

MAKALAH SEMINAR TUGAS AKHIR
PERANCANGAN APLIKASI PENDAMPING PEMBELAJARAN BAHASA INGGRIS
MENGUNAKAN MICROSOFT SPEECH API

Karnadi*, Agung Budi Prasetijo**, R. Rizal Isnanto**

Abstrak --- Penguasaan bahasa Inggris sebagai bahasa asing yang dominan dipakai dalam pergaulan internasional merupakan salah satu akses untuk meraih kesuksesan di segala bidang. Salah satu tujuan kurikulum nasional pendidikan bahasa Inggris adalah mengembangkan kemampuan berkomunikasi yang meliputi menyimak (*listening*), berbicara (*speaking*), membaca (*reading*), dan menulis (*writing*) dalam bahasa tersebut, dalam bentuk lisan dan tulis.

Teknologi Pengenalan Suara Otomatis (*Automatic Speech Recognition*) pada *Microsoft Speech API* dapat digunakan untuk membuat aplikasi pendamping untuk mempraktikkan keempat keterampilan bahasa tersebut. Aplikasi dibangun berbasis GUI (*Graphical User Interface*) agar menarik dan ramah pengguna (*user friendly*). Aplikasi ini dapat menyimpan teks sehingga dapat diubah dan digunakan berulang-ulang.

Dalam tugas akhir ini, implementasi GUI menggunakan bahasa pemrograman visual Delphi 7.0 pada sistem operasi *Windows XP* yang dihubungkan dengan basis-data *Firebird 1.5* sehingga berkas dapat ditambah, diperbarui dan dipakai secara bersamaan dalam satu jaringan. Aplikasi ini dapat digunakan secara interaktif untuk melatih empat dasar keterampilan berbahasa Inggris ditambah dengan aktivitas latihan untuk mempertajam pemahaman.

Kata-kunci : komunikasi, pengenalan suara otomatis, *microsoft speech API*, GUI, basis-data.

I. PENDAHULUAN

1.1 Latar Belakang

Bahasa memiliki peran sentral dalam perkembangan intelektual, sosial, dan emosional siswa dan merupakan kunci penentu menuju keberhasilan dalam mempelajari semua bidang studi. Penguasaan bahasa asing, terutama yang dominan dalam pergaulan internasional, merupakan salah satu akses untuk meraih keberhasilan dalam berbagai bidang^[18]. Kemampuan berkomunikasi meliputi mendengar atau menyimak (*listening*), berbicara (*speaking*), membaca (*reading*), dan menulis (*writing*)^[15]. Kenyataan di lapangan menunjukkan bahwa sebagian besar tamatan sekolah menengah tidak dapat berkomunikasi dalam bahasa Inggris walaupun mereka telah belajar bahasa Inggris

selama 6 (enam) tahun di SLTP dan SMU. Hal tersebut terjadi karena pengajaran bahasa Inggris di sekolah lebih banyak terfokus pada pengajaran tata-bahasa dan kurang memberikan kesempatan pada siswa untuk berlatih berbicara dan mendengarkan dalam bahasa Inggris.

Menurut Dale (1969)^[11], pada umumnya siswa hanya 10% mengingat dari apa yang mereka baca, 20% dari apa yang mereka dengar, 30% dari apa yang mereka lihat, 50% dari apa yang mereka dengar dan lihat, 70% dari apa yang mereka katakan dan tulis, dan 90% dari apa yang mereka katakan seperti yang mereka lakukan. Untuk itu diperlukan media pendamping untuk melatih semua keterampilan komunikasi tersebut sehingga dapat memudahkan guru dalam memberikan pengajaran dan memudahkan siswa dalam menangkap isi pembelajaran sehingga kualitas pembelajaran menjadi lebih baik. Tujuan tersebut dapat dipenuhi dengan bantuan teknologi pengenalan suara otomatis (*automatic speech recognizer*). Menurut Supriyono(2004)^[13], teknologi tersebut memungkinkan komputer untuk mengenali suara manusia sesuai dengan bahasa yang digunakan sehingga komunikasi antara manusia dan komputer dapat dilakukan se-natural mungkin seperti komunikasi antara manusia dan manusia. Teknologi pengenalan suara otomatis tersebut belum dapat digunakan secara langsung dalam aplikasi, karena masih membutuhkan implementasi pada aras sistem operasi. Oleh Microsoft teknologi pengenalan suara otomatis digabungkan dengan teknologi sintesis suara (*speech synthesis*), yaitu teknologi yang dapat mensintesis suara manusia menjadi suatu API (*Application Programming Interface*) yang disebut *Microsoft Speech API*.

Untuk memudahkan penggunaannya maka aplikasi dibuat menggunakan GUI (*Graphical User Interface*). Dengan GUI semua pesan dari komputer ke pengguna dilakukan dengan menggunakan visualisasi gambar atau tulisan yang mudah dimengerti oleh manusia sebagai pengguna. Agar dapat digunakan secara berulang-ulang sebagai latihan maka berkas yang digunakan dapat disimpan dalam sebuah basis-data.

1.2 Tujuan

Merancang aplikasi pendamping pembelajaran bahasa Inggris berbasis GUI

* Mahasiswa Jurusan Teknik Elektro UNDIP

** Staf Pengajar Jurusan Teknik Elektro UNDIP

(*Graphical User Interface*) pada sistem operasi *Windows XP* dengan menggunakan *Microsoft Speech API* dengan tujuan untuk mempermudah pembelajar menguasai keterampilan komunikasi menggunakan bahasa Inggris.

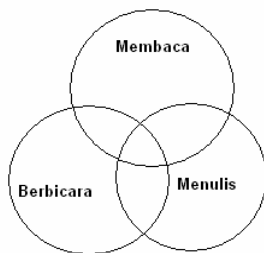
1.3 Batasan Masalah

1. Perancangan aplikasi menggunakan metode perancangan berorientasi objek **Coad-Yourdon**.
2. Pemodelan analisis, perancangan, dan pengujian menggunakan kaskas pemodelan **UML versi 1.5**.
3. Implementasi model rancangan dipetakan menggunakan bahasa pemrograman **Delphi 7.0** dan basis-data **Firebird 1.5.2** yang dijalankan pada sistem operasi **Windows XP**.
4. Pengujian menggunakan konsep pengujian berorientasi objek, hanya diterapkan pada strategi **pengujian validasi** skenario dengan menggunakan **metode black box** pada setiap utilitas *use case* yang diidentifikasi pada tahap analisis kebutuhan.
5. Tidak membahas proses pengenalan suara (pengenalan pola) dan proses sintesis suara.
6. Tidak dibahas algoritma kecerdasan buatan yang digunakan oleh **Microsoft Speech API 1.5**.
7. Tidak dibahas struktur dan jenis basis-data yang digunakan oleh **Microsoft Speech API 1.5**.
8. Pengguna aplikasi ini ditujukan bagi siswa sekolah menengah dan yang sederajat.

II. TINJAUAN TEORI

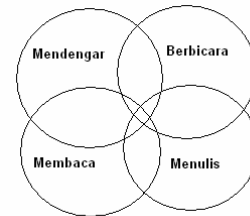
2.1 Perspektif Keterampilan Berbahasa

Salah satu lingkup keterampilan berbahasa, yaitu mendengarkan, berbicara, membaca, dan menulis. Keterampilan mendengarkan, berbicara, membaca dan menulis tidak diposisikan secara linier, berjajar, melainkan terpadu sebagai struktur yang mewarnai rancangan proses belajar dan mengajar. Kern(2000:132) merepresentasikan gagasannya dalam tiga lingkaran berbicara (*talking*), membaca (*reading*), dan menulis (*writing*) pada seperti Gambar 2.1.



Gambar 2.1 Diagram Kern

Dalam konteks pengajaran bahasa asing, seringkali kegiatan mendengarkan dilakukan terpisah dari kegiatan berbicara karena kegiatan tersebut difokuskan kepada latihan mengucapkan atau menirukan bunyi-bunyi, kata dan sebagainya. Oleh karenanya model Gambar 2.1 di atas dapat dimodifikasi menjadi empat lingkaran seperti pada Gambar 2.2.



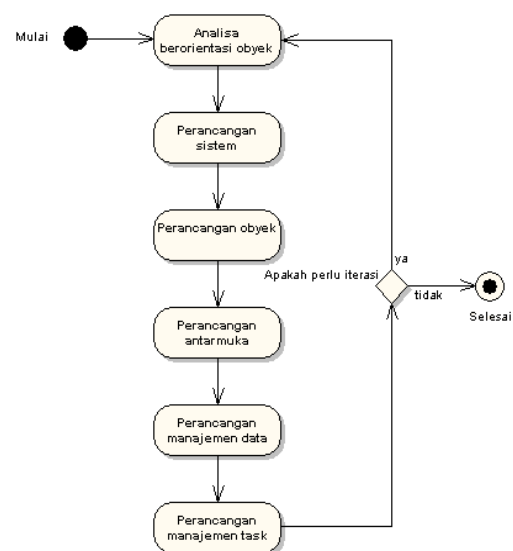
Gambar 2.2 Modifikasi Diagram Kern

2.2 Metodologi Perancangan Berorientasi Obyek Coad-Yourdon

Metode ini memfokuskan perancangan pada empat komponen yaitu:

1. *Problem domain component*, yang terbagi menjadi dua yaitu:
 - a. Perancangan sistem (arsitektur yang membangun sistem)
 - b. Perancangan obyek (arsitektur struktur logika dan algoritma pemrograman)
2. *Human interaction component* adalah perancangan antarmuka manusia dengan sistem
3. *Data management component* adalah perancangan manajemen data
4. *Task management component* adalah perancangan manajemen task

Dari deskripsi di atas dapat ditunjukkan tahap-tahap perancangan metode Coad-Yourdon seperti pada Gambar 2.3.



Gambar 2.3 Tahapan metodologi perancangan Coad-Yourdon

2.3 Teknologi-Teknologi yang Mendasari Aplikasi

Aplikasi yang akan dibuat ini bukanlah aplikasi yang berdiri sendiri. Aplikasi ini tersusun atas teknologi-teknologi lainnya yang memungkinkan aplikasi ini berjalan sebagaimana mestinya. Teknologi-teknologi yang mendasari Aplikasi ini adalah :

1. Pengolahan Bahasa Alami (*Natural Language Processing*)
2. Pengenalan Ucapan Automatis (*Automatic Speech Recognition*)
3. Sintesis Ucapan
4. Windows OLE
5. Windows API

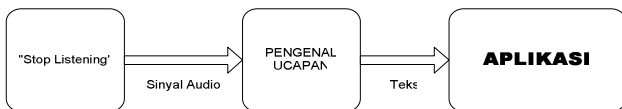
2.3.1 Pengolahan Bahasa Alami

Teknologi Pengolahan Bahasa Alami dikembangkan dengan tujuan supaya diperoleh suatu teknologi yang dapat menganalisis, mengerti dan menghasilkan ucapan yang digunakan oleh manusia sealam mungkin. Sehingga diharapkan manusia dapat berkomunikasi dengan komputer seperti manusia berkomunikasi dengan manusia.

2.3.2 Pengenalan Ucapan Automatis

Pengenalan Ucapan Automatis merupakan teknologi yang dikembangkan dari teknologi pengolahan bahasa alami. Pengenalan Ucapan Automatis adalah teknologi yang dapat mengenali ucapan manusia dalam bahasa tertentu dan mengubah ucapan yang dikenalnya kedalam bentuk teks^[19,20].

Proses pengenalan ucapan ini sangat bergantung pada bahasa yang digunakan, karena setiap bahasa memiliki cara pengucapan yang berbeda. Sehingga teknologi pengenal ucapan otomatis ini bersifat bergantung bahasa^[20]. Skema prinsip kerja pengenal ucapan otomatis dapat dilihat pada Gambar 2.4



Gambar 2.4 Diagram prinsip kerja Pengenal Ucapan Automatis^[13]

2.3.3 Sintesis Ucapan

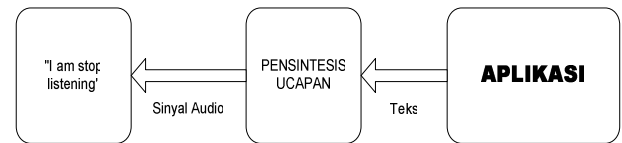
Sintesis Ucapan juga merupakan pengembangan dari teknologi pengolahan bahasa alami. Sintesis Ucapan adalah teknologi yang mampu membuat sintesis ucapan dalam bahasa

tertentu berdasarkan teks yang diberikan kepadanya^[19,20].

Skema prinsip kerja Pensintesis Ucapan dapat dilihat pada

Gambar 2.5. Secara umum Sintesis Ucapan terdiri dari dua subsistem, yaitu :

1. Teks-ke-fonem, berfungsi untuk mengubah masukan (berupa teks) dalam bahasa tertentu menjadi rangkaian kode-kode bunyi yang direpresentasikan sebagai kode fonem, lama dan keras-lemahnya. Bagian ini sangat tergantung dari bahasa yang digunakan.
2. Fonem-ke-ucapan, berfungsi untuk mengubah kode-kode fonem, durasi dan keras-lemahnya, dari sub-sistem sebelumnya, menjadi ucapan.



Gambar 2.5 Diagram prinsip kerja Pensintesis Ucapan^[13]

2.3.4 OLE

OLE merupakan pengembangan dari teknologi Pertukaran Data Dinamis (*Data Dynamic Exchange*). OLE adalah teknologi yang dikembangkan oleh Microsoft untuk mendukung pertukaran informasi antar aplikasi, dimana informasi yang dikirimkan masih dapat dimodifikasi^[5]. Konsep dasarnya adalah informasi direpresentasikan sebagai suatu objek yang menghubungkan antar aplikasi atau keseluruhan dokumen objek tersebut ditempelkan ke aplikasi tujuan. Dari sinilah muncul istilah *Object Linking and Embedding*.

2.3.5 Windows API

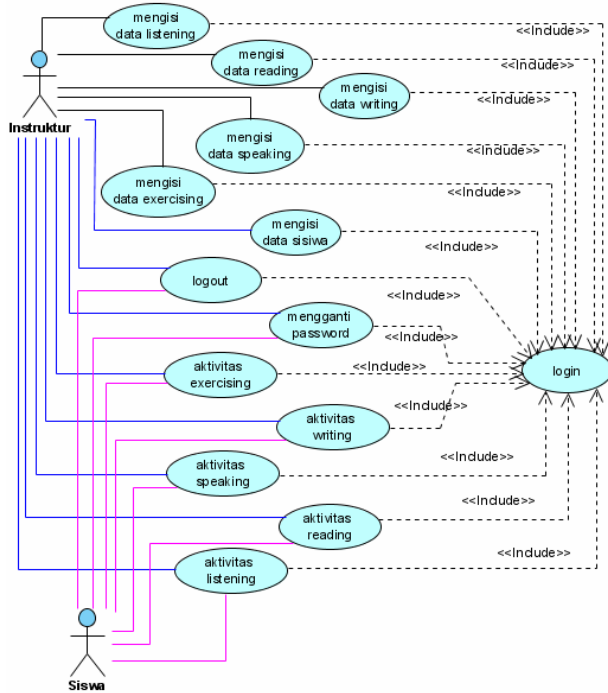
Windows API adalah sekumpulan fungsi dan konstanta yang terdapat dalam berkas-berkas DLL yang menyusun sistem operasi Windows^[7].

III. PERANCANGAN SISTEM BERORIENTASI OBYEK

3.1 AnalisisKebutuhan Sistem

Hasil dari analisis berorientasi objek merupakan masukan bagi perancangan berorientasi objek. Model analisis menyediakan informasi mendalam mengenai skenario penggunaan sistem dan menetapkan landasan untuk model implementasi. Aktivitas analisis dimulai dengan mengidentifikasi aktor, menemukan *uses case* dan mendeskripsikan proses bisnis dari sistem dan hasilnya

didokumentasikan ke dalam model analisis kebutuhan dan model proses bisnis sistem seperti ditunjukkan pada Gambar 3.1.



Gambar 3.1 Model analisis kebutuhan sistem

3.2 Perancangan Sistem

3.2.1 Arsitektur Komponen Perangkat Keras

Aplikasi yang akan dirancang merupakan aplikasi berbasis *client-server*. Dalam rancangan ini, tipe server yang akan digunakan adalah *database server*, sedangkan *client* dari sistem ini diimplementasikan dengan suatu *aplikasi GUI (Graphical User Interface)*. Implementasi aplikasi *client-server* dapat dilakukan dalam satu mesin yang sama ataupun terpisah. Model arsitektur komponen perangkat keras dari aplikasi yang akan dirancang ini dapat dilihat pada diagram *deployment* Gambar 3.2.

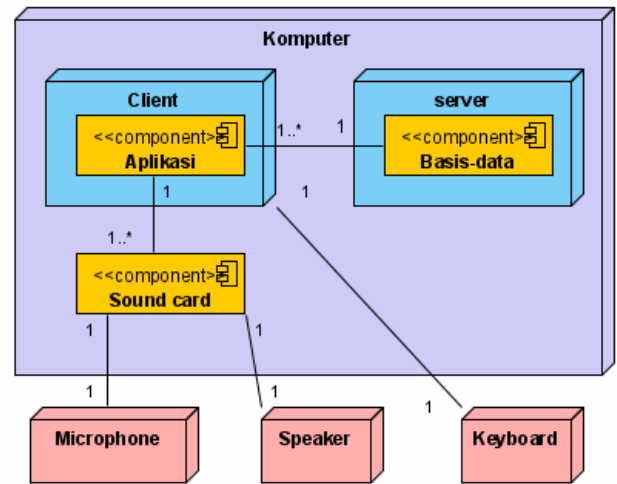
3.2.2 Arsitektur Komponen Perangkat Lunak

Diagram Komponen menggambarkan alokasi semua kelas dan objek kedalam komponen-komponen dalam rancangan fisik sistem aplikasi. Diagram ini memperlihatkan pengaturan dan kebergantungan antara komponen-komponen aplikasi, seperti kode sumber, kode biner dan komponen tereksekusi (*executable components*). Gambar 3.3 menunjukkan diagram komponen dari rancangan aplikasi yang akan dibuat.

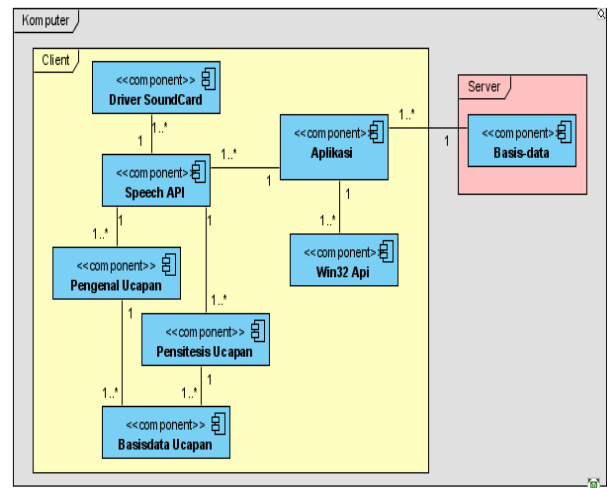
3.2.3 Arsitektur Logika Kelas

Didasarkan pada model-model aliran kendali pada tiap *use case* yang dideskripsikan

sebelumnya, maka Gambar 3.4 menjelaskan rancangan arsitektur dari kelas-kelas yang dibutuhkan oleh aplikasi. Gambar 3.4 menunjukkan daftar lengkap dari kelas-kelas yang terdapat dalam rancangan aplikasi ini.



Gambar 3.2 Model rancangan distribusi komponen perangkat keras sistem



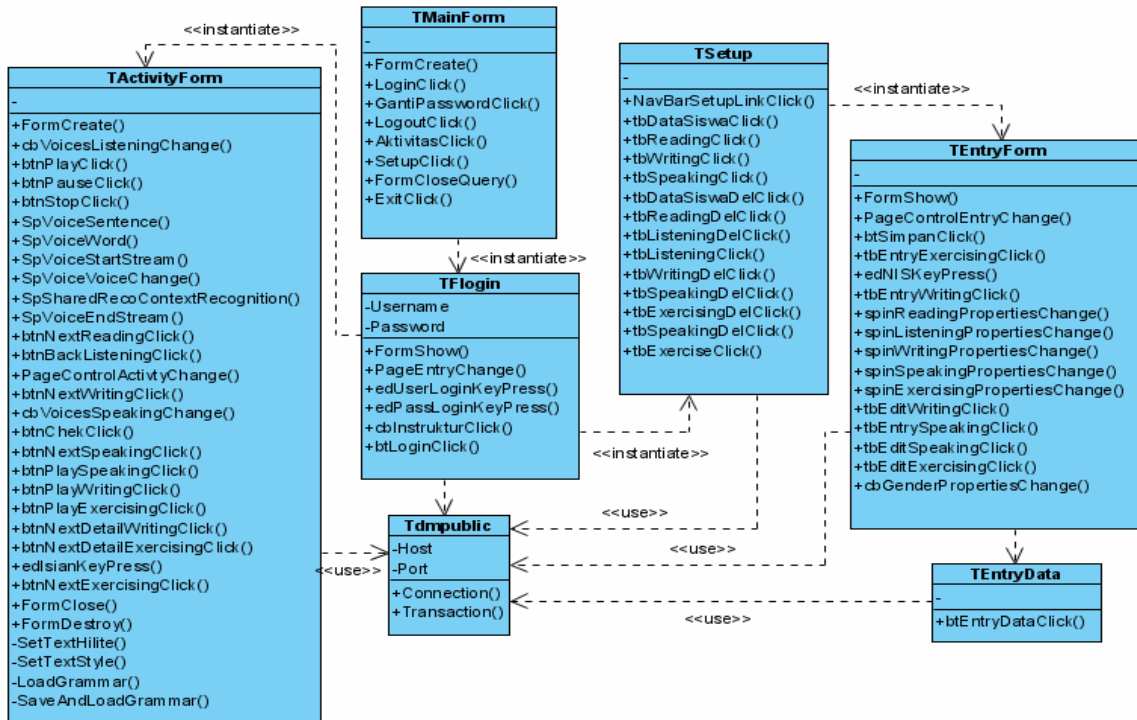
Gambar 3.3 Model rancangan distribusi komponen perangkat lunak sistem

3.3 Perancangan Antarmuka Manusia

Rancangan antarmuka pemakai merupakan hal yang kritis untuk penerimaan dan kesuksesan pemakaian produk. Tanpa antarmuka pemakai yang dirancang bagus, bahkan sistem-sistem dengan fitur yang lengkap akan menjadi tidak berhasil.

Langkah-langkah yang disarankan oleh Coad-Yourdon dalam merancang antarmuka manusia adalah:

1. Identifikasi pemakai
2. Perancangan rincian interaksi perintah
3. Perancangan hirarki perintah
4. Pembuatan prototipe tampilan antarmuka



Gambar 3.4 Model rancangan instansiasi antar kelas

3.3.1 Deskripsi Pengguna

Deskripsi tentang pengguna aplikasi dapat dilihat pada Tabel 3.1.

Tabel 3.1 Deskripsi Pengguna Aplikasi

Aktor : <i>User</i>
Peran : Melakukan peran sesuai dengan hak yang diberikan yaitu: 1. Instruktur berperan untuk melakukan fungsi administrasi basisdata yang meliputi <i>insert</i> , <i>update</i> dan <i>delete</i> pada data siswa, data <i>listening</i> , data <i>reading</i> , data <i>writing</i> , data <i>speaking</i> dan data <i>exercising</i> . Instruktur juga diberi hak untuk menjalankan peran siswa. 2. Siswa berperan untuk menjalankan fungsi aplikasi yang meliputi aktivitas <i>listening</i> , aktivitas <i>reading</i> , aktivitas <i>writing</i> , aktivitas <i>speaking</i> , aktivitas <i>exercising</i> dan mengganti <i>password</i> .
Karakteristik : 1. Mengetahui dasar-dasar komputer. 2. Bisa melakukan komunikasi melalui mikrofon. 3. Mampu memberi tanggapan suara dari pengeras suara atau <i>headset</i> . 4. Bisa memberikan masukan teks dari papan-ketik.
Faktor kesuksesan kritis : Melakukan komunikasi interaktif dengan komputer menggunakan teknologi teks-ke-suara dan suara-ke-teks dengan konsep pembelajaran Bahasa Inggris.
Level : Dasar – Menengah -Ahli
Skenario tugas yang dijalankan : 1. Sebagai Instruktur 2. Sebagai Siswa

3.3.2 Rincian Interaksi Perintah

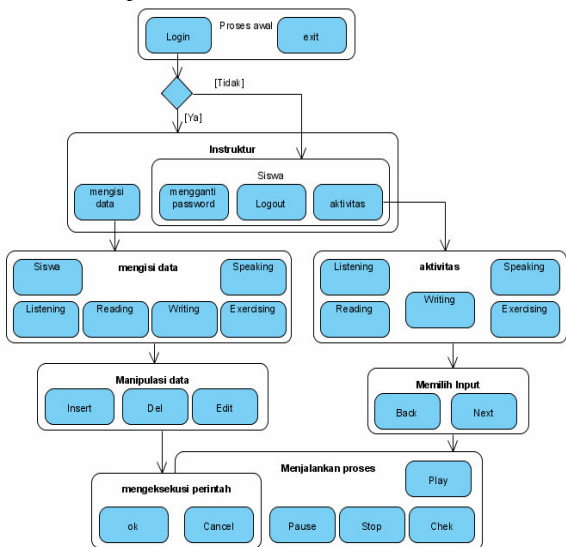
Rancangan rincian perintah yang akan ditampilkan secara konsisten pada aplikasi diperlihatkan pada **Error! Not a valid bookmark self-reference.**

Tabel 3.2 Rincian perintah dalam rancangan aplikasi

Aktivitas	Perintah
<i>Login</i>	<i>Login</i>
Mengisi data siswa	Mengisi DataSiswa
Mengisi data <i>listening</i>	Mengisi DataListening
Mengisi data <i>reading</i>	Mengisi DataReading
Mengisi data <i>writing</i>	Mengisi DataWriting
Mengisi data <i>speaking</i>	Mengisi DataSpeaking
Mengisi data <i>exercising</i>	Mengisi DataExercising
Melakukan aktivitas <i>listening</i>	AktivitasListening
Melakukan aktivitas <i>reading</i>	AktivitasReading
Melakukan aktivitas <i>writing</i>	AktivitasWriting
Melakukan aktivitas <i>speaking</i>	AktivitasSpeaking
Melakukan aktivitas <i>exercising</i>	AktivitasExercising
Mengisi data	<i>Insert</i>
Menghapus data	<i>Del</i>
Memodifikasi data	<i>Edit</i>
Menyimpan data	<i>OK</i>
Membatalkan proses	<i>Cancel</i>
Menjalankan proses	<i>Play</i>
Menghentikan sementara	<i>Pause</i>
Menghentikan proses	<i>Stop</i>
Meneruskan ke data selanjutnya	<i>Next</i>
Kembali ke data sebelumnya	<i>Back</i>
Membandingkan jawaban	<i>Chek</i>
Mengganti <i>password</i>	Ganti Password
<i>Logout</i>	<i>Logout</i>

3.3.3 Rincian Hirarki Perintah

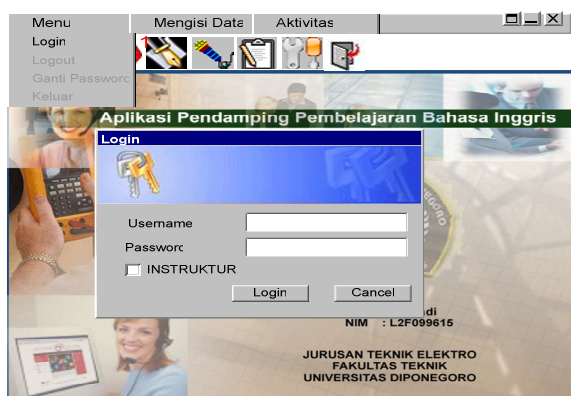
Berdasarkan model analisis kebutuhan sistem pada Gambar 3.1 dan penetapan rincian perintah pada Tabel 3.2, maka dapat dibuat model hirarki perintah yang disajikan dalam diagram *statechart* pada Gambar 3.5.



Gambar 3.5 Model rancangan hirarki perintah

3.3.4 Pembuatan Prototipe Antarmuka

Tampilan aplikasi dirancang dalam sebuah jendela yang berfungsi sebagai media untuk meletakkan menu dan kontrol-halaman (*page control*). Menu berisi perintah-perintah yang jika dieksekusi akan ditanggapi oleh aplikasi dengan menampilkan *form* baru yang berfungsi sebagai media interaksi pengguna dengan aplikasi. Contoh sebuah prototipe tampilan antarmuka yang menunjukkan perintah *Login* diperlihatkan pada



Gambar 3.6 Prototipe tampilan perintah *login*

3.4 Perancangan Manajemen Data

Tujuan dari perancangan manajemen data adalah memberikan deskripsi yang jelas kepada implementator rancangan (*programmer*) dalam menuliskan kode program yang sesuai dengan spesifikasi kriteria pengaksesan basisdata yang

diinginkan oleh perancang aplikasi. Tabel 3.3 menunjukkan spesifikasi rancangan manajemen data yang digunakan oleh aplikasi.

Tabel 3.3 Deskripsi Manajemen Data

Kriteria	Spesifikasi
Jenis manajemen data	Basisdata relasional
Bahasa pendefinisian dan manipulasi	SQL
Manajemen koneksi	<i>Non persistent connection</i>
Manajemen Transaksi	<i>Commit on success</i>
Strategi interaksi	<i>Common application programming interface (API)</i>

3.5 Perancangan Manajemen Task

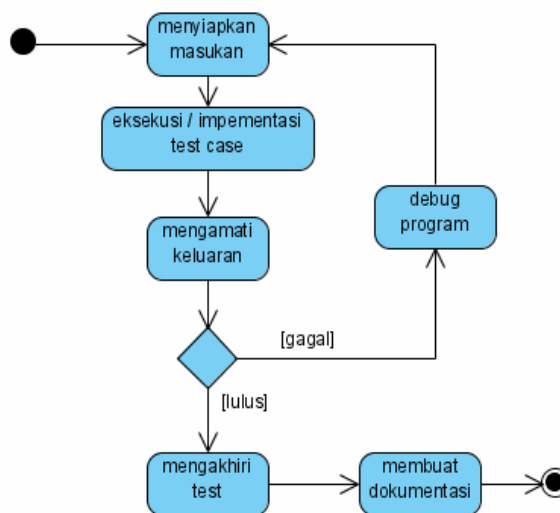
Perancangan inisiator aplikasi ini menggunakan task dituntun kejadian (*event-driven task*). Pada pemrograman visual, *event-driven task* dikenal dengan pemrograman *event*. *Event* pada dasarnya merupakan suatu aksi yang akan dilakukan oleh program jika terjadi suatu pemicu (*trigger*) baik dari pengguna program tersebut atau dari sistem lain^[10].

IV. Pengujian Berorientasi Obyek

4.1 Pengantar

Pengujian adalah proses pemeriksaan atau evaluasi sistem atau komponen sistem secara manual atau otomatis untuk memverifikasi apakah sistem memenuhi kebutuhan-kebutuhan yang dispesifikasikan atau mengidentifikasi perbedaan-perbedaan antara hasil yang diharapkan dengan hasil yang terjadi. Dalam hal ini dipakai rencana sebagai berikut.

1. Strategi pengujian : Pengujian Validasi
2. Metode : *Black Box*
3. Item-item pengujian : *Use case* dalam model kebutuhan sistem
4. Prosedur pengujian : Gambar 4.1



Gambar 4.1 Model aliran proses prosedur pengujian

4.2 Rancangan Pengujian

Tujuan perancangan pengujian adalah mendefinisikan kasus-kasus (*test case*) dan sampel data yang akan digunakan dalam melaksanakan prosedur-prosedur pengujian.

4.3 Laporan Ringkasan Pengujian

Setelah dilakukan proses implementasi dari semua *test case* yang telah dirancang maka didapatkan kesimpulan seperti yang terlihat pada Tabel 4.1.

Tabel 4.1 Ringkasan Hasil Pengujian

Test case	Use case	Status	
		gagal	lulus
Test case 1	Login		✓
Test case 2	Mengisi data siswa		✓
Test case 3	Mengisi data <i>listening</i>		✓
Test case 4	Mengisi data <i>reading</i>		✓
Test case 5	Mengisi data <i>writing</i>		✓
Test case 6	Mengisi data <i>speaking</i>		✓
Test case 7	Mengisi data <i>exercising</i>		✓
Test case 8	Melakukan aktivitas <i>listening</i>		✓
Test case 9	Melakukan aktivitas <i>reading</i>		✓
Test case 10	Melakukan aktivitas <i>writing</i>		✓
Test case 11	Melakukan aktivitas <i>speaking</i>		✓
Test case 12	Melakukan aktivitas <i>exercising</i>		✓
Test case 13	Mengganti <i>password</i>		✓
Test case 14	Logout		✓

V. Kesimpulan dan Saran

5.1 Kesimpulan

1. Telah berhasil dikembangkan aplikasi pembelajaran bahasa Inggris menggunakan Microsoft Speech API dengan metodologi perancangan berorientasi objek **Coad-Yourdon** menggunakan bahasa pemodelan **UML 1.5** pada aplikasi **Visual-Paradigm**. Aplikasi diimplementasikan menggunakan alat bantu bahasa pemrograman visual **Delphi 7.0** dan basis-data **Firebird 1.5** pada sistem operasi **Windows XP**.
2. Hasil analisis kebutuhan perangkat lunak mengidentifikasi dua peran yaitu Instruktur dan Siswa yang disebut Pengguna (*User*). Pada tingkat identifikasi kebutuhan sistem diperoleh empat-belas skenario penggunaan sistem (*use case*) yaitu: *login*, mengisi data siswa, mengisi data *listening*, mengisi data *reading*, mengisi data *writing*, mengisi data *speaking*, mengisi data *exercising*, aktivitas *listening*, aktivitas *reading*, aktivitas *writing*, aktivitas *speaking*, aktivitas *exercising*, mengganti *password* dan *logout* dari aplikasi.
3. Pengujian berorientasi objek yang diterapkan pada aplikasi menggunakan strategi pengujian **validasi** skenario dengan menerapkan metode

black box untuk setiap *test case* yang dilaksanakan. Semua *test case* lulus dalam pengujian tingkat validasi skenario. Sehingga semua *use case* dinyatakan berhasil melaksanakan skenario sesuai dengan yang dirancang dalam analisis kebutuhan sistem.

5.2 Saran

1. Aplikasi dapat dirancang selain dengan menggunakan metodologi perancangan berorientasi objek Coad-Yourdon, misal: Metode Booch, Metode Rumbaugh (*OMT-Object Modeling Technique*), Metode Jacobson (*OOSE-Object Oriented Software Engineering*), Metode Wirfs-Brock dan sebagainya.
2. Fitur rancangan dapat dilengkapi dengan skenario *use case* melihat nilai untuk setiap aktivitas yang dilakukan.
3. Implementasi perancangan sistem dapat diujicobakan menggunakan bahasa pemrograman visual selain Delphi 7.0, misal: Visual C++, Visual .Net, Visual Basic dan sebagainya dengan basis-data selain Firebird 1.5, misal: My-SQL, PostgreSQL, Interbase, MicrosoftSQL dan sebagainya.
4. Strategi pengujian dapat diperluas dengan menggunakan pengujian unit dan pengujian integrasi serta dengan menggunakan metode kotak-putih (*white box*), uji puncak-kebawah (*top-down test*), uji dasar-keatas (*bottom-up test*), uji penerimaan (*acceptance test*), uji kerusakan (*defect test*), uji banding (*benchmark test*), uji penekanan (*stress test*) dan sebagainya.

DAFTAR PUSTAKA

- [1] Booch, G., I. Jacobson, and J. Rumbaugh, *The Unified Modeling Language User Guide*, Addison-Wesley Longman Inc., Massachusetts, USA, 1999.
- [2] Fowler, M. *UML Distilled: Panduan Singkat Tentang Bahasa Pemodelan Objek Standar*, edisi ketiga, Penerbit Andi, Yogyakarta, 2005.
- [3] Hariyanto, B. *Rekayasa Sistem Berorientasi Objek*, Informatika, Bandung, 2004.
- [4] Long, B. *Adding Speech Synthesis and Speech Recognition capabilities into Delphi applications using The Microsoft Speech API (SAPI)*, <http://bdn.borland.com/article/0,1410,29580,00.html>, Januari 2004.
- [5] Long, B. *Automation In Delphi 5*, <http://www.blong.com/articles/AutomationDelphi5.html>, Januari 2004.
- [6] Long, B. *Speech Synthesis & Speech Recognition Using SAPI 5.1*, <http://bdn.borland.com/article/0,1410,29583,00.html>, Januari 2004.
- [7] Long, B. *Using the Windows API in Delphi*, <http://www.blong.com/articles/UsingWinAPI.html>, Januari 2004.

- [8] Martina, I. *Database Menggunakan Delphi*, Elex Media Computindo, Jakarta, 2001.
- [9] Miller, R. *Practical UML™: A Hands-On Introduction for Developers*, <http://bdn.borland.com/article/0,1410,31863,00.html>, Januari 2006.
- [10] Sudargo, P. *Pemrograman Berorientasi Objek Menggunakan Delphi*, Penerbit Andi, Yogyakarta, 2004.
- [11] Sudaryono. *Penggunaan "Authentic Materials" dalam Pengajaran Bahasa Indonesia bagi Penutur Asing*. <http://www.ialf.edu/bipafjan2003/authenticmaterials.html>, Januari, 2006.
- [12] Suhendar, A dan H. Gunadi, *Visual Modelling Menggunakan UML dan Rational Rose*. Informatika, Bandung, 2002.
- [13] Supriyono, B. *Perancangan Aplikasi VUI (Voice User Interface) Menggunakan Microsoft Speech API*, Skripsi S-1, Universitas Diponegoro, Semarang, 2004.
- [14] Warschauer, M. *Computer-assisted language learning: An introduction, Multimedia language teaching (pp. 3-20)*, Logos International, Tokyo, 1996.
- [15] -----, *Kurikulum 2004: Standar Kompetensi Mata Pelajaran Bahasa Inggris Sekolah Menengah Atas dan Madrasah Aliyah*, http://www.puskur.or.id/data/2004/StandarKompetensi/StandarKompetensi_SMA-MA/04BahasaInggris.pdf, Juli 2005.
- [16] -----, *Microsoft Speech SDK 5.1 Help*, Microsoft, Juli 2005.
- [17] -----, *Natural Language Processing*, <http://research.microsoft.com/nlp/>, Januari 2006.
- [18] -----, *Pengajaran Bahasa Asing Antara Sekolah dan Kursus*, <http://www.kompas.com/kompascetak/0407/08/PendIN/1129942.htm>, Juli 2005.
- [19] -----, *Speech Analysis Tutorial*, <http://www.ling.lu.se/research/spechtutorial/tutorial.html>, Januari 2006.
- [20] -----, *Speech Recognition*, <http://www.computerworld.com/SpeechRecognition.htm>, Januari 2006.
- [21] -----, *Speech Synthesis & Speech Recognition: Using SAPI 5.1*, <http://bdn.borland.com/article/0,1410,31863,00.html>, Januari 2004.
- [22] -----, *Win32 Developer's Reference*, <http://www.msdn.microsoft.com>, Januari 2006.



Karnadi, Lahir di Sragen 30 April 1981. Saat ini sedang mendalami konsentrasi Informatika dan Komputer di Teknik Elektro Universitas Diponegoro.

Mengetahui dan Mengesahkan

Pembimbing I

Agung Budi Prasetyo, S.T., MIT

NIP : 132 137 932

Tanggal:

Pembimbing II

R. Rizal Isnanto S.T., M.M., M.T.

NIP 132 288 515

Tanggal: