.

2.6.3 Arsitektur Aplikasi

Pada arsitektur sistem konversi *short message service* (SMS) ke Audio menjelaskan bahwa *user* mengirim SMS ke nomor *handphone* yang telah ditentukan, *handphone* menerima sms kemudian program mengecek isi sms dan isi sms ke diubah ke audio.

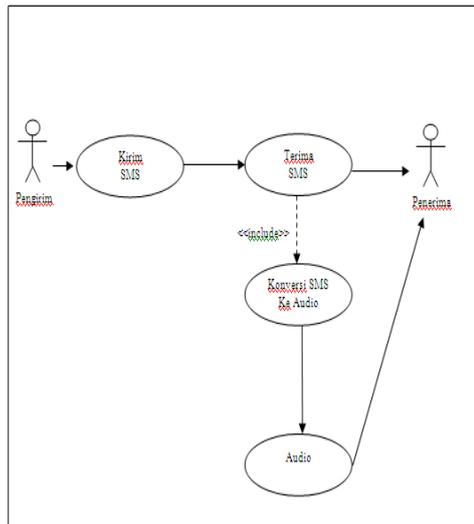


Gambar 3.1 Arsitektur Aplikasi

2.6.4 Unified Modeling Language (UML)

1. Use Case Diagram (UCD)

Use case Diagram (UCD) menjelaskan apa yang akan dilakukan oleh sistem yang akan dibangun dan siapa yang berinteraksi dengan sistem. UCD menjadi dokumen kerja dari admin dan *user*.



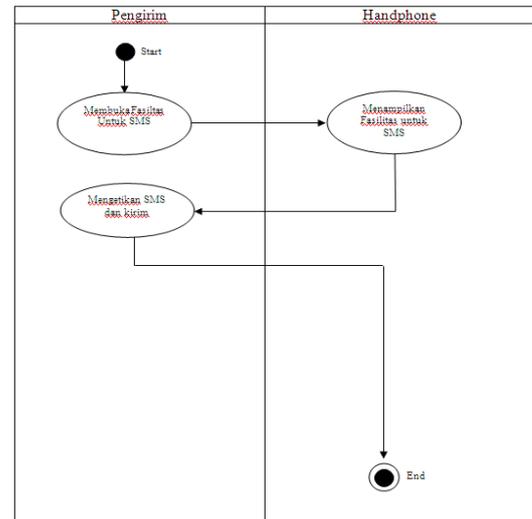
Gambar 3.2 Use Case Diagram

Pengirim mengirimkan SMS ke *handphone* penerima, perangkat *handphone* menerima informasi SMS tersebut, kemudian perangkat lunak pengucapan teks sms pada telepon seluler yang sudah di *install* di *handphone* akan mengkonversikan teks SMS ke audio dan user mendapatkan informasi berupa suara dari SMS tersebut.

2.7. Diagram Activity

a. Diagram Activity Pengirim

Pada diagram *activity* user menjelaskan tentang proses pengirim dengan mengirim sms dengan dengan *handphone*.



Gambar 3.3 Diagram Activity Pengirim

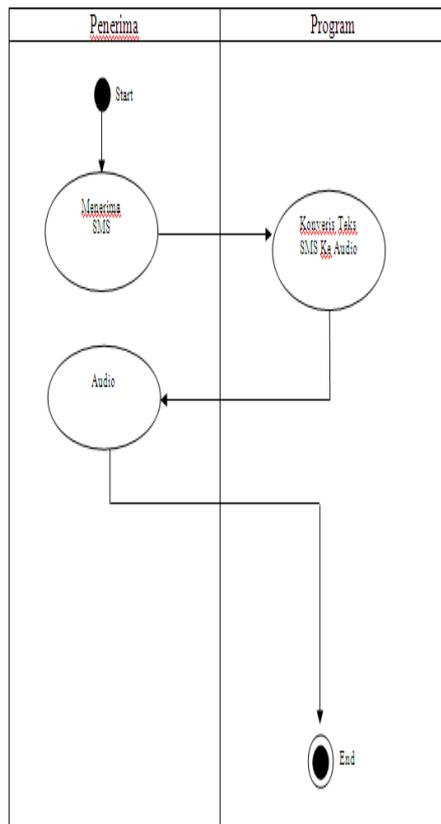
Pengirim menampilkan atau membukan fasilitas untuk SMS pada *handphone*. Pengirim mengetikkan SMS dan mengirim SMS.

b. Diagram Activity Penerima

Pada diagram *activity* penerima menjelaskan tentang proses penerima mengenai tentang proses penerima SMS dengan perangkat lunak pengucapan teks SMS pada telepon seluler yang mana penerima akan menerima sms dan program melakukan konversi teks sms ke audio dan secara otomatis.

Penerima membuka perangkat lunak pengucapan teks SMS pada telepon seluler. Penerima menerima SMS,

aplikasi secara otomatis akan konversi teks SMS ke audio



Gambar 3.4 Diagram Activity Penerima

2.7.1 Desain Tampilan

1. Rancangan Halaman Pembuka

Rancangan halaman pembuka merupakan halaman pertama ketika program dijalankan tampilannya seperti gambar dibawah.

Pada halaman pembuka ini akan tampil jika pada *handphone* di klik *icon* aplikasi ini, fasilitas *library text to speech* android pada halaman pembukaan ini secara otomatis sudah dijalankan,

sehingga jika halaman pembuka ini di tampilkan dan ada SMS yang masuk ke *handphone* yang sudah ada aplikasi ini sudah di instal maka secara otomatis akan mengkonversi teks SMS tersebut menjadi suara sesuai dengan tulisan pada SMS tersebut

2.7.2 Rancangan Halaman Profil

Rancangan profil merupakan rancangan halaman dimana user mendapatkan informasi tentang perangkat lunak pengucapan teks sms pada telepon seluler, tampilannya seperti gambar dibawah ini.



Gambar 3.6 Rancangan Halaman Profil

Untuk menampilkan tombol profil dan tombol keluar, pengguna *handphone* harus klik tombol menu, maka secara otomatis dua tombol tersebut akan tampil, jika di klik tombol profil secara otomatis akan keluar layar baru yang isinya tentang profil perangkat lunak pengucapan teks SMS pada telepon seluler dan juga membahas tentang

library TTS (*text to speech*) android dan juga menjelaskan tentang bahasa-bahasa yang *suport* dengan *library* TTS (*text to speech*) android.

2. Rancangan Tampilan Keluar

Aplikasi

Rancangan tampilan keluar aplikasi merupakan rancangan tampilan keluar aplikasi jika di klik tombol keluar halaman dimana user dapat menjalankan aplikasi agar SMS yang masuk dapat dikonversi menjadi suara dengan



Gambar 3.7 Rancangan Tampilan Keluar Aplikasi perangkat lunak pengucapan teks sms pada telepon seluler, tampilanya seperti gambar dibawah ini.

3. HASIL DAN PEMBAHASAN

3.1 Hasil

Hasil dari rancangan program pada pembahasan bab III yang dibuat skripsi ini adalah tampilan dari masing-masing halaman, bagaimana cara penggunaanya, adapun hasil dari rancangan program ini adalah sebuah perangkat lunak pengucapan teks SMS pada telepon seluler berbasis android dengan menggunakan *library text to speech android*. Perangkat lunak ini digunakan dapat membantu pengguna telepon seluler dalam mendapat informasi dari suatu teks SMS (*Short Message Service*) tanpa harus membuka SMS (*Short Message Service*) tersebut pada telepon seluler dan masih bisa melakukan aktivitas yang lain seperti menyetir, tuna netra dan juga penyiar radio dalam mendapatkan SMS masuk.

3.2 Pembahasan

Pembahasa ini menjelaskan tentang cara *install* perangkat lunak seperti *java*, *android* dan *eclipse*. Cara penggunaan perangkat lunak pengucapan teks SMS pada telepon seluler berbasis *android* dengan menggunakan *library text to speech android*.

3.2.1 Install Perangkat Lunak Pendukung

1. Instal Java (*jdk-6u22-windows-i586*)

- a. Klik dua kali file *jdk-6u22-windows-i586*, seperti gambar berikut.



Gambar 3.1 File *jdk-6u22-windows-i586*

- b. Maka akan tampil jendela *welcome* seperti berikut :



Gambar 3.2 Welcome J2SDK 6

Tampilan jendela *welcome* merupakan tampilan pertama ketika di klik dua kali icon file *jdk-6u22-windows-i586*.

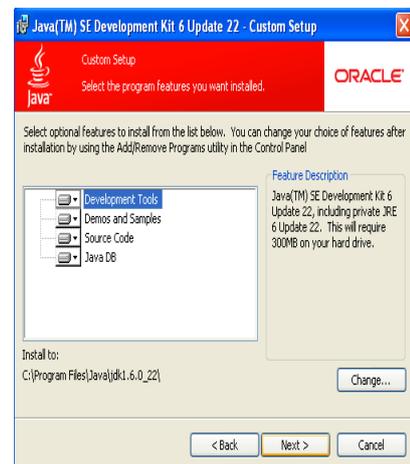
- c. Di jendela *welcome* klik tombol *next*, maka akan tampil seperti berikut :



Gambar 3.3 Welcome

Tampilan jendela *welcome* akan tampil setelah klik tombol *next* untuk menampilkan jendela berikutnya dan tombol *cancel* untuk membatalkan proses berikutnya.

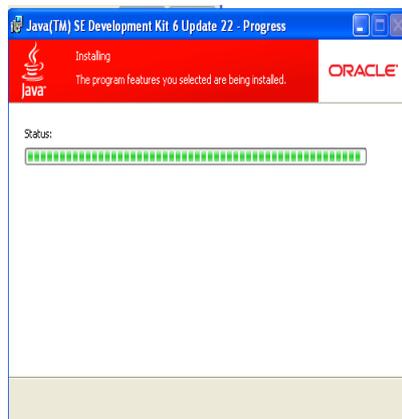
- d. Muncul jendela *custom setup*, klik tombol *next*.



Gambar 3.4 Custom Setup

Tampilan jendela *custom* setup akan tampil setelah klik tombol *next* untuk menampilkan jendela berikutnya, tombol *back* untuk kembali ke jendela sebelumnya, tombol *cancel* untuk membatalkan proses berikutnya.

- e. Klik tombol *next*, muncul proses instalasi J2SDK 6.



Gambar 3.5 Proses Instalasi

Tampilan proses instalasi akan merupakan proses setelah setelah klik tombol *next* pada jendela sebelumnya.

- f. Setelah instal berakhir dengan sukses, muncul jendela *complete*.

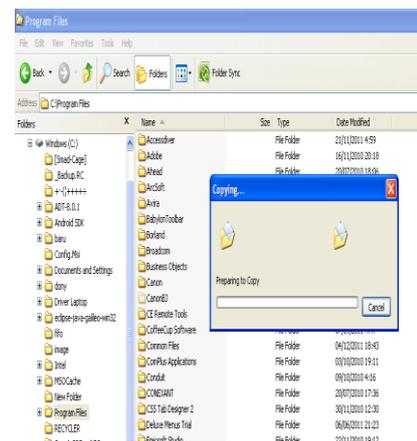


Gambar 3.6 Instalasi *Complete*

Tampilan jendela instalasi *complete* akan tampil setelah klik tombol *next* untuk menampilkan jendela berikutnya, tombol *finish*.

2. Instal Android SDK (*Software Development Kit*)

Untuk menginstal android sdk (*software development kit*) hanya di copy folder android-sdks ke C:\Program Files seperti gambar dibawah ini.

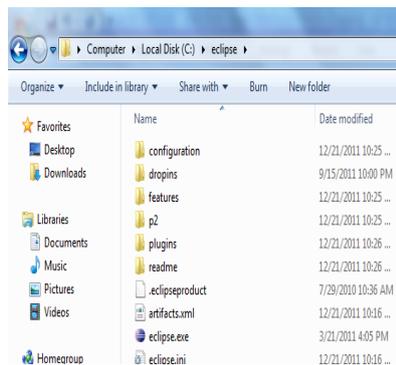


Gambar 3.7 *Copy Folder* Android-SDKS

Tampilan *copy folder* Android-SDKS merupakan proses duplikasi file android SDKS seperti gambar diatas.

3. Instal *Eclipse IDE for Java Developers*

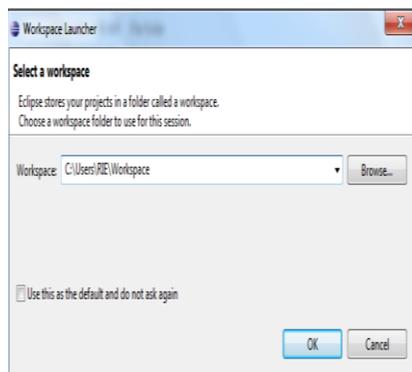
- a. Klik dua kali *file eclipse.exe*, seperti gambar berikut.



Gambar 3.8 File Eclipse.exe

Tampilan diatas menjelaskan tentang menjalankan *eclipse* dengan cara klik dua kali *file* yang diatas.

- b. Maka akan tampil *workspace* klik tombol OK seperti gambar berikut. d.



Gambar 3.9 Workspace

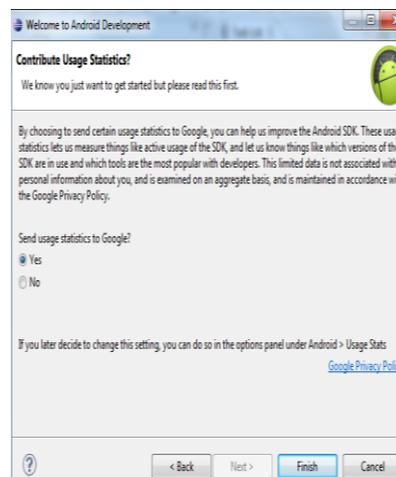
Tampilan diatas menjelaskan tentang menjalankan lokasi *workspace*, terdapat tombol OK untuk proses selanjutnya dan tombol *cancel* untuk membatalkan proses yang ada.

- c. Tampilan untuk mencari lokasi android-sdks, kemudia klik tombol next seperti gambar berikut ini.



Gambar 4.0 Lokasi Android-sdks

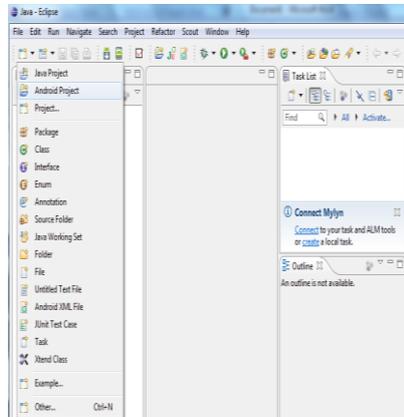
Tampilan diatas menjelaskan tentang menjalankan lokasi *existing location*, terdapat tombol *next* untuk proses selanjutnya dan tombol *cancel* untuk membatalkan proses yang ada. Install *complete* seperti gambar berikut ini.



Gambar 4.1 Complete Install Eclipse

Tampilan diatas menjelaskan tentang menjalankan lokasi *existing location*, terdapat tombol *next* untuk proses selanjutnya dan tombol *cancel* untuk membatalkan proses yang ada.

- e. Tampilan Eclipse seperti gambar dibawah ini.



Gambar 4.2 Tampilan *Eclipse*

Tampilan diatas menjelaskan tentang menjalankan program android, terdapat menu-menu untuk membuat project android, bisa dilihat seperti gambar diatas.

3.2.2 Tampilan Program

Pada tampilan program ini menampilkan seperti halaman pembuka yang menjelaskan tentang gambar dan judul penelitian, halaman profil yang isinya tentang perangkat lunak yang menggunakan *library text to speech android* dan halaman keluar.

1. Halaman Pembuka

Halaman pembuka merupakan halaman pertama ketika program dijalankan, pada halaman pembuka ini tampilannya seperti gambar dibawah.



Gambar 4.3 Halaman Pembuka

Pada halaman pembuka ini akan tampil jika pada *handphone* di klik *icon* aplikasi ini, fasilitas *library text to speech android* pada halaman pembukaan ini secara otomatis sudah dijalankan, sehingga jika halaman pembuka ini di tampilkan dan ada SMS yang masuk ke *handphone* yang sudah ada aplikasi ini sudah di *instal* maka secara otomatis akan menkonversi teks SMS tersebut menjadi suara sesuai dengan tulisan pada SMS tersebut.

2. Halaman Profil

Halaman profil merupakan halaman dimana user mendapatkan informasi tentang perangkat lunak pengucapan teks sms pada telepon seluler, di halaman profil ini menjelaskan tentang perangkat lunak pengucapan teks sms pada telepon seluler menggunakan *library text to speech android* dan hanya

suport terhadap bahasa Inggris, Jerman, Perancis, Italia dan Spanyol tampilanya seperti gambar dibawah ini.



Gambar 4.4 Halaman Profil

Untuk menampilkan tombol profil dan tombol keluar, pengguna *handphone* harus klik tomo menu, maka secara otomatis dua tombol tersebut akan tampil, jika di klik tombol profil secara otomatis akan keluar layar baru yang isinya tentang profil, seperti gambar dibawah ini.

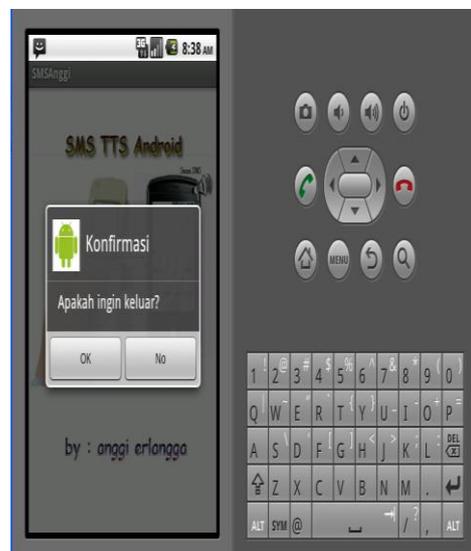


Gambar 4.5 Halaman Isi Profil

Halaman profil yang isinya tentang perangkat lunak yang menggunakan *library text to speech android*. Untuk membuka tombol profil ini klik menu pada *handphone* dan keluar halaman baru.

3. Tampilan Keluar Aplikasi

Tampilan keluar aplikasi merupakan tampilan keluar aplikasi jika di klik tombol keluar halaman dimana user dapat menjalankan aplikasi agar SMS yang masuk dapat dikonversi menjadi suara dengan perangkat lunak pengucapan teks sms pada telepon seluler, tampilanya seperti gambar dibawah ini.



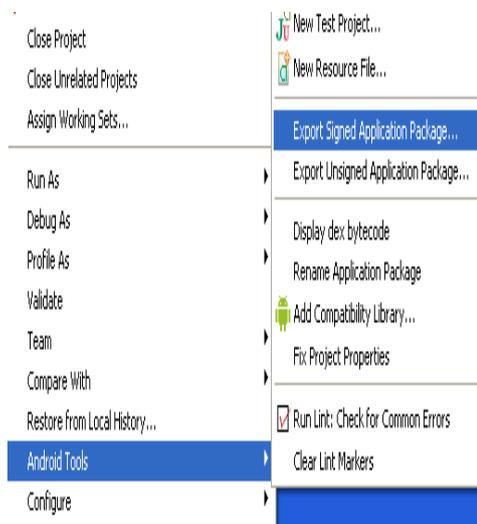
Gambar 4.6 Tampilan Keluar Aplikasi

Tampilan keluar aplikasi merupakan tampilan keluar. Terdapat tombol OK untuk keluar dari perangkat lunak pengucapan teks SMS pada telepon seluler berbasis android dengan

menggunakan *library free java text to speech* dan tombol NO untuk membatalkan keluar dari aplikasi.

4.2.3 Membuat File APK Untuk Install di Handphone

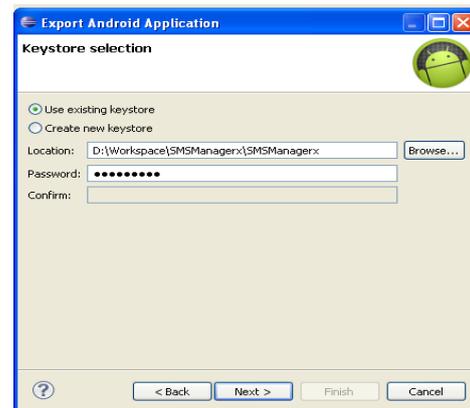
1. Untuk membuat file APK, maka file android yang telah dibuat di eclipse di klik kanan kemudian pilih android tools seperti gambar dibawah ini.



Gambar 4.7 Menu Membuat File APK

Tampilan diatas untuk untuk membuat file APK pada aplikasi eclipse pada perangkat lunak pengucapan teks SMS pada telepon seluler berbasis android dengan menggunakan *library free java text to speech*.

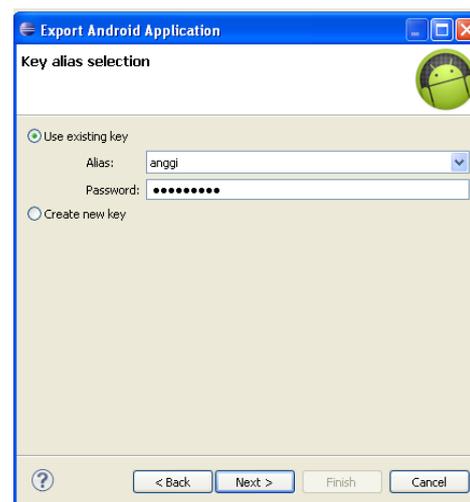
2. Cari lokasi *folder* dari android yang ingin kita buat file APK dan buat *password* sesuai dengan keinginan kita seperti gambar dibawah ini.



Gambar 4.8 Lokasi File Android

Tampilan diatas untuk mencari lokasi file java dengan project android pada proses membuat file APK pada perangkat lunak pengucapan teks SMS pada telepon seluler berbasis android dengan menggunakan *library free java text to speech*.

2. Setelah menemukan lokasi *folder* dari android yang ingin kita buat file APK dan buat *password*, kemudian membuat alias key seperti gambar dibawah ini.



Gambar 4.9 Membuat Alias Key

Tampilan diatas untuk membuat alias dan mengisi *password* pada perangkat lunak pengucapan teks SMS pada telepon seluler berbasis android dengan menggunakan *library free java text to speech*.

3 Setelah menemukan lokasi *folder* dari android yang ingin kita buat file APK dan buat *password*, kemudian membuat alias key seperti gambar dibawah ini.



Gambar 5.0 Install Complete

Tampilan diatas untuk menyimpan file APK pada direktori yang telah ditentukan dan terdapat

tombol finis untuk selesai dari pembuatan pada perangkat lunak pengucapan teks SMS pada telepon seluler berbasis android dengan menggunakan *library free java text to speech*.

4. KESIMPULAN

4.1 Kesimpulan

Berdasarkan dari penelitian yang telah dilaksanakan dan sudah diuraikan dalam perangkat lunak pengucapan teks SMS pada telepon seluler berbasis android, maka penulis dapat menarik kesimpulan sebagai berikut :

1. Penelitian ini menghasilkan perangkat lunak pengucapan teks SMS pada telepon seluler.
2. Perangkat lunak pengucapan teks SMS pada telepon seluler di rancang dan dibuat dengan bahasa pemrograman android dan menggunakan *library text to speech* android .

DAFTAR PUSTAKA

1. Febrian, J. (2002). *Kamus Komputer dan Teknologi Informasi*. Bandung: Informatika.
2. Kristanto, A. (2004). *Rekayasa Perangkat Lunak*. Yogyakarta: Gava Media.
3. Pressman, Roger.S. (2002). *Rekayasa Perangkat Lunak*. Yogyakarta: ANDI.
4. Roliadi. (2010). *Sistem Konversi Short Message Service (SMS) ke Sistem Audio*. Universitas Bina Darma, Palembang.
5. Shalahuddin. (2008). *Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile*. Bandung: Informatika.
6. Sudarmo, P. (2006). *Kamus Istilah Komputer, Teknologi Informasi dan Komunikasi*. Bandung: Yrama Widya.
7. Susetyo, R. (2010). *Aplikasi Mobile Pembaca Short Message Service (SMS) Untuk Tunanetra Menggunakan Model Intonasi Dan Ekspresi Bicara Bahasa Indonesia*. Universitas Pendidikan Indonesia, Bandung.
8. Zakaria, M. (2006). *Aplikasi SMS Untuk Beberapa Keperluan*. Bandung: Informatika.

