

APLIKASI KONVERSI SUARA KE TEKS BERBASIS ANDROID MENGGUNAKAN GOOGLE SPEECH API

Supriyanta, Pudji Widodo dan Bekti Maryuni Susanto
 Program Studi Manajemen Informatika
 AMIK “BSI Yogyakarta”
 Ringroad Barat, Ambarketawang, Gamping, Sleman, Yogyakarta
 Telp (0274) 4342536
 e-mail:supri.spt@bsi.ac.id, pudji.piw@bsi.ac.id, bekti.bms@bsi.ac.id

Abstrak

Memasukkan teks ke dalam perangkat mobile lambat dan rentan kesalahan dibandingkan dengan mengetik pada keyboard ukuran penuh. Google menawarkan fitur pada Android yang bertujuan untuk membuat suara sebagai alternatif metode input, misal penelusuran menggunakan suara, input suara ke dalam text field tertentu dan Application Programming Interface (API) suara untuk pengembang aplikasi. Untuk menelusuri menggunakan suara, pengguna secara sederhana mengetuk ikon mikropon pada kotak pencarian desktop atau dengan menahan tombol pencarian fisik. Penelitian ini bertujuan untuk membangun aplikasi yang mengkonversi suara ke dalam teks dengan Bahasa Indonesia. Aplikasi dibuat menggunakan API Google Speech, bahasa pemrograman yang digunakan bahasa pemrograman Java. Editor Java yang digunakan Eclipse. Aplikasi diujicoba pada smartphone Android. Manfaat dari aplikasi ini adalah mengurangi kesalahan memasukkan input pada bidang teks. Selain itu aplikasi ini juga bisa digunakan untuk merekam pembicaraan yang hasil rekamannya sudah langsung berbentuk teks.

Kata Kunci : Speech To Text, Google API

PENDAHULUAN

Kemajuan di bidang prosesor, khususnya prosesor daya rendah atau embedded processor dan kecerdasan buatan membuat *Human Computer Interaction* (HCI) menjadi ada dimana-mana. *Automatic speech recognition* adalah bagian penting dari HCI dan komputasi *ubiquitous*, mempunyai beberapa keuntungan menarik ketika digunakan sebagai I/O perangkat sebuah sistem. Biasanya, orang dapat berbicara lebih cepat dari pada dia mengetik (Chandra, 2007). *Automatic speech recognition* juga bermanfaat pada aplikasi seperti *automated customer service*, *video game* interaktif, dan untuk mengendalikan kendaraan tanpa awak (unmanned vehicle). Dengan kehadiran baterai yang dioperasikan pada ponsel, PDA/smartphone, *speech recognition* dapat digunakan dalam metode speech to text untuk menulis e-mail/SMS pada PDA/smartphone, resep medis dan sebagainya layaknya I/O perangkat mobile yang dioperasikan dengan baterai yang tidak ramah terhadap pengguna.

Smartphone dan komputer tablet saat ini menjadi alat komunikasi dan asisten pribadi yang sangat penting. Bentuknya yang portabel dan mudah dibawa membuat orang menyukainya sebagai alat utama dalam kehidupan sehari-hari. Berbagai macam aplikasi tersedia di *smartphone* maupun komputer tablet, baik yang memiliki antarmuka grafis maupun dengan antarmuka berbasis suara. *Smartphone* dan komputer tablet berusaha agar orang bisa senyaman mungkin menggunakannya. Sebagai contoh misalnya aplikasi Siri, aplikasi asisten pribadi pada smartphone

Apple. Pada aplikasi siri orang bisa memberi perintah ke *smartphone* melalui suara. Misal mengirim sms, mengirim e-mail, membaca sms, membaca e-mail, mendengarkan musik, mencari web dan sebagainya (Reddy & Mahender, 2013). Namun, sampai saat ini belum ada aplikasi sejenis yang mampu mengenali suara yang Berbahasa Indonesia.

Penggunaan *smartphone* dan komputer tablet ini membawa paradigma baru (Meisel, 2010), yaitu:

1. Smartphone mengarahkan kita untuk terhubung dimanapun kita berada, baik data maupun channel suara.
2. Smartphone benar-benar telepon pribadi, bahkan lebih pribadi dari pada komputer pribadi. Sebagai perangkat pribadi fitur dan layanan disesuaikan dengan kebutuhan dan kesukaan pribadi.
3. Smartphone mendukung semua jenis komunikasi, suara, teks, e-mail dan sebagainya.
4. Smartphone bisa menjadi perangkat bisnis dan perangkat pribadi, kita tidak perlu membawa dua perangkat.
5. Panggilan dan pengiriman data yang tidak terbatas dengan biaya bulanan yang tetap, dengan koneksi Internet biaya panggilan telepon bisa gratis.

Smartphone dan komputer tablet menjadi populer akhir-akhir ini serta menyediakan aplikasi menggunakan input suara (*speech recognition*). Beberapa metode *speech recognition* beroperasi secara langsung pada terminal dan ada yang beroperasi pada server (Iizuka, Tsujino, Oguri, &

Furukawa, 2012). *Speech recognition* yang beroperasi pada server mengirim sinyal suara atau fitur suara ke server, yang menjalankan mesin *speech recognition* dan mengembalikan hasil teks kembali ke terminal. *Speech recognition* pada terminal terbatas pada kecilnya vocabulari karena keterbatasan pemrosesan dan konsumsi daya, tetapi tidak dipengaruhi oleh kondisi komunikasi seperti delay atau tidak terjangkau. *Speech recognition* ini diaplikasikan ke aplikasi seperti operasi terminal, yang terbatas tetapi tersedia sepanjang waktu. Sebaliknya, *speech recognition* berbasis server dipengaruhi oleh kondisi komunikasi tetapi dapat menggunakan teknik yang membutuhkan pemrosesan yang lebih kompleks. Metode ini cocok untuk aplikasi seperti pencarian atau input teks, yang harus mendukung vocabulari yang besar.

Speaker recognition adalah sebuah proses yang memungkinkan mesin untuk memahami dan menginterpretasikan suara manusia dengan menggunakan algoritma tertentu dan memverifikasi kebenaran dari manusia yang mengucapkannya (Kumar & Rao, 2011). Oleh karena itu, *speaker recognition* atau identifikasi pada dasarnya adalah sebuah metode yang secara otomatis mengidentifikasi speaker dari sinyal suara yang direkam atau langsung dengan menganalisa parameter sinyal suara. Pertama, suara manusia dikonversi ke format yang bisa dibaca oleh mesin, setelah itu mesin memproses data. Pemrosesan data berhubungan dengan ekstraksi fitur dan pencocokan fitur. Kemudian, berdasarkan data yang diproses, tindakan yang tepat dilakukan oleh mesin. Tindakan ini tergantung pada aplikasinya.

Google sebagai pengembang sistem operasi Android mempunyai visi agar layanan bisa dinikmati dimana saja (*ubiquitous*) dan tersedia dalam jumlah yang besar. Agar bisa dilayani dimana saja (*ubiquitous*), Google menerapkan layanan berbasis cloud. Dengan layanan berbasis cloud ini, layanan Google bisa diakses dimana saja dan dalam skala yang luas (Schalkwyk, et al., 2010), (Schuster, 2010). Berdasarkan penelitian yang dilakukan oleh Gartner, Inc menunjukkan bahwa pada tahun 2013 sistem operasi Android mempunyai marketshare terbesar (79%), mengungguli pesaingnya, Apple (14,2%), Microsoft (3,3%), BlackBerry (2,7%). Market share ini meningkat 14,8 % dari tahun sebelumnya (Gartner, 2013).

Memasukkan teks ke dalam perangkat mobile sering lambat dan rentan kesalahan dibandingkan dengan mengetik pada keyboard ukuran penuh (Ballinger, Allauzen, Gruenstein, & Schalkwyk, 2010). Google menawarkan beberapa fitur pada Android yang bertujuan untuk membuat suara sebagai alternatif metode input yang layak, misal penelusuran menggunakan suara, input suara ke dalam text field tertentu dan *Application*

Programming Interface (API) suara untuk pengembang aplikasi. Untuk menelusuri menggunakan suara, pengguna secara sederhana mengetuk ikon mikropon pada kotak pencarian desktop atau dengan menahan tombol pencarian fisik. Pengguna dapat berbicara permintaan apapun, dan kemudian akan ditampilkan hasil pencarian Google. Untuk menggunakan fitur input suara, pengguna mengetuk kunci mikropon pada layar keyboard dan kemudian berbicara untuk memasukkan teks secara virtual dimanapun mereka ingin mengetik secara normal. Pengguna bisa memerintah pesan e-mail dan sms, mengisi form pada halaman web, atau memasukkan teks ke dalam aplikasi apapun. API *speech* Android merupakan alat yang sederhana bagi pengembang untuk mengintegrasikan kemampuan pengenalan suara ke dalam aplikasi mereka.

Proses mengkonversi input suara menjadi output digital seperti teks yang dikenal dengan *speech recognition* menyediakan sebuah komputer dengan fitur mendengarkan suara pengguna, memahaminya dan mengubahnya ke dalam bentuk output yang diinginkan dan juga melakukan tindakan yang dibutuhkan (Patel, Patel, & Virparia, 2013). Proses pengenalan suara terdiri dari empat tahapan, yaitu mengubah suara ke dalam sinyal digital, mengekstraksi suara nyata, membuat kerangka suara dan membandingkan kerangka suara dengan kata di dalam grammar untuk menentukan kata yang tepat. *Speech recognition* melibatkan pengucapan ke dalam perangkat mikropon dan memungkinkan software untuk mendeteksi suara, menginterpretasikan suara, mengubah suara ke dalam string dan layanan *speech recognition* ini ditujukan untuk digunakan dengan perintah pendek, seperti frase tanpa jeda.

Penelitian ini bertujuan untuk membangun sebuah aplikasi yang mengkonversi suara ke dalam teks dengan Bahasa Indonesia. Aplikasi dibuat menggunakan API Google Speech, bahasa pemrograman yang digunakan bahasa pemrograman Java. Editor Java yang digunakan adalah Eclipse. Aplikasi diujicoba pada perangkat smartphone Android. Manfaat dari aplikasi ini adalah mengurangi kesalahan memasukkan input pada bidang teks. Selain itu aplikasi ini juga bisa digunakan untuk merekam pembicaraan yang hasil rekamannya sudah langsung berbentuk teks.

TINJAUAN PUSTAKA ANDROID

Android adalah sistem operasi mobile berdasarkan modifikasi versi Linux. Android awalnya dikembangkan oleh sebuah perusahaan pengembang yang namanya sama, Android, Inc. pada tahun 2005, sebagai bagian strateginya dalam memasuki pasar mobile, Google membeli Android

dan mengambil alih pengembangannya sampai sekarang.

Google ingin Android terbuka dan bebas, oleh karena itu sebagian besar kode Android dilepas di bawah lisensi open source Apache, yang berarti bahwa siapapun yang ingin menggunakan Android dapat mendownload sumber kode Android secara penuh. Terlebih lagi bagi vendor (biasanya pabrikan hardware) dapat menambahkan ekstensi propietari pada Android dan menyesuaikan Android untuk membedakan produk Android mereka dengan lainnya. Model pengembangan yang sederhana ini membuat Android sangat menarik dan telah mengisik ketertarikan banyak vendor. Terutama bagi perusahaan yang terpengaruh dengan fenomena iPhone Apple, sebuah kesuksesan produk yang luar biasa yang merevolusi industri smartphone. Perusahaan termasuk Motorola dan Sony yang selama bertahun-tahun mengembangkan sistem operasi mobile sendiri. Ketika iPhone diluncurkan, banyak perusahaan seperti ini berjuang untuk menemukan cara baru untuk merevitalisasi produk mereka. Pabrikan ini melihat Android sebagai sebuah solusi, mereka meneruskan untuk mendesain hardware mereka dan menggunakan Android sebagai Sistem Operasi yang memberikan kekuasaanya.

Keuntungan utama mengadopsi Android adalah Android menawarkan pendekatan terpadu pada pengembangan aplikasi. Pengembang hanya perlu mengembangkan untuk Android, dan aplikasi mereka dapat dijalankan pada banyak perangkat yang berbeda, sepanjang perangkat tersebut menggunakan Android. Pada dunia smartphone, aplikasi adalah bagian rantai sukses yang paling penting. Oleh karena itu pabrikan hardware melihat Android sebagai harapan terbaiknya untuk menantang serangan hebat iPhone, yang sudah memiliki dasar aplikasi yang besar.

Android telah mengalami sejumlah update sejak pertama kali diluncurkan pada tahun 2007 sampai yang diluncurkan terakhir tahun 2013. Tabel 2 berikut menunjukkan berbagai macam versi Android dan kode namanya. Kode nama versi Android menggunakan nama-nama kue agar mudah diingat.

Android bersifat open source dan bebas tersedia bagi pabrikan untuk penyesuaian, sehingga tidak ada konfigurasi hardware dan software yang tetap. Akan tetapi Android sendiri mendukung fitur-fitur berikut:

1. Storage, menggunakan SQLite sebuah database relasional yang ringan.
2. Connectivity, mendukung GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wifi, LTE dan WiMAX
3. Messaging, mendukung SMS dan MMS.
4. Web Browser, berdasarkan open source WebKit.

5. Media Support, mendukung berbagai tipe media, MPEG, MP4, 3GP, JPEG, PNG dan lain-lain.
6. Hardware support, accelerometer sensor, camera, digital compass, proximity sensor, dan GPS.
7. Multi touch, multi tasking, mendukung flash dan tethering, shairng koneksi internet.

Secara umum sistem operasi Android terbagi ke dalam 4 lapisan, yaitu kernel linux, libraries dan android runtime, application framework dan application. Kernel linux merupakan dasar dari sistem operasi Android. Lapisan di atasnya adalah lapisan libraries. Pada lapisan ini berisi semua kode yang menyediakan fitur utama sistem operasi Android. Sebagai contoh, library SQLite menyediakan dukungan database sehingga sebuah aplikasi dapat menggunakannya untuk penyimpanan data. Library WebKit menyediakan fungsionalitas web browsing. Pada lapisan yang sama terdapat Android Runtime, yang menyediakan seperangkat library inti yang memungkinkan pengembang untuk menulis aplikasi android menggunakan bahasa pemrograman java. Android runtime juga termasuk Dalvik virtual machine, yang memungkinkan setiap aplikasi android menjalankan prosesnya sendiri, dengan instance Dalvik virtual machine-nya (aplikasi android dikompilasi ke dalam dalvik executable). Dalvik adalah virtual machine yang khusus digunakan pada Android dan optimal untuk perangkat mobile dengan baterai dengan memori dan CPU yang terbatas. Lapisan di atasnya adalah application framework, yang menampilkan berbagai macam kemampuan sistem operasi Android kepada pengembang aplikasi sehingga mereka dapat menggunakan pada aplikasi mereka. Lapisan yang paling atas adalah aplikasi, pada lapisan ini kita bisa menemukan aplikasi yang dikapalkan bersama dengan perangkat Android, seperti contacts, browser serta aplikasi-aplikasi yang didownload dari Play Store.

Algoritma Hidden Markov Model adalah algoritma yang paling efisien dan paling akurat dalam pengenalan suara. Hidden Markov Model terdiri dari dua bagian, yaitu proses Markov dan hidden model. Proses Markov adalah proses stokastik dengan properti Markov. Properti Markov mengatakan bahwa probabilitas kondisional dari kenampakan kondisi proses yang akan datang, ditentukan kondisi masa lampau dan saat ini, hanya tergantung pada kondisi saat ini dan bukan kondisi masa lampau. Pengenalan suara berbasis HMM menggunakan konsep bahwa ucapan terdiri dari urutan suara dasar. Kata apapun di dalam kamus terdiri dari urutan suara dasar. Masing-masing suara dasar ini mempunyai model statistik. Oleh karena itu sebuah kata dapat diekspresikan sebagai urutan model statistik. Suara dijadikan sampel oleh

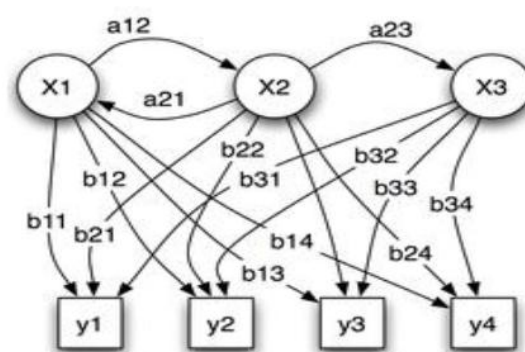
sistem pengenalan suara untuk membentuk sebuah urutan vektor fitur suara (parameter numerik). Urutan vektor fitur suara ini adalah urutan observasi. Kemudian pengenal menentukan secara probabilistik model yang mana yang kemungkinan besar menghasilkan vektor suara. Mengidentifikasi urutan model menghasilkan identifikasi kata yang diucapkan (Chandra, 2007).

HMM merupakan sebuah model statistik dimana suatu sistem yang dimodelkan diasumsikan sebagai markov proses dengan kondisi yang tidak terobservasi (Prasetyo, 2010). Suatu HMM dapat dianggap sebagai jaringan Bayesian dinamis yang sederhana (simplest dynamic bayesian network). Dalam markov biasa, smetiap keadaan dapat terlihat langsung oleh pengamat. Oleh karena itu kemungkinan dari transisi antar kondisi menjadi satu-satunya parameter teramati. Dalam HMM keadaan tidak terlihat secara langsung. Tetapi output yang bergantung terhadap keadaan itu terlihat.

Setiap kondisi memiliki distribusi kemungkinan di setiap output yang mungkin. Oleh karena itu urutan langkah yang dibuat oleh HMM memberikan suatu informasi tentang urutan dari keadaan. Perlu dipahami bahwa sifat hidden ‘tersembunyi’ menunjuk pada kondisi langkah yang dilewati model, bukan kepada parameter dari model tersebut. Walaupun parameter model diketahui model tersebut tetap tersembunyi.

Hidden markov model dapat digunakan untuk aplikasi di bidang temporal pattern recognition atau pengenalan pola temporal seperti pengenalan suara, tulisan, gesture, bioinformatika, kompresi kalimat, computer vision, ekonomi, finansial, dan pengenalan not balok. HMM adalah variasi dari finite state machine yang memiliki kondisi tersembunyi Q , suatu nilai output O (observasi), kemungkinan transisi A , kemungkinan output B , sebuah kondisi awal q_0 . Kondisi saat ini tidak terobservasi. Tetapi setiap keadaan menghasilkan output kemungkinan B . Biasanya Q dan O dimengerti, jadi HMM disebut triple (A, B, q_0) (Prasetyo M. E., 2010).

1. Himpunan observed state: $O = \{o_1, o_2, \dots, o_n\}$
2. Himpunan hidden state: $Q = \{q_1, q_2, \dots, q_n\}$
3. Probabilitas transisi: $A = \{a_{01}, a_{02}, \dots, a_{n1}, \dots, a_{nm}\}$; a_{ij} adalah probabilitas untuk berpindah dari state i ke j .
4. Probabilitas emisi atau observation likelihood: $B = \{b_i(o_t)\}$, merupakan probabilitas observasi O_t dibangkitkan oleh kondisi i .
5. State awal dan akhir: q_0, q_{end} , yang tidak terkait dengan observasi.



Sumber: (Prasetyo, 2010)

Gambar 1. Hidden Markov Model

Keterangan:

- X = kondisi
- Y = observasi yang mungkin
- A = kemungkinan keadaan transisi
- B = kemungkinan output

HMM mampu menyelesaikan tiga permasalahan khusus, yaitu evaluasi, inferensi dan learning. Evaluasi menggunakan HMM dengan menghitung probabilitas dari urutan nilai observasi yang diberikan oleh HMM. Inferensi menggunakan HMM dilakukan dengan menarik kesimpulan berdasarkan asumsi yang diperoleh dari nilai probabilitas observasi yang didapat sebelumnya pada operasi evaluasi. Learning menggunakan HMM dilakukan dalam dua tahapan yaitu, menghitung nilai probabilitas forward dan backward untuk setiap statement dan menentukan frekuensi dari pasangan transisi emisi dan membaginya dengan nilai probabilitas semua observasi.

Java Virtual Machine

Inovasi bahasa komputer didorong oleh dua faktor, yaitu peningkatan seni pemrograman dan perubahan lingkungan komputasi (Schildt, 2005). Java tidak terkecuali. Dibangun berdasarkan warisan kaya C dan C++, java menambahkan perbaikan dan fitur yang merefleksikan kondisi seni pemrograman saat ini. Merespon peningkatan lingkungan online, Java menawarkan fitur yang menyederhanakan pemrograman untuk arsitektur yang terdistribusi.

Java disusun oleh James Gosling, Patrick Naughton, Chris Warth, Ed Frank dan Mike Sheridan pada perusahaan Sun Microsystem pada tahun 1991. Bahasa ini awalnya namanya “Oak” tetapi dinamai kembali “Java” pada tahun 1995. Sesuatu yang mengejutkan dorongan awal Java bukan Internet, melainkan dorongan utamanya adalah kebutuhan bahasa yang multi platform yang dapat digunakan untuk menciptakan software untuk ditanamkan pada berbagai macam perangkat elektronik, seperti pemanggang, oven microwave, remote control dan lain-lain. Seperti yang dapat ditebak, banyak CPU berbeda yang digunakan

sebagai pengendali. Permasalahannya adalah sebagian besar bahasa pemrograman dikompilasi untuk target tertentu, misal C++.

Meskipun mungkin untuk mengkompilasi sebuah program C++ untuk hampir semua jenis CPU, untuk melakukannya membutuhkan kompiler C++ untuk CPU tersebut. Permasalahannya, kompiler tersebut mahal dan memakan waktu untuk membuatnya. Dalam usaha untuk menemukan solusi yang lebih baik, Gosling dan teman-temannya bekerja pada perangkat portabel, lintas platform yang dapat menghasilkan kode yang berjalan pada berbagai macam CPU pada lingkungan yang berbeda. Usaha inilah yang menuntun penciptaan Java. Dorongan yang kedua adalah World Wide Web. Java menjadi garis terdepan pada pemrograman berbasis Web karena Web menginginkan program yang portabel.

Java secara langsung berhubungan dengan C dan C++. Java menurunkan sintaknya dari C. Model objeknya diadaptasi dari C++. Hubungan Java dengan C dan C++ penting untuk beberapa alasan. Pertama, banyak programmer yang familiar dengan sintaks C dan C++. Ini membuat mudah bagi programmer C/C++ untuk belajar Java, begitu juga sebaliknya programmer Java yang belajar C/C++. Alasan yang kedua, Java bukan sebuah penemuan kembali, melainkan sebuah perbaikan dari paradigma pemrograman yang sudah sukses. Pemrograman modern dimulai dari C kemudian C++ dan sekarang Java. Dengan menurunkan dan membangun berdasarkan warisan kaya, Java menyediakan lingkungan pemrograman yang konsisten dan powerful yang menjadikan terbaik dan menambahkan fitur baru pada lingkungan online. Karena kemiripan mereka, C, C++ dan Java menetapkan sebuah framework yang umum dan konseptual untuk programmer profesional. Programmer tidak menghadapi celah yang besar ketika berpindah dari satu bahasa ke bahasa lain.

Java merupakan salah satu bahasa pemrograman yang berorientasi objek (Object Oriented Programming). Java mempunyai tiga konsep dasar dalam pemrograman berorientasi objek, yaitu encapsulation, polymorphism dan inheritance. Encapsulation adalah mekanisme pemrograman yang membungkus kode dan data bersama dan menjaganya dari gangguan luar. Data dan kode dibungkus ke dalam sebuah *blackbox* yang disebut dengan objek. Objek ada yang bersifat private dan publik. Objek yang bersifat private hanya bisa diakses di seluruh bagian di dalam objek. Objek yang bersifat publik dapat diakses oleh seluruh bagian dalam satu *package*. Satuan dasar encapsulasi Java adalah *class*. Kelas merupakan bentuk cetak biru sebuah objek, satu kelas bisa diturunkan menjadi beberapa objek. Dengan kata lain, objek adalah bentuk instansiasi dari sebuah kelas. Kelas tidak bisa diakses sebelum dibendakan atau diinstansiasi menjadi objek.

Konsep yang kedua dari object oriented Java adalah *polymorphism*, yang artinya banyak bentuk. Objek hasil instansiasi dari sebuah kelas memiliki karakteristik atau atribut dan metode atau tingkah laku. Atribut merupakan ciri-ciri yang menempel pada sebuah objek, sedangkan metode adalah tingkah laku yang bisa dilakukan oleh sebuah objek. Namun, kadang kala objek yang diturunkan dari kelas yang sama memiliki metode yang berbeda. Misal kelas hewan diinstansiasi menjadi objek hewan harimau dan ular. Harimau dan ular memiliki metode yang sama yaitu berjalan, namun cara berjalan harimau berbeda dengan cara berjalan ular.

Konsep yang ketiga adalah *inheritance* atau pewarisan sifat. Sebuah objek yang diturunkan dari sebuah kelas memiliki atribut dan metode yang sama dengan kelas yang menurunkannya. Misal sebuah kelas mobil yang memiliki atribut merk, nomor, jumlah roda dan metode maju, mundur, belok kanan dan belok kiri. Sedan adalah bentuk instansiasi dari kelas mobil. Maka berdasarkan konsep dasar object oriented programming (oop) maka sedan juga memiliki atribut dan metode yang sama dengan kelas mobil.

GOOGLE SPEECH API

Google speech API atau Google Voice search diluncurkan pada tahun 2008 di Amerika Serikat untuk beberapa tipe smartphone. Google speech API adalah sebuah framework yang dikembangkan oleh Google untuk mengenali suara, mengubahnya menjadi string (teks) dan memasukkannya ke dalam halaman pencarian Google sehingga akan tampil hasil pencarian berdasarkan input suara. Pengenalan suara dilakukan pada server Google menggunakan algoritma Hidden Markov Model (HMM). Dengan kata lain input suara yang diterima oleh perangkat Android (smartphone) akan dikirimkan ke server Google, yang selanjutnya server Google melakukan pengenalan dan mengubahnya menjadi teks menggunakan algoritma HMM. Hasil konversi suara menjadi teks kemudian dimasukkan dalam halaman pencarian Google kemudian server Google akan mengirimkan hasil pencariannya tersebut ke perangkat Android (Reddy & Mahender, 2013).

Antarmuka web pencarian menyediakan semua jenis informasi dan layanan di Internet, tetapi pengguna harus mencari melalui daftar hasil untuk hasil yang diinginkan. Ini dapat menjadi membebani pada perangkat yang memiliki ukuran layar yang relatif kecil. Pengembangan pencarian berdasarkan suara memiliki keuntungan (Iizuka, Tsujino, Oguri, & Furukawa, 2012):

1. Memungkinkan memanggil aplikasi tertentu pada fungsi terminal atau layanan web.
2. Menggunakan teknologi pemrosesan bahasa dan rumus penggolongan untuk menentukan

secara otomatis kategori mana yang termasuk ucapan pengguna dan menyarankan aplikasi yang tepat.

3. Menggabungkan layar yang menyediakan akses yang mudah ke aplikasi lain pada kategori yang lain yang berhubungan dengan ucapan

Speech recognition system dapat dibagi menjadi beberapa blok, yaitu feature extraction, acoustic model ndatabase yang dibangun berdasarkan data training, dictionary, model bahasa dan algoritma pengenalan suara. Sinyal suara analog dianalisa bahkan dalam interval. Periode ini biasanya 20 ms karena sinyal ini dianggap statis. Ekstraksi fitur suara melibatkan pembentukan vektor diskrit spasi sama dari karakteristik suara. Vector fitur dari database pelatihan digunakan untuk memperkirakan parameter model akustik. Model akustik menjelaskan properti elemen dasar yang dapat dikenali. Elemen dasar ini dapat berupa fonem untuk suara yang terus menerus dan kata untuk pengenalan kata terisolasi.

Dictionary digunakan untuk menghubungkan model akustik dengan vocabulari kata. Model bahasa mengurangi jumlah kombinasi kata yang disetujui berdasarkan aturan bahasa dan informasi statistik dari teks yang berbeda. *Speech recognition system* berdasarkan hidden markov model saat ini digunakan luas pada teknologi modern. Sistem ini menggunakan fonem atau kata untuk pemodelan. Model output merupakan fungsi kondisi propabilitas tersembunyi dan tidak dapat ditentukan secara spesifik. *Speech recognition system* pada umumnya mengasumsikan bahwa sinyal suara adalah realisasi dari beberapa pesan yang tersandikan sebagai urutan satu atau lebih simbol (Reddy & Mahender, 2013).

Android Application Development

Untuk membuat aplikasi berbasis Android, kita perlu mendownload dan menginstal beberapa software yang diperlukan, yaitu Java Virtual Machine, Android SDK dan Eclipse. Ketiga software tersebut bersifat open source dan kita bisa mendapatkannya dengan gratis, tinggal mendownloadnya dari Internet. Java bisa didownload pada <https://java.com/en/download/index.jsp>. Setelah didownload, klik dua kali file hasil download untuk menginstalnya dan pilih install, next sampai selesai.

Langkah yang kedua mendownload Android SDK pada link <https://developer.android.com/sdk/index.html>. Ada beberapa pilihan Android SDK yang bisa didownload, yaitu Android SDK untuk Windows baik yang 32 bit maupun 64 bit, untuk Linux 32 bit dan 64 bit, untuk Mac 32 bit dan 64 bit serta kita bisa mendownload adt-bundle baik untuk Windows, Linux maupun Mac. Adt-bundle adalah beberapa software yang sudah dikonfigurasi secara

otomatis, sehingga kita tidak perlu mendownload update Android SDK lagi. Namun, android sdk pada software ini hanya android sdk terbaru, yaitu android kitkat. Klik dua kali file hasil download untuk menginstal Android SDK.

Pada saat menginstal Android SDK, akan diperiksa apakah Java sudah terinstal atau belum. Karena pemrograman Android menggunakan bahasa pemrograman java. Jika belum kita harus menginstal Java terlebih dahulu, jika sudah kita bisa melanjutkan untuk menginstal Android SDK.

Setelah Java dan Android SDK kita membutuhkan satu software lagi yaitu Eclipse, tempat kita mebuliskan programnya. Eclipse dapat didownload pada link <http://www.eclipse.org/downloads/>. Setelah didownload klik dua kali file eclipse.exe untuk menjalankan Eclipse. Selanjutnya kita perlu mengkonfigurasi Eclipse agar bisa digunakan untuk membuat aplikasi berbasis Android. Kita tambahkan plugin Android SDK untuk Eclipse dengan menambahkan repositorinya.

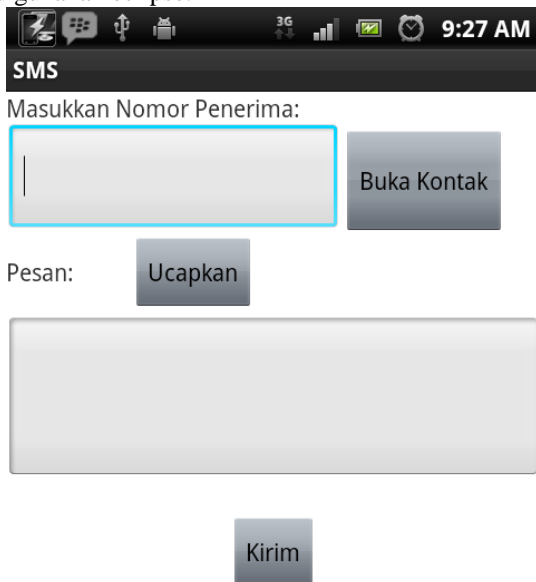
Setelah selesai maka Eclipse kita sudah terinstal plugin Android SDK. Kita bisa membuat program berbasis Android menggunakan Eclipse. Selanjutnya kita perlu mengupdate Android SDK, untuk mendapatkan Android SDK terbaru. Setelah itu kita buat device virtual Android, untuk menjalankan aplikasi yang kita buat pada Eclipse. Kita juga bisa menjalankan langsung aplikasi yang kita buat pada Smartphone android, tetapi cara ini terlalu beresiko karena kita harus mengubah setting di handphone maupun di komputer tablet agar bisa menginstal software yang didapatkan bukan dari play store. Apabila aplikasi kita mengandung malware, bisa merusak perangkat. Untuk lebih aman kita gunakan perangkat virtual yang sudah disediakan oleh Android SDK walaupun agak lambat untuk dijalankan di komputer tertentu, khususnya komputer yang layar kecil.

Setelah semuanya terinstal dan dikonfigurasi, maka Android Development Tool siap digunakan untuk membuat aplikasi berbasis Android. Aplikasi berbasis Android terdiri dari dua file utama yaitu file xml dan file java. File xml berisi layout dari aplikasi kita sedangkan file java berisi logika program kita. Keduanya menjadi satu kesatuan yang tidak bisa dipisahkan.

Hasil Dan Pembahasan

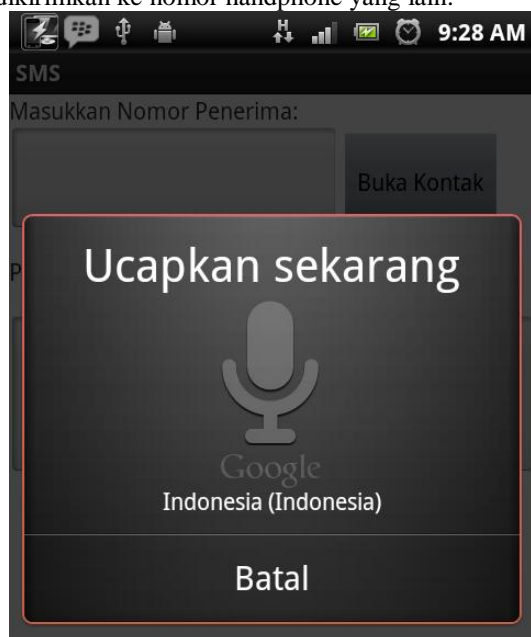
Penelitian ini adalah penelitian eksperimen, dimana penelitian melibatkan investigasi hubungan sebab akibat menggunakan tes yang dikendalikan oleh peneliti (Dawson, 2009). Aplikasi dibuat menggunakan bahasa pemrograman java dan xml. Java digunakan untuk menangani logika program dan xml digunakan untuk membuat layout aplikasi.

Antarmuka lingkungan pengembangan yang digunakan eclipse.



Gambar 2. Desain antar muka aplikasi speech to text Android

Google Speech API dipanggil menggunakan framework ACTION_RECOGNIZE_SPEECH. Framework ini diimplementasikan menggunakan metode onActivityResult(). Selanjutnya library ini akan menangkap ucapan melalui mic pada *smartphone*. Ucapan yang ditangkap akan dikirimkan ke server Google untuk diterjemahkan ke dalam teks. Google menerjemahkan ucapan ke dalam teks menggunakan algoritma Hidden Markov Model (HMM). Hasil terjemahan dalam bentuk teks dikirimkan kembali ke *smartphone*, dalam hal ini dalam field tulis sms dan siap untuk dikirimkan ke nomor handphone yang lain.

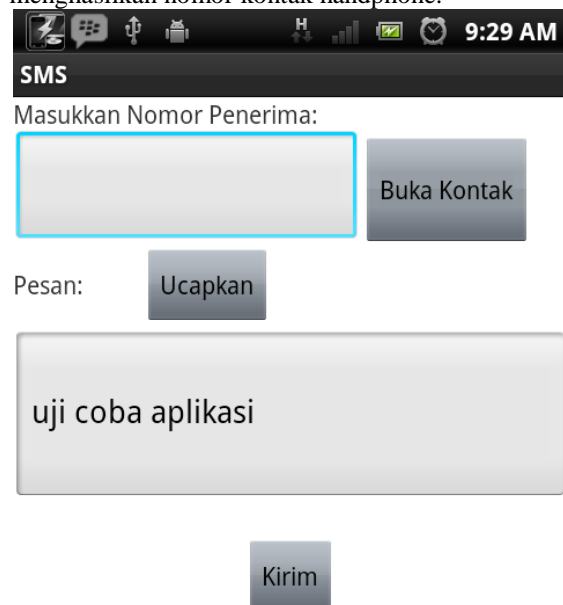


Gambar 3. Tampilan proses mengkonversi suara menjadi teks

Google mampu menerjemahkan kata dalam ratusan bahasa termasuk Bahasa Indonesia. Agar bisa menerjemahkan kata dalam Bahasa Indonesia dengan baik sebelumnya *smartphone* diatur ke dalam Bahasa Indonesia. Pengaturan dalam Bahasa Indonesia dilakukan oleh library EXTRA_LANGUAGE_MODEL.

Pada aplikasi ini terdapat tiga bagian utama yaitu membaca kontak telepon, menerjemahkan suara menjadi teks dan mengirim SMS. Pada file AndroidManifest.xml perlu ditambahkan dua permission, yaitu android.permission.READ_CONTACTS dan android.permission.SEND_SMS. Notifikasi permission ini akan muncul ketika kita akan menginstal aplikasi ini pada handphone Android.

Pada metode onActivityResult() terdapat dua case, yaitu case RESULT_SPEECH dan case CONTACT_PICKER_RESULT. Case RESULT_SPEECH akan menghasilkan terjemahan ucapan dalam bentuk teks. Sedangkan case CONTACT_PICKER_RESULT akan menghasilkan nomor kontak handphone.



Gambar 4. Hasil konversi suara menjadi teks yang diterapkan dalam aplikasi sms

Pengujian dilakukan pada handphone Sony Xperia Ray dengan sistem operasi Android Gingerbread atau Android versi 2.3.4. Hasil pengujian menunjukkan bahwa aplikasi ini mampu mengenali suara yang diucapkan dan berhasil dikirimkan melalui sms. Karena penterjemahan ucapan dilakukan oleh server Google maka handphone harus terkoneksi Internet. Pengujian dilakukan menggunakan koneksi internet dengan kecepatan 2Mbps. Pada koneksi ini penterjemahan dapat dilakukan dengan cepat. Selanjutnya pengujian dilakukan menggunakan koneksi provider Internet yang lain dengan kecepatan kurang dari 2 Mbps. Penterjemahan dilakukan lebih

lambat. Aplikasi ini berjalan dengan baik pada lingkungan dimana kecepatan Internet di atas 2 Mbps. Kecepatan dibawahnya menyebabkan aplikasi ini kurang berjalan maksimal.

Gambar 5. Tabel pengujian black box

No	Fitur	Validasi
1	Membaca kontak handphone	Sukses
2	Memanggil Framework Google Speech API	Sukses
3	Memasukkan teks hasil konversi suara pada bidang teks	Sukses
4	Mengirim sms	Sukses

KESIMPULAN Dan SARAN

Aplikasi ini mampu menggantikan tugas tulis sms, sehingga kita tidak perlu menulis sms pada layar ketik melainkan cukup dengan mengucapkannya. Penterjemahan ucapan menjadi teks dilakukan oleh server Google. Aplikasi mengirimkan ucapan ke server Google melalui koneksi Internet, dan selanjutnya server Google mengirimkan kembali hasil penterjemahan ke smartphone. Kelemahan aplikasi ini adalah membutuhkan koneksi Internet, aplikasi tidak berjalan maksimal pada koneksi Internet yang lambat. Penelitian selanjutnya membuat mesin penterjemah ucapan menjadi teks pada handphone, sehingga penterjemahan tidak perlu dilakukan oleh server.

DAFTAR PUSTAKA

Ballinger, B., Allauzen, C., Gruenstein, A., & Schalkwyk, J. (2010). On-Demand Language Model Interpolation for Mobile Speech Input. *Interspeech*.

Chandra, D. (2007). *Speech Recognition Co-Processor*. North Carolina: Dissertation Doctor of Philosophy North Carolina State University.

Dawson, C. W. (2009). *Projects In Computing And Information Systems A Students Guide*. Essex: Pearson Education Limited.

Gartner. (2013, Agustus 14). *Gartner Says Smartphone Sales Grew 46.5 Percent in Second Quarter of 2013 and Exceeded Feature Phone Sales for First Time*. Dipetik

Desember 6, 2013, dari Gartner: <http://www.gartner.com/newsroom/id/2573415>

Iizuka, S., Tsujino, K., Oguri, S., & Furukawa, H. (2012). Speech Recognition Technology and Applications for Improving Terminal Functionality and Service Usability. *NTT DOCOMO Technical Journal Vol. 13 No. 4*, 79-84.

Kumar, C. S., & Rao, P. M. (2011). Design of Automatic Speaker Recognition System Using MFCC, Vector Quantization and LBG Algorithm. *International Journal on Computer Science and Engineering Vol. 3 No. 8*, 2942-2954.

Lee, W. M. (2012). *Beginning Android 4 Application Development*. Indianapolis, USA: John Wiley & Sons, Inc.

Meisel, W. (2010). Life on-the-GO: The Role of Speech Technology in Mobile Applications. *Advances in Speech Recognition: Mobile Environments*, DOI 10.1007/978-1-4419-5951-5_1 (ISBN: 978-1-4419-5950-8 (Print) 978-1-4419-5951-5 (Online)), 3-18.

Patel, P. N., Patel, J. K., & Virparia, P. V. (2013). Generating Select Query form Spoken Words on Android Smartphone. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) Vol. 2 No.3*, 91-94.

Petersen, K., Wohlin, C., & Baca, D. (2009). The Waterfall Model in Large-Scale Development. *Lecturer Notes in Bussiness Information Processing Vol. 32*, 386-400.

Prasetyo, M. E. (2010). *Index of /~rinaldi.munir/Probstat/2010-2011/Makalah2010*. Dipetik Desember 10, 2013, dari Sekolah Teknik Elektro dan Informatika ITB: <http://informatika.stei.itb.ac.id/~rinaldi.munir/Probstat/2010-2011/Makalah2010/MakalahProbstat2010-025.pdf>

Reddy, R. B., & Mahender, E. (2013). Speech To Text Conversion Using Android Paltform. *Internaitonal Journal of Engineering Research and Applicaiton (IJERA) Vol. 3 No.1*, 253-258.

Schalkwyk, J., Beeferman, D., Beaufays, F., Byrne, B., Chelba, C., Cohen, M., et al. (2010).

- Google Search By Voice: A Case Study. *Springer* .
- Schildt, H. (2005). *Java A Beginner's Guide Third Edition*. New York: McGraw-Hill.
- Schuster, M. (2010). Speech Recognition for Mobile Device at Google. *Springer* , 8-10.
- Singh, Y. K. (2006). *Fundamental of Research Methodology and Statistic*. New Delhi: New Age International (P), Ltd.