Faculty of Computer Science and Information Technology
University Malaya

NG KOON LEE

WEK 990231

SPEECH RECOGNITION AND SPEECH SYNTHESIS REPOSITORY SYSTEM

Dissertation submitted by in partial fulfillment of the requirements for the Degree of Bachelor of Computer Science

Supervised by
Pn. Norizan Mohd. Yasin

Moderated by
Pn. Fariza Hanum Md. Nasaruddin

Contents

Chapter 3: Methodology

Chapter 4: System Design

Chapter 5: System Implementation and Testing

Chapter 6 Project Evaluation

Example Source Code

List Of Figure

List Of Table

Abstract

One of the advance computer technologies today perhaps is the system that able to communicate with the user via voice. From speak, user need not use hand frequently to control the system but with voice, all possible instruction can handle by the system do for user.

This project main purpose is to build a system that able to show this technology especially in database. This project named Speech Recognition and Synthesis Repository System (SRn'SSRS). With this application, user can retrieve data from the database after setup. It works just like a search engine like Yahoo! but the information will be limited to faculty student information.

This application will be support Malay language and also pointing device for user to interact with the computer. Development tools are from Microsoft Speech SDK5.x together with Microsoft Platform SDK and Microsoft Visual Studio. The programming language for development including Visual C++, COM, XML and HTML.

For this system's function, it can let user to control 6 type of function, which include: student registration information, student name list based on major, subject and year, list of subject in different semester and also student result.

After the development, a lot of test cases and testing method have been done. The strength and limitation of system conclude during using the system. Problems and solution was list out during testing. Future enhancement also was including in the project in order to enhance the system performance.

Acknowledgement

First of all, I would like to thanks to Puan Norizan Mohd. Yasin as my supervisor, which giving me a lot of suggestion for this project and guided me in the time when prepared the Latihan Ilmiah Tahap Akhir II. Without her helped, this project wouldn't possible can finish.

I would like to thanks also to Puan Fariza Hanum that spending time to hear my presentation and also give me some idea in presentation. It did help me to know problems could I face with when I am ready.

Also, thanks to my family for provided some of the facility for my convenient in development phase.

For my friends also that giving me some of the idea of programming language and database fields. Their help save my time in design phase.

And also to FSKTM (Fakulti Sains Komputer dan Teknologi Maklumat) that provided the related source and facility to me.

Chapter 1
Introduction

Chapter 1
Introduction

1.1 Project Definition

The Speech Recognition and Speech Synthesis Repository System (SRn'SSRS) is a computer program that allows a user to communicate with the computer in database field through human voice. It can be done when user has taking part in training or enrollment for a minimum time for system to record the user voice profile. After the lesson, user can always communicate with the system through human natural speaking as an alternative input device, with the help of recognition engine's technology, At this stage, it is called Speech Recognition (SR). At the same time, the computer also will respond to the user using synthesis engine's technology through speaker to inform the status. At this stage, it is called Test-To-Speech (TTS) or Speech Synthesis (SS). This system is just behaved like a search engine but the different is only searching data for certain field and control via voice.

With the complete of this system, It can delivery some of the advantages over other input device. The advantages of this system is it help the user to get data from the database, at the same time user can continue in their daily works while waiting the system to generate information from the database. When it's done, the system will inform the user in time. It's all done without any movement of hand since mouse is no a main input anymore after the activation of this system. Because it also support both Malay and English language, It help the user more easily communicate with the computer since Malay is knowing by every one.

On the other hand, this system has its uncertainty and limitation. The environment takes a lot of impact to the system's performance. One good example was the sound from the background and the quality of the microphone. Each different user also require to set up a new account, which is training for a minimum lesson for the system to recognize the

sound of the user. It was which is very time consuming for the first time user. Anyway, this system will get more accurate in recognition the sound if user completed the whole lesson.

1.2 Objective

The objective of this program:

1. First, to build up a system, which can use Malay language in the method of voice to communicate with the computer to get the information that user needs. In this system, user can gave a set of instruction to the system via voice command. Instead of giving the instruction in a formal way, users can also speaking in a more natural command just like a daily communication with each other. This system does not fully function with the normal database program like create a new format of database management since the system main purpose was helping user to interact with the computer in retrieval information or checking from the existing database. It will just function just like a search engine.

2. Second, the program is as an alternative for user to interact with the computer via voice. It let user to try a new method since using pointing device or mouse sometimes quiet troublesome for certain case, need space for mouse pad or mouse not function properly.

3. Third, which is more to faculty, is to help the faculty have more advance technology to use or test before it is become wide use. It let Fakulti Sains Komputer dan Teknologi Maklumat (FSKTM) able to stand still on top of every faculty in Information Technology field.

4. Fourth, with the support of Malay, it also means the product is suitable for our country culture and user friendly.

1.3 Scope

Generally, Speech Recognition and Speech Synthesis Repository System (SRn'SSRS) provided a convenience way of retrieve information from database using human natural speaking concept. With the help of this system, user will be able to support Open Database Connectivity (ODBC) structure. It will be performance in Microsoft SQL environment. With the first time setup for choosing the type of database, file that store the information to retrieve and record user voice profile, the system ready to operate.

In voice command environment, user can talk to the system, which the process called Speech Recognition (SR). The system will get the voice and generate information according to the command and display the result on the screen. The system will also notice the user in voice feedback when the process finishes, either pass or fail. The process where system talks to the user is called Test-To-Speech (TTS).

User will also be able to speak to the system in Malay language or English. In some situation, it also able to detect when user using both Malay and English at the same time.

This system also supports the function for pointing device even though user is encourage to control via voice. The purpose for including the extra support is for helping the user able to continue to retrieve the information when the system failed to recognize the command.

Te system development purpose was mainly in speech technology with support-limited function of database, which is only searching and retrieve data. So, the system will not be able to create a new format of database of environment in voice command. It is certainly suitable using the traditional way, which is pointing device and keyboard in this issue.

The scope for the system currently only support registration data of FSKTM student. It means that user only can search all related information for FSKTM student but not support for the other faculty currently.

As the other way to explain, this system is using voice command to searching related information from FSKTM database, besides using other input device.

The target user for this system is for the staff or lecturer of FSKTM of University Malaya who is responsible helping student in registration problems or as references to certain information. So, all the presentation or interface is only based on the registration environment in FSKTM. Through the system, the user can check some data with giving the condition and command to the system. System will generate and respond to the user with showing the result on the screen. When the system failed to recognize the user command or failed to get data from the database, system will also feedback to user on the screen and voice out as a warning or message.

Beside, every time system generate the information after receiving the result, system will also respond back to the user telling that the process has successfully and ready for the next command. It helped user to continue other works without missed the process.

1.4 Project Significant

Since in this country still doesn't have a database system that support speech technology, so it is possible that the local market need this technology, especially can communicate in Malay language and with some modification, it still can communication in English.

Another reason is the product which using the similar technology likes IBM ViaVoice or Dragon NaturallySpeaking was so expensive for user. Users who pay that amount of money sometimes never need to use some of the function. They only used some of the basic function like process speaking or asking computer to read only. User who needs only basic function or just focusing in certain field with a light package will help in their finance.

Finally the product in the market just mention usually requirement a powerful processor and hungry memory because of their fully package and multifunction. It comes to the problems that not every user has that such of advance of computer.

1.5 Project Timeline

Below is the timetable and Gantt chart for the project timeline:

Activity	Duration (Days)	Start	Finish
System Study	44	1 Jun 2001	1 August 2001
Requirement Analysis	31	20 Jun 2001	1 August 2001
Design	43	1 August 2001	28 Sept 2001
Coding	80	1 Oct 2001	18 Jan 2002
Testing	80	1 Oct 2001	18 Jan 2002
Review	35	3 Dec 2001	18 Jan 2002
Documentation	143	9 July 2001	23 Jan 2002

Table 1.1 Project Timetable

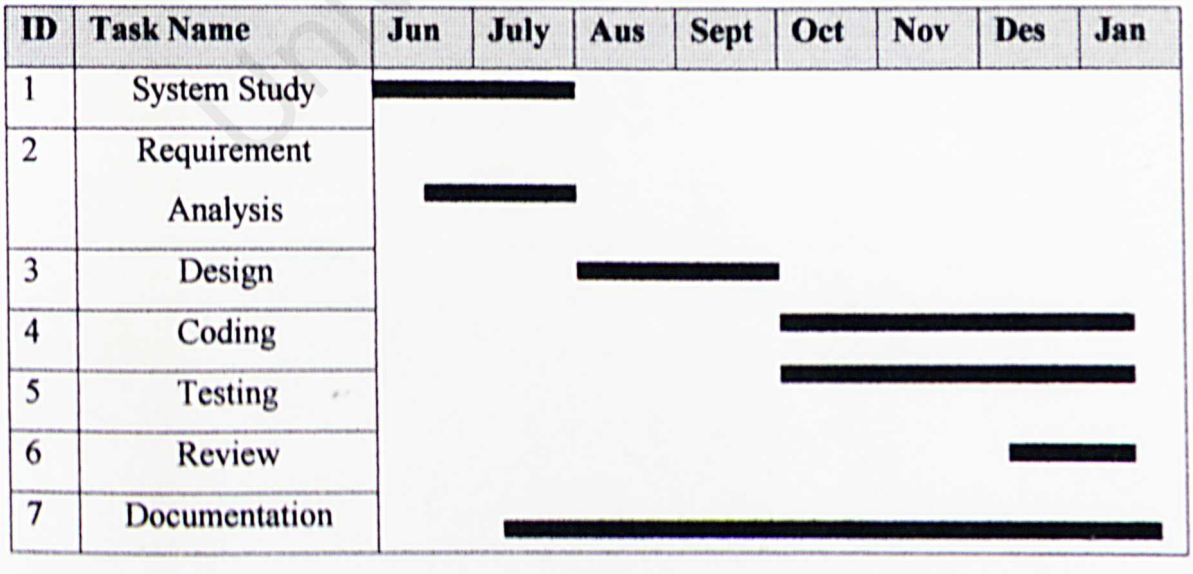


Figure 1.1 Gantt chart

1.6 Summary

This chapter explains the overview of the project definition for function of Speech Recognition and Speech Synthesis Repository System. The system works like a search engine on the Internet but it using the different technology, which is the speech technology for control and interact with the system. While the data store is also specific, that's only data of FSKTM student information. It also defines the objective to develop due to requirement and market needs. The project limitation also covers in this section, same with the scope of the function and the target user.

Chapter 2
Literature Review

Chapter 2
Literature Review

2.1 Speech Recognition and Speech Synthesis Repository System

Speech technology is one of the most popular for computer fields. Even people with no interest in technology became familiar with the idea of talking to the computer. This can be seen in a lot of science fiction movie like Star Wars and Star Trek. For a deeper knowledge, those system that's support the speech technology have the following characteristics:

- The system able to recognition the voice from the outside
- The system able to detect either the source is a command for the system or just a background noise.
- The system also able to respond to the user via voice

The idea behind of creating the Speech Recognition and Speech Synthesis Repository System (SRn'SSRS) is the possibility of enabling the user to get the alternative way to search data from database beside using keyboard and pointing device. In this system, there are 2 modules involved:

- Speech engine
- Database

The speech engine further divided to 2 main engines:

- Speech Recognition engine (SR)
- Text-to-Speech engine (TTS)

While the database contain the FSKTM (Fakulti Sains Komputer dan Teknologi Maklumat) student information. It includes student metrics number, year of study, semester, course taking and etc.

The Speech Recognition engine or sometimes called *Speech-to-Text* is the engine that running in background when user interact with the system via voice. This engine enables the system to generate the voice that receives. It then converts the information to the command that system understand and respond with showing the result on the screen. It means when user speak through microphone, the microphone converts the voice into an analog signal and feeds it to the PC's sound card. An analog-to-digital converter takes the signal and converts it to a stream of digital data (ones and zeros). Then the software goes to work with the corresponding output.

The Text-to-Speech engine on the other hand run the process opposite with the speech recognition engine. In this stage, the engine will convert the text, which the system would like to say to voice according to different cases with the help of engine.

Example case is if a user need to retrieve information about the how many student taking course name "Latihan Ilmiah Tahap I", user only need to speak to the computer with a suitable command asking the system to generate this information for him/her. The system starts working after recognized user voice and getting the correct command and display the result on the screen. After the process, system will ask user to do the further task, if user wish to.

2.2 Varieties of Speech Software

There are a lot of speech products on the market. These speech products for the big company have a different of choice in supporting different operating system, different programming language and different characteristic and use. Some product can be run in Linux platform, some ca be running in Window platform. Some of the program only can be use for reading e-mail and some more advance help user to control every command on the computer. Example is launching a program. For a small and simple product, the language support was limited to 1 or 2 most common languages that are worldwide known. As for a bigger and multiple use products, it could be support up to more than 5

languages. Table 2.1 shows the description of some of the product that's currently on the market.

Product	Company	Function
IBM Via Voice	IBM	Supports speech recognition in Brazilian, Portuguese, Chinese (Simplified and Traditional), French, German, Italian, Japanese, Spanish, U.K. English, and U.S. English languages
Dragon NaturallySpeaking	L&H	Support speech recognition in British English, French, Italian, German, and Spanish
L&H Voice Xpress	L&H	Specialized dictionary of legal and medical terminology
AOL IM Speech 2000	AOL	This program enhances instant messaging even more by providing text-to-speech capabilities
ReadPlease	Money Tree Software	Read the document for user
Game Commander	Microsoft	Control games with voice
FreeSpeech 2000	Philips	Able to control consumer electronics and automotive device & application

Table 2.1 Speech's product in the market

2.3 Programming Language Support in Existing Product

A lot of related product exists today use Visual C++ and Visual Basic programming language. Using Java was also found in some of the mega product that support in helping user to develop a new program. These packages of program including the speech program and their SDK (Software Development Kit) in order to let programmer to develop a new function in suitable language. The purpose for these companies to provide is to help the company able to continue produce a powerful of program with the help of programmer and continue to exist in the market place.

For IBM Via Voice Development tools, it enable the programmer to choose variety of programming language including Microsoft Visual C++, Visual J++, Visual Basic, Borland Delphi, Borland C++ Builder, or IBM VisualAge C++. Any development platform that allows access to C++ APIs and OCX containment is a viable platform to develop for Via Voice technologies [13].

As for Dragon NaturallySpeaking Developer Suite, it contains ActiveX controls that make speech-enabling applications easier than ever before. Developers can integrate the full capabilities of Dragon NaturallySpeaking directly into their programs using Visual Basic, Visual C++, Delphi, and other development environments supporting the industry-standard ActiveX specification. Support is also provided for the Microsoft Speech Application Programming Interface (SAPI), which is based on the Component Object Model (COM) architecture under Win32. The component objects can be accessed through C/C++ or Visual Basic Automation [5].

2.3.1 Introduction of Visual C++ and the Advantages

Visual C++ is an application development tool developed by Microsoft for C++ programmers [6,14,15]. C++ is a high-level programming language developed by Bjarne Stroustrup at Bell Labs. C++ adds object-oriented features to its predecessor, C. C++ is one of the most popular programming language for graphical applications, such as those that run in Windows and Macintosh environment.

2.3.2 Introduction of Visual Basic and the Advantages

A programming language and environment developed by Microsoft [6,14,15]. Based on the BASIC language, Visual Basic was one of the first products to provide a graphical programming environment and a paint metaphor for developing user interface. Instead of worrying about syntax details, the Visual Basic programmer can add a substantial amount of code simply by dragging and dropping controls, such as buttons and dialog boxes, and then defining their appearance and behavior.

Although not a true object-oriented programming language in the strictest sense, Visual Basic nevertheless has an object-oriented philosophy. It is sometimes called an event-driven language because each object can react to different events such as a mouse click.

Since its launch in 1990, the Visual Basic approach has become the norm for programming languages. Now there are visual environments for many programming language, including C, C++, Pascal and Java. Visual Basic is sometimes called Rapid Application Development (RAD) system because it enables programmers to quickly build prototype application. RAD is a programming system that enables programmers to build a working program in a short period. In general, RAD system provided a number of tools help build graphic user interface that would normally take a large environment effort. Two of the most popular RAD system for Windows is Visual Basic and Delphi.

2.4 Structure of Speech Recognition and Synthesis System

For developing the speech repository system, one of the tools that use to develop the system was the Microsoft Speech Software Development Kit (Microsoft Speech SDK) 5.0. With the help of this SDK, the system will be able to integrate the speech recognition a speech synthesis engines with the Microsoft Speech API 5.0 (SAPI 5.0)[18]. SAPI 5.0 consists of 2 interfaces [19]:

- Application Programming Interface (API)
- Device Driver Interface (DDI)

The SAPI 5.0 API dramatically reduces the code overhead required for an application to use speech recognition and synthesis, making speech more useable for more applications by developer.

2.4.1 API and DDI Overview

The SAPI 5.0 DDI and API remove the implementation details such as multi-threading and audio device management making speech synthesis and recognition engines and applications convenient.

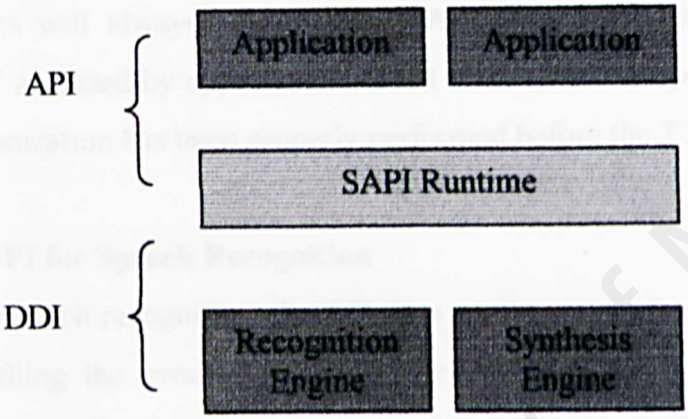


Figure 2.1 API and DDI

2.4.2 API Events

In both speech recognition and speech synthesis, application receives events. For example, when words of text spoken by user to system have been recognized or when system spoken the words. SAPI components that have generated the events will start running to notify the source. When the application receives a notification, it will either retrieves or remove the events [19].

2.4.3 API for Speech Synthesis

For basic speech synthesis, and application need to use an interface for synthesize speech for the text passed in as a parameter. In addition to generating speech from plain text, an application can place synthesis markup language within the text passed into interface.

This markup language is based on the XML format. The format of the text (either Unicode or ANSI) will automatically be detected [19].

2.4.4 DDI for Speech Synthesis

In Device Driver Interface (DDI), a SAPI speech synthesis engine also implements a different interface in this case. SAPI, rather than the engine, performs XML parsing of the input text stream. The Speak method is handed a linked list of text fragments with their associated XML attribute state. The TTS engine uses this interface to queue events and write output data. Even though SAPI is a free-threaded architecture, the TTS engine instances will always be called by SAPI on a single thread. TTS engines are never directly accessed by applications. SAPI ensures that all parameter validation and thread synchronization has been properly performed before the TTS engine was called [19].

2.4.5 API for Speech Recognition

As for speech recognition, the API uses another type of interface mainly for this purpose in handling the events. This interface works as the speech application's vehicle for receiving notifications for the requested speech recognition events [19].

2.4.6 DDI for Speech Recognition

At the most basic level, the SAPI 5.0 Recognition DDI provides the functionality for an engine to receive audio data from SAPI and return phrase recognitions. There are 2 interfaces that use for it: one is implemented by the engine and another one is implemented by SAPI. The SAPI interface also provides speech engine methods for communicating more detailed information on what the engine recognizes. Grammars and speakers provide information to engines that better help them recognize speech and are a critical part of the communication between SAPI and speech engines [19].

2.5 Characteristics of Speech Recognition And Synthesis Repository System

- **Separates Speech Engine from Repository System**

  With the help of interface connecting between a database and a speech system, it make possible for the system engine separate with the database. Separating the speech engine and the database makes the speech engine to easily support any data of the new database, and any changes of data will not be affected to speech engine. The same also came to speech engine, where any maintenance of the program will also safe for the data in database.

- **Need to Record User Voice Profile**

  All user need to follow a minimum of 20 minutes training period in order to let the system to recognized user voice. To get more accurate result, sometimes the user needs to follow the lesson for an hour long.

- **Performance Affected by Background Environment and Hardware**

  If the system place in a silent room, the performances will better than in an open noisy area. The microphones that use to communicate with the computer suggest to place near the user mouth and recommend using the headset microphone.

- **The Speed of User Speaking**

  In most of the speech system, it all needs the user to speak in a gentle with a clear voice in order to let system manage to catch the entire message.

- **Continuous vs. Discrete**

  If speech recognition is continuous [1,12], users can speak to the system naturally. If it's discrete, users need to pause between each word. Obviously, continuous recognition is preferred over discrete recognition, but continuous recognition takes more processing power.

- **Vocabulary Size**

  Speech recognition can support a small or large vocabulary [1,12]. Small-vocabulary recognition allows users to give simple commands to their computers. To dictate a document, the system must have large-vocabulary recognition. Large-vocabulary recognition takes a lot more processor power and memory than small-vocabulary recognition

- **Combination of Characteristics**

  "Command and Control" speech recognition is continuous, small vocabulary, and speaker independent [1,12]. This means that users can use several hundred different commands or phrases. If a user says a command that is not in the list, the speech-recognition system will return either "not recognized," or will think it heard a similar-sounding command. Because users of Command and Control can say only specific phrases, the phrases must be either visible on the screen--so intuitive that all users will know what to say--or the users must learn what phrases they can say. "Discrete Dictation" speech recognition is discrete, large vocabulary, and speaker dependent. It's used to dictate text into word processors or mail, or for natural-language commands. Although users may say anything they wish, they must leave pauses between words, making the speech unnatural.

- **Makes Mistakes**

  Since this technology still in development phase, it still could make mistake. 80 percent of accuracy is still the target in 10 commands. So the overall convenience in correcting errors is just as important as the recognition itself in the overall goal of producing an error-free system

- **Matching Techniques**

  Speech-recognition engines match a detected word to a known word using one of these techniques [12]:

- **Whole-word matching**. The engine compares the incoming digital-audio signal against a prerecorded template of the word. This technique takes much less processing than sub word matching, but it requires that the user (or someone) prerecord every word that will be recognized - sometimes several hundred thousand words. Whole-word templates also require large amounts of storage (between 50 and 512 bytes per word) and are practical only if the recognition vocabulary is known when the application is developed.

- **Sub word matching**. The engine looks for sub words - usually phonemes - and then performs further pattern recognition on those. This technique takes more processing than whole-word matching, but it requires much less storage (between 5 and 20 bytes per word). In addition, the pronunciation of the word can be guessed from English text without requiring the user to speak the word beforehand.

2.6 Speech Recognition Mode

A speech recognition system can be used in many different modes [1,12]. There are:

1. Speaker-dependent or independent
2. Isolated word recognition
3. Continuous speech recognition
4. Keyword spotting

❖ **Speaker Dependent / Independent System**

A speaker-dependent system is a system that must be trained on a specific speaker in order to recognize accurately what has been said. To train a system, the speaker is asked to record predefined words or sentences that will be analyzed and whose analysis results will be stored. This mode is mainly used in dictation systems where a

single speaker is using the speech recognition system. On the contrary, any speaker without any training procedure can use speaker-independent systems. Those systems are thus used in applications where it is not possible to have a training stage (telephony applications, typically). It is also clear that the accuracy for the speaker-dependent mode is better compared to that of the speaker-independent mode [1].

❖ **Isolated Word Recognition**

This is the simplest speech recognition mode and the less greedy in terms of CPU requirement. Each word is surrounded by a silence so that word boundaries are well known. The system does not need to find the beginning and the end of each word in a sentence. The word is compared to a list of words models, and the model with the highest score is retained by the system. This kind of recognition is mainly used in telephony application to replace traditional method [1].

❖ **Continuous Speech Recognition**

Continuous speech recognition is much more natural and user-friendly. It assumes the computer is able to recognize a sequence of words in a sentence. But this mode requires much more CPU and memory, and the recognition accuracy is really inferior compared with the preceding mode [1]. Continuous speech recognition more difficult than isolated word recognition because:

- Speakers pronunciation is less careful
- Speaking rate is less constant
- Word boundaries are not necessarily clear

❖ **Keyword Spotting**

This mode has been created to cover the gap between continuous and isolated speech recognition [1,12]. Recognition systems based on keyword spotting are able to identify in a sentence a word or a group of words corresponding to a particular command. For example, in the case of a virtual kiosk providing any customer with the way to a special department in a supermarket, there are many different ways of asking

this kind of information. One possibility could be "Hello, can you please give me the way to the television department". The system should be able to extract from the sentence the important word "television" and to give the associated information to the customer. This is also the mode to use in this system development.

2.6.1 Speech Recognition Process Flow

Figure 2.3 showed the process flow of speech recognition [10]:

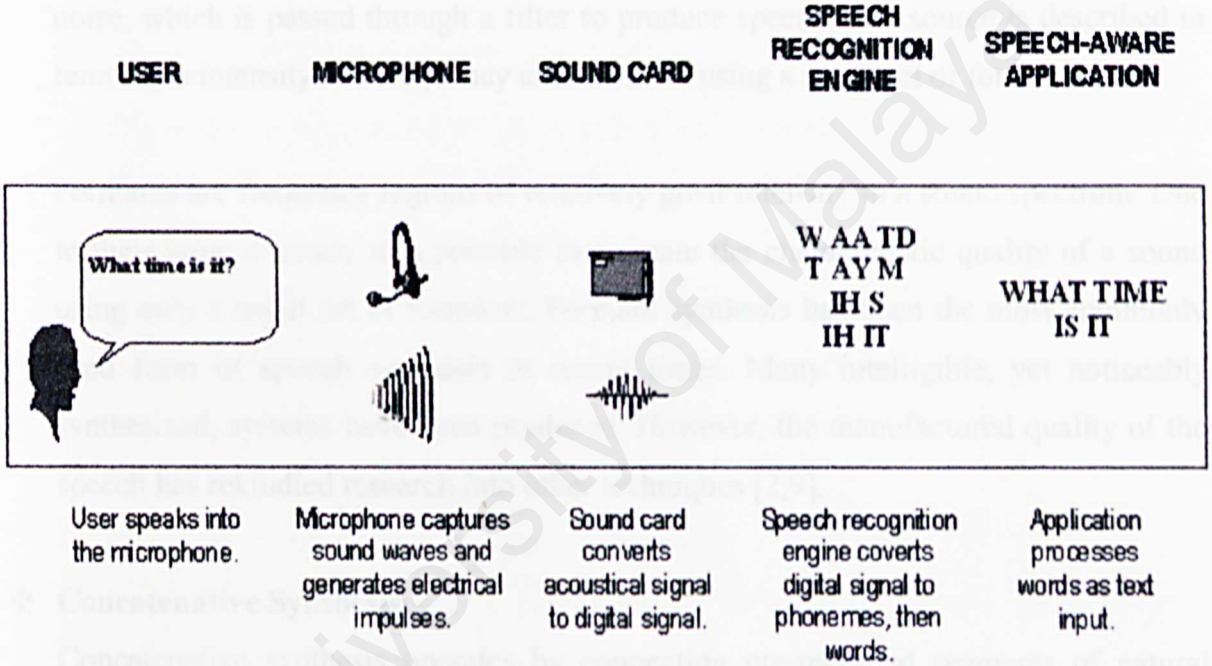


Figure 2.2 Speech recognition process flow

2.7 Speech Synthesis Mode

For speech synthesis module, there are mainly two mode or method used, which are [2,9]:

1. Formant Synthesis
2. Concatenative Synthesis

❖ **Formant Synthesis**

Formant synthesis is based on the source-filter model. A source provides a constant noise, which is passed through a filter to produce speech. The source is described in term of its intensity and frequency and the filter using a small set of formants.

Formants are frequency regions of relatively great intensity in a sound spectrum. Due to their large intensity it is possible to recreate the characteristic quality of a sound using only a small set of formants. Formant synthesis has been the most commonly used form of speech synthesis in recent times. Many intelligible, yet noticeably synthesized, systems have been produced. However, the manufactured quality of the speech has rekindled research into other techniques [2,9].

❖ **Concatenative Synthesis**

Concatenative synthesis operates by connecting pre-recorded segments of natural language. For easy explanation, it is record own the human voice and cut it into small segment for different words. This tends restrict these method to a single voice.

Key to performing concatenative synthesis is selecting the length of each recorded segment. Using short segments, such as words, will sound disjointed. However, using larger segments, such as sentences, will require large storage requirements.

There is interest in using concatenative synthesis where the quality of speech is of high priority and the breadth of vocabulary required is quite small. To overcome the

disjoint flow of combined sentences a "smoothing out" processing phase is sometimes employed [2,9].

2.7.1 Advantages and Disadvantages

- **Advantages of Formant Synthesis**

  Formant synthesis is very flexible, able to produce an infinite number of sounds. Additionally, it has comparatively reduced complexity compared to other methods [9].

- **Advantages of Concatenative Synthesis**

  The quality of speech is very good, as it exactly mirrors that of a human [9].

- **Disadvantages of Formant Synthesis**

  The quality of a given formant synthesis machine tends to favor a small subset of sound. To favorably cover all the sounds necessary for human speech can require quite sophisticated combinations of these individual machines [9].

- **Disadvantages of Concatenative Synthesis**

  It can be difficult to find a good balance between the storage space of large segments and the quality of the natural flow of speech [9].

2.7.2 Speech Synthesis Process Flow

Figure 2.3 below show the process flow of speech synthesis involved [10]:

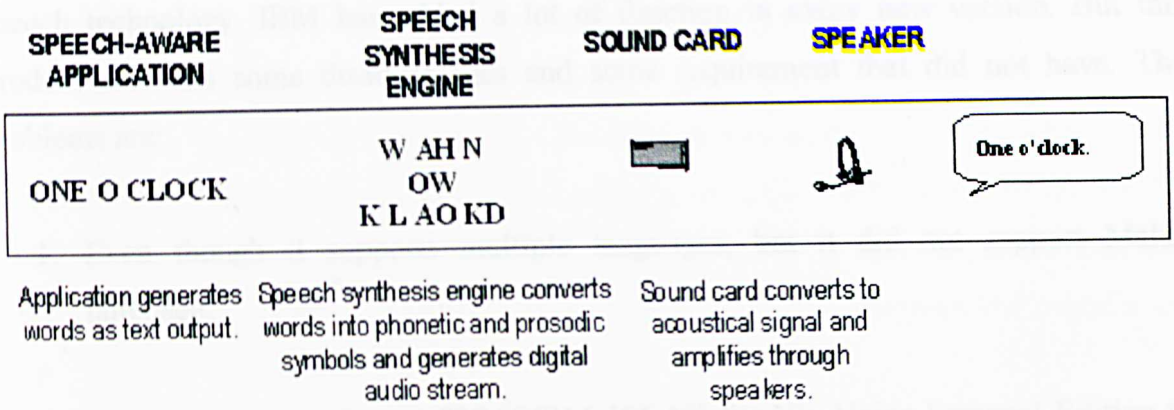


Figure 2.3 Speech synthesis process flow

2.8 Comparison Existing Product

The objective of comparison the product in the market is to check if the idea of product now can have any function that's needed to be added or modify in the system that will develop. In this comparison, IBM Via Voice has been chosen as the target of comparison.

No doubt that IBM Via Voice comes in a lot of function and advance function and performance. As a leading Information Technology company, IBM began selling commercial speech recognition products using its own technology in 1992. In 1993,IBM released its first personal speech product, the IBM Personal Dictation System for OS/2. This product was the first high accuracy speech recognition product available commercially. In 1996 IBM broke ground again, this time with IBM VoiceType Simply Speaking - the world's first commercially available retail offering of high-accuracy spoken dictation technology. In 1997 IBM raised the bar once more and released Via Voice - the first continuous speech technology spoken dictation product in the family. With the advent of Via Voice technology, users no longer had to place short pauses between words as they spoke, instead they could speak freely and normally. With Via Voice IBM has a set of technologies that have actually started us on a journey towards meeting our set expectations from science fiction. Until recently, IBM has release the

IBM Via Voice Version 8. With more then 10 years in research and development in speech technology, IBM has added a lot of function in every new version. But this product also has some disadvantages and some requirement that did not have. The problems are:

1. Even though it supports multiple languages, but it did not support Malay language.

2. It was expensive. It cost $28.00(RM 106.40) for Via Voice Personal Edition to $190.00 (RM722) for Via Voice Pro Edition [13].

3. Users pay more but just never use all the functions provide. They usually only need some of it for daily task.

4. With the multifunctional package, it also needs higher processor clock speed and hungry memory. But not every user have advance computer to have good performance. It means it slow down the user's computer.

With the problems above, the system that will be develop will try to solve some of the problems:

1. The system will only support speech recognition and synthesis in database field. In the other words it just like a search engine but input keyword via voice.

2. The system will support Malay language and pointing device for certain case when system failed to recognition the command.

3. The program will be small and with just some basic daily function

4. User can speak with the computer with more natural.

2.9 Microsoft Access SQL

Most relational database management system (RDBMS) today provided some type of query or data management language to access database. It allows users to access information in database in a simple and flexible manner and one of the most popular query languages called Structured Query Language (or SQL). SQL is part of the DBMS (Database Management System). Users can access a database using SQL and they can write program, which can include SQL commands. Figure 2.4 illustrates this explanation [18]:

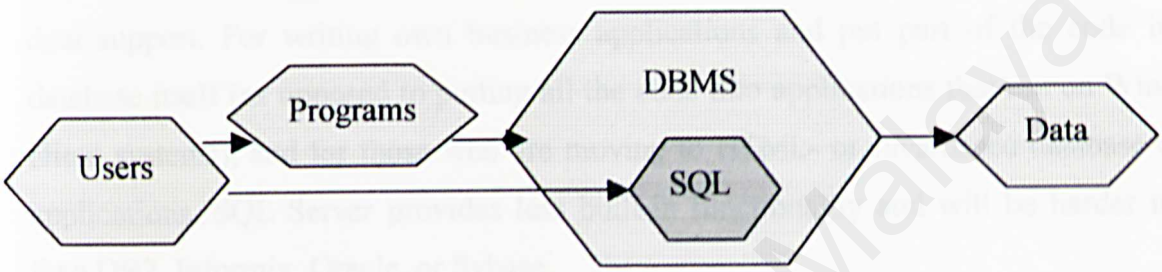


Figure 2.4 Place of SQL in DBMS

SQL has 2 components, which are:

1. Data Definition Language (DDL)
2. Data Manipulation Language (DML)

DDL is use to define the database structure while DML used to manipulate the data.

2.9.1 SQL Commands

SQL commands are not case sensitive. They can be typed in upper, lower or mixed case. For example, users can type SELECT, Select or select. But one exception in this rule: the literal characters must be entered exactly as they appear in the database. For example, if the database stores "Kad Matriks" and users search "kad matriks" the record would not be found.

The following conventions are used in the literature:

- Uppercase letters are used to represent reserved words.
- Lower case is used to represent user-defined words.
- Vertical bars (|) are used to represent optional elements.
- Square brackets [] are used to indicate optional elements.
- Curly brackets {} are used to indicate optional elements.
- Ellipsis (...) are used to represents optional repetition of an item (0 or more times).

Anyway, this product has some of the disadvantages: programmability and multimedia data support. For writing own business applications and put part of the code in the database itself (as opposed to putting all the code into applications that run on Windows client systems), and for those who are moving to HTML- or Java-based database client applications, SQL Server provides less built-in functionality and will be harder to use than DB2, Informix, Oracle, or Sybase.

2.10 Summary

This chapter has discussed the concept how speech recognition and synthesis works with the database. The speech technologies have 2 fields, which are recognition and synthesis, that using the different engine. This chapter also discusses how the process works for these 2 engines in flow chart. Further explanation including method or mode of speech recognition and speech synthesis used. For speech recognition are: speaker-dependent or independent, isolated word recognition, continuous speech recognition and keyword spotting. While for speech synthesis are: Formant synthesis and Concatenative synthesis.

During the development, references of the product in the market also involved. IBM Via Voice has been chosen for purpose of comparison with the system will be developed. Study of other product (including IBM) in their type of programming language also needed to help in process development.

The main tools: Microsoft Speech SDK 5.0 also including in these chapter in explanation the structure and interface involved, API (Application Program Interface) and DDI (Device driver Interface). Together with this chapter also introduce some of the programming language in the market for development this speech technology. The introduction of the programming language cover in this chapter including Visual C++ and Visual Basic.

This system mainly focusing on speech technology. Since it involve database, explanation of Microsoft SQL as an example for SQL also cover in this chapter.

Survey in the market and internet also help me notice that currently there is no other similar system that using Malay Language to control the searching from the database that is suitable for the university, while the existing product like IBM Via Voice cost a lot for personal and corporate user. Study the advantages and the disadvantages of the product and technology also helped me to know which technology is suitable use in speech recognition and synthesis. For speech recognition, I will using the Keyword Spotting

technique since it allow user to speak naturally while system still able to work. For speech synthesis I use the Formant Synthesis method due to its flexibility.

Chapter 3
Methodology

Chapter 3
Methodology

3.1 Methodologies In Development System

There are some methodologies to development a system [8]. To name a few, there are:

1. Waterfall Model
2. V-Shape Model
3. Prototyping Model
4. Incremental Model
5. Spiral Model

❖ **Waterfall Model**

This model is well-define development in which one phase has to be finished before the next phase. It is simple to use. Waterfall model is used when the problem is very well understood. Usually, designers use the waterfall model to develop a simple system because it is hard to change if the model is used. If customers come by and ask for changing requirements, designers will have to start from the scratch because there is no fast way to design the system with the new requirement. Also there is a big problem for testing later if there is a change in requirement. The testers do not have well-define set of test cases, so the test is easy to fail. In brief, waterfall model is used if designers have a well-defined list of user requirements.

❖ **V-Shape Model**

This model is similar to waterfall model and the only different is that each test phase matches each development phase: requirements with system testing, high-level design with integration testing, and detailed design with unit testing. V-shape model is an improved version of waterfall model. V-shape model does not run into the problem that the software is impossible to be tested because system test, integration test, and unit test are planed ahead. For example, when we plan the requirement, we

also plan for system testing. Therefore, when the system is built, we have a whole set of test cases for system testing. By that way, the system does meet user requirements.

❖ **Prototyping Model**

This model is the technique, which helps designers and users to clarify the requirement of the system. A testing model or prototype is developed by the designer and evaluated by users and from the feedback, the designer will be able to improve the program when redesign the model. The Prototype model is a good model for the project, which has unambiguous user requirement. The model will help users to understand what they actually want. A throw-away prototype is developed so that users can realize what the system like. The prototype model is also good for deploying the new technology. Before the technology is used, users are interested in know whether the technology works or not. Therefore, the prototype is a neat way to demonstrate the idea to users or customers.

❖ **Incremental Model**

This model needs the designer to develop the software in a number of stages and is able to deliver the product early. At each phase the designer have a goal to deliver certain feature to customer. Incremental model is good for fast delivering product to the marker place. Incremental model has many advantages over the other techniques. One of advantages is that the system can be developed at several stages. Each stage has its own requirement; usually it has certain features or core of the system. Each stage can use V-shape, prototype or waterfall model to develop the requirement for this stage. Regardless what kind of model is used in each stage, the product with certain features must be done at the end of the stage. Incremental model satisfies the requirement of fast delivery to the market place, so business people are interested in this model.

❖ **Spiral Model**

This model is using an interactive method. The model carefully takes risks into account. The designers develop a small part of the project and evaluate the risks. If the risk is low, designers keeps developing more features. Spiral model is heavily involved in risks management. If designer have a project with a very high risks, the designer should use spiral model. Every iteration, designer has a chance to evaluate the risks and to forecast whether the project keeps going or stops. For each iteration, similar to incremental model stage, designer can use V-shape, prototype, or waterfall. Spiral model is usually used in the large project such as financial system, wireless cellular communication management system, network management system.

Waterfall model, V-shape model, and prototyping model are three basic models, which can be used, in small project. Incremental model and spiral model are usually used in the large project. When the large project is divided into well-defined small project (or phase or stage or iteration), a small project can use waterfall, V-shape, or prototyping model.

3.1.1 Explanation of Waterfall Model

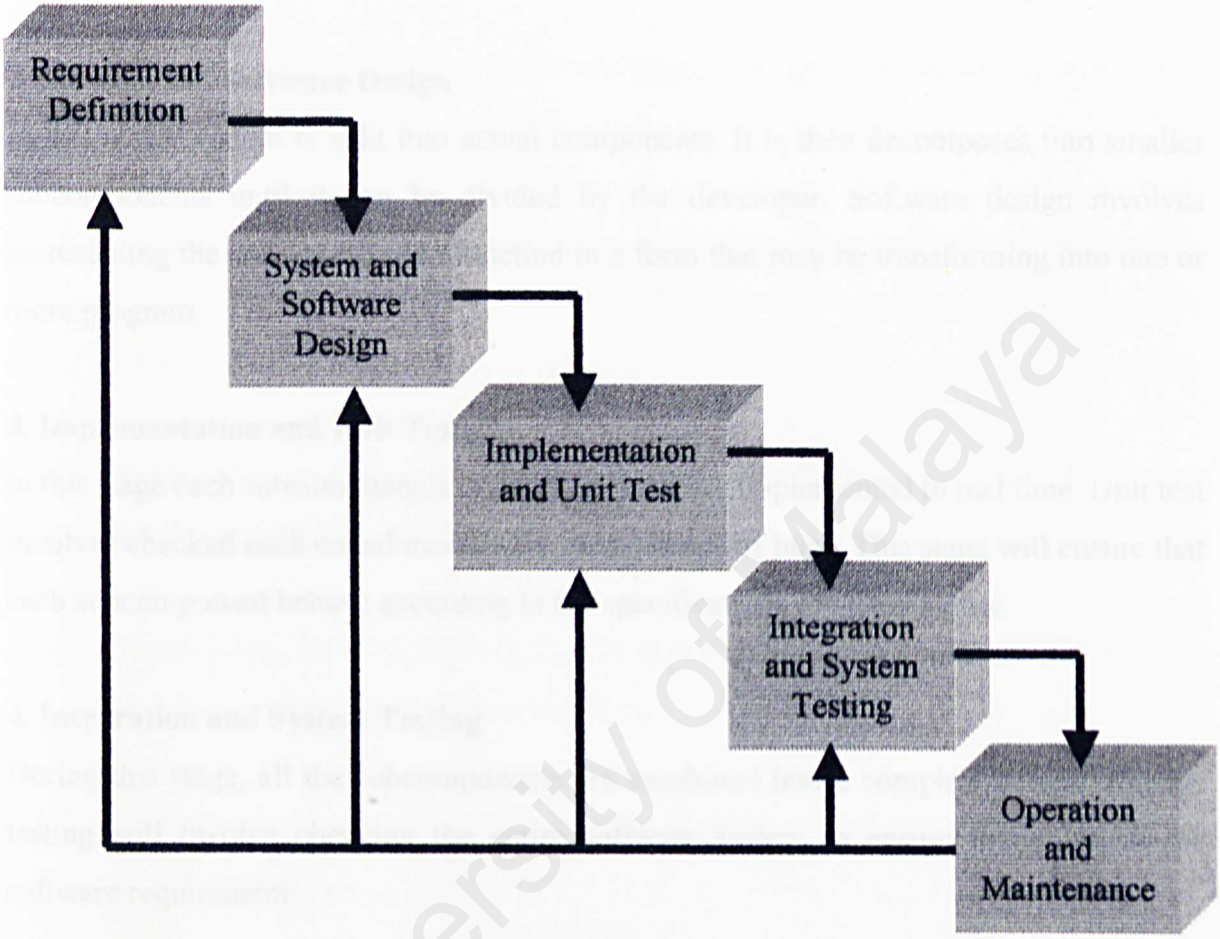


Figure 3.1 Waterfall Model

1. Requirements and Definition

The requirement includes analyzing the software problems. Complete of the software specification of the design external behavior of the software system are built in this stage.

2. Systems and Software Design

In this stage, system is split into actual components. It is then decomposes into smaller subcomponents until it can be divided by the developer. Software design involves representing the software system function in a form that may be transforming into one or more program.

3. Implementation and Unit Testing

In this stage each subcomponents or program units are implemented in real time. Unit test involves checked each coded module for the presence of bugs. This stage will ensure that each subcomponent behave according to the specifications as stated before.

4. Integration and System Testing

During this stage, all the subcomponents are combined into a complete system. System testing will involve checking the entire software system to ensure that it fulfill the software requirement.

5. Operations and Maintenance

Maintenance means continue detecting and repairing bugs after deployment. It will improve the implementation of the subcomponents and enhancing the system services as new requirement are discovered.

3.2 Justification of Development Methodology

After define the method above, Waterfall Model is the most suitable for develop this system. It is because:

1. This system has 3 phases, which are speech recognition, speech synthesis and database.

2. Develop for speech recognition and synthesis needs to be separate for the first stage and only join together when two different engines can run according to user need.

3. Before the speech synthesis combine together with the speech recognition, testing need to do for each phase.

4. Development for database will join the speech system (recognition and synthesis) when the database successfully builds up.

5. When the database built up, testing also needed in a normal format without the speech system.

6. After the combination of all the phase, again process testing will test to see any error occur.

7. Any requirement to improvement the system will make the system to test again and check for new bugs.

3 Information Gathering

fore design this Speech Recognition and Speech Synthesis Repository System Rn'SSRS), the first task for me to do is to collect the information on how the similar hnology worked. That's information is for me to know about the process involved and e tools to be use. I also collect information about the example coding provided by the ols in order to help me understand the explanation of the system deeper. Information ught for this project from 3 main resources, there are:

1. Internet
2. Bilik Dokumen from FSKTM (Fakulti Sains Komputer dan Teknologi Maklumat)
3. References Books

hese 3 methods have helped me a lot in gather related information regarding knowledge f programming, tools and database.

3.3.1 Internet

Now a day, a lot of information flooding in the cyberspace. To get related information is easier than before. But due to so many information in the Internet, getting relevant information become harder. Therefore, a good search engine need to help to solve this problem. The searching that I use is:

1. Google (http://www.google.com)
2. MSN (http://www.msn.com)
3. Yahoo! (http://www.yahoo.com)

The reason to use search engine:

1. Find out the existing product
2. Find out the software available
3. Find out the programming tools for development
4. Find out step in development a good system
5. Get help on the Internet from newsgroup and references

3.3 Information Gathering

Before design this Speech Recognition and Speech Synthesis Repository System (SRn'SSRS), the first task for me to do is to collect the information on how the similar technology worked. That's information is for me to know about the process involved and the tools to be use. I also collect information about the example coding provided by the tools in order to help me understand the explanation of the system deeper. Information sought for this project from 3 main resources, there are:

1. Internet
2. Bilik Dokumen from FSKTM (Fakulti Sains Komputer dan Teknologi Maklumat)
3. References Books

These 3 methods have helped me a lot in gather related information regarding knowledge of programming, tools and database.

3.3.1 Internet

Now a day, a lot of information flooding in the cyberspace. To get related information is easier than before. But due to so many information in the Internet, getting relevant information become harder. Therefore, a good search engine need to help to solve this problem. The searching that I use is:

1. Google (http://www.google.com)
2. MSN (http://www.msn.com)
3. Yahoo! (http://www.yahoo.com)

The reason to use search engine:

1. Find out the existing product
2. Find out the software available
3. Find out the programming tools for development
4. Find out step in development a good system
5. Get help on the Internet from newsgroup and references

3.3.2 Bilik Dokumen

Bilik Dokumen also is another source for me to gather information. Although I can't find any related topic for my system, I do get some of the information regarding development methodology and the documents they're also giving me a brief idea in develop a system. The documents there also help me getting information in choosing related programming with the analysis provided.

3.3.3 References Books

References books that I refer were the one related with my system and can giving me a quick understanding on time. The references are programming and database references books.

3.4 Development Tools

After gather information, I need to decide which programming language and tools to use for development purpose. For programming language, after trying Visual Basic and Java, I found that I need to choose the tools for development. It is because different development tools support different language and giving different information and help in the language they prefer. I have try the Chant's SpeechKit, Microsoft Speech SDK5.x and Microsoft Platform SDK 2000, I decided to use Microsoft product as the tools to help me. The reason for me to choose it because:

1. These Microsoft product (Microsoft Speech SDK5.x and Microsoft Platform SDK 2000) are free to download. Although the size of the programs were big, but I was able to download it from faculty that provided faster Internet connection speed.

2. Microsoft product object oriented programming like Visual Basic and Visual C++.

3. Chant's SpeecKit also giving free download service but I only giving 60 days of development. It was not enough for me since I need to take days to familiar with

the tools and method it given. These problems will only solved if I pay for minimum of $29 (RM 120) [11].

Since the tools I choose is Microsoft product and it support Visual basic and Visual C++, I decided to use Visual C++ because the help comes with Visual C++ explanation and focusing on that language. Another reason I using Visual C++ due to its object oriented programming, which enable programmers to create modules that do not need to be changed when a new type of object is added. A programmer can simply create a new object that inherits many of its features from existing objects. This makes object-oriented programs easier to modify.

I also need to learn XML (eXtensible Markup Language) and COM (Component Object Model)[18] for speech recognition and synthesis purpose. Introduction and reason of using XML and COM will explain a short well. I also decided using HTML as part of creating the help file.

As for database tools, I have try Microsoft Access 2000 just to see the environment of database. But since the system need to be able to display as user request and support a large database, so I decided to use Microsoft SQL so the system can perform, as user likes to. I have also gather some information from the faculty staff in order to see the currently display method of student information. The reason using SQL because:

1. It is easy to learn as its syntax is similar of the English language
2. It uses a free format
3. Several people such as end-users, database designers and database administrators can use it.
4. It is a non-procedural language. That means users can specify what information that want, rather than how to get that information.

The reason of choosing Microsoft Access SQL because:

1. This system is using a Microsoft Windows platform to develop. So it is wisely to use the program that compatible with the operating system

2. Microsoft supports ODBC (Open Database Connectivity, a Microsoft standard defining how Windows applications access database data) and OLE DB (Object Linking and Embedding Database), a new Microsoft standard defining how applications access database data) because the system to develop will involved 1 of these 2 technologies [3].

3. Microsoft SQL can run in Windows 9x and above. It did help the developer to get less trouble in order to upgrade to Window 2000.

Its user-friendly environment help beginner user who familiar with the Windows to learn faster.

3.4.1 XML

XML is subset of the Standard Generalized Markup Language (SGML) defined in ISO standard 8879:1986 that is designed to make it easy to interchange structured documents over the Internet. XML files always clearly mark where the start and end of each of the logical parts (called elements) of an interchanged document occurs. XML restricts the use of SGML constructs to ensure that fallback options are available when access to certain components of the document is not currently possible over the Internet. It also defines how Internet Uniform Resource Locators can be used to identify component parts of XML data streams [4,6,14,15,16].

By defining the role of each element of text in a formal model, known as a *Document Type Definition* (DTD), users of XML can check that each component of document occurs in a valid place within the interchanged data stream. An XML DTD allows computers to check, for example, that users do not accidentally enter a third-level

heading without first having entered a second-level heading, something that cannot be checked using the Hypertext Markup Language (HTML) previously used to code documents that form part of the World Wide Web (WWW) of documents accessible through the Internet.

However, unlike SGML, XML does not require the presence of a DTD. If no DTD is available, either because all or part of it is not accessible over the Internet or because the user failed to create it, an XML system can assign a default definition for undeclared components of the markup.

XML allows users to:

- Bring multiple files together to form compound documents
- Identify where illustrations are to be incorporated into text files, and the format used to encode each illustration
- Provide processing control information to supporting programs, such as document validations and browsers
- Add editorial comments to a file.

It is important to note, however, that XML is not:

- A predefined set of tags, of the type defined for HTML, that can be used to markup documents
- A standardized template for producing particular types of documents.

XML was not designed to be a standardized way of coding text: in fact it is impossible to devise a single coding scheme that would be suit all languages and all applications. Instead XML is formal language that can be used to pass information about the component parts of a document to another computer system. XML is flexible enough to be able to describe any logical text structure, whether it is a form, memo, letter, report, book, encyclopedia, dictionary or database.

Why the system needs to use XML? It is because SAPI uses XML content in the following two methods [19]:

1. The SAPI context-free grammar compiler compiles the XML grammar into a binary grammar format. The compiled binary grammar is loaded into the SAPI runtime environment from a file, memory, or object (.DLL) resource.
2. The speech recognition (SR) engine queries the runtime environment for available grammar information.

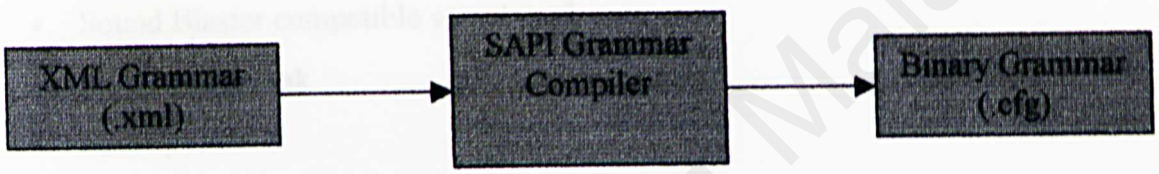


Figure 3.2 How XML grammars compiles to binary grammar

To makes thing clear, even though XML have a relationship with HTML and web pages, but when using the XML to develop the speech system, the format that SAPI 5.0 uses is ***not*** placed inside web pages. SAPI application that is synthesizing text from a web page will "render" HTML+ACSS into SAPI's synthesis markup format. Programs apply a default ACSS file when synthesizing web pages that do not have an associated ACSS file.

3.4.2 COM

COM is a model for binary code developed by Microsoft. The Component Object Model (COM) enables programmers to develop objects that can be access by any COM-compliant application. This architecture can use for defining interfaces and interaction among objects implemented by widely varying software applications. A COM object

instantiates one or more interfaces, each of which exposes zero or more properties and zero or more methods. All COM interfaces are derived from the base class IUnknown. Technologies built on the COM foundation include ActiveX, MAPI and OLE [14,15].

3.5 Hardware Requirement

3.5.1 Development Environment

- AMD TunderBird 1.1 GHz Processor
- 256MB RAM
- 32M Graphic Card
- 17" inch Monitor with 1024 X 768 resolution
- Sound Blaster compatible sound card
- 20 GB Hard Disk
- Microphone
- Speaker

3.5.2 Runtime Environment

- A Pentium-equivalent or later processor at 200 MHz or higher
- 32 MB RAM (Recommend 128MB RAM)
- Around 80 MB of hard disk space
- Minimum 15" inch monitor
- Sound Blaster compatible sound card
- Microphone
- Mouse
- Speaker

3.6 Software Requirement

3.6.1 Development Environment

- Microsoft Visual C++ 6 with Service Pack 3 or later version
- Microsoft Speech SDK 5.0
- Microsoft Platform SDK April 2000 or later version
- Microsoft Access 2000
- Microsoft Windows 98 Second Edition
- Microsoft Office 2000
- InstallShield 5.5 Professional Edition
- Microsoft Internet Explorer 5.0 or Netscape 4.7x

3.6.2 Runtime Environment

- Window 98 and later, Window NT and Window 2000
- Database Program with ODBC support
- Microsoft Internet Explorer 5.0 or Netscape 4.7x
- Microsoft Visual C++ 6 with Service Pack 3 or later version
- Microsoft Speech SDK 5.0 or later

3.7 Summary

This chapter has explained some of the methodologies in development a system. Using Waterfall Model is a suitable for development in Speech Recognition and Speech Synthesis Repository System" because this system needs to break to 3-sub system and then break again to sub module. It is good since all the requirement and study of the system have been done before starting the design phase. With this systematic and planning, the system can be done more quickly and any changes will only involved in improvement of the system.

Method of information gathering also touches in this chapter. The sources I get mainly fro Internet, Bilik Dokumen and references book.

Programming tools used in development this system also explain here. IBM ViaVoice SDK 8.0, SpeechKit and Microsoft Speech SDK 5.0 are three development tools used to compare each other. Microsoft Speech SDK5.0 has been chosen since it is free and all the development environment process will be test in Windows environment. Using Microsoft product will help in development because stable and compatible issue wills no more an important problems. It also helps the developer to check or debug the system correctly when notice the program cannot run properly. Developer only need to check the script and time for checking will be reduce.

Since I have choosing the Microsoft Speech SDK5.0 and Microsoft Platform SDK 2000, I need to use Visual C++, XML and COM for development my system. XML and COM also introduced in this section.

For the database part, Microsoft SQL has been chosen. Microsoft SQL is one of the powerful, multi functional database management tools that help user to management and control the large system easily and its user-friendly interface helped beginner user to learn faster.

For the software and hardware requirement, those references are from Microsoft Speech SDK5.0 help file. Anyway, process speech recognition and speech synthesis require different of minimum resource in terms of processor power and memory. The list here is only a references and base on the more hungry resource, which is speech recognition.

Chapter 4
System Design

Chapter 4
System Design

4.1 System Design

Speech Recognition and Speech Synthesis Repository System (SRn'SSRS) can be divided to 2 main programs:

1. Speech application
2. Database

Further, the speech application has 2 main functions, as showed in figure 4.1, which are:

1. Speech recognition function (engine)
2. Speech synthesis function (engine)

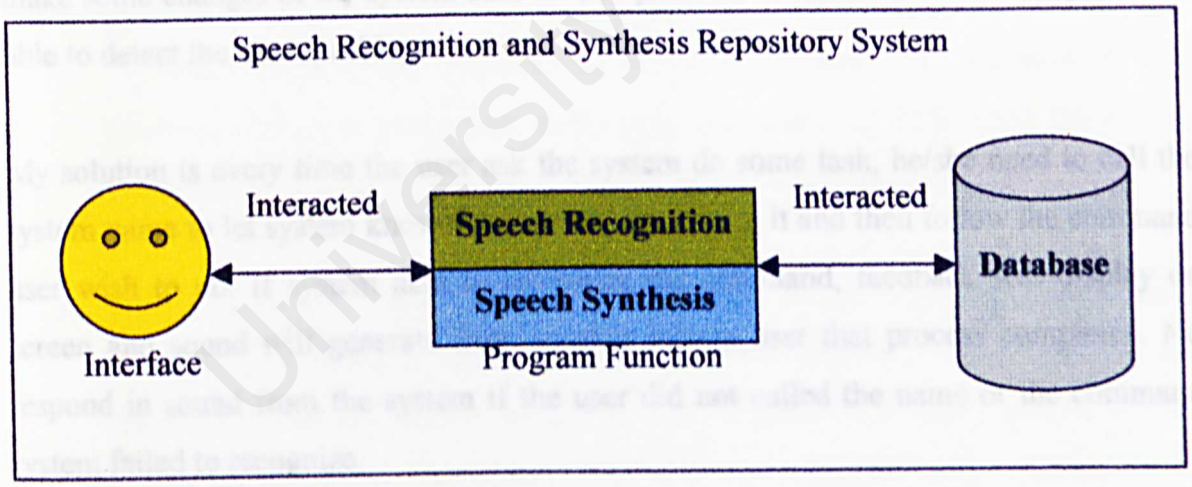


Figure 4.1 System overview

4.2 Design Overview

The system design can be divided to 3 parts, which are:

1. Speech recognition program design
2. Speech synthesis program design
3. Database design

The design basically start with create user interface and after that only move to create database function. This system function will be added with the help of Microsoft Speech SDK5.x, Microsoft Platform SDK and Microsoft Visual C++.

4.3 Program Structuring Design

To make the system work well, I have tried some of the example provided Microsoft Speech SDK5.x. I found that the program example was responded sensitively until the sound of fan will make the system respond in the way user never want to. I decided to make some changes of my system base on this problem to reduce the sensitively but still able to detect the user speaking.

My solution is every time the user ask the system do some task, he/she need to call the system name to let system know that he/she is talking to it and then follow the command user wish to do. If system able to recognize the command, feedback will display on screen and sound will generate from speaker inform user that process completed. No respond in sound from the system if the user did not called the name or the command system failed to recognize.

4.3.1 Program Structuring

The design structuring will only can understand with the helped of structure chart. With the help of it, process design becomes easier to develop because it can display the module involved in the development phase.

For the SRn'SSRS, the program structure is as showed in figure 4.2:

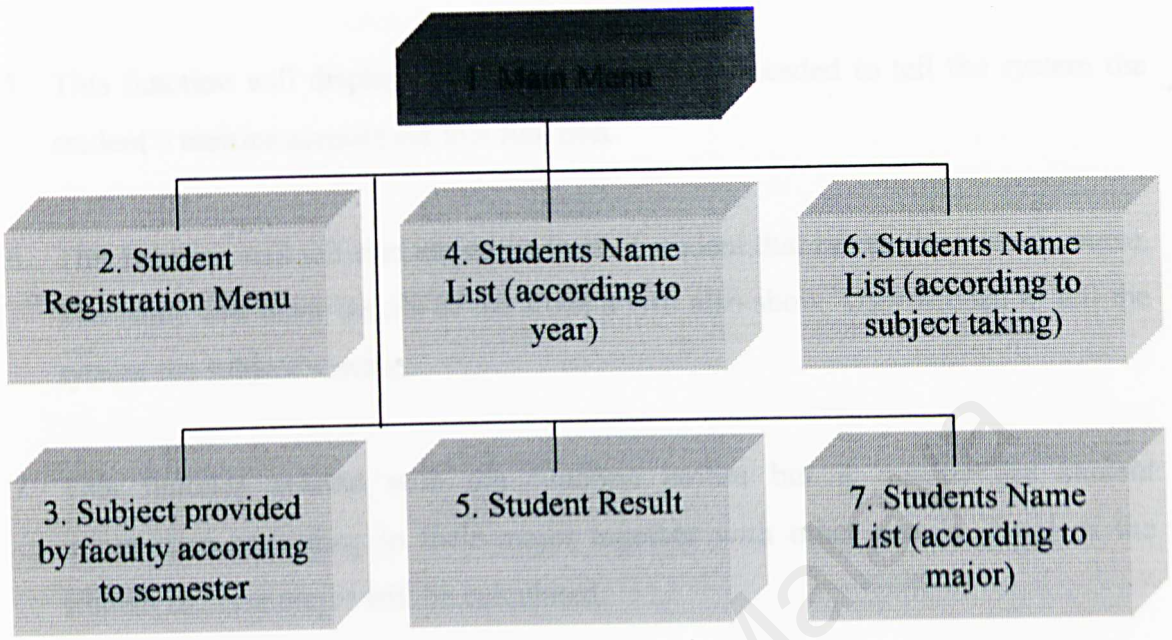


Figure 4.2 Program Structure Chart

Explanation:

1. The main menu is the introduction interface for user who activated it. This interface will greet to the user every time user launched it. It listed the 6 main functions for user to control either via speech or mouse pointing.

2. This function will show the student information related with the courses taken. The data including year, semester, metrics number, student name, IC, address and the courses that the student took. Fro searching the data, user needs to tell the system the student metrics number.

3. This function provided information about which subject will be provided for which semester. User needed to tell the system the year and semester for searching.

4. This function lists the entire student in faculty according the year intake. User needs to tell the system which year he/she wants to keep track.

5. This function will display the student result. User needed to tell the system the student's metrics number for this function.

6. This function will tell user about the total of student that taking the related course. The name and other details of the student will also show. User needed to tell the system the subject's name.

7. This function related with the function before but it display the student information according to their major together with other details. Total of the student in same major will be calculated.

4.4 User Interface Design

A good user interface is one of the design objectives that contain consistent, user friendly, easy to use and precision. Some of the guide has proposed to reach this stage:

1. User interface design must be attractive and fulfill user requirement. For this system, due to speech technology require an amount of memory and processor power, so I will only comes with a normal interface without a lot of animation and colouring. Bt the function should fulfill the user needed.

2. User interface needed to be easy to understand and easily saw by the user. For this system, it will use Malay language and everybody understands that.

3. The icon should be consistent and terms of colour and words.

4. The confirmation's message should respond with the function that has been chosen. This system will tell user when a process successfully done with a voice and no respond to the user command when it failed to recognize the command.

5. Avoid using short form for the button. The system button comes with the explanation beside the button (can be seen in the main menu).

6. The button and the function should be arranged in a tidy way. Example is create a function tool bar for locate the function. This system will put the entire important button like open or print in a toolbar and left no other function outside of it.

7. Opposite function should put separate. One of the examples is the delete and save button need to separate by either function.

8. The interface should be able to support different type of input device, like keyboard and pointing device. This system support speech and pointing device.

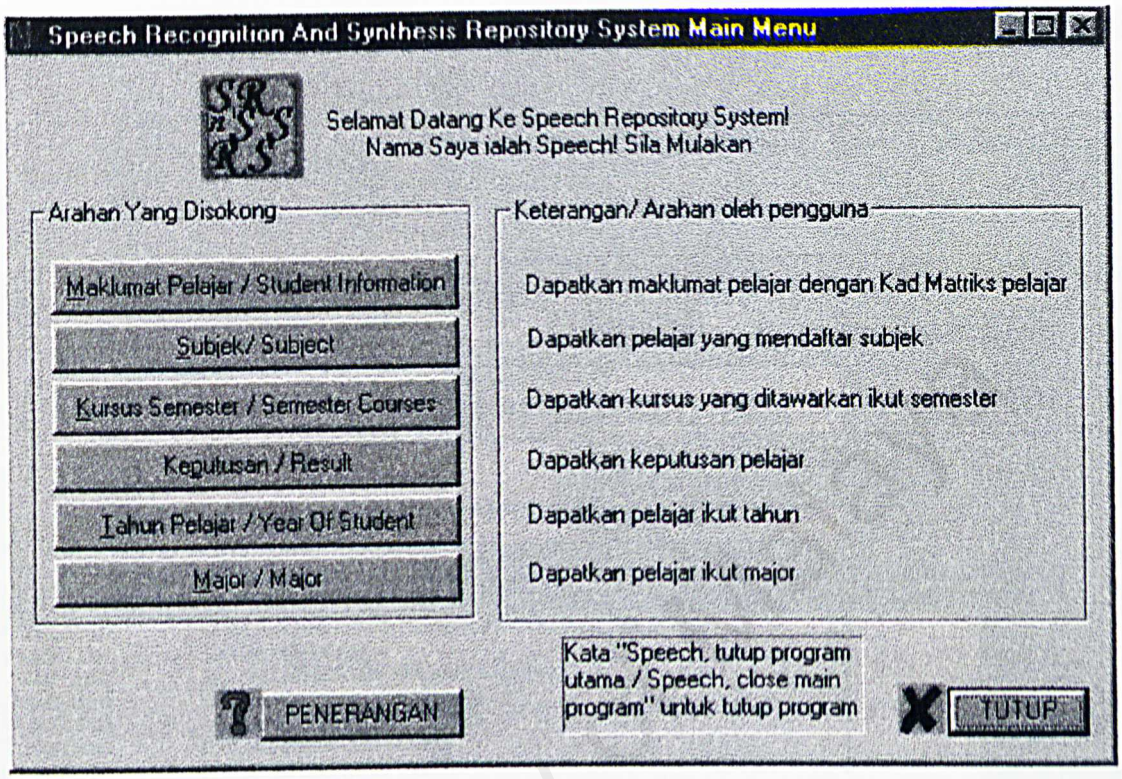


Figure 4.3 System's Main Menu

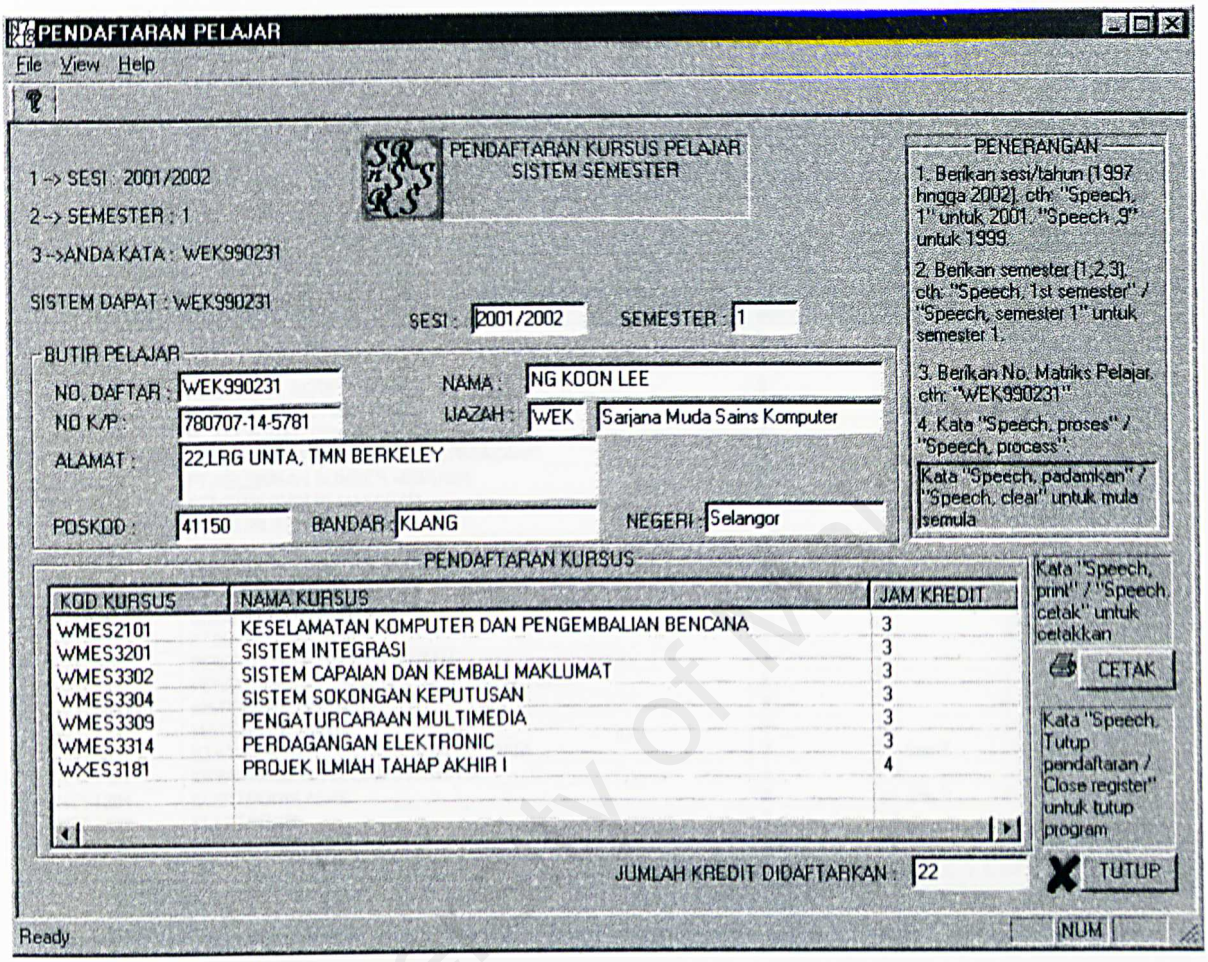


Figure 4.4 System's Student Information Menu

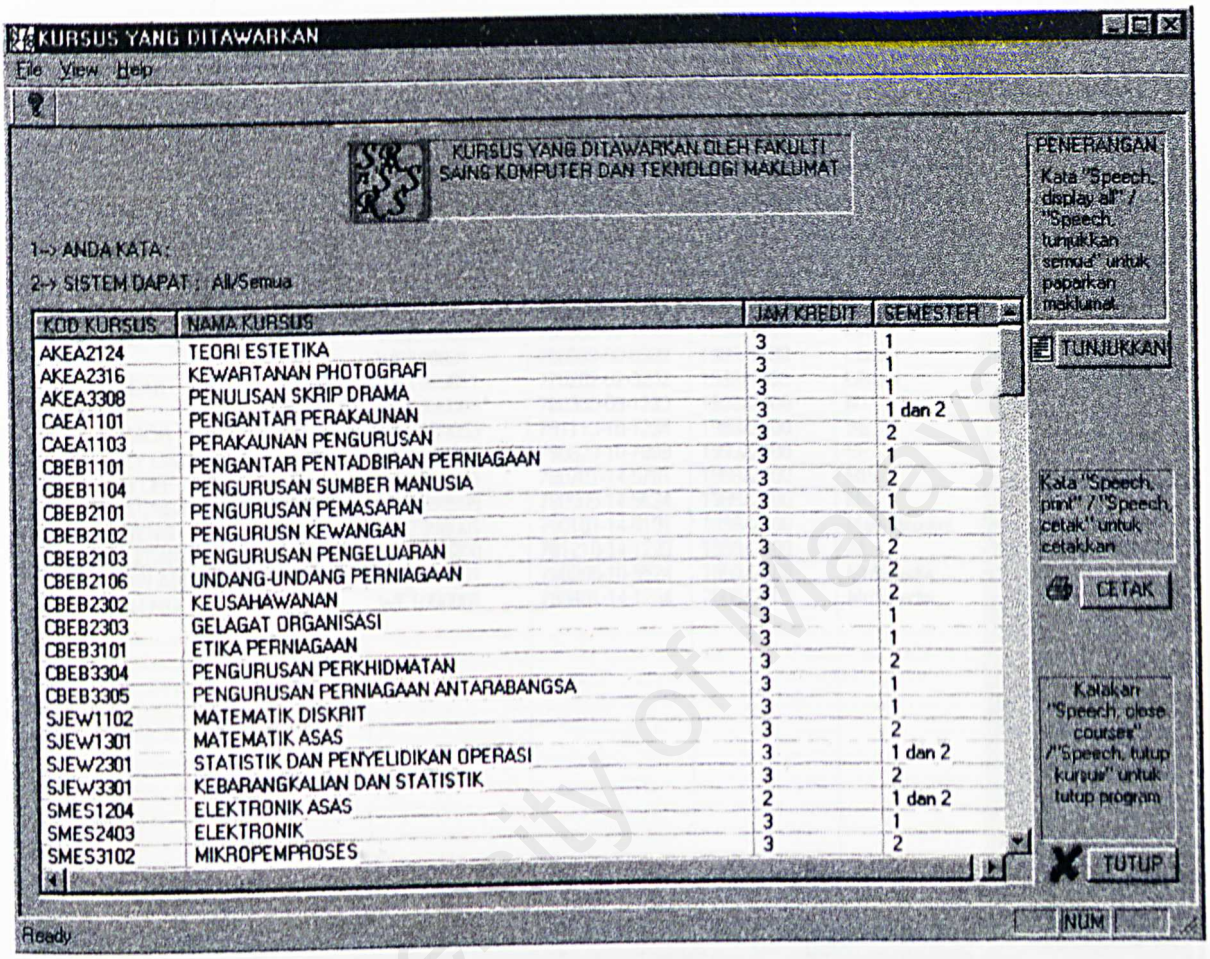


Figure 4.5 Subject provided by faculty according to semester

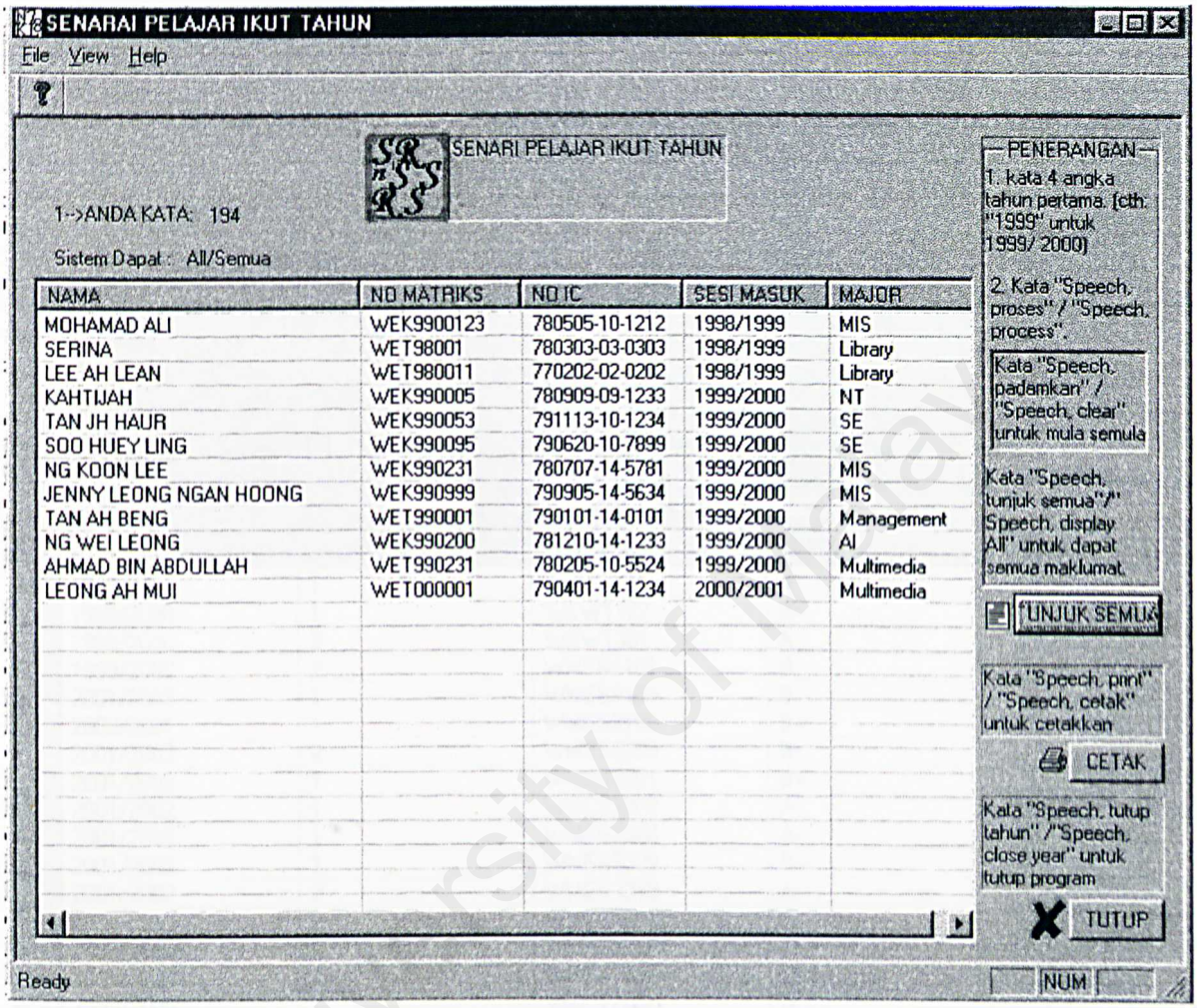


Figure 4.6 Students Name List (according to year)

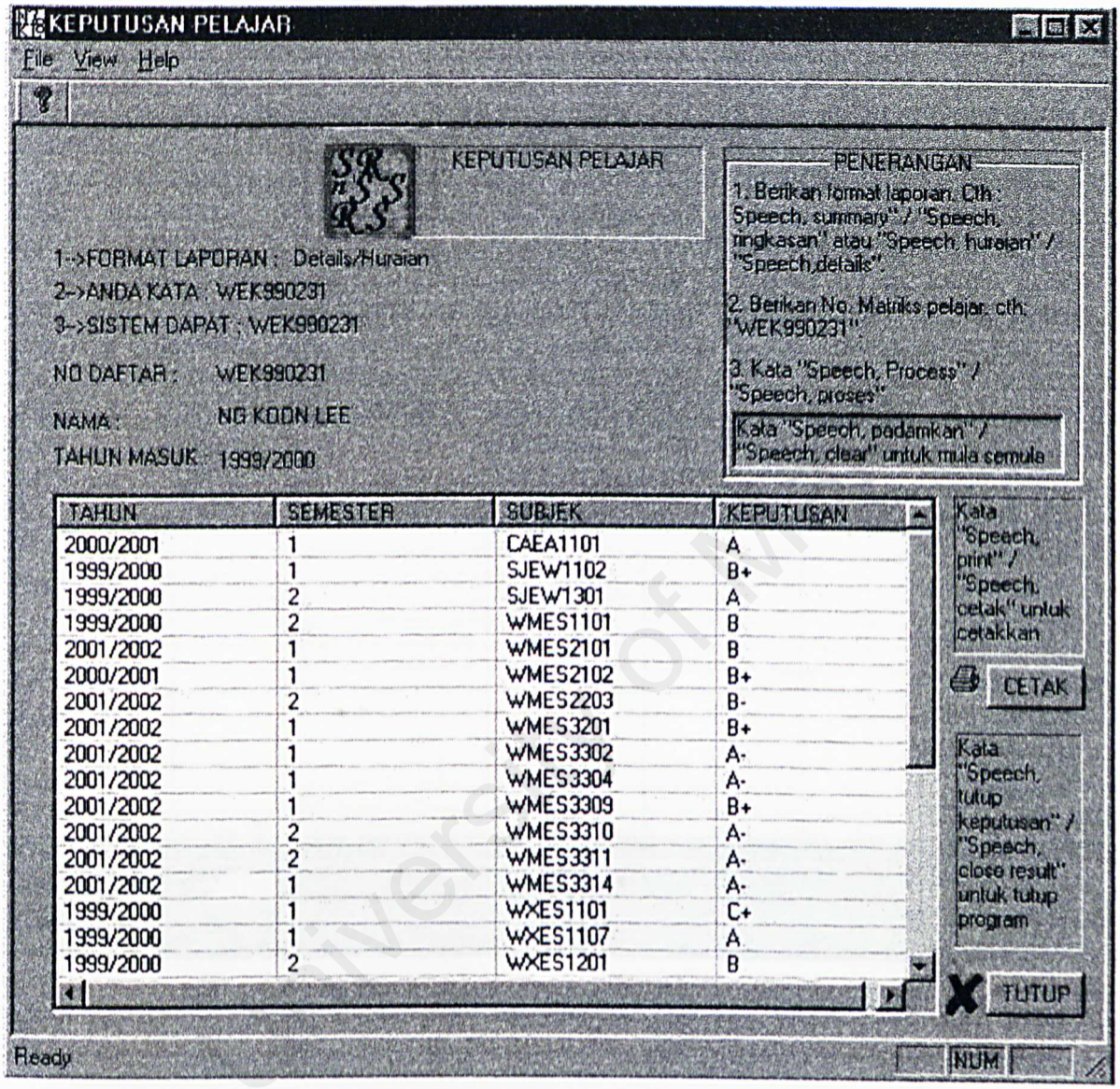


Figure 4.7 Student Result Menus

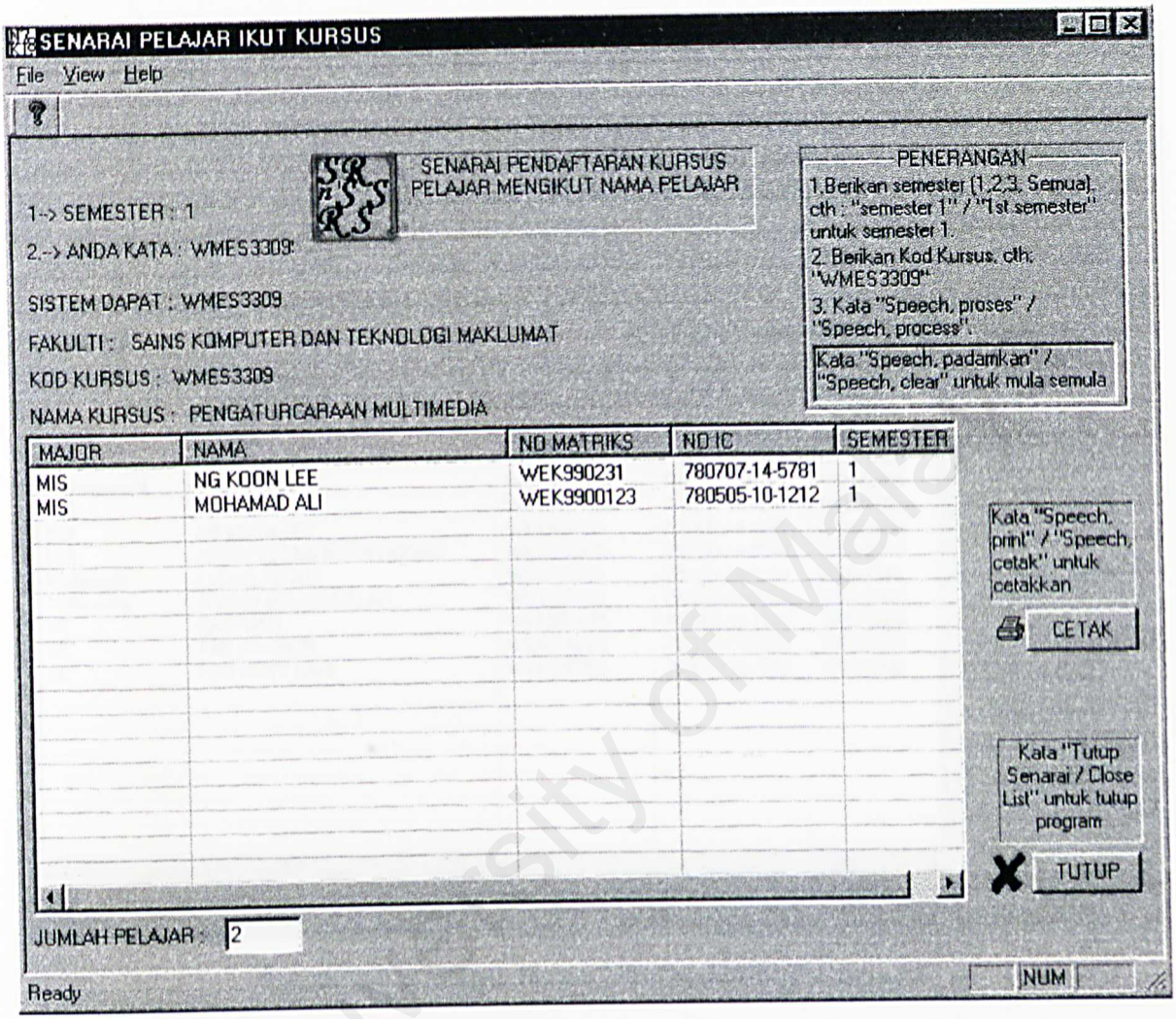
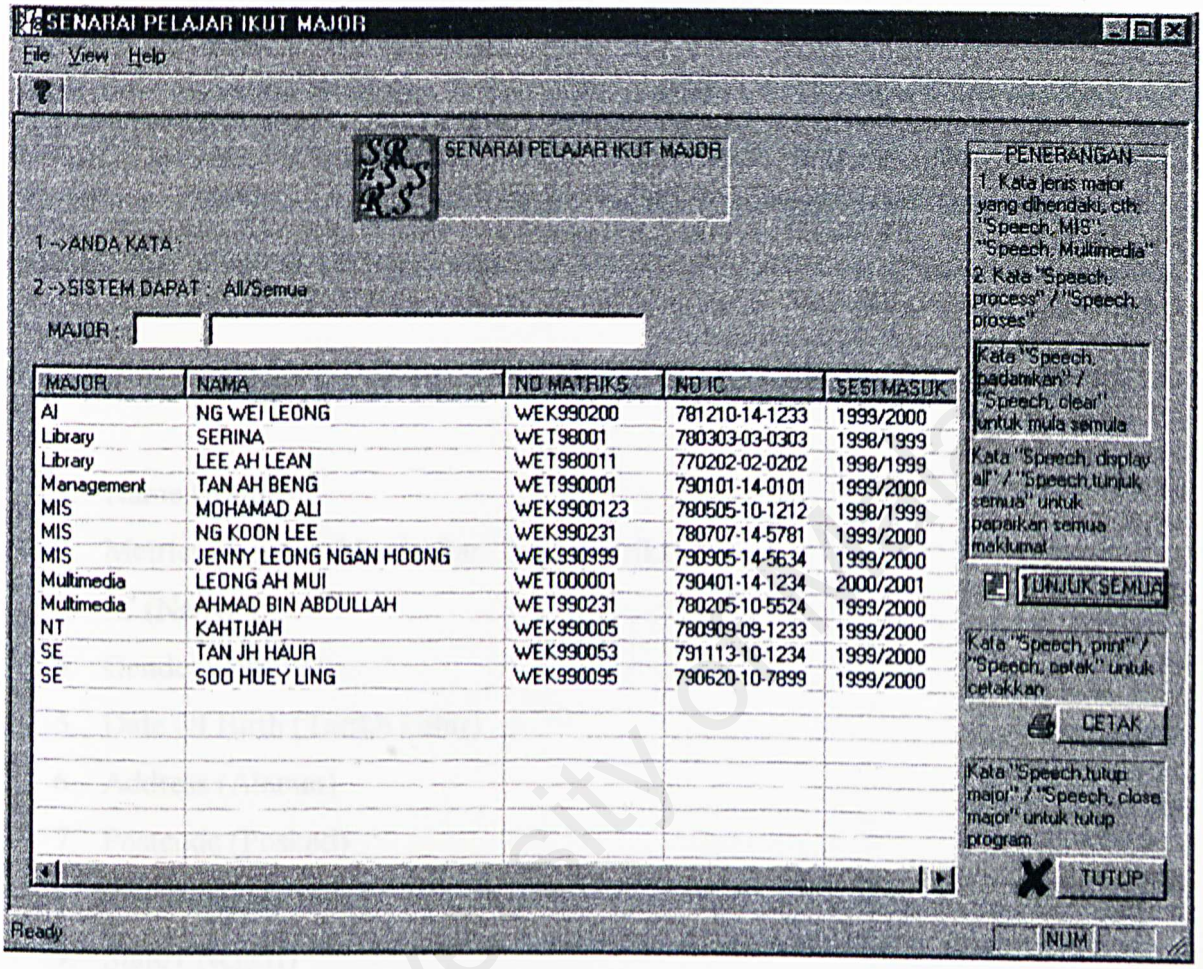


Figure 4.8 Students Name List (according to subject taking)



MAJOR	NAMA	NO MATRIKS	NO IC	SESI MASUK
AI	NG WEI LEONG	WEK990200	781210-14-1233	1999/2000
Library	SERINA	WET98001	780303-03-0303	1998/1999
Library	LEE AH LEAN	WET980011	770202-02-0202	1998/1999
Management	TAN AH BENG	WET990001	790101-14-0101	1999/2000
MIS	MOHAMAD ALI	WEK9900123	780505-10-1212	1998/1999
MIS	NG KOON LEE	WEK990231	780707-14-5781	1999/2000
MIS	JENNY LEONG NGAN HOONG	WEK990999	790905-14-5634	1999/2000
Multimedia	LEONG AH MUI	WET000001	790401-14-1234	2000/2001
Multimedia	AHMAD BIN ABDULLAH	WET990231	780205-10-5524	1999/2000
NT	KAHTIJAH	WEK990005	780909-09-1233	1999/2000
SE	TAN JH HAUR	WEK990053	791113-10-1234	1999/2000
SE	SOO HUEY LING	WEK990095	790620-10-7899	1999/2000

Figure 4.9 Students Name List (according to major)

4.5 Database Design

Database design consists of table design. These tables keep data in the database and are referred to as metrics consisting of a series of column and row intersection. Table, also known as relation, are related to each other by sharing a common entity characteristic. There are 4 important table uses in this system, which are STUDENT_INFO (Table 4.2), STU_CONTACT_INFO (Table 4.3), COURSES_INFO (Table 4.4) and RESULT_INFO (Table 4.5). The following section below is my consideration of information of the database that will store for my system:

Note: words in bracket are the Malay Words that will use in the program interface

1. Name (Nama)
2. Metrics Number (No. Daftar / Kad Matriks)
3. IC (No KP)
4. Gender (Jantina)
5. Date Of Birth (Tarikh Lahir)
6. Address (Alamat)
7. Postcode (Poskod)
8. City (Bandar)
9. State (Negeri)
10. Phone Number (No. Telefon)
11. Year Intake (Sesi Masuk)
12. Major (Major)
13. Year (Sesi Pendaftaran / Sesi)
14. Semester (Semester)
15. Courses Taking (Nama Ijazah)
16. Subject Code (Kod Kursus)
17. Subject Name (Nama Kursus)
18. Credit Hour (Jam Kredit)
19. Result (Keputusan)

Table 4.1 list the field's name needed by the function.

Function	Field Name
Student Registration Menu	Name, Metrics Number, IC, Gender, Date Of Birth, Address, Postcode, City, State, Phone Number, Year Intake, Major, Year, Semester, Courses Taking, Subject Code, Subject Name, Credit Hour
Subject provided by faculty according to semester	Year, Semester, Subject Name, Subject Code, Credit Hour
Students Name List (according to year)	Year Intake, Name, Metrics number, IC, Gender, Major
Student Result	Metrics Number, Name, Result, Year, Semester
Students Name List (according to subject taking)	Subject Code, Subject Name, Year, Semester, Name, Metrics Number, Gender, IC, Year Intake,
Students Name List (according to major)	Major, Name, Metrics Number, IC, Gender, Year Intake,

Table 4.1 Field name in the function

A) STUDENT_INFO

This table contain only the student information in their name, metrics number, IC, date of birth, and year intake.

Field Name	Data Type	Description
Name	Text (50)	Student's name
Metrics Number	Text (15)	Student's metrics number
IC	Text (20)	Student Identification card number
Gender	Text (10)	Student's gender
Date Of Birth	Text (10)	Student's date of birth
Year Intake	Text (20)	The year student start study in

		university

Table 4.2 STUDENT_INFO

B) STU_CONTACT_INFO

This table contents the student contact information, which included: name, metrics number, address, postcode, city, state and phone number.

Field Name	Data Type	Description
Name	Text (50)	Student's name.
Metrics Number	Text (15)	Student's metrics number.
Address	Text (100)	Student's permanent address as shown in IC.
Postcode	Number (10)	Address's postcode.
City	Text (30)	Name of city as shown in the student IC (example: Klang).
State	Text (20)	Name of state as shown in the student IC (example: Selangor).
Phone Number	Text (15)	Student contact number (home).

Table 4.3 STU_CONTACT_INFO

C) COURSES_INFO

This table contents all the courses information in the faculty, which included: subject code, subject name, major, courses taking, year, semester and credit hour.

Field Name	Data Type	Description
Subject Code	Text (10)	Subject code according to the student registration book (example WXES 3181)

Subject Name	Text (50)	Subject name according to the student registration book (example: Latihan Ilmiah Tahap Akhir I).
Major	Text (50)	Student major (example: MIS, NT)
Courses Taking	Text (50)	Either in Science Computer or Information Technology
Year	Text (10)	The register year for current subject.
Semester	Text (10)	The register semester for current subject
Credit Hour	Text (10)	The subject credit hour

Table 4.4 COURSES_INFO

D) RESULT_INFO

This table related to student passes result that content their name, metrics number, year, semester and result.

Field Name	Data Type	Description
Name	Text (50)	Student's name.
Metrics Number	Text (15)	Student's metrics number.
Year	Text (10)	The register year for current subject.
Semester	Text (10)	The register semester for current subject
Result	Text (5)	Result for 1 semester

Table 4.5 RESULT_INFO

4.5.1 Data Flow Diagram

Data flow diagrams are part of a structured model in the development of software. Data flow diagrams are a graphical technique that depicts information flow and the transforms that are applied as data move from input to output.

Basically, the function of Data Flow Diagrams is to show the user a graphical analysis of a software system. It is kind of like a flowchart, except Data Flow Diagrams show the flow of data throughout the system

The table below shows the basic of notation of Data Flow diagram (DFD)

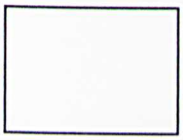
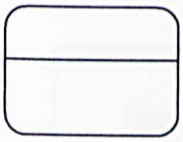
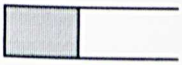
Symbol	Name	Description
	Source or destination of data	It is used to depict an external sources that interact with system but are outside the boundary.
	Process	It is used to show the occurrence of a transforming process.
	Data store	It is used to show where data is stored or referenced by a process.
	Data flow	It is used to show movement of data from one point to another, with head of arrow pointing toward the data's destination

Table 4.6 Basic DFD Notation

Data Flow Diagrams are also broken up into levels. The designer begins with a Level 0 (which is the general software system). The example below shows a simple example of a Level 0 Data Flow Diagram. Based on how many inputs and outputs (processes or number of circles) a system may have will determine how many levels a user will have to go to get a detailed enough design of the software system. Normally, a designer will go

three levels deep in the Data Flow Design. At Level 0 no details about the system are given, just inputs and outputs. From here the system would be broken down further and more specifically in Levels 1,2,3. Levels 1,2,3 are the same notation as Level 0, except they are more specific and give more details about the system.

Below show the level 1 system's data Flow diagram:

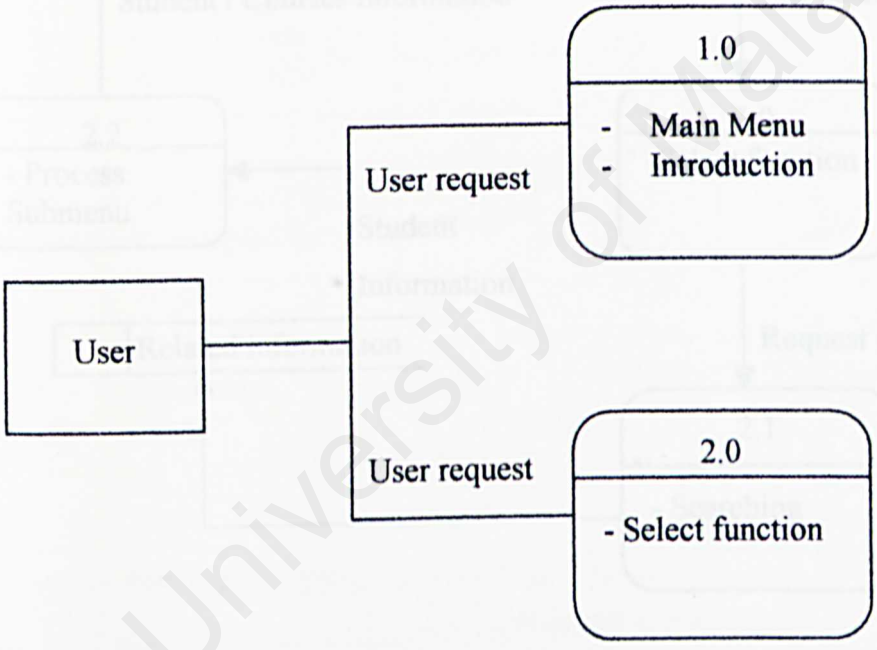


Figure 4.10 Level 1 Data Flow Diagram

Below shows level 2 system's Data Flow Diagram

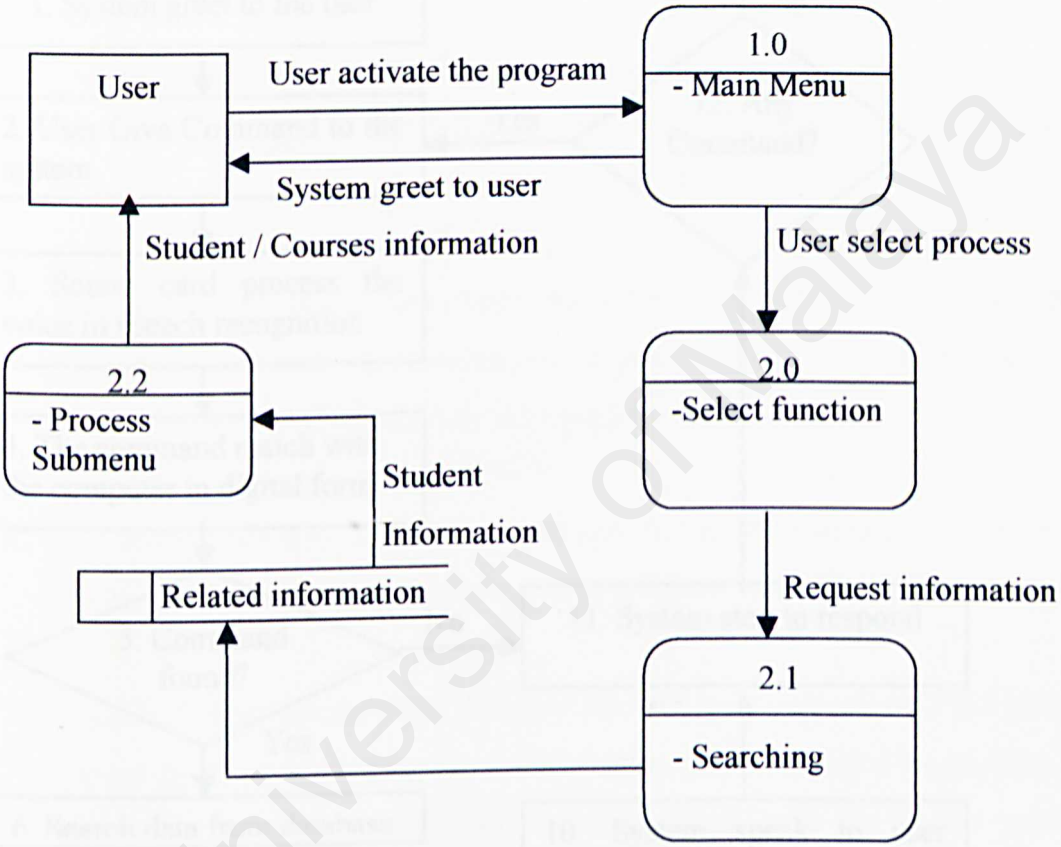


Figure 4.11 level 2 Data Flow Diagram

4.6 How System Works

Figure 4.12 show how the full system works when integrated with the database.

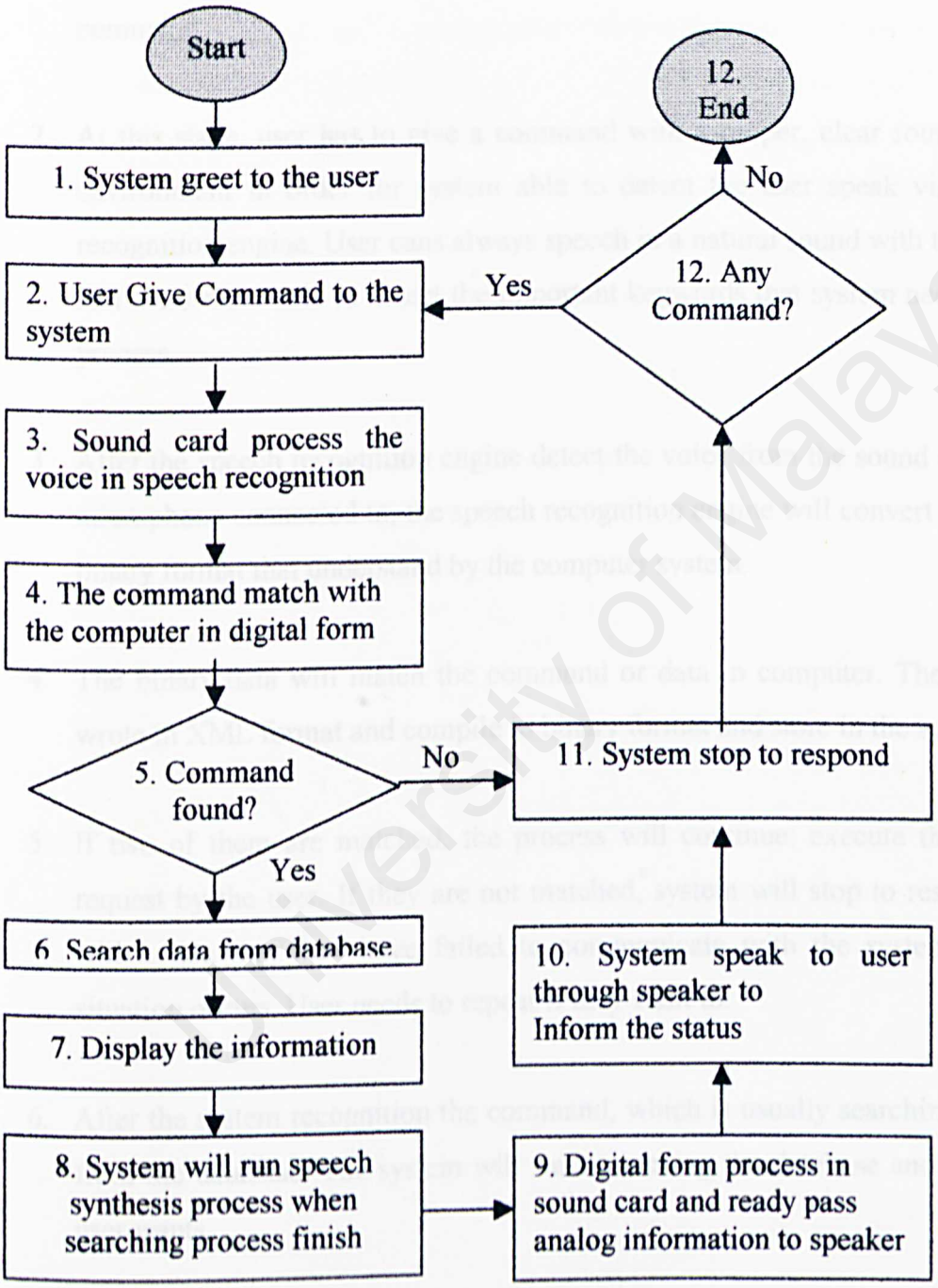


Figure 4.12 how system works

Explanation:

1. When user executes the program, this program will greet to the user. User will hear "Selamat Datang ke Speech Repository System! Nama saya ialah Speech! Sila Mulakan!". This starts the speech system and it is ready for user to give the command.

2. At this stage, user has to give a command with a proper, clear sound and quiet environment in order for system able to detect the user speak via its speech recognition engine. User cans always speech in a natural sound with the condition that the system able to detect the important keywords that system need to run the process.

3. After the speech recognition engine detect the voice from the sound card that the microphone connected to, the speech recognition engine will convert the sound to binary format that understand by the computer system.

4. The binary data will match the command or data in computer. These data was wrote in XML format and compile to binary format and store in the system.

5. If two of them are matched, the process will continue: execute the command request by the user. If they are not matched, system will stop to respond. Users will notice that they have failed to communicate with the system when this situation occurs. User needs to repeat if they want to.

6. After the system recognition the command, which is usually searching some data from the database. The system will start searching the database and get the data user wants.

7. All the related data will display in a form format in the user monitor.

8. When the process finishes, system will inform the user that the process finished via voice.

9. This process now involved converts the binary data (has process before that and store in the system) to the analog information and passes it to the speaker, which also connected to the sound card.

10. Speaker now informs the user via voice that the process user request has been done.

11. System will stop.

12. System will only start again if the users give another command. If user have no more command to give. User needs to close the program.

4.7 Summary

This chapter explained about the design information for Speech Recognition and Speech Synthesis System. This system involved 2 main programs, which are the system application and the database to present this application. Further, the system divides to 2 modules, which are: speech recognition and speech synthesis.

Explanation of each speech function (speech recognition and synthesis) has show step-by-step in previous chapter. But of the complete system, this chapter explained the details from starting (system greet to the user every time user activate the program) to retrieved data from database and present to user in different interface.

This system has a main menu where user will see every time activate the program. Inside the menu, user can choose any 6 of the function in 1 time for retrieved student and courses information. The function including:

- Student registration menu
- Subject provided by faculty according to semester
- Student name list according to year (1st year, 2nd year)
- Student result
- Student name list according to subject taking
- Student name list according to major

The user interface design also cover in this chapter. Note that the interface will be change from time to time to suit the requirement. Characteristic of the interface design have been taking a serious step in order to show a friendly function to the user since it is still new for this technology in this field.

The database design and Data Flow Diagram (DFD) explain what is the data will store in the database and how the program retrieve the data and display I the suitable menu. The description of the field name and the data type decided in this design level helped the development process because all the related information can be prepare before going to

the next step. When in the development phase, I can concentrate more on the coding phase without reverse back every time. This also fulfills the Waterfall model method.

Chapter 5
System Implementation and Testing

Chapter 5
System Implementation and Testing

5.1 Introduction

System implementation involves the translation of the software representation produced by the design into a computer-readable form. The major tasks in this phase are coding (program a new system), testing (test a new system) and debugging (fix the bug occurred in the program).

Implementation activities are primarily "environmental". They deal with the realities of particular machines, system, language, compilers, tools, developers and clients necessary to translate a design into working codes.

The goals of implementation phase are to implement a system correctly, efficiently and quickly on particular tools. This phase is a set of activities with:

- Input: Design, environmental and performance requirements.
- Output: A working system.
- Technique: Reconciliation, transformation, conversion, monitoring and testing.

5.2 Coding Methodology

The coding methodology used in the development of this system is top-down approach.

This approach allows the higher-level modules to be coded first before the lower modules. The coded in the lower modules contain only an entry and an exit. A module with such characteristic is called a shell. The higher module will reference the lower ones if they are coded and available. Reference to a shell will result in an empty action.

The top-down approach will ensure that the most important modules will be developed and tested first. It also gives a preliminary version of the system sooner.

5.3 Testing

Testing is the process of determining whether a program or a system performs the desired process. System testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. However, testing cannot show the absence of defects, it can only show that software defects are present. Hence, it is very important to keep this in mind while testing is being conducted.

There are a number of rules that can serve well as testing objectives:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one has a high probability of finding an as yet undiscovered error.
- A successful test is one that uncovers on as yet undiscovered error.

Three type of testing strategies has been used to test the system namely:

1. Unit testing
2. Integration testing
3. System testing

Each of the testing strategy will be explained in their respective section later in this chapter. Besides, the test case used will be discussed.

5.3.1 Unit Testing

In unit testing, the developer tests the programs making up the system. The units in a system are the modules and routines that are assembled and integrated to performed a specific function.

In this system, units or modules that's form the system are mostly from modules, which perform a specific function. Each of this form modules will contain many sub-function or action commands.

Some of the unit testing aspect taken into account are as follows:

- **Modules interface** - purpose is to ensure that information property flows into and out of the program unit under test.
- **Boundary conditions** – purpose is to ensure that the modules operate properly at boundaries established to limit or restrict processing.
- **Independent paths through the control structure** – purpose is to ensure that all statements in a module have been executed at least once.
- **Error-handling paths** – purpose is to ensure that the error handling routine can handle expected and unexpected errors.

5.3.2 Integration Testing

The integration testing is the testing of two or more modules or units combined together to got errors associated with the interfacing. Examples of errors associated with interfacing are:

- Data can be lost.
- One module can have inadvertent, adverse affect on another.
- Sub-functions may not produce the desired major function when combined.
- Individually acceptable imprecision maybe magnified to unacceptable levels.

For this system, the bottom-up integration approach is used when it begin construction are testing with modules at the lowest levels and then move upward to the higher level modules. Several lower level forms can be combined together for testing first before testing could be done on the higher level ones.

5.3.3 System Testing

System testing is series of different tests conducted to verity that all system elements such as hardware, software and information have been properly integrated and performed allocated functions.

In this system, system testing is done by testing all of the available function modules, which make up the whole system. Once the system is tested, it will be expected to executable files, which enable them to run on the stand-alone computer.

For hardware testing, one of the purposes was to ensure that most of the computer in different setting and environment could be run. The setting here including the monitor screen area (800 x 600, 1024 x 768 using 15' and 17' monitor), colour depth (32 bits and 24 bits). The test cases also including testing the system running in minimum system requirement. The testing tells that this system can run in Intel Pentium 200 MMX with 32 EDO RAM when user communicate with it.

5.4 Test Cases

These are some of the test cases that I have been done in different situation and environment.

5.4.1 Operating System Testing

- The operating system used when development was Windows 98 SE.
- For testing purpose, Windows NT 4 and Windows 2000 Professional also been used.
- Windows 95 was not compatible due to some of the development tools was not support.
    - After testing, it runs successfully in Windows 98 SE, NT 4 and 2000 professional

5.4.2 Software Testing

- In these testing, it was all done in home environment, which can be consider quiet.

- The number of training lesson was 2. The first lesson was when the "Introduction of Microsoft Speech Recognition" when install Microsoft Speech SDK. The second lesson was "Aesop's Fables".
- The hardware using was Athlon TunderBird 1.1G with 256 SDRAM.
- Each test will repeat 10 times. If the respond was more then 5, it consider pass, else failed.

1. Calling the common function

Function	Speech (BM/BI)	Comment
From main menu to other program / sub function	Pass	I have testing this with Malay and English, most of the time it showed the program I call. ➢ In Malay, 7 out of 10 test successfully. Sometimes the system confuse with "Pendaftaran Pelajar" with "Tahun Pelajar". ➢ In English, 9 out of 10 test successfully.
Getting help	Pass	It will prompt out the help menu **every time** I test it with Malay and English. The help command will only accept in Main Menu via voice. While other help file in sub function need mouse to control. ➢ In Malay and English, 10 out of 10 test successfully.
Close program	Pass	Since I have give all the program with a name, most of the time the system able to recognize which program I asked to close. ➢ In Malay, 8 out of 10 test successfully.

		➢ In English, 10 out of 10 test successfully.
Minimize, maximize, restore	Pass	Since I have give all the program with a name, most of the time the system able to recognize which program I asked to minimize. ➢ In Malay, 8 out of 10 test successfully. ➢ In English, 10 out of 10 test successfully.
Print	Pass	This command more accurate in English compare to Malay. ➢ In Malay, 6 out of 10 test successfully. ➢ In English, 9 out of 10 test successfully.
Clear buffer	Pass	Most of the time the system able to recognize this command and clear out the test area. ➢ In Malay, 9 out of 10 test successfully. ➢ In English, 9 out of 10 test successfully.
Process command	Pass	The system can process in any situation **every time** even when the data given was not correct. ➢ In Malay and English, 10 out of 10 test successfully.

Table 5.1 Basic Function Testing

2. Menu Student Information

Instruction	Speech (BM/BI)	Comment
Get semester	Pass	The system able to recognize the different of the semester. The semester tested was in 1st semester and 2nd semester. ➢ In Malay, 8 out of 10 test successfully. ➢ In English, 9 out of 10 test successfully.
Get year	Pass	The system able to recognize the year's number in Malay and English. The year's number tested start from 1997 until 2002. ➢ In Malay, 7 out of 10 test successfully. ➢ In English, 9 out of 10 test successfully.
Get metrics number	Pass	The system gets the metrics number more accurate in English compare to Malay. Anyway, the system hard to differentiates between "WEK" and "WET". ➢ In Malay, 5 out of 10 test successfully. ➢ In English, 6 out of 10 test successfully.

Table 5.2 Menu Student Information Testing

3. Menu Subject

Instruction	Speech (BM/BI)	Comment
Get semester	Pass	The system able to recognize the semester. The semester tested were 1st semester, 2nd semester, 3rd semester and "all semester". ➢ In Malay, 7 out of 10 test successfully. ➢ In English, 8 out of 10 test successfully.
Get subject code	Pass	The system able to recognize the subject code in either the alphabet or number. The alphabet was more accurate that the number. ➢ In Malay, 8 out of 10 test successfully. ➢ In English, 9 out of 10 test successfully.

Table 5.3 Menu Subject Testing

4. Menu Semester Courses

Instruction	Speech (BM/BI)	Comment
Get subject code	Pass	The system able to recognize the subject code in either the alphabet or number. The alphabet was more accurate that the number. ➢ In Malay, 8 out of 10 test successfully. ➢ In English, 9 out of 10 test successfully.
Display all the subject	Pass	In this command, system can recognize in any situation **every time** the user speak. ➢ In Malay and English, 10 out of 10 test successfully.

Table 5.4 Menu Semester Courses Testing

5. Menu Result

Instruction	Speech (BM/BI)	Comment
Get report format	Pass	The system able to differentiate between "Ringkasan/Summary" and "Huraian/Details". ➢ In Malay, 7 out of 10 test successfully. ➢ In English, 9 out of 10 test successfully.
Get metrics number	Pass	The system gets the metrics number more accurate in English compare to Malay. Anyway, the system hard to differentiates between "WEK" and "WET". ➢ In Malay, 5 out of 10 test successfully. ➢ In English, 6 out of 10 test successfully.

Table 5.5 Menu Result Testing

6. Menu Year Of Student

Instruction	Speech (BM/BI)	Comment
Get year intake	Pass	The system able to recognize the year's number in Malay and English. The year's number tested start from 1997 until 2002. ➢ In Malay, 7 out of 10 test successfully. ➢ In English, 9 out of 10 test successfully.
Display all the student	Pass	In this command, system can recognize in any situation **every time** the user speak. ➢ In Malay and English, 10 out of 10 test successfully.

Table 5.6 Menu Year Of Student Testing

7. Menu Major

Instruction	Speech (BM/BI)	Comment
Get major name	Pass	The system have no problems differentiates the major name likes "MIS" "NT" or "Multimedia". ➢ In Malay, 5 out of 10 test successfully. ➢ In English, 6 out of 10 test successfully.
Display all the student	Pass	In this command, system can recognize in any situation **every time** the user speak. ➢ In Malay and English, 10 out of 10 test successfully.

Table 5.7 Menu Major Testing

5.4.3 Hardware and Environment Testing

With these test cases, different computer with different of hardware specification was test in different environment. The purpose of hardware testing is to ensure that the minimum requirement for the system to run. While environment was used to know the effect for the system performance.

Hardware / Environment	Place	Comment
Intel Pentium 200 MHz MMX	Home	Speech technology needs a lot of processor power. The faster processor speed, the better performances. The system load slowly but still able to recognize user command slowly. Having no problems in

		speech synthesis but slowly in respond. Decrease the friendliness and performance.
Intel Pentium III 600 MHz	Lab	It performed better then the system before. System able to load quickly and no problems at all in speech recognition and speech synthesis.
AMD Athlon TunderBird 1.1GHz	Home	The system load quickly and the voice recognition and synthesis can be done quickly. Increase the friendliness and performance.
32 EDO RAM	Home	Speech technology again was a memory hungry system. The more memory computer the better performances. The system load slowly but still able to recognize user command slowly. Having no problems in speech synthesis but slowly in respond. Decrease the friendliness and performance.
128 SDRAM	Lab	This was the system hardware recommended in Microsoft Speech SDK 5.x. It performed better then the system before. System able to load quickly and no problems at all in speech recognition and speech synthesis.
256 SDRAM	Home	The system load quickly and the voice recognition and synthesis can be done quickly. Increase the friendliness and

		performance.
15" inch and 17" inch with 800X600 monitor resolution	Home / Lab	The menu look bigger until almost close the whole screen. Most of the program menu still did not appear the scroll bar when the program activate. It helps user to see all part of the interface without using mouse. If the data display was a lot, a scroll bar will appear inside the display box.
15" inch and 17" inch with 1024X768 monitor resolution	Home / Lab	The menu look nicer and smaller. No scroll bar appears when the program activates. . If the data display was a lot, a scroll bar will appear inside the display box.
Printer	Home / Lab	System able to print either the printer was stand alone (only connect to 1 computer) or with network (many computer share 1 printer).
Creative Sound Blaster Live! Value sound card	Home	With newer and better technology, the sound generate was more soft. The different between "Mike" and "Sam" can be differentiates easily.
Creative Sound Blaster AWE 64 sound card	Home	This card can generate the sound normally. The sound of "Mike" and "Sam" can only is differentiates when the speaker sound's was loud.
Take for 1	Home /	The lesson was "Introduction of Microsoft

training lesson	Lab	Speech Recognition". It was hard for system to perform accurate in recognize user command.
Take for 2 training lesson	Home / Lab	The lessons were "Introduction of Microsoft Speech Recognition" and "Aesop's Fables". It increase the accuracy of the system in speech recognition.
Quiet environment	Home	The result above in "Software Testing" area explained the effect in this environment.
Noisy environment	Lab	The system almost hard to recognize the command especially in student metrics number. User needs to speak and clear the buffer when mistake occur. It makes the system performs badly. Most of the time the system **failed** to recognizes the command. It either appears wrong information or does nothing at all.

Table 5.8 Hardware and Environment Testing

5.5 Summary

This chapter explained some of the testing method and its definition and purpose. When testing this system, I have try different platform and environment in order to capture the limitation and error that may occur.

First of all, I try the operating system that could be run and not. The operating system I did not face any problems because the tools I use require Windows 98 and above. When I create the system into a setup package, I also selected the system to make sure it could be install in these platform.

I also try the system function by function for capture the different error. When error occurs, I have made some adjustment in the coding phase and then try the system again. This step has to repeat until the error could be solved.

Hardware testing I for checking the minimum hardware requirement for the system able to run. Although the system still run successfully with a computer that comes with 200MHz processor and 32 RAM, I do not recommend user run under this condition. With this condition. It makes the system performance slowly and hard to use. User also cannot do multiple tasks at the same time.

Chapter 6
Project Evaluation

Chapter 6
Project Evaluation

6.1 Features and Strength of Speech Recognition and Speech Synthesis Repository System

- **Voice control**

  Speech Recognition and Speech Synthesis Repository System enable user control the system via voice, unlike the normal system that need to use keyboard. User can access the database and getting information when the program was activated. Every information user required will be display. User only needs to give the desire information when searching via voice without using keyboard.

- **Two language support**

  This system enable user control not just using Malay, but also English. User free to speak with the system with any of these two languages in any situation, either in full Malay, full English or even Malay and English use together.

- **Two way input support**

  Beside the system support voice command, user also able control the system using pointing device to activate the sub-function, close it, access help file or do printing works.

- **Speech engine separate with SRn'SSRS**

  During development and creating the setup package, I found that it is better for separate the speech engine (recognition / synthesis) with the system that I developed. It is because with the speech engine install different or separate with the SRn'SSRS (Speech Recognition And Speech Synthesis Repository System), user can train the speech recognition anytime without cares much of the version of SRn'SSRS. In other

words, when SRn'SSRS need some changes, it can change anytime and install the new system to the computer. User need not train again the system to recognize their voice since the speech engine was the same. It saves a lot of time compare with the system that integrated with speech engine. In this integrated cases, when user uninstall the old version and install the new one, they need to train again the voice recognition process start from beginning.

- **Support different version of Windows operating system**

  This system can run in more then 1 type of Windows operating system, including Windows 98, Windows 98 SE, Windows ME and Windows 2000.

6.2 Limitations and Constrains of Speech Recognition and Speech Synthesis Repository System

- **Sensitive with the environment**

  The system once activate, will detect the sound comes from user. It's fine when the environment was quietly. Anyway, if the environment full with noise like sound from fans or paper works, the system will assume that a command was given and try to process that command.

- **Required a big storage of hard disk**

  The system was small for install, but the speech engine required for the computer to recognize sound and synthesis from text was huge.

- **Not user friendly**

  The speech technology was still new to human. The working process was hard to beat the traditional input device because of it user friendliness issue. User feel more comfortable control the computer works without notice of other people. With voice, people beside user able to detect what function user using or information was access. The respond was also faster in access the information or function using keyboard

compare to voice control. It is because the keyboard comes with a lot of different function and character for user to choose, unlike voice control input, user only can use microphone to control different kind of function and access information. Imagine the microphone have to do a lot of different works. It was just like a "1-button keyboard", user press 1 time for typing alphabet "a", 2 times quickly for getting "b" or 26 times just to getting "z".

- **Required multimedia support**

  This system controlled by voice. It means the computer should have a compatible sound card plug-in in the computer to process the voice that receive or send. User also needs to have a set of microphone and speaker for input and output voice.

6.3 Proposal For Future Enhancement

There is still much room for improvement on this system. The possible future enhancements are listed below:

- **Sound frequency analyzer**

  If the system comes with a sound frequency analyzer, user will able to detect the noise of the environment. The wave show in the program will notice the user switch to other alternative control device if the environment was noisy.

- **Improve user interface**

  The program will only accept by user if it gives the clear information to user to control the system. The interface could be added more graphics and multimedia to attract the interest of the system.

- **Microphone's Quality**

  A good quality of a microphone able to filter out noise and increase user voice command. It helps to improve the system friendliness and speed performance.

- **System Sensitivity Adjustable**

  The system can be add with more function likes let user adjust the system recognition sensitivity within the interface. Currently this function could only be found in control panel at Speech.

6.4 Summary

There are a lot of potential for this system. First of all, it could be control without a mouse and keyboard. This could be user for the user that facing some difficulties in using the traditional input device. When controlling the system. User also gets notice from system with voice telling that the status of the system, either run successfully or failed.

Any way, this system still faces some of the problems that are hard to solve. The system was not as user friendly as traditional input device. User needs to repeat the command when system failed to recognize to command. The environment also affected the system performance. System will fail to do the task if the surrounded was noisy. The hardware requirement was also a big issue here. The computer was recommend comes with 128 RAM memories for system run smoothly. Luckily now the memory was quiet cheap. Most of the user able to upgrade their computer without any problems.

Some of the suggestion also point out for future enhancement. This including sound frequency analyzer. With the sound analyzer that appears beside the program, user will notice the changes of the environment. It could help user make any adjustment likes using a better microphone or put the system in a quieter place.

Chapter 7
Discussion and Conclsion

Chapter 7
Discussion and Conclusion

7.1 Problems and Solution

During the process of developing, number of problems has been encountered. The main problems and possible solution are listed below:

- **Problem: Lack of references book regarding speech technology in Visual C++**

  In the market, there is a lot of references book like Visual Basic compare to Visual C++. There are also more people using Visual Basic compare to Visual C++ in developing a system. It became worse when the market lack of the speech technology solution in any programming language.

  - **Solution:**

    Internet becomes the main source for me to gain the knowledge. Some of the discussion board I attend did help me to find the solution in term of programming problems or speech related. One of the main discussion boards I attend in solving programming problems was Codeguru.com and Codeproject.com. While Generation5.com help me in speech recognition and synthesis field.

- **Problem: Dilemma in designing the interface**

  Speech technology is still new for user. The program should also support different kind of input if the system failed to detect user voice.

  - **Solution:**

    The program needs to have simple and straight interface for user. Mouse as backup control device should support beside 2 different language.

❖ **Problems: Selecting software for development**

The selection of development software was one of the crucial factors in determining the success of the project. Beside Microsoft Speech SDK, other like Chant Speech, IBM Via Voice should also be taken into account.

➢ **Solution:**

To develop the latest and advance system, a substantial task research has been carried out. Book and Internet are two main source used to evaluate the feature and capabilities of the system. Beside, help and advice are also sough from my supervisor and friends who have used and familiar with this software. Base on the requirements for Speech Recognition and Speech Synthesis Repository System, Microsoft Speech SDK finally chosen.

❖ **Problem: Slow processing time**

Speech Recognition and Speech Synthesis Repository System that's deals with voice and text processing, which is multimedia elements. As such, the application created using this software requires more memory to execute. When this system starts to grow, it was found that the system took up longer to process.

➢ **Solution:**

More memory and processor speed will have to install to increase the processing speed.

❖ **Problem: Require a number of support software**

Some of the software required use as a support during software development. Example of the program need to install in developer computer including Visual Studio 6 with service pack 3 or later for programming, Microsoft Platform SDK and Microsoft Speech SDK for support speech programming and provided speech engine, Installed Shield for turning the development program to an installer setup program.

➢ **Solution:**

Download the Microsoft Speech SDK and Platform SDK and install it in the computer. While the creation of setup program can be done in the faculty's lab.

- **Problems: the system was hard to detect user's command**

  The system was not sensitive or too sensitive for different environment. The command user given wasn't clear to the application affected the application to get the correct command.

  - **Solution:**

    Use good quality microphone to filter out the noise. User require to patient in training the voice detection tutorial. More tutorial user takes, the more accurate for system detect the command. Changes of coding also been done in order to increase the accuracy.

7.2 Overall Conclusion

Upon the completion of this project, the Speech Recognition and Speech Synthesis Repository System is consider to have achieved its objectives as well as the functional and non-functional requirements as planned earlier. This application will serve as the basic knowledge for future Speech technology developer and user. Hopefully it will also able to gain more confidence among Malaysian toward the Speech technology development fields.

A lot of valuable knowledge was gained throughout the development of this project. The knowledge gained can be categorized into a few group such as programming in speech technology, Windows programming skills, packaging software and database related knowledge. Below list out the knowledge and experience gained throughout the project:

- Speech recognition technique like "whole-word matching", "sub word matching", "isolated word recognition" or "keyword spotting".
- Speech synthesis technique like Formant synthesis and Concatenative synthesis.
- Visual C++, XML, HTML and COM programming language.
- Microsoft Access and the SQL knowledge.
- Windows API (Application Program Interface) and DDI (Device Driver Interface).
- Building a setup program.

Computer sciences studies have provided valuable theories and knowledge like system analysis, design and software engineering that can be put into practice in the project.

Although speech technologies (speech recognition and speech synthesis) was not as convenience as the traditional input device like keyboard and mouse, and user need to repeat the command in some situation, there are still a potential for this technology in the future. Speech Recognition and Speech Synthesis Repository System will become more powerful and complete as the new enhancements and features are incorporated in the near future.

Bibliography

Bibliography

Internet :

1. A Short Introduction to Speech Recognition
   http:// tcts.fpms.ac.be/asr/introduction.html
2. A Short Introduction to Text-to-Speech Synthesis
   http:// tcts.fpms.ac.be/synthesis/synthesis.html
3. Comparison of Oracle and Microsoft Access SQL
   http://www.cse.bris.ac.uk/~ccmjs/ora_sql.htm#Introduction
4. DevGuru XML DOM Introduction
   http://www.devguru.com/Technologies/xmldom/quickref/xmldom_intro.html
5. Dragon Naturally Speaking Developer Suite
   http://www.synapseadaptive.com/naturallyspeaking/developer.htm
6. Encarta Encyclopedia
   http://encarta.msn.com/
7. generation5.org
   http://www.generation5.org/
8. Software Life Cycle
   http://sern.ucalgary.ca/~tma/seng611/SLC.htm
9. Speak & Spell assignment
   http://www.cee.hw.ac.uk/~cmj/projects/Speak/speachsyn/speachsyn.html
10. Speech 2000
    http://microsoft.com/speech
11. Speechkit.com
    http://www.speechkit.com/
12. Speech Primer
    http://www.o2a.com/SpeechPrimer.html
13. Voice System IBM Software
    http://www-4.ibm.com/software/speech/
14. Whatis.com

http://www.whatis.com/

15. Webopedia Online Computer Dictionary for Internet Term and Technical Support
    http://webopedia.internet.com/
16. xmlpitstop
    http://www.xmlpitstop.com/

Books:

17. Davis Chapman. "Teach Yourself Visual C++ 6 in 21 Days".SAMS Publishing
18. Dr. P. Sellappan. "Access 2000 Through Example". Federal Publications Sdn. Bhd.
19. Microsoft Speech SDK5.0 Help File

Appendix

User Manual

User Manual Part 1

Speech Recognition and Speech Synthesis Repository System

System Requirement and Installation

1.0 System Requirement

1.0.1 Operating Systems

Supported operating systems are:

- Microsoft Windows 2000 Professional Workstation or Server; all language versions.
- Microsoft Windows Millennium edition.
- Microsoft Windows 98 all editions.
- Microsoft Windows ® NT Workstation or Server 4.0, service pack 6a, English, Japanese, or Simplified Chinese edition.
- Windows 95 or earlier is not supported.

1.0.2 Software Requirement

- Microsoft Internet Explorer 5.0 or later version. Users of Windows NT 4 with any version of the service packs require Microsoft Internet Explorer 5.5 or later.
- Microsoft Visual C++ 6.0, service pack 3 or later version is needed to run the SAPI 5 SDK. In general, any 32-bit C compiler will work for writing SAPI applications.
- Platform SDK is generally not needed although some samples and functionality may require it.

1.0.3 Hardware Requirement

- A Pentium-equivalent or later processor at 200 MHz with 128 megabytes (MB) of RAM is recommended.
- SAPI 5 can now take advantage of a computer and operating system that supports multiple processors, including all those mentioned above. Additionally, you can use SAPI 5 in a distributed application environment.
- A microphone or some other sound input device to receive the sound is required for speech recognition. In general, the microphone should be a high quality device with noise filters built in. The speech recognition rate is directly related to the quality of the input. The recognition rate will be significantly lower or perhaps even unacceptable with a poor microphone.
- Not all sound cards or sound devices are supported by SAPI 5, even if the operating system supports them otherwise.
- The following table outlines the RAM usage:

Component	Minimum RAM	Recommended RAM
TTS Engine	14.5 MB	32 MB
SR Command and Control	16 MB	32 MB
SR Dictation	25.5 MB	128 MB
SR Both	26.5 MB	128 MB

Table A1 RAM Usage

As a result, the user hardware should have:

- Pentium-equivalent or later processor at 200 MHz
- Recommended 128 megabytes (MB) of RAM
- 350MB of hard disk space
- Microsoft compatible sound card
- Microphone

- Speaker
- 15" inch monitor or larger

1.1 Installation

1. Put the CD into CD-ROM.
2. Go to SRn'SSRS folder.
3. Double click the setup.exe file. A setup wizard will guide user through the installation.
4. User will see at the Start->Programs, there is a shot-cut for activate the system, the shot-cut is "SRn'SSRS".
5. Click the shot-cut, the main menu will prompt out.

If user wish to see the help file after the installation, user can follow these step:

1. Open Window Explorer.
2. Select the drive that program Install, or drive C.
3. Select Programs folder
4. Select WEK 990231 folder.
5. Select SRn'SSRS
6. There are **2 help file in Malay and English version**.
7. Select any one.

User Manual Part 2

Speech Recognition and Speech Synthesis Repository System

System Setting

2.0 Voice Training

After complete install of Microsoft Speech SDK, a wizard will prompt out and user to adjust the microphone and speaker in order to perform better. User can start training on this step or after the complete of all of the installation. User also can go to Control Panel to do the adjustment afterwards.

Go to Start -> Settings -> Control Panel.

Activate Speech icon there.

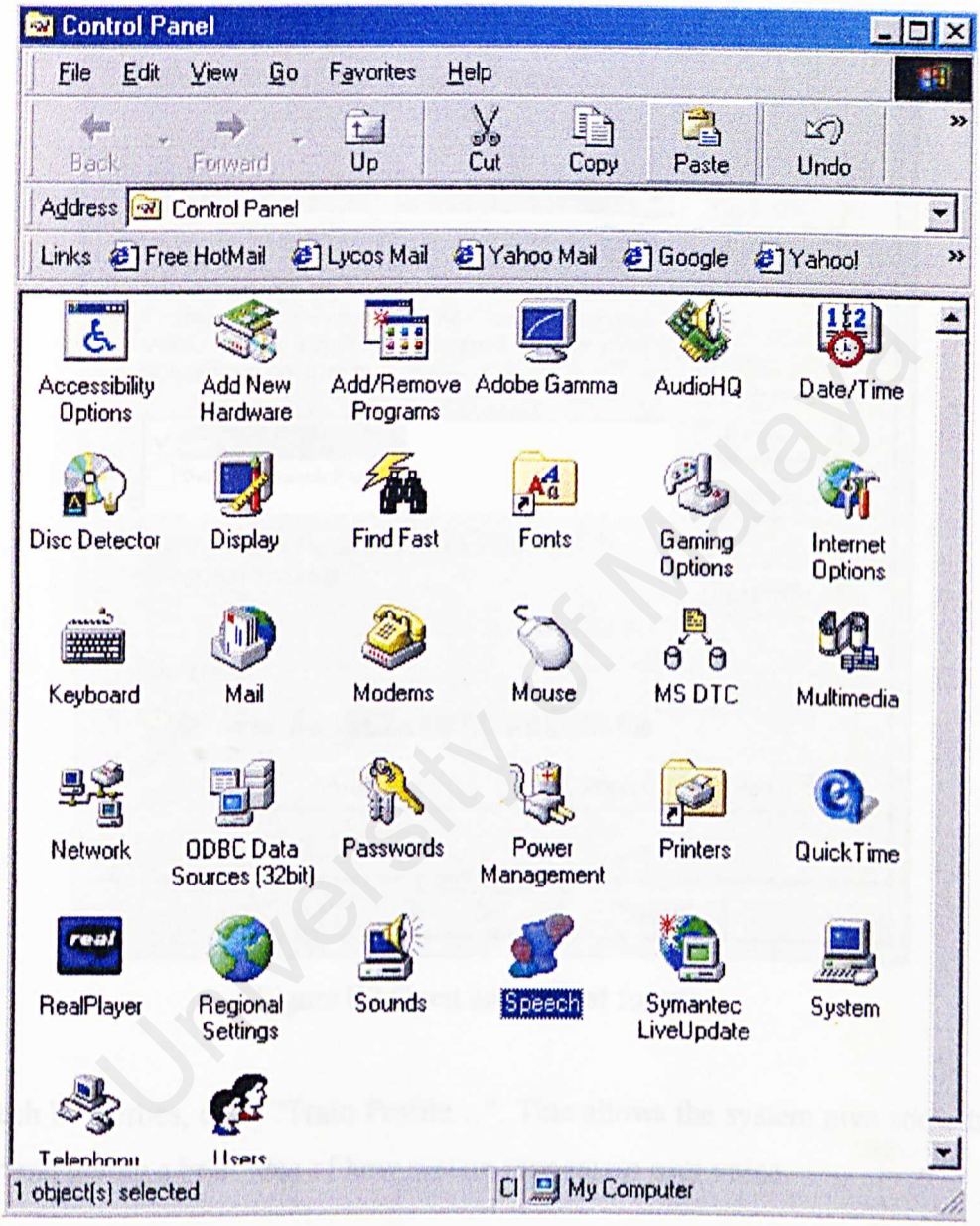


Figure B1 Control Panel

A menu named Speech properties will prompt to user. User needs to open an account for system recognizes their voice. After that, user needs to train the voice for system to recognize it.

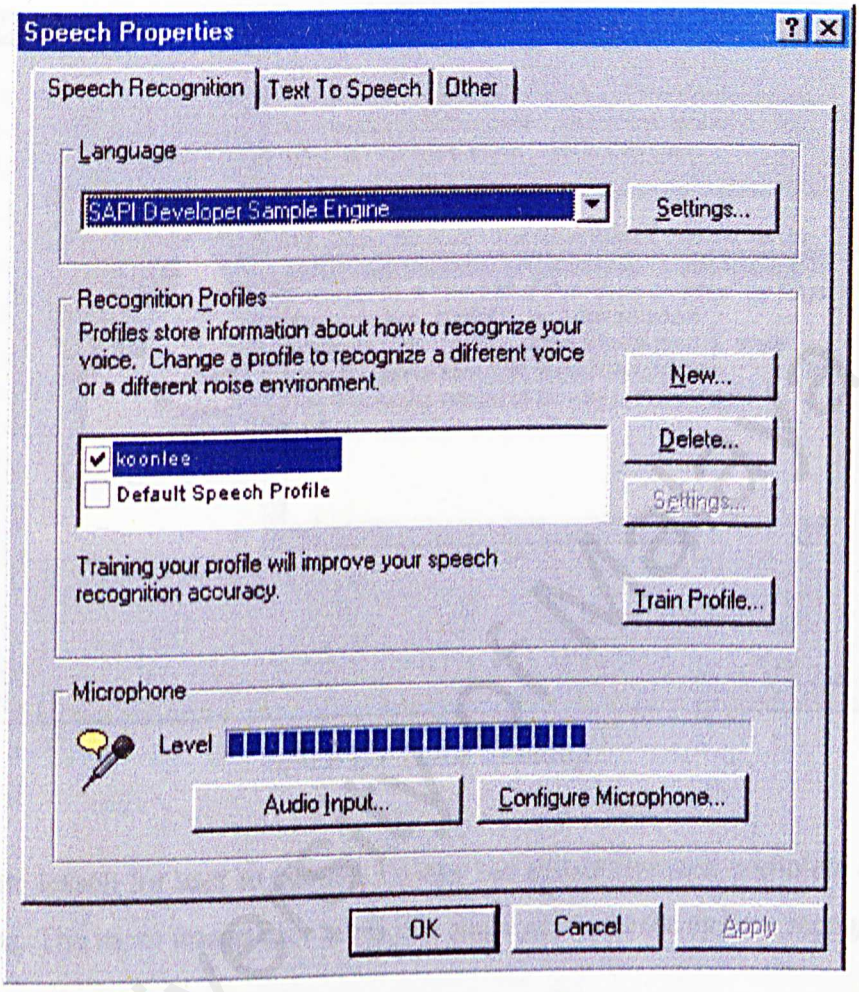


Figure B2 Open an account for user

At Speech Properties, click "Train Profile…". This allows the system give some training for user and giving a brief idea of how system recognizes user voice.

A menu named Voice Training will appear.

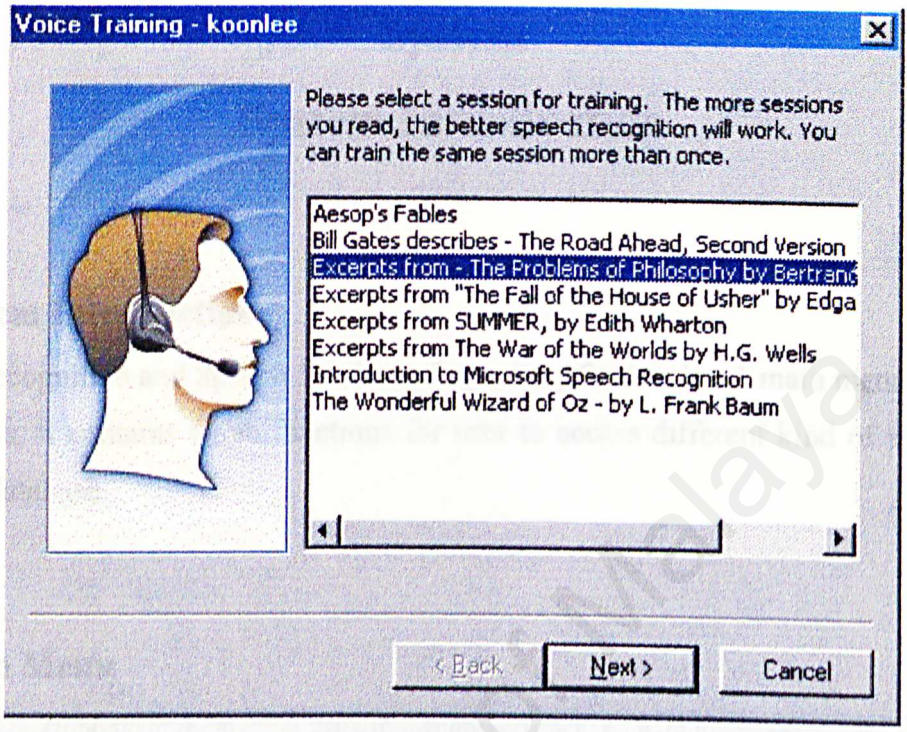


Figure B3 Voice Training

There is some lesson for user to choose. Follow the instruction and complete at least one of the lessons. The more lesson user takes, the more accurate the system recognize user.

User Manual Part 3

Speech Recognition and Speech Synthesis Repository System

System Introduction

3.0 System Introduction

Speech Recognition and Speech Synthesis Repository System has 1 main menu. From the main menu, it contains 6 sub functions for user to access different kind of information from the database.

3.1 Main Menu

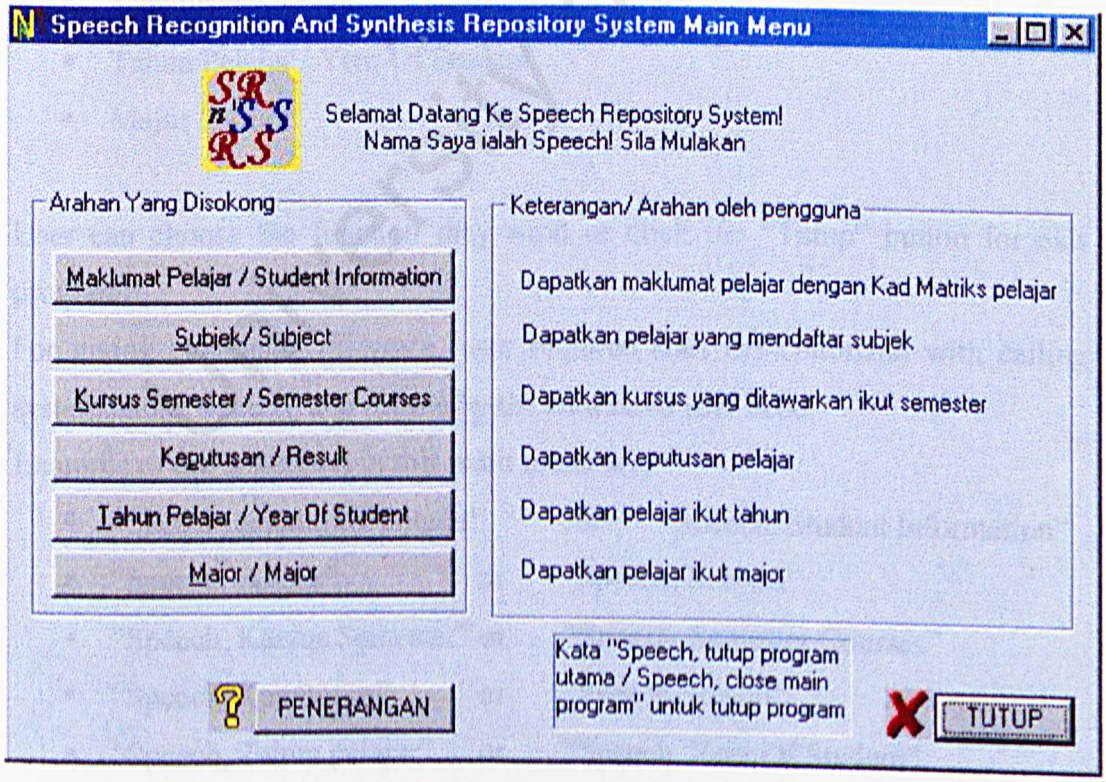


Figure C1 Main Menu

- This main menu for Speech Recognition and Speech Synthesis Repository System. When user activate from Start->Programs->SRn'SRS, it will prompt out.
- When the main menu appears, user would hear system greet "Selamat datang ke Speech Repository System! Nama saya ialah Speech! Sila mulakan" to user.
- This interface will always appear either at the background of other program or in front of the monitor, depends the situation.
- From this menu, user can control in 2 method, which are:
  - Speech
  - Pointing device (mouse)

- This menu enable user gets to use other function of this system. The function included:
  - Maklumat Pelajar / Student Information
  - Subjek / Subject
  - Kursus Semester / Semester Courses
  - Keputusan / Result
  - Tahun Pelajar / Year Of Student
  - Major / Major

- User can choose the function they want or click the "Tutup" button for exit the program.
- For giving command via voice, user required start the command with calling the system name, Speech, and follow by the command user need.
- Example of the command in this main menu are:
  - "Speech, Maklumat Pelajar" or "Speech, Student Information"
  - "Speech, Subjek" or "Speech, Subject"
  - "Speech, Kursus Semester" or "Speech, Semester Courses"
  - "Speech, Keputusan" or "Speech, Result"
  - "Speech, Tahun pelajar" or "Speech, Year Of Student"
  - "Speech, Major"

- After user activated selected function, user will heard a respond via voice to confirm that the system receive user command and was in the progress. The sound was:
    - "Proses" follow by the name of the command or function user selected.

- If user planned to quit the program, user can give the command by speaking "Speech, tutup menu utama" or "Speech, close main menu".
- Use can always asked for helped when necessary. User can say "Speech, tolong" or "Speech, help" and a dialog description about the program will prompt out.

3.2 Common Command and Situation

3.2.1 Window Size Control

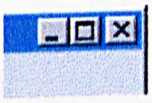


Figure C2 The place to control the window's size

- User free to either click the logo or control via voice
- The command can be use are:
    - "Speech, minimize program" follow by the program's name
    - "Speech, restore program" follow by the program's name
    - "Speech, maximize program" follow by the program's name

3.2.2 Print and Close

Figure C3 Print data and close program

- User free to either click the button or control via voice
- The command can be use are:
  - "Speech, print" or "Speech cetak"
  - "Speech, tutup" follow by the program's name. In this case, we can say "Speech, tutup pendaftaran or Speech, close register".

3.2.3 Command Description Box

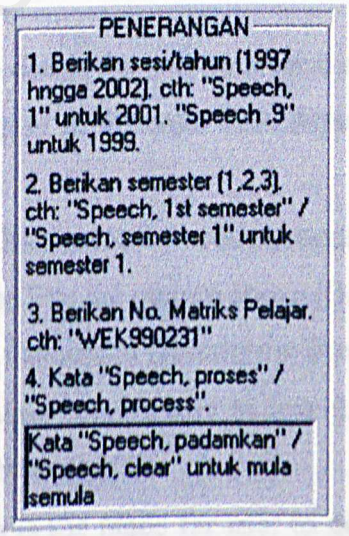


Figure C4 Command description box

- This is the step-by-step box for helping user to control the system.
- In order to let system capture user command, user required following the step list in the box.

3.2.4 Help File

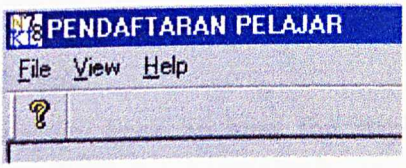


Figure C5 Help file logo

- The question mark indicated the help file for the specific program. For this request, user only able to control via the pointing device and not the voice.

3.2.5 System Detection Area

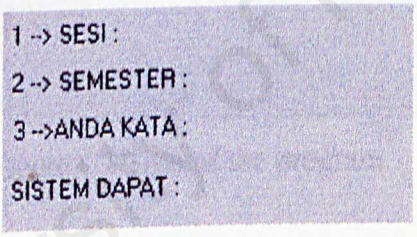


Figure C6 System detection area

- This is the area where system detect the user command and display the command that user said. The result will display in different field depends the user command given and the program that user run.
- Example, when user activate Student Information program, user can see on top left of the interface, there was an area like the picture above (System Detection Area) and on top right, there is an area like Command Description Box.
- In "1.-> SESI:" there is the place require user to give the information about the year user wants to get, like 1999, 2000 or 2001.
- In "2.-> SEMESTER :" there is the place for user to give the information about the semester number, like 1,2 or 3.
- In "ANDA KATA:" this is the place for user to give the student metrics number.

- When all the data given was correct, user can said, "Speech, proses" or "Speech, process" to display the student information.
- If the data detected was not the data user wanted, or user wishes to retrieve other information, ser need to said "Speech, padamkan" or "Speech, clear" to clean all the detected area.
- Notes that the running method was same for the entire program, the different was only the required data that it need.
- That's the reason every program have an area for display the description of the program and a file for it.

3.2.6 Title of the Program

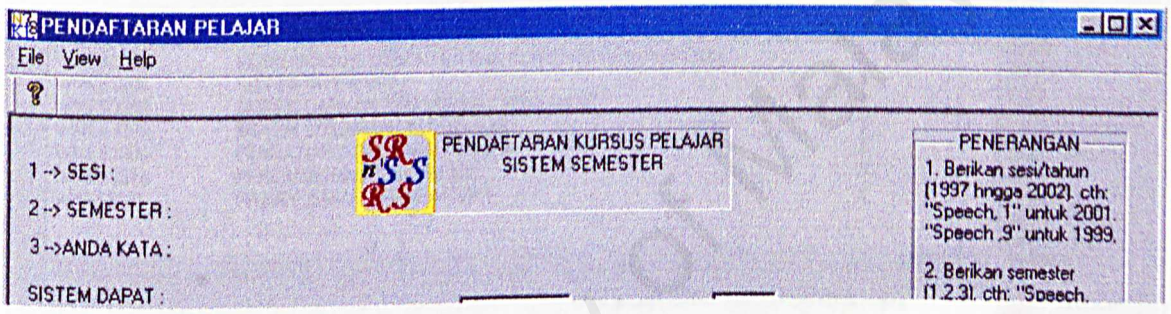


Figure C7 Title of the program

- In every program, the title of the program will be display on the top left side and the center of the interface.
- This help user to know which they are currently.

3.2.7 Inform of Process Status

- If user chooses any of the function 6 function from main menu, user will heard "proses". It means that the system had detected user command and start processing.
- When user giving data to system, and asking for process, user would also heard the respond from system, example:
  - "Pendaftaran pelajar ditunjukkan" when the information exists
  - ""Pendaftaran pelajar tidak dapat ditunjukkan" when the information wasn't correct or error exists.

3.3 Student Registration Menu

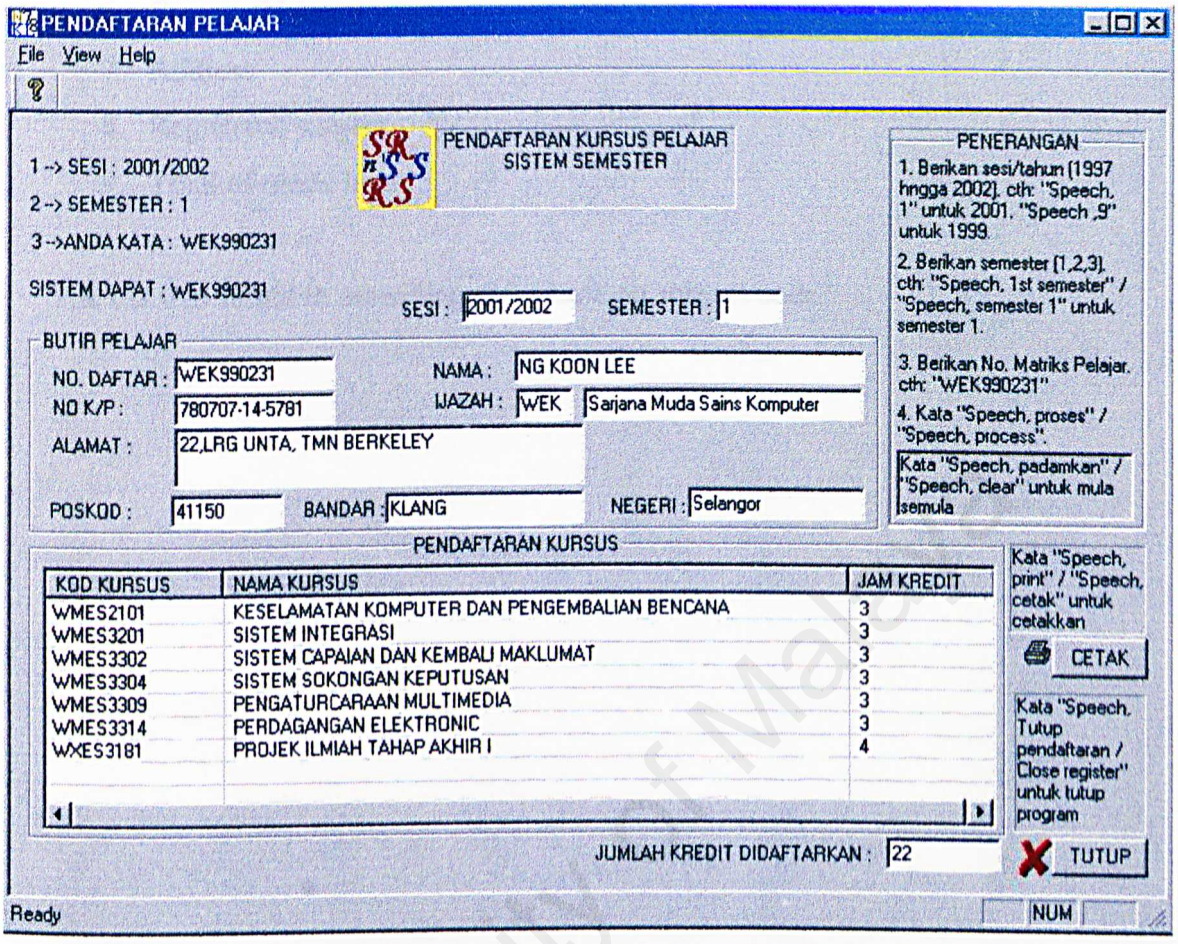


Figure C8 Student Registration Menu

- In Student Registration menu, user need to give 3 data for the system:
    1. Year from 1997 until 2002
    2. Semester from 1,2 or 3
    3. Student metrics number

- System will process if the data given exists in the database.
- As return, the information display to the user including
    1. Year
    2. Semester
    3. Metrics number
    4. Name

5. IC number
6. Courses Name
7. Address
8. Registered subject
9. Total of credit hour

- The list was sorted in ascending order base on subject code.

3.4 Subject Menu

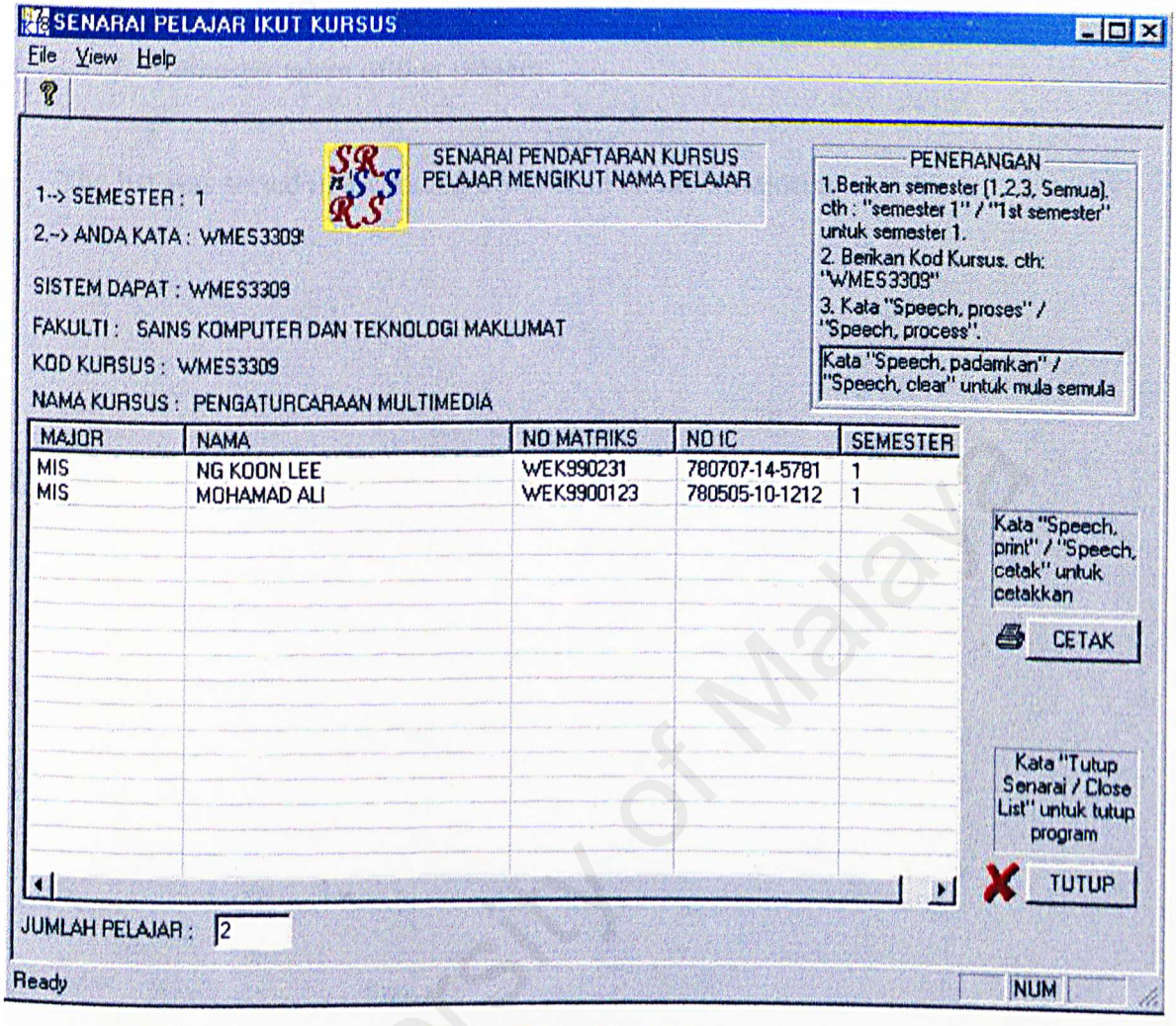


Figure C9 Subject Menu

- In Subject menu, user need to give 2 data for the system:
    1. Semester from 1,2,3 for all the semester
    2. Subject code

- System will process if the data given exists in the database.
- As return, the information display to the user including
    1. Subject code
    2. Subject Name
    3. Student's major
    4. Student's name

5. Student's metrics number
6. Student's IC number
7. Semester taken of that subject

- The list was sorted in ascending order base on their majoring.

3.5 Semester Courses Menu

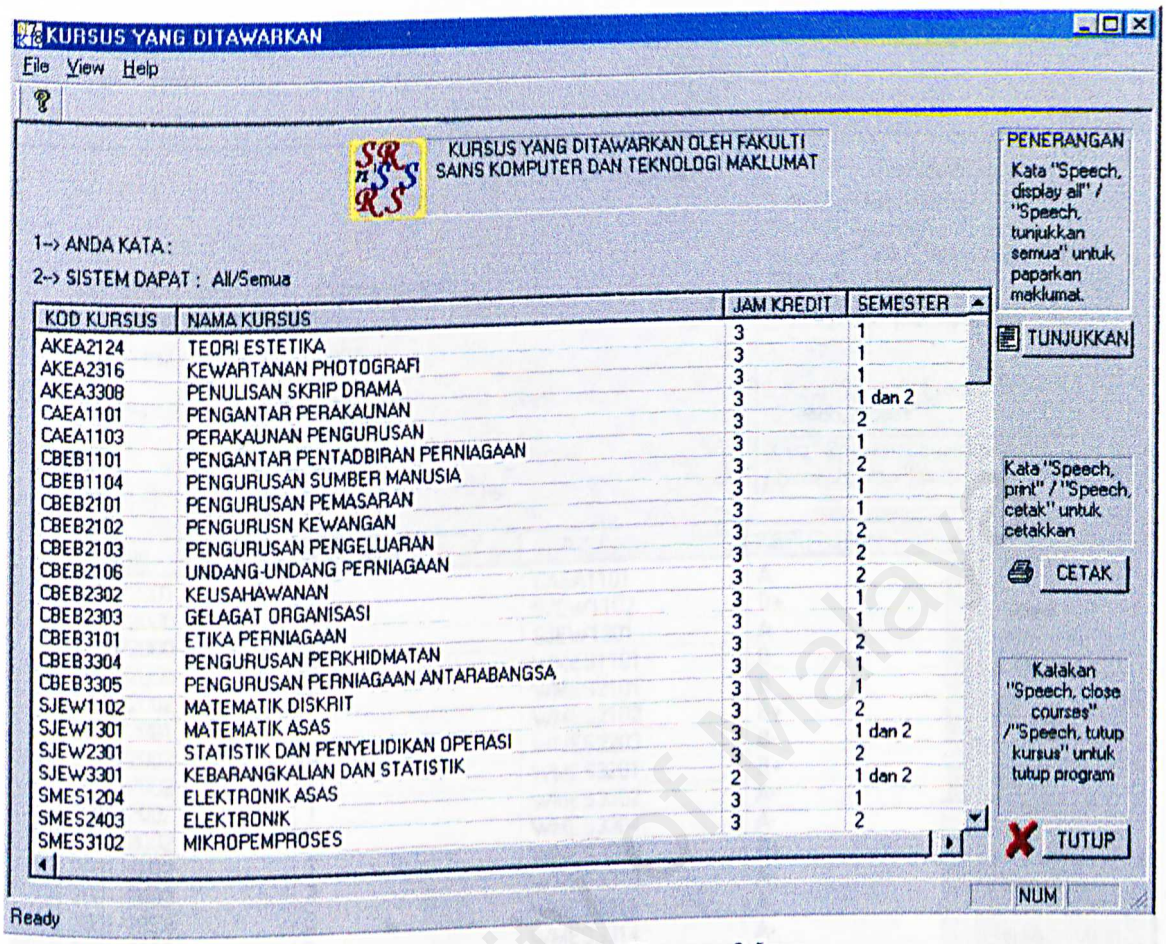


Figure C10 Semester Courses Menu

- In Semester Courses menu, user need to give 1 data for the system
    1. Subject code
- The list was sorted in ascending order base on subject code.

3.6 Result Menu

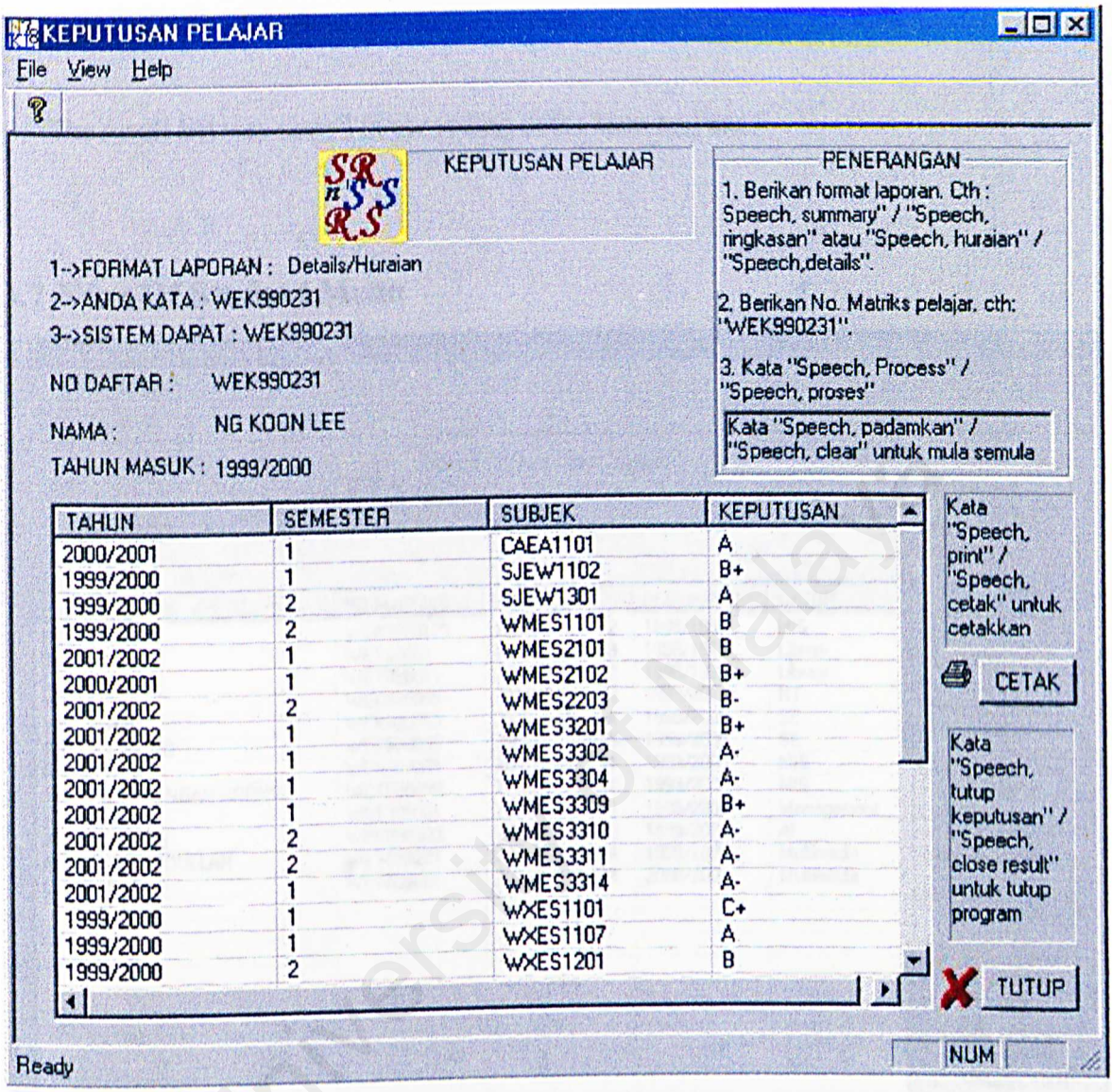


Figure C11 Result Menu

- In Result menu, user need to give 2 data for the system:
    1. Format of report – details or summary
    2. Student metrics number

- System will process if the data given exists in the database.
- As return, the information display to the user including
    1. Student's metrics number
    2. Name

3. Student's year intake
4. Result in GPA

- The result list was sorted in ascending order base on year.

3.7 Year Of Student Menu

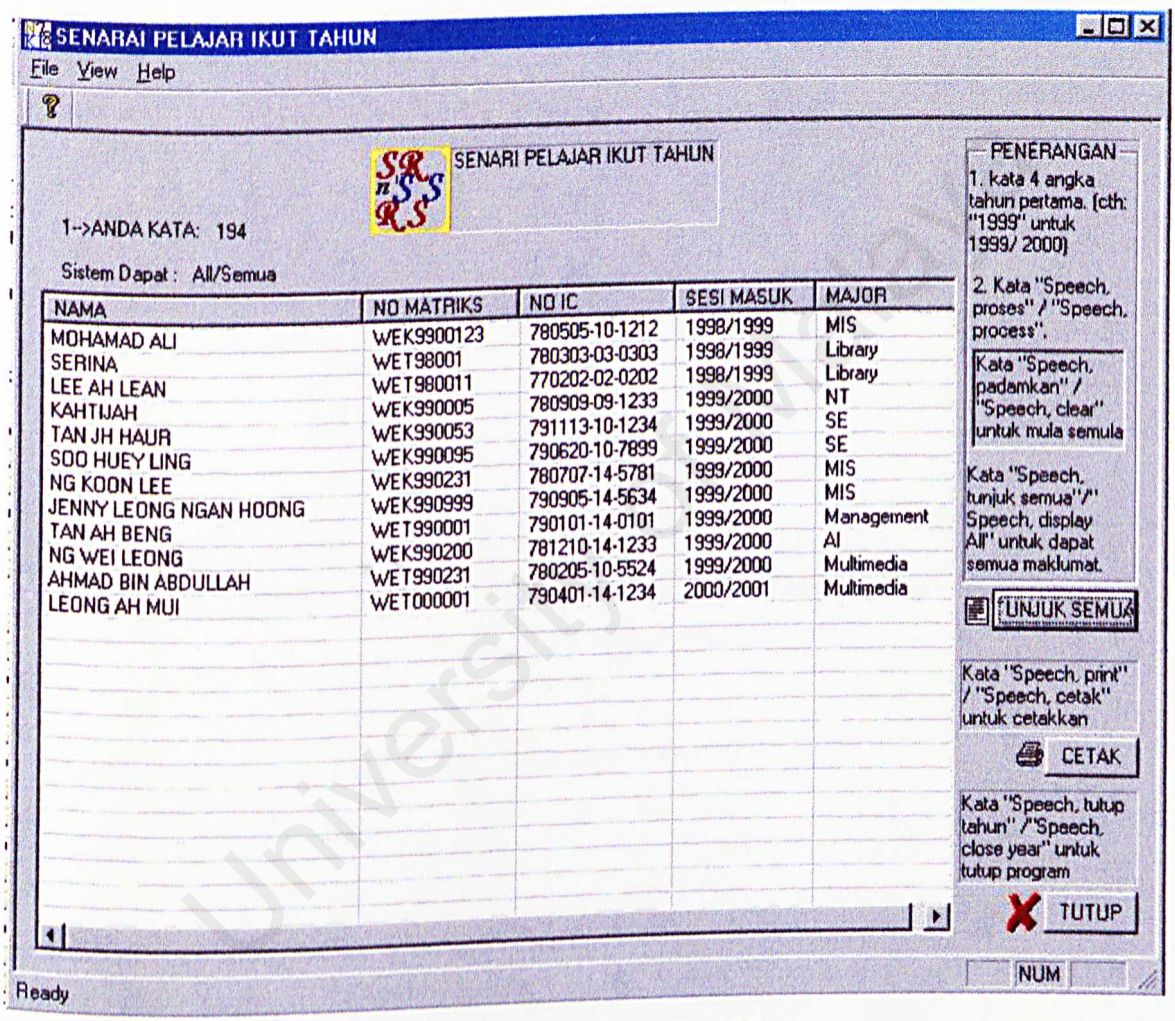


Figure C12 Year Of Student Menu

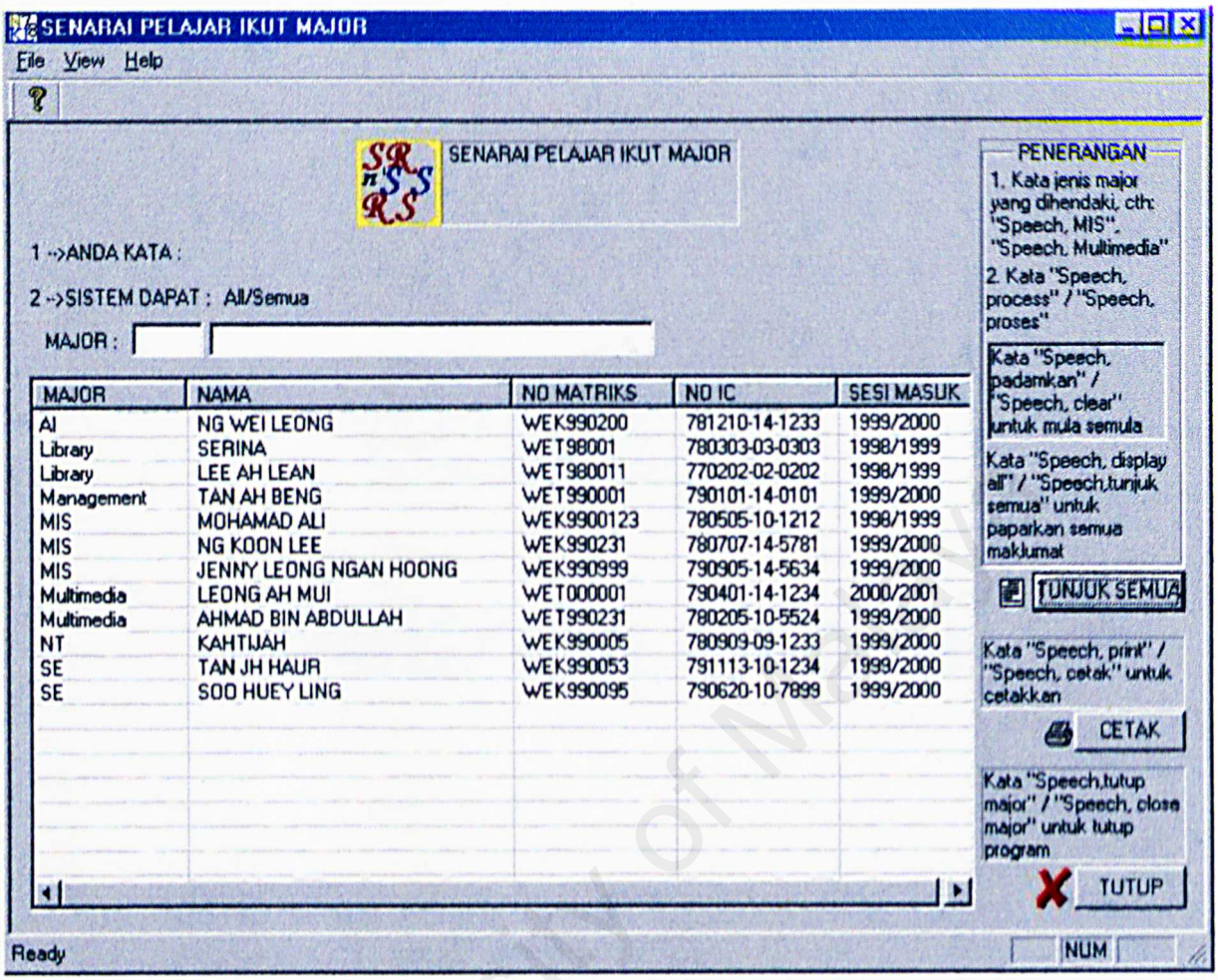
- In this menu, user need to give 1 data for the system:
    1. The year intake from 1997 to 2002

- System will process if the data given exists in the database.

- As return, the information display to the user including
    1. Student's Name
    2. Metrics number
    3. IC number
    4. Year intake
    5. Student's major

- The result list was sorted in ascending order base on year intake.

3.8 Major Menu



MAJOR	NAMA	NO MATRIKS	NO IC	SESI MASUK
AI	NG WEI LEONG	WEK990200	781210-14-1233	1999/2000
Library	SERINA	WET98001	780303-03-0303	1998/1999
Library	LEE AH LEAN	WET980011	770202-02-0202	1998/1999
Management	TAN AH BENG	WET990001	790101-14-0101	1999/2000
MIS	MOHAMAD ALI	WEK9900123	780505-10-1212	1998/1999
MIS	NG KOON LEE	WEK990231	780707-14-5781	1999/2000
MIS	JENNY LEONG NGAN HOONG	WEK990999	790905-14-5634	1999/2000
Multimedia	LEONG AH MUI	WET000001	790401-14-1234	2000/2001
Multimedia	AHMAD BIN ABDULLAH	WET990231	780205-10-5524	1999/2000
NT	KAHTIJAH	WEK990005	780909-09-1233	1999/2000
SE	TAN JH HAUR	WEK990053	791113-10-1234	1999/2000
SE	SOO HUEY LING	WEK990095	790620-10-7899	1999/2000

Figure C13 Major Menu

- In major menu, user need to give 1 data for the system:
    1. The major name

- System will process if the data given exists in the database.
- As return, the information display to the user including
    1. Student's major
    2. Student's Name
    3. Metrics number
    4. IC number
    5. Year intake

- The result list was sorted in ascending order base on majoring.

Example Source Code

Example Source Code

4.0 Main Menu

4.0.1 MainMenu.h

```
// MainMenu.h : main header file for the MAINMENU application
//

#if !defined(AFX_MAINMENU_H__7D0CDD65_8338_11D5_A1DE_C2E5AB5D252B__INCLUDED_)
#define AFX_MAINMENU_H__7D0CDD65_8338_11D5_A1DE_C2E5AB5D252B__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#ifndef __AFXWIN_H__
	#error include 'stdafx.h' before including this file for PCH
#endif

#include "resource.h"		// main symbols

/////////////////////////////////////////////////////////////////////////////
// CMainMenuApp:
// See MainMenu.cpp for the implementation of this class
//

class CMainMenuApp : public CWinApp
{
public:
	void TestSapi();
	CMainMenuApp();

// Overrides
	// ClassWizard generated virtual function overrides
	//{{AFX_VIRTUAL(CMainMenuApp)
	public:
	virtual BOOL InitInstance();
	//}}AFX_VIRTUAL

// Implementation

	//{{AFX_MSG(CMainMenuApp)
		// NOTE - the ClassWizard will add and remove member functions here.
		//    DO NOT EDIT what you see in these blocks of generated code !
	//}}AFX_MSG
	DECLARE_MESSAGE_MAP()
};


/////////////////////////////////////////////////////////////////////////////

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX_MAINMENU_H__7D0CDD65_8338_11D5_A1DE_C2E5AB5D252B__INCLUDED_)
```

4.0.2 MainMenu.cpp

```
// MainMenu.cpp : Defines the class behaviors for the application.
//

#include "stdafx.h"
#include "MainMenu.h"
#include "MainMenuDlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CMainMenuApp

BEGIN_MESSAGE_MAP(CMainMenuApp, CWinApp)
        //{{AFX_MSG_MAP(CMainMenuApp)
                // NOTE - the ClassWizard will add and remove mapping macros here.
                //    DO NOT EDIT what you see in these blocks of generated code!
        //}}AFX_MSG
        ON_COMMAND(ID_HELP, CWinApp::OnHelp)
        ON_MESSAGE(WM_USER+100, TestSapi)
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CMainMenuApp construction

CMainMenuApp::CMainMenuApp()
{
        // TODO: add construction code here,
        // Place all significant initialization in InitInstance
}

/////////////////////////////////////////////////////////////////////////////
// The one and only CMainMenuApp object

CMainMenuApp theApp;

/////////////////////////////////////////////////////////////////////////////
// CMainMenuApp initialization

BOOL CMainMenuApp::InitInstance()
{
                                                        //236,233,216
        SetDialogBkColor(RGB(200, 225, 255), RGB(0, 0, 0)); //background color  //text color
        // i start add my speech code here
        AfxOleInit();  // this allows the dialog to use COM objects

        AfxEnableControlContainer();

        // Standard initialization
        // If you are not using these features and wish to reduce the size
        //  of your final executable, you should remove from the following
        //  the specific initialization routines you do not need.

#ifdef _AFXDLL
        Enable3dControls();                     // Call this when using MFC in a shared DLL
#else
        Enable3dControlsStatic();       // Call this when linking to MFC statically
#endif

        CMainMenuDlg dlg;
        m_pMainWnd = &dlg;
        int nResponse = dlg.DoModal();
        if (nResponse == IDOK)
        {
                // TODO: Place code here to handle when the dialog is
                //  dismissed with OK
```

```
        }
        else if (nResponse == IDCANCEL)
        {
                // TODO: Place code here to handle when the dialog is
                //  dismissed with Cancel
        }

        // Since the dialog has been closed, return FALSE so that we exit the
        //  application, rather than start the application's message pump.
        return FALSE;
}

void CMainMenuApp::TestSapi()
{

        AfxMessageBox("Got It");
}
```

4.0.3 MainMenuDlg.h

```
// MainMenuDlg.h : header file
//

#if !defined(AFX_MAINMENUDLG_H__7D0CDD67_8338_11D5_A1DE_C2E5AB5D252B__INCLUDED_)
#define AFX_MAINMENUDLG_H__7D0CDD67_8338_11D5_A1DE_C2E5AB5D252B__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#include <sphelper.h>

/////////////////////////////////////////////////////////////////////////////
// CMainMenuDlg dialog

class CMainMenuDlg : public CDialog
{
// Construction
public:
        void SpeechRestore();
        void SpeechMaximize();
        void SpeechMinimize();
        CMainMenuDlg(CWnd* pParent = NULL);  // standard constructor

// Dialog Data
        //{{AFX_DATA(CMainMenuDlg)
        enum { IDD = IDD_MAINMENU_DIALOG };
                // NOTE: the ClassWizard will add data members here
        //}}AFX_DATA

        // ClassWizard generated virtual function overrides
        //{{AFX_VIRTUAL(CMainMenuDlg)
        protected:
        virtual void DoDataExchange(CDataExchange* pDX);  // DDX/DDV support
        //}}AFX_VIRTUAL

// Implementation
protected:
        void OnDestroy();
        void FinishSapi();
        void ExecuteCommand(ISpPhrase *pPhrase);
        bool InitializeSapi();

        // i put the speech code here
        CComPtr<ISpRecognizer> g_cpEngine;      // speech engine
        CComPtr<ISpRecoContext> g_cpRecoCtxt;   // speech recognition context
        CComPtr<ISpRecoGrammar> g_cpCmdGrammar;  // speech grammar
        CComPtr<ISpVoice>               g_cpVoice;      // TTS
```

```
        HICON m_hIcon;

        // Generated message map functions
        //{{AFX_MSG(CMainMenuDlg)
        virtual BOOL OnInitDialog();
        afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
        afx_msg void OnPaint();
        afx_msg HCURSOR OnQueryDragIcon();
        virtual void OnCancel();
        afx_msg void OnDataPelajar();
        afx_msg void OnSubjek();
        afx_msg void OnKursusSemester();
        afx_msg void OnKeputusan();
        afx_msg void OnTahunPelajar();
        afx_msg void OnMajor();
        afx_msg void OnPenerangan();
        //}}AFX_MSG
        afx_msg LRESULT OnRecoEvent(WPARAM,LPARAM);  // i add the speech code here,for SAPI handle
        DECLARE_MESSAGE_MAP()
};

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX_MAINMENUDLG_H__7D0CDD67_8338_11D5_A1DE_C2E5AB5D252B__INCLUDED_)
```

4.0.4 MainMenuDlg.cpp

```
 // MainMenuDlg.cpp : implementation file
//

#include "stdafx.h"
#include "MainMenu.h"
#include "MainMenuDlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

#include "main.h"

#define WM_RECOEVENT WM_USER+1

/////////////////////////////////////////////////////////////////////////////
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
        CAboutDlg();

// Dialog Data
        //{{AFX_DATA(CAboutDlg)
        enum { IDD = IDD_ABOUTBOX };
        //}}AFX_DATA

        // ClassWizard generated virtual function overrides
        //{{AFX_VIRTUAL(CAboutDlg)
        protected:
        virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
        //}}AFX_VIRTUAL

// Implementation
protected:
        //{{AFX_MSG(CAboutDlg)
```

```
	//}}AFX_MSG
	DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
	//{{AFX_DATA_INIT(CAboutDlg)
	//}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
	//{{AFX_DATA_MAP(CAboutDlg)
	//}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
	//{{AFX_MSG_MAP(CAboutDlg)
		// No message handlers
	//}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CMainMenuDlg dialog

CMainMenuDlg::CMainMenuDlg(CWnd* pParent /*=NULL*/)
	: CDialog(CMainMenuDlg::IDD, pParent)
{
	//{{AFX_DATA_INIT(CMainMenuDlg)
		// NOTE: the ClassWizard will add member initialization here
	//}}AFX_DATA_INIT
	// Note that LoadIcon does not require a subsequent DestroyIcon in Win32
	m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CMainMenuDlg::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
	//{{AFX_DATA_MAP(CMainMenuDlg)
		// NOTE: the ClassWizard will add DDX and DDV calls here
	//}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CMainMenuDlg, CDialog)
	//{{AFX_MSG_MAP(CMainMenuDlg)
	ON_WM_SYSCOMMAND()
	ON_WM_PAINT()
	ON_WM_QUERYDRAGICON()
	ON_BN_CLICKED(IDC_BUTTON1, OnDataPelajar)
	ON_BN_CLICKED(IDC_BUTTON2, OnSubjek)
	ON_BN_CLICKED(IDC_BUTTON3, OnKursusSemester)
	ON_BN_CLICKED(IDC_BUTTON4, OnKeputusan)
	ON_BN_CLICKED(IDC_BUTTON5, OnTahunPelajar)
	ON_BN_CLICKED(IDC_BUTTON6, OnMajor)
	ON_WM_DESTROY()  // need to add
	ON_BN_CLICKED(IDC_PENERANGAN, OnPenerangan)
	//}}AFX_MSG_MAP
	ON_MESSAGE(WM_RECOEVENT,OnRecoEvent) // add manually
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CMainMenuDlg message handlers

BOOL CMainMenuDlg::OnInitDialog()
{
	CDialog::OnInitDialog();

	// Add "About..." menu item to system menu.
```

```
	// IDM_ABOUTBOX must be in the system command range.
	ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
	ASSERT(IDM_ABOUTBOX < 0xF000);

	CMenu* pSysMenu = GetSystemMenu(FALSE);
	if (pSysMenu != NULL)
	{
		CString strAboutMenu;
		strAboutMenu.LoadString(IDS_ABOUTBOX);
		if (!strAboutMenu.IsEmpty())
		{
			pSysMenu->AppendMenu(MF_SEPARATOR);
			pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
		}
	}

	// Set the icon for this dialog.  The framework does this automatically
	//  when the application's main window is not a dialog
	SetIcon(m_hIcon, TRUE);			// Set big icon
	SetIcon(m_hIcon, FALSE);		// Set small icon

	if(!InitializeSapi())
	{
		FinishSapi();
		PostMessage(WM_CLOSE);
	}

	USES_CONVERSION;
	CString welcome = "Selamat Datang Ke Speech Repository System";
	CString welcome2= "Nama Saya ialah Speech";
	CString welcome3= "Sila Mulakan";
	g_cpVoice->Speak(T2W(welcome),SPF_ASYNC,NULL);
	g_cpVoice->Speak(T2W(welcome2),SPF_ASYNC,NULL);
	g_cpVoice->Speak(T2W(welcome3),SPF_ASYNC,NULL);


	// TODO: Add extra initialization here

	return TRUE;  // return TRUE  unless you set the focus to a control
}

void CMainMenuDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
	if ((nID & 0xFFF0) == IDM_ABOUTBOX)
	{
		CAboutDlg dlgAbout;
		dlgAbout.DoModal();
	}
	else
	{
		CDialog::OnSysCommand(nID, lParam);
	}
}

// If you add a minimize button to your dialog, you will need the code below
//  to draw the icon.  For MFC applications using the document/view model,
//  this is automatically done for you by the framework.

void CMainMenuDlg::OnPaint()
{
	if (IsIconic())
	{
		CPaintDC dc(this); // device context for painting

		SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

		// Center icon in client rectangle
		int cxIcon = GetSystemMetrics(SM_CXICON);
		int cyIcon = GetSystemMetrics(SM_CYICON);
		CRect rect;
```

```
                GetClientRect(&rect);
                int x = (rect.Width() - cxIcon + 1) / 2;
                int y = (rect.Height() - cyIcon + 1) / 2;

                // Draw the icon
                dc.DrawIcon(x, y, m_hIcon);
        }
        else
        {
                CDialog::OnPaint();
        }
}

// The system calls this to obtain the cursor to display while the user drags
//  the minimized window.
HCURSOR CMainMenuDlg::OnQueryDragIcon()
{
        return (HCURSOR) m_hIcon;
}

void CMainMenuDlg::OnCancel()
{
        // TODO: Add extra cleanup here

        CDialog::OnCancel();
}

void CMainMenuDlg::OnDataPelajar()
{
        // TODO: Add your control notification handler code here
        USES_CONVERSION;
        g_cpVoice->Speak(T2W("process"),SPF_ASYNC,NULL);
        ShellExecute(NULL,"Open","C:\\Program Files\\WEK 990231\\SRn'SSRS
1.0\\PendaftaranPelajar.exe",NULL,NULL,SW_SHOWNORMAL);

}

void CMainMenuDlg::OnSubjek()
{
        // TODO: Add your control notification handler code here
        USES_CONVERSION;
        g_cpVoice->Speak(T2W("process"),SPF_ASYNC,NULL);
        ShellExecute(NULL,"Open","C:\\Program Files\\WEK 990231\\SRn'SSRS
1.0\\SenaraiPelajarIkutKursus.exe",NULL,NULL,SW_SHOWNORMAL);
}

void CMainMenuDlg::OnKursusSemester()
{
        // TODO: Add your control notification handler code here
        USES_CONVERSION;
        g_cpVoice->Speak(T2W("process"),SPF_ASYNC,NULL);
        ShellExecute(NULL,"Open","C:\\Program Files\\WEK 990231\\SRn'SSRS
1.0\\KursusDitawarkan.exe",NULL,NULL,SW_SHOWNORMAL);
}

void CMainMenuDlg::OnKeputusan()
{
        // TODO: Add your control notification handler code here
        USES_CONVERSION;
        g_cpVoice->Speak(T2W("process"),SPF_ASYNC,NULL);
        ShellExecute(NULL,"Open","C:\\Program Files\\WEK 990231\\SRn'SSRS
1.0\\KeputusanPelajar.exe",NULL,NULL,SW_SHOWNORMAL);
}

void CMainMenuDlg::OnTahunPelajar()
{
        // TODO: Add your control notification handler code here
        USES_CONVERSION;
        g_cpVoice->Speak(T2W("process"),SPF_ASYNC,NULL);
```

```
	ShellExecute(NULL,"Open","C:\\Program Files\\WEK 990231\\SRn'SSRS
1.0\\PelajarIkutTahun.exe",NULL,NULL,SW_SHOWNORMAL);
}

void CMainMenuDlg::OnMajor()
{
	// TODO: Add your control notification handler code here
	//MessageBox("Kamu telah memilih  fungsi <<Major>>.");
	USES_CONVERSION;
	g_cpVoice->Speak(T2W("process"),SPF_ASYNC,NULL);
	ShellExecute(NULL,"Open","C:\\Program Files\\WEK 990231\\SRn'SSRS
1.0\\PelajarIkutMajor.exe",NULL,NULL,SW_SHOWNORMAL);
}

bool CMainMenuDlg::InitializeSapi()
{
	if (FAILED(CoInitialize(NULL))) {
		AfxMessageBox("Error starting COM");
		return false;
	}

	HRESULT hRes = g_cpEngine.CoCreateInstance(CLSID_SpSharedRecognizer);

	if (FAILED(hRes)) {
		AfxMessageBox("Error starting SAPI");
		return false;
	}

	hRes = g_cpEngine->CreateRecoContext(&g_cpRecoCtxt);

	if (FAILED(hRes)) {
		AfxMessageBox("Error creating context");
		return false;
	}

	hRes = g_cpRecoCtxt->SetNotifyWindowMessage(m_hWnd, WM_RECOEVENT, 0, 0);

	if (FAILED(hRes)) {
		AfxMessageBox("Error creating notification window");
		return false;
	}

	hRes = g_cpRecoCtxt->SetInterest(SPFEI(SPEI_RECOGNITION), SPFEI(SPEI_RECOGNITION));

	if (FAILED(hRes)) {
		AfxMessageBox("Error creating interest...seriously");
		return false;
	}

	hRes = g_cpRecoCtxt->CreateGrammar(0, &g_cpCmdGrammar);
	if (FAILED(hRes)) {
		AfxMessageBox("Error creating grammar");
		return false;
	}

	hRes = g_cpCmdGrammar->LoadCmdFromResource(
		NULL,
		MAKEINTRESOURCEW(IDR_SAPI0),
		L"SRGRAMMAR",
		MAKELANGID( LANG_NEUTRAL, SUBLANG_NEUTRAL), SPLO_DYNAMIC);

	if (FAILED(hRes)) {
		AfxMessageBox("Error creating grammar rules");
		return false;
	}

	// Set rules to active, we are now listening for commands
	hRes = g_cpCmdGrammar->SetRuleState(NULL, NULL, SPRS_ACTIVE );
	if (FAILED(hRes)) {
		AfxMessageBox("Error setting rule state");
```

```
                return false;
        }

        hRes = g_cpRecoCtxt->GetVoice(&g_cpVoice);
        if(FAILED(hRes))
        {
                AfxMessageBox("Error creating voice");
                return false;
        }

        return true;

}

LRESULT CMainMenuDlg::OnRecoEvent(WPARAM wParam, LPARAM lParam) {
   // Event helper class
   CSpEvent event;

   // Loop processing events while there are any in the queue
   while (event.GetFrom(g_cpRecoCtxt) == S_OK)
   {
      // Look at recognition event only
      switch (event.eEventId)
      {
                case SPEI_RECOGNITION:
                        ExecuteCommand(event.RecoResult());
                        break;
      }
   }

        return 0;
}

void CMainMenuDlg::ExecuteCommand(ISpPhrase *pPhrase)
{
        SPPHRASE *pElements;

        // Get the phrase elements, one of which is the rule id we specified in
   // the grammar.  Switch on it to figure out which command was recognized.
   if (SUCCEEDED(pPhrase->GetPhrase(&pElements)))
   {
      switch ( pElements->Rule.ulId )
      {
        case VID_MainCommand:
          {
                        const SPPHRASEPROPERTY *pProp = pElements->pProperties;

                        while (pProp) {
                                switch(pProp->vValue.ulVal )
                                {
                                        case VID_DataPelajar:
                                        case VID_StudentInformation:
                                                OnDataPelajar();
                                                break;

                                        case VID_Subjek:
                                        case VID_Subject:
                                                OnSubjek();
                                                break;

                                        case VID_KursusSemester:
                                        case VID_SemesterCourses:
                                                OnKursusSemester();
                                                break;

                                        case VID_Keputusan:
                                        case VID_Result:
                                                OnKeputusan();
                                                break;
```

```
                                        case VID_TahunPelajar:
                                        case VID_TahunPelajar2:
                                        case VID_YearOfStudent:
                                                OnTahunPelajar();
                                                break;

                                        case VID_Major:
                                                OnMajor();
                                                break;

                                        case VID_Close:
                                        case VID_Tutup:
                                                OnCancel();
                                                break;

                                        case VID_Minimize:
                                        case VID_MinimizeMalay:
                                                SpeechMinimize();
                                                break;

                                        case VID_Maximize:
                                        case VID_MaximizeMalay:
                                                SpeechMaximize();
                                                break;

                                        case VID_Restore:
                                        case VID_RestoreMalay:
                                                SpeechRestore();
                                                break;

                                        case VID_Help:
                                        case VID_Tolong:
                                                OnPenerangan();
                                                break;
                                }

                                pProp = pProp->pNextSibling;
                        }
        } break;
    }

    // Free the pElements memory which was allocated for us
    ::CoTaskMemFree(pElements);
  }
}
void CMainMenuDlg::FinishSapi()
{
        // Release grammar, if loaded
  if (g_cpCmdGrammar) {
    g_cpCmdGrammar.Release();
  }
        // Release recognition context, if created
  if (g_cpRecoCtxt) {
    g_cpRecoCtxt->SetNotifySink(NULL);
    g_cpRecoCtxt.Release();
  }
        // Release recognition engine instance, if created
        if (g_cpEngine) {
                g_cpEngine.Release();
        }
}
void CMainMenuDlg::OnDestroy()
{
        CDialog::OnDestroy();
```

```
        FinishSapi();
}

void CMainMenuDlg::SpeechMinimize()
{
        AfxGetMainWnd()->ShowWindow(SW_SHOWMINIMIZED);  // minimize mainframe
}

void CMainMenuDlg::SpeechMaximize()
{
        AfxGetMainWnd()->ShowWindow(SW_SHOWMAXIMIZED);    // maximize mainframe
}
void CMainMenuDlg::SpeechRestore()
{
        AfxGetMainWnd()->ShowWindow(SW_RESTORE);
}

void CMainMenuDlg::OnPenerangan()
{
        // TODO: Add your control notification handler code here
        CAboutDlg dlg;
        dlg.DoModal();
}
```

4.1 Header Used for Speech

4.1.1 Main.h

```
#define VID_DataPelajar 91
#define VID_Subjek 92
#define VID_KursusSemester 93
#define VID_Keputusan 94
#define VID_TahunPelajar 95
#define VID_Major 96
#define VID_TahunPelajar2 97
#define VID_StudentInformation 101
#define VID_Subject 102
#define VID_SemesterCourses 103
#define VID_Result 104
#define VID_YearOfStudent 105
#define VID_MainCommand 150
#define VID_Function 13
#define VID_Help 214
#define VID_Tolong 215
#define VID_Minimize 220
#define VID_Maximize 221
#define VID_MinimizeMalay 222
#define VID_MaximizeMalay 223
#define VID_Restore 224
#define VID_RestoreMalay 225
#define VID_Close 255
#define VID_Tutup 256
```

Main.h was used for speech to recognize the "VID_Name" in MainMenu program. For different of function there are different of name. Example like:

- PendaftaranSpeech.h

- SenaraiSpeech.h
- KursusDitawarkanSpeech.h
- KeputusanSpeech.h
- PelajarTahunSpeech.h
- PelajarMajorSpeech.h

This kind of header file create by grammar compiler. It will translate user's XML source code to *.cfg (context free grammar) and *.h header file. *.cfg file was in binary format and can't be show here.

4.1.2 Main.xml

```
<GRAMMAR LANGID="409">
  <DEFINE>
        <ID NAME="VID_DataPelajar" VAL="91"/>
        <ID NAME="VID_Subjek" VAL="92"/>
        <ID NAME="VID_KursusSemester" VAL="93"/>
        <ID NAME="VID_Keputusan" VAL="94"/>
        <ID NAME="VID_TahunPelajar" VAL="95"/>
        <ID NAME="VID_Major" VAL="96"/>
        <ID NAME="VID_TahunPelajar2" VAL="97"/>

      <ID NAME="VID_StudentInformation" VAL="101"/>
        <ID NAME="VID_Subject" VAL="102"/>
        <ID NAME="VID_SemesterCourses" VAL="103"/>
        <ID NAME="VID_Result" VAL="104"/>
        <ID NAME="VID_YearOfStudent" VAL="105"/>

        <ID NAME="VID_MainCommand" VAL="150"/>
        <ID NAME="VID_Function" VAL="13"/>

        <ID NAME="VID_Help" VAL="214"/>
        <ID NAME="VID_Tolong" VAL="215"/>

        <ID NAME="VID_Minimize" VAL="220"/>
        <ID NAME="VID_Maximize" VAL="221"/>
        <ID NAME="VID_MinimizeMalay" VAL="222"/>
        <ID NAME="VID_MaximizeMalay" VAL="223"/>
        <ID NAME="VID_Restore" VAL="224"/>
        <ID NAME="VID_RestoreMalay" VAL="225"/>

        <ID NAME="VID_Close" VAL="255"/>
      <ID NAME="VID_Tutup" VAL="256"/>
  </DEFINE>

<RULE ID="VID_MainCommand" TOPLEVEL="ACTIVE">
  <P>Speech</P>
  <O>please</O>
  <O>open</O>
  <O>run</O>
  <O>execute</O>
    <O>sila</O>
    <O>mulakan</O>
    <O>mula</O>
```

```
  <P>
    <RULEREF REFID="VID_Function"/>
  </P>
</RULE>

<RULE ID="VID_Function">
  <L PROPID="VID_Function">
    <P VAL="VID_DataPelajar">maklumat<O>per la jar</O></P>
    <P VAL="VID_Subjek">Subjek</P>
    <P VAL="VID_KursusSemester">Kursus Semester</P>
    <P VAL="VID_Keputusan">Keputusan</P>
    <P VAL="VID_TahunPelajar">Tar phun<O>per la jar</O></P>
    <P VAL="VID_TahunPelajar2">Tar phune</P>
    <P VAL="VID_Major">Major</P>

    <P VAL="VID_StudentInformation">Student Information</P>
    <P VAL="VID_Subject">Subject</P>
    <P VAL="VID_SemesterCourses">Semester Courses</P>
    <P  VAL="VID_Result"><O>Student</O>Result</P>
    <P VAL="VID_YearOfStudent"> Year Of Student</P>

    <P VAL="VID_Close">Close main<O> program</O></P>
    <P VAL="VID_Tutup">Tutup<O> program </O>utama</P>

        <P VAL="VID_Minimize">minimize main<O> program</O></P>
        <P VAL="VID_MinimizeMalay">minimize <O> program </O>utama</P>

        <P VAL="VID_Maximize">maximize main<O> program</O></P>
        <P VAL="VID_MaximizeMalay">maximize <O> program </O>utama</P>

        <P VAL="VID_Restore">restore main<O> program</O></P>
        <P VAL="VID_RestoreMalay">restore <O> program </O>utama</P>

        <P VAL="VID_Help">Help <O>me</O></P>
        <P VAL="VID_Tolong">Tor llong <O> saya</O></P>
  </L>
</RULE>


</GRAMMAR>
```

In order to create Main.h and Main.cfg for system to recognize our command, we need to use Grammer Compiler and compile it. Next we should repeat the compile process but this time we run in dos mode. Go to the program path where the Main.xml place, and type this command:

gc.exe /o main.cfg /h main.h main.xml

If compile success, it will appear main.h and main.cfg inside the same directory as main.xml placed. So it means before the compilation, main.h and main.cfg was not appear in that path.