

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# โปรแกรมการสร้างภาพเคลื่อนไหวจากข้อความภาษาไทย

## GENERATING 2D GRAPHIC ANIMATION FROM THAI TEXT

โดย



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต  
สาขาวิชาเทคโนโลยีสารสนเทศ  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ภาคเรียนที่ 2 ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# GENERATING 2D GRAPHIC ANIMATION FROM THAI TEXT



A PROJECT SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY  
FACULTY OF INFORMATION TECNOLOGY  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/ 2006

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**COPYRIGHT 2007**

**FACULTY OF INFORMATION TECHNOLOGY**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองปริญญาโท ประจำปีการศึกษา 2549  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมการสร้างภาพเคลื่อนไหวจากข้อความภาษาไทย  
GENERATING 2D GRAPHIC ANIMATION FROM THAI TEXT

ผู้จัดทำ

1. นางสาววรรณพร เตียเจริญกิจ รหัสประจำตัว 46060012
2. นายวิรัช อัครวิธานนท์ รหัสประจำตัว 46060040
3. นางสาวราภรณ์ พันโท รหัสประจำตัว 46060084

.....อาจารย์ที่ปรึกษา  
(อาจารย์สุพัฒนา โขติพันธ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	โปรแกรมสร้างภาพเคลื่อนไหวจากข้อความภาษาไทย (Generating 2D Graphic Animation from Thai Text)
นักศึกษา	นางสาว วรรณพร เตียเจริญกิจ 46060012 นาย วีรธร อัสวธิตานนท์ 46060040 นางสาว วราภรณ์ พันโท 46060084
ปริญญา	วิทยาศาสตรบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
ปีการศึกษา	2549
อาจารย์ที่ปรึกษา	อาจารย์สุพัฒน์ดา โชติพันธ์

### บทคัดย่อ

โครงการฉบับนี้เป็นการนำเสนอโปรแกรมการแสดงผลภาพเคลื่อนไหว 2 มิติจากข้อความภาษาไทย เมื่อโปรแกรมรับข้อมูลเข้ามาเป็นประโยคภาษาไทย โปรแกรมจะทำการตีความด้วยการวิเคราะห์คำและวิเคราะห์ไวยากรณ์ของประโยค แล้วแสดงผลภาพเคลื่อนไหวที่สอดคล้องตามความหมายของประโยคนั้นๆ ที่สำคัญ โปรแกรมสามารถรับชุดประโยคแล้วแสดงผลออกเป็นเรื่องราวได้ ด้วยความสามารถดังกล่าว จึงสามารถนำโปรแกรมนี้ไปประยุกต์ใช้เป็นสื่อการเรียนการสอนเพื่อความบันเทิงได้ แต่ถึงอย่างไรก็ตาม โปรแกรมนี้สามารถรองรับได้เฉพาะประโยคภาษาไทยที่มีรูปแบบอย่างง่ายเท่านั้น

<b>Title</b>	Generating 2D Graphic Animation from Thai Text	
<b>Student</b>	Ms. Wannapor Tiacharoenkit	46060012
	Mr. Weerathorn Asavathitanonta	46060040
	Ms. Waraporn Punto	46060084
<b>Degree</b>	Bachelor of Science	
<b>Program</b>	Information Technology	
<b>Academic Year</b>	2006	
<b>Advisor</b>	Ms. Supannada Chotiphan	

## ABSTRACT

This project proposes the application that can generate 2D graphic animation from Thai text. After Thai sentences are input into the application, it will be interpreted by using lexical and syntax analysis in order to create animation relevant to the true meaning of those sentences. Importantly, it can receive a series of sentences to automatically illustrate the picture story. This capability can be applied to use in telling a tale multimedia entertainer. Unfortunately, the limitation of this application is to support only the simple sentences.

## กิตติกรรมประกาศ

โครงการโปรแกรมสร้างภาพเคลื่อนไหว จากข้อความภาษาไทยนี้ ประสบความสำเร็จได้ ด้วยความช่วยเหลือและการสนับสนุนจากบุคคลหลายท่าน ผู้เขียนใคร่ขอแสดงความระลึกถึงบุคคลสำคัญผู้อยู่เบื้องหลัง ดังต่อไปนี้

คุณพ่อและคุณแม่ ที่เปิดโอกาสให้ได้ทำงานอย่างเต็มที่ และคอยให้การสนับสนุนตลอดมา อาจารย์สุพัฒน์ดา ชาติพันธ์ อาจารย์ที่ปรึกษาโครงการ ที่กรุณาให้คำปรึกษา ให้กำลังใจ และให้คำแนะนำต่างๆที่เป็นประโยชน์ ให้ความช่วยเหลือในทุกๆด้าน ทำให้โครงการนี้สำเร็จลุล่วงไปได้ด้วยดี

อาจารย์ธนิศา เครือไวยวรรณ ที่คอยเป็นที่ปรึกษา และช่วยเหลือด้านโปรแกรม เพื่อน ๆ ทุกคนที่คอยเป็นห่วงเป็นใย คอยดูแลกันและกัน ให้ความช่วยเหลือมาตลอด คณาจารย์คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆ ท่านที่ให้คำปรึกษาและคำชี้แนะในการพัฒนาโครงการนี้ให้สำเร็จลุล่วงลงด้วยดี

จึงใคร่ขอขอบคุณบุคคลดังกล่าวข้างต้นมา ณ โอกาสนี้

วรรณพร เตี้ยเจริญกิจ

วีรธร อัสวิตานนท์

วราภรณ์ พันโท

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญรูป.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของ โครงการงาน.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบเขต.....	1
1.4 กรอบแนวคิด.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 การใช้ภาษาจาวา.....	4
2.1.1 หลักการทำงานของภาษาจาวา.....	5
2.1.2 ภาษาจาวากับการเขียน โปรแกรมเชิงวัตถุ.....	6
2.1.3 ข้อดีของการใช้ภาษาจาวา.....	6
2.2 การใช้งานฐานข้อมูล.....	6
2.2.1 ความหมายของระบบฐานข้อมูล.....	8
2.2.2 โครงสร้างของฐานข้อมูล.....	8
2.2.3 รูปแบบของระบบฐานข้อมูล.....	9
2.2.4 ระบบการจัดการฐานข้อมูล (Database Management System, DBMS).....	10
2.2.5 ข้อดีข้อเสียของการใช้ฐานข้อมูล.....	11
2.3 การนำภาษาจาวามาใช้กับฐานข้อมูล.....	12
2.3.1 การทำงานที่สำคัญของเจดีบีซี.....	12
2.3.2 ลักษณะที่สำคัญของเจดีบีซี.....	12
2.3.3 ประเภทตัวติดตั้งของเจดีบีซี.....	13
2.3.4 ข้อดีของเจดีบีซี.....	13

## สารบัญ (ต่อ)

	หน้า
2.4 โครงสร้างของคอมไพเลอร์ .....	14
2.4.1 ขั้นตอนการทำงานของคอมไพเลอร์ .....	14
2.4.2 โครงสร้างของคอมไพเลอร์ .....	15
2.5 ส่วนการวิเคราะห์ของคอมไพเลอร์ .....	17
2.5.1 คำจำกัดความ .....	17
2.5.2 นิพจน์สามัญ (Regular Expression) .....	18
2.6 การวิเคราะห์รูปแบบประโยค (Syntax analysis) .....	19
2.6.1 บทบาทของพาร์เซอร์ (Parser) .....	19
2.6.2 วัตถุประสงค์ของส่วนจัดการข้อผิดพลาดในพาร์เซอร์ .....	20
2.6.3 หลักการคอนเท็กซ์ฟรีแกรมมา (Context-Free Grammar หรือ CFG) .....	20
2.6.4 การแจงส่วนชนิดล่างสู่ขึ้นบน (Bottom-Up Parsing) .....	20
2.7 การใช้งานไลบรารีเจเนอรัลและไลบรารีคัพ (Library JFlex, LibraryCUP) .....	21
2.7.1 โครงสร้างของไฟล์เจเนอรัล .....	22
2.7.2 ขั้นตอนการใช้งานของไลบรารีเจเนอรัลและไลบรารีคัพ .....	23
2.7.3 ตัวอย่าง code ของการทำคอนเท็กซ์ฟรีแกรมมา .....	24
2.8 หลักภาษาไทย .....	26
2.8.1 การแบ่งคำของภาษาไทย .....	26
2.8.2 การแบ่งประโยคของ .....	26
2.8.3 การเรียงคำในประโยค .....	27
2.8.4 การเรียงคำในประโยคของภาษา .....	28
บทที่ 3 การวิเคราะห์และออกแบบโครงงาน .....	29
3.1 วิเคราะห์ลักษณะการทำงานจากมุมมองผู้ใช้ .....	29
3.2 วิเคราะห์ส่วนแนวคิดของการทำงานระบบ .....	30
3.2.1 ส่วนรับข้อมูล .....	30
3.2.2 ส่วนขบวนการคิดของระบบ .....	30
3.2.3 ส่วนแสดงผล .....	31

## สารบัญ (ต่อ)

	หน้า
3.3 วิเคราะห์ส่วนแนวคิดขั้นตอนการทำงานของระบบ .....	31
3.3.1 แผนภาพการไหลของการทำงาน ( Flowchart Diagram ) .....	31
3.4 วิเคราะห์และออกแบบการพัฒนาโปรแกรม .....	35
3.4.1 แผนภาพของคลาส ( Class Diagram ) .....	35
3.4.2 แผนภาพแสดงลำดับ ( Sequence Diagram ) .....	43
บทที่ 4 การพัฒนาโปรแกรม .....	46
4.1 การพัฒนาคลาสของส่วนการทำงานต่างๆ .....	46
4.1.1 ส่วนรับค่า .....	47
4.1.2 ส่วนวิเคราะห์และแปลความประโยค .....	48
4.1.3 ส่วนติดต่อกับฐานข้อมูล .....	50
4.1.4 ส่วนการแสดงผล .....	51
4.2 การกำหนดรูปแบบไวยากรณ์ของประโยค .....	57
4.2.1 การกำหนดค่าและชนิดของค่า .....	58
4.2.2 การกำหนดรูปแบบของประโยค .....	59
บทที่ 5 คู่มือและตัวอย่างการทำงานของโปรแกรม .....	62
5.1 ส่วนใช้งานหลัก .....	62
5.2 การใช้งานเพิ่มเติมของโปรแกรม .....	68
5.2.1 ส่วนอธิบายการทำงานของโปรแกรม .....	68
5.2.2 ส่วนแนะนำตัวละคร .....	68
5.2.3 ส่วนคู่มือการใช้งาน .....	69
5.2.4 ส่วนแนะนำผู้จัดทำ .....	70
บทที่ 6 สรุปการทำงานของระบบ .....	71
6.1 ประโยชน์ของการสร้างโปรแกรมขึ้นมาใหม่ .....	71
6.2 ข้อดีของระบบ .....	72
6.3 ข้อจำกัดของระบบ .....	72

## สารบัญ (ต่อ)

	หน้า
6.4 ข้อเสนอแนะ.....	72
บรรณานุกรม .....	73
ภาคผนวก .....	74
ประวัติผู้จัดทำ.....	100



## สารบัญตาราง

ตารางที่	หน้า
2.1 ตารางพนักงาน.....	10
3.1 ตารางแสดงคลาส Interface.....	36
3.2 ตารางแสดงคลาส Flex.....	37
3.3 ตารางแสดงคลาส MapWord .....	38
3.4 ตารางแสดงคลาส ConnectDB.....	39
3.5 ตารางแสดงคลาส Manage.....	40
3.6 ตารางแสดงคลาส InformationActor .....	41
3.7 ตารางแสดงคลาส Animation.....	42
4.1 ตารางฐานข้อมูลเก็บข้อมูลของคำศัพท์ภาษาไทย และภาษาอังกฤษ.....	50



# สารบัญรูป

รูปที่	หน้า
1.1 แผนภาพแสดงแนวคิดในการสร้างโปรแกรม.....	2
2.1 แสดงการทำงานของเครื่องจักรสมมุติ .....	5
2.2 แสดงการทำงานของเครื่องดึงข้อมูลของฐานข้อมูล .....	7
2.3 ลำดับขั้นของหน่วยข้อมูลในระบบฐานข้อมูล .....	8
2.4 หลักการทำงานของคอมพิวเตอร์ .....	14
2.5 ขั้นตอนการทำงานของคอมพิวเตอร์ .....	15
2.6 โครงสร้างของคอมพิวเตอร์ .....	17
2.7 การทำงานของไลบรารีเจแปล็กซ์ (Library JFlex).....	21
2.8 การทำงานของ คอนเทคพีแกรมมา .....	23
3.1 การใช้งานระบบ .....	29
3.2 กรอบแนวคิด .....	30
3.3 แผนภาพการไหลของการทำงานโดยภาพรวมของระบบ .....	31
3.4 แผนภาพการไหลของการทำงานในส่วนการรับค่าจากผู้ใช้ .....	32
3.5 แผนภาพการไหลของการทำงานของส่วนวิเคราะห์และแปลความประโยค .....	32
3.6 แผนภาพการไหลของการทำงานของการติดต่อการดึงค่าจากฐานข้อมูล .....	33
3.7 แผนภาพการไหลของส่วนการแสดงผลภาพ .....	34
3.8 แผนภาพของคลาสของระบบ .....	35
3.9 แผนภาพแสดงลำดับของระบบ .....	43
4.1 แสดงหลักการการทำงานของโปรแกรม .....	47
4.2 แสดงการทำงานของส่วนการรับค่า .....	47
4.3 แสดงการทำงานของส่วนแปลความการรับค่า .....	48
4.4 แสดงการทำงานของส่วนการรับค่าคลาส Flex .....	49
4.5 แสดงการทำงานของส่วนติดต่อกับฐานข้อมูล .....	50
4.6 รูปแสดงการทำงาน คลาส ConnectDB .....	50
4.7 แสดงการทำงานของส่วนจัดการค่า .....	52
4.8 แสดงการทำงานของส่วนการแสดงผลภาพ .....	54
4.9 การจำลองรูปแบบการแสดงผลภาพลงบนหน้าจอ .....	55
4.10 แสดงโครงสร้างของการเก็บภาพ .....	56

## สารบัญรูป (ต่อ)

รูปที่	หน้า
5.1 หน้าจอของโปรแกรม .....	63
5.2 แสดงการเลือกภาพพื้นหลัง.....	63
5.3 แสดงตัวอย่างการใช้งานโปรแกรม .....	64
5.4 แสดงผลลัพธ์ที่เป็นภาพเคลื่อนไหวของประโยค “วัฏกระโดดที่ต้นไม้”.....	64
5.5 แสดงผลลัพธ์ที่เป็นภาพเคลื่อนไหวของประโยค “วัฏเดินไปหาแกะ” .....	65
5.6 แสดงผลลัพธ์ที่เป็นภาพเคลื่อนไหวของประโยค “วัฏเดินไปหาแกะ” (ภาพต่อเนื่อง).....	65
5.7 แสดงผลลัพธ์เมื่อกดปุ่ม “เริ่มใหม่” .....	66
5.8 แสดงข้อมูลที่ผู้ใช้กรอกทั้งหมด.....	66
5.9 แสดงการบันทึกข้อมูลทั้งหมด.....	67
5.10 แสดงการเปิดไฟล์และข้อมูลที่ได้จากการเปิดไฟล์ .....	67
5.11 แสดงหลักการทำงานของโปรแกรม .....	68
5.12 แสดงตัวละครในโปรแกรม .....	69
5.13 แสดงคู่มือการใช้งานโปรแกรม.....	69
5.14 แนะนำผู้จัดทำ.....	70

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของโครงการ

เนื่องจากเห็นว่าการสร้างโปรแกรมสร้างภาพเคลื่อนไหว จากข้อความภาษาไทย (Generating 2D Graphic Animation from Thai Text) จะสร้างภาพตามที่ผู้ใช้ป้อนข้อมูลเข้ามาในลักษณะภาพเคลื่อนไหว โดยโปรแกรมจะรับข้อความทีละประโยค แล้วสามารถนำเสนอให้เป็นภาพต่อเนื่อง หรือในลักษณะการเล่าเรื่อง หรือจะทำเป็นนิทานได้

โปรแกรมสร้างภาพเคลื่อนไหว จากข้อความภาษาไทยนี้สามารถอำนวยความสะดวกให้กับผู้ใช้ที่ต้องการสร้างสื่อการสอนที่ใช้รูปภาพในการบรรยาย หรือนิทานสอนใจ เพราะนิทานส่วนใหญ่เป็นเรื่องราวที่ถูกแต่งไว้แล้วจึงจะนำมาพิมพ์เป็นหนังสือออกขายได้ ซึ่งทำให้ไม่สามารถที่จะแต่งเพิ่มเติม หรือเปลี่ยนแปลงใดๆ ลงไปในนิทานเหล่านั้นได้ หรือถ้าหากจะแต่งเรื่องราวขึ้นมาเอง ก็จะต้องมีการสร้างขึ้นมาใหม่เองก็จะเสียค่าใช้จ่ายและเวลาเป็นจำนวนมาก ดังนั้น โปรแกรมสร้างภาพเคลื่อนไหว จากข้อความภาษาไทยนี้จะช่วยให้ผู้ใช้สามารถสร้างเรื่องราวขึ้นมาใหม่ได้อย่างง่าย เพียงแค่ป้อนข้อมูลลงไปไม่ต้องเสียเวลาและช่วยลดค่าใช้จ่ายได้

### 1.2 วัตถุประสงค์

- 1) เพื่อแสดงภาพเคลื่อนไหวจากข้อความภาษาไทยได้ และแปลงประโยคหลายๆประโยครวมเป็นเรื่องราวต่อเนื่อง แล้วแสดงออกมาในรูปแบบนิทาน
- 2) เพื่อแนวคิดในการสร้างความสัมพันธ์ของคำกับรูปภาพ แล้วนำไปประยุกต์ใช้พัฒนาเป็นโปรแกรมอื่นต่อไป

### 1.3 ขอบเขต

- 1) ข้อความที่ป้อนลงโปรแกรมต้องมีความถูกต้องด้านหลักภาษาไทยทั้งไวยากรณ์และรูปประโยค หรือก็คือข้อความสามารถตีความหมายได้ เช่น
  - กระต่ายกินผัก เป็นประโยคที่ถูกต้อง
  - กระต่ายกินผก เป็นประโยคที่ผิด คำที่เป็นกรรมไม่สามารถหาความหมายได้
  - กระต่ายกินขาว เป็นประโยคที่ผิด ทุกคำมีความหมายแต่เมื่อรวมประโยคแล้วหาความหมายไม่ได้ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) ต้องเป็นประโยคอย่างง่าย ประโยคไม่ซับซ้อน คำในประโยคต้องไม่กำกวม
- 3) รับผิดชอบได้แต่ประโยคภาษาไทย ไม่สามารถรับภาษาอังกฤษ
- 4) แสดงภาพเป็นภาพ 2 มิติ
- 5) ไม่สามารถรับชื่อเฉพาะ จำพวกเช่น เคโระ กาฬิลาวัณย์ คิตตี้ จำพวกชื่อตัวละคร

#### 1.4 กรอบแนวคิด

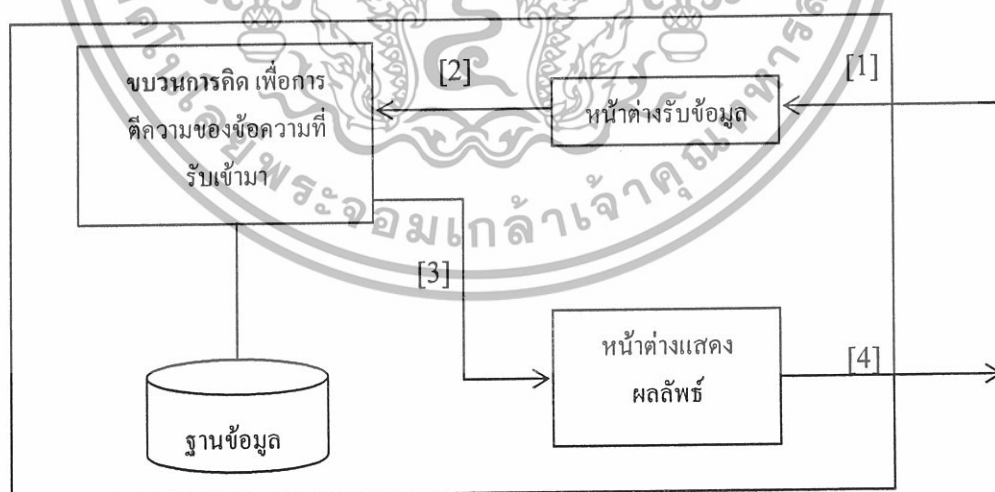
จากการศึกษาเอกสารและงานวิจัยที่เกี่ยวข้องกับการออกแบบและพัฒนาโปรแกรม แนวคิดที่เกี่ยวข้อง มีตัวแปรที่ใช้ศึกษาได้แก่

1) แนวทางการแปลงรูปประโยคภาษาไทยให้เป็นภาพเคลื่อนไหว และแสดงออกมาเป็นเรื่องราวได้

2) หลักภาษาไทย ไวยากรณ์ต่างๆ เพื่อใช้ในการแยกประเภทของคำต่างๆ ในประโยค ให้เป็นหมวดหมู่ จะได้ทราบถึงหน้าที่ของแต่ละคำว่าทำหน้าที่เป็นส่วนไหนของประโยค อย่างเช่น ประธาน กริยา หรือกรรม

3) แนวทางการแบ่งคำในประโยค การนำประโยคที่ได้มาแยกออกเป็นคำ เพื่อที่จะได้สามารถนำไปพิจารณาว่าแต่ละคำนั้นจัดอยู่ในกลุ่มไหนต่อไปได้

ต้องอาศัยความเข้าใจด้านหลักภาษาไทย เพื่อให้ได้แผนภาพเคลื่อนไหวที่สื่อความหมายอย่างถูกต้อง รวมทั้งแนวทางที่ใช้ในการจำแนกคำจากประโยคเพื่อให้ตรงกับภาพและการกระทำต่างๆ แสดงได้ดังแผนภาพด้านล่าง



รูปที่ 1.1 แผนภาพแสดงแนวคิดในการสร้างโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### อธิบายขั้นตอนตามรูปที่ 1.1

- ขั้นตอนที่ 1 โปรแกรมรับข้อมูลที่ผู้ใช้ป้อนเข้ามาผ่านหน้าจอ(Interface)
- ขั้นตอนที่ 2 เมื่อได้ข้อมูลก็ทำการส่งต่อไปที่ขบวนการคิด ซึ่งจะมีขั้นตอนในการตีความรูปประโยค โดยจะต้องมีการติดต่อกับฐานข้อมูลเพื่อแบ่งกลุ่มแต่ละคำในประโยค
- ขั้นตอนที่ 3 จากนั้นเมื่อตีความเสร็จก็ทำการแปลงให้ออกมาเป็นภาพเคลื่อนไหว
- ขั้นตอนที่ 4 แสดงภาพเคลื่อนไหวสู่หน้าจอแสดงผล

### อธิบายขั้นตอนภายในขบวนการคิด

1. ตัดคำ (Token) หาแนวความคิดหรือโปรแกรมที่จะใช้ในการแบ่งประโยคให้เป็นคำๆ เพื่อนำมาพิจารณาต่อไป
2. ตรวจสอบความถูกต้อง (Check Syntax) เพื่อให้คำแต่ละคำนั้นสามารถตีความหมายเพื่อที่จะนำไปแปลงเป็นภาพเคลื่อนไหวต่อไปได้จริง
3. ตีความหมาย หรือการแปลความหมาย แต่ละคำสามารถตีความหมายได้จริง และเมื่อนำมารวมประโยคแล้วหาความหมายได้จริง
4. แสดงภาพเคลื่อนไหว ต้องมีโครงสร้างเก็บความสัมพันธ์ของวัตถุต่างๆ ที่นำมาแสดงเพื่อที่จะนำมาแปลงข้อความที่ผู้ใช้ป้อนเข้ามา

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) เป็นสื่อในการสร้างความบันเทิง เนื่องจากภาพเคลื่อนไหวที่ได้นั้นสามารถนำมาประยุกต์เพื่อสร้างความสนุกสนานเพลิดเพลินให้แก่ผู้ใช้งานหรือผู้ที่รับชม
- 2) เป็นสื่อในการเล่านิทาน เมื่อใช้งานโปรแกรมแล้วผลลัพธ์ที่ได้ออกมานั้นสามารถแสดงเป็นภาพต่อเนื่องจึงสามารถนำมาประยุกต์เป็นการนำเสนอประกอบการเล่านิทานได้
- 3) เรียนรู้หลักการในการแปลภาษา ทำให้สามารถนำไปประยุกต์ใช้กับการประโยคที่ซับซ้อนต่อไปภายหลังได้
- 4) นำไปประยุกต์เป็นสื่อการสอนให้กับเด็กเล็ก และผู้ที่มีปัญหาในการเรียนรู้ซ้ำ เนื่องจากมีภาพแสดงให้เห็นชัดเจน
- 5) ทำให้สามารถนำไปพัฒนาแนวคิดในการสร้างภาพเคลื่อนไหวต่อไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# เอกสารและงานวิจัยที่เกี่ยวข้อง

โปรแกรมสร้างภาพเคลื่อนไหวจากข้อความภาษาไทยเป็น โปรแกรมที่สามารถนำมาประยุกต์ เข้ากับการเขียนนิทานตามจินตนาการ โดยผู้เขียนสามารถกรอกข้อความที่แต่งขึ้น โปรแกรมจะทำการตัดคำจากข้อความนั้นๆ แล้วนำข้อความเหล่านั้นมาใช้ในการตีความหมาย และแสดงผลลัพธ์ที่ ลักษณะภาพเคลื่อนไหวให้สอดคล้องกัน การพัฒนาโปรแกรมห่วงต้องมีความรู้พื้นฐานเรื่องใน หลายส่วน

ทฤษฎีพื้นฐานในการทำโครงการและงานวิจัยที่เกี่ยวข้องที่ใช้ในการศึกษา เพื่อนำมาประยุกต์ ใช้ในการพัฒนาโปรแกรมสร้างภาพเคลื่อนไหวจากข้อความภาษาไทย การศึกษาข้อมูลดังนี้

- 2.1 การใช้ภาษาจาวา (JAVA Language)
- 2.2 การใช้งานระบบฐานข้อมูล (Database)
- 2.3 การนำภาษาจาวามาใช้กับฐานข้อมูล (ODBC)
- 2.4 โครงสร้างของคอมไพเลอร์ (Compiler)
- 2.5 ส่วนการวิเคราะห์ของคอมไพเลอร์ (Lexical Analysis)
- 2.6 ส่วนการวิเคราะห์รูปแบบประโยค (Syntax Analysis)
- 2.7 การใช้งานไลบรารีเจฟเล็กซ์ และ คัพ (Library JFlex และ CUP)
- 2.8 การประมวลภาษาธรรมชาติ (Natural Language)

### 2.1 การใช้ภาษาจาวา

ภาษาสำหรับเขียนโปรแกรม (Application Programming) ซึ่งเป็นลักษณะของโปรแกรมเชิง วัตถุ (Object Oriented Programming) ซึ่งสามารถใช้งานบนเว็บได้ด้วย ผู้พัฒนาต้องมีวิสัยทัศน์เชิง วัตถุ เพื่อหาความต่างของอ็อบเจกต์ (Object) แต่ละอย่างหรือคุณลักษณะประจำ (Attribute) ออกมา ขณะเดียวกันต้องพิจารณาพฤติกรรมของอ็อบเจกต์ เพื่อนำไปสร้างเมธอด (Method) และนำมาเขียน เป็นคลาส (Class) คลาสจะเป็นแม่แบบหรือพิมพ์เขียวใน การสร้างอ็อบเจกต์ ให้เกิดขึ้นและมีชีวิต ในระบบ โดยอ็อบเจกต์ที่สร้างจะมีรายละเอียดที่ต่างกันไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

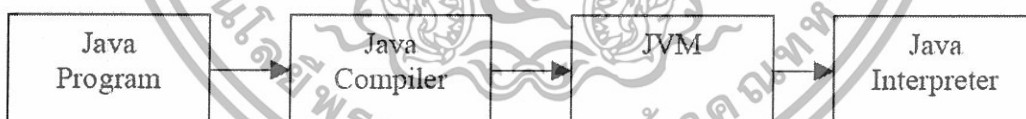
ตัวอย่าง กลาสของรถ ตัวแปรหรือคุณลักษณะที่ใช้อธิบาย ความต่างของรถแต่ละคัน คือ ยี่ห้อ เช่น โตโยต้า ฮอนด้า หรือนิสสัน เป็นต้น สีของรถ หรือจำนวนคนที่สามารถโดยสารได้ ก็เป็นอีกตัวแปรหนึ่งที่ใช้อธิบายความต่างของรถ เป็นต้น ส่วนพฤติกรรมของรถ เช่น การขับเคลื่อนไปข้างหน้า ถอยหลัง หรือหยุด เป็นต้น จะถูกรวบรวมเป็นเสมือนพิมพ์เขียวในการผลิตรถ โดยรถแต่ละคันที่ผลิตออกมาเรียกว่าเป็น อินสแตนซ์ (Instance) ซึ่งจะมีลักษณะประจำและพฤติกรรม

การสร้างโปรแกรมสร้างภาพเคลื่อนไหวจากข้อความภาษาไทยนี้ ได้เลือกใช้ภาษาจาวาในการพัฒนาโปรแกรม เนื่องจากภาษาจาวานั้นเป็นภาษาที่เป็นที่นิยม และมีโปรแกรมที่ใช้ในการเขียนอยู่หลายโปรแกรม ซึ่งได้เลือกใช้โปรแกรมเนตบีน (NetBeans) มาใช้ในการพัฒนาโปรแกรมสร้างภาพเคลื่อนไหวจากข้อความภาษาไทย การเขียนภาษาจาวาจำเป็นที่จะต้องศึกษาพื้นฐาน คุณสมบัติ หรือข้อดี และคำสั่งต่างๆ ดังนี้

ภาษาจาวาถูกพัฒนาขึ้น โดยบริษัท ซันไมโครซิสเต็มส์ (Sun Microsystems) โดยมีจุดเด่นที่เมื่อสร้างขึ้นบนเครื่องคอมพิวเตอร์เครื่องหนึ่งแล้ว ทำให้สามารถนำไปทำงานบนเครื่องคอมพิวเตอร์ที่ระบบแตกต่างกันได้ โดยไม่ต้องคอมไพล์โปรแกรมใหม่

### 2.1.1 หลักการทำงานของภาษาจาวา

ภาษาจาวานั้นใช้วิธีการแปลภาษาทั้งแบบคอมไพล์และแบบอินเทอร์พรีท (Interpret) ภาษาจาวา โดยจะนำแนวคิดของการสร้างเครื่องจักรสมมติ (Java Virtual Machine) มาใช้ เพื่อให้โปรแกรมไม่ขึ้นกับระบบ โดยคอมไพเลอร์จะทำการแปลภาษาให้เป็นโปรแกรมของเครื่องจักรสมมติ แล้วนำโปรแกรมนั้นมาทำงานด้วยเครื่องจักรสมมติที่จำลองขึ้นด้วยจาวาอินเทอร์พรีทเตอร์ (Java Interpreter)



รูปที่ 2.1 แสดงการทำงานของเครื่องจักรสมมติ

ซึ่งวิธีนี้โปรแกรมภาษาจาวาจะถูกคอมไพล์ โดยจาวาคอมไพล์เลอร์ได้เป็นโปรแกรมของเครื่องจักรสมมติ สามารถนำไปทำงานบนเครื่องใดๆ ที่มีจาวาอินเทอร์พรีทเตอร์ได้ จึงมีคุณสมบัติไม่ขึ้นกับระบบ โปรแกรมของเครื่องจักรสมมติจะทำงานได้เร็วกว่าการใช้จาวาอินเทอร์พรีทเตอร์เพียงอย่างเดียว โดยการคอมไพล์ถูกแยกออกไปจากการประมวลผล และด้วยการออกแบบคำสั่งของเครื่องจักรสมมติ ให้ใกล้เคียงกับคำสั่งของหน่วยประมวลผลทั่วไป จาวาอินเทอร์พรีทเตอร์จึงเปลี่ยนคำสั่งของเครื่องจักรสมมติไปสู่คำสั่งของหน่วยประมวลผลที่ใช้งานได้ง่าย การทำการแปลภาษาด้วยอินเทอร์พรีทเตอร์ของภาษาจาวาจึงทำงานได้เร็วกว่าอินเทอร์พรีทเตอร์ของภาษาอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดคำสั่งของเครื่องจักรสมมติถูกออกแบบมาเพื่อสนับสนุนการทำงานของเครื่องเขียนโปรแกรมเชิงวัตถุซึ่งมีคำสั่งเกี่ยวกับอ็อบเจกต์และการอ้างถึงสมาชิกในอ็อบเจกต์ซึ่งไม่มีในหน่วยประมวลผลทั่วไปภาษาจาวาเป็นภาษาที่เน้นความถูกต้องเกี่ยวกับชนิดข้อมูล จึงมีคำสั่งสำหรับการคำนวณชนิดข้อมูลพื้นฐานแต่ละชนิด

### 2.1.2 ภาษาจาวากับการเขียนโปรแกรมเชิงวัตถุ

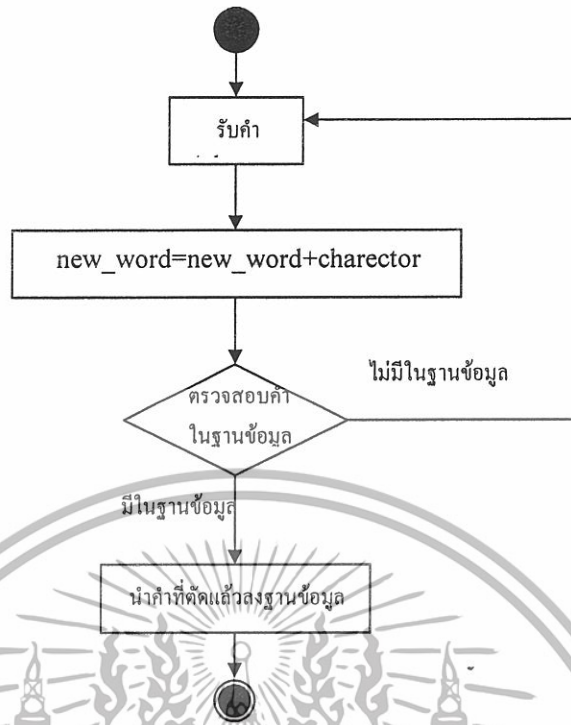
การเขียนโปรแกรมโดยให้มีลักษณะเชิงวัตถุ นั้น ต้องอาศัยภาษาคอมพิวเตอร์ที่ถูกออกแบบมาให้มีลักษณะพิเศษ และเอื้อประโยชน์สำหรับการเขียนโปรแกรมแบบเชิงวัตถุ โดยเฉพาะภาษาเชิงวัตถุมีคลาสเป็นโครงสร้างหลักของโปรแกรม แต่ละคลาสอาจมีสมาชิกเป็นข้อมูลหรือฟังก์ชัน โดยที่ฟังก์ชันในภาษาเชิงวัตถุ จะเรียกว่า เมธอด (Method) หรือ โอเปอเรชัน (Operation)

### 2.1.3 ข้อดีของการใช้ภาษาจาวา

- 1) เป็นภาษาที่ง่ายในการเรียนรู้และใช้งานใน เนื่องจากเลียนแบบภาษา C/C++ ไม่มีคำสั่ง GOTO และ pointer
- 2) มีการป้องกันการผิดพลาด (Robust) การกำจัดสิ่งที่ไม่ใช้ (Garbage Collection) แบบอัตโนมัติและการจัดการความผิดพลาด (Exception Handling) แบบอัตโนมัติ
- 3) มีความปลอดภัยในการที่จะทำงาน โดยไม่ทำลายทรัพยากรของเครื่อง
- 4) ความสามารถในการประมวลผลหลายงานพร้อมกัน
- 5) สามารถเคลื่อนย้ายได้ง่าย รวมทั้งใช้งานในสถานะแวดล้อมที่ต่างกัน และใช้กับคอมพิวเตอร์ได้ทุกระบบ (Multi Platform) โดยไม่ต้องปรับแต่ง
- 6) การเชื่อมโยงแบบพลวัต (Dynamic Link) และการประมวลผลลักษณะการกระจาย (Distributed) ในระบบเครือข่าย

## 2.2 การใช้งานฐานข้อมูล

ในการตัดค่าจะทำได้โดยมีขบวนการไล่ค้นหาค่าในข้อความโดยเริ่มพิจารณาที่อักขระตัวแรกแล้วเปรียบเทียบกับฐานข้อมูลว่ามีค่านี้หรือไม่ ถ้าไม่มีก็เพิ่มอักขระตัวถัดไปเข้ากับค่าเดิม แล้วดูว่าค่าใหม่นั้นมีในฐานข้อมูลหรือไม่ ทำไปเรื่อยๆ จนกว่าจะได้ค่าตรงตามฐานข้อมูลดังรูปที่ 2.2



รูปที่ 2.2 แสดงการทำงานของ การดึงข้อมูลของฐานข้อมูล

การทำงานดังกล่าวจะเห็นว่าการตัดคำต้องเปรียบเทียบข้อมูลจากฐานข้อมูลเสมอ จึงมีความจำเป็นที่ต้องมีความรู้ในเรื่องการติดต่อกับฐานข้อมูลอย่างหลีกเลี่ยงไม่ได้ นอกจากนี้ยังต้องใช้ฐานข้อมูลเก็บคำที่ตัดมา รวมทั้งความถี่ และลักษณะของตัวละครต่างๆ ที่ปรากฏในเรื่องราวที่ผู้ใช้อ่านลงใน โปรแกรมแสดงภาพเคลื่อนไหวนี้

การจัดโครงสร้างข้อมูลให้เป็นแบบฐานข้อมูลกำลังเป็นที่นิยม ซึ่งเกือบทุกหน่วยงานที่มีการใช้ระบบสารสนเทศจะจัดทำข้อมูลให้เป็นแบบฐานข้อมูล เนื่องจากปริมาณข้อมูลมีมากถ้าจัดข้อมูลเป็นแบบเพิ่มข้อมูลจะทำให้มีเพิ่มข้อมูลมีเป็นจำนวนมากจะทำให้เกิดข้อมูลที่ซ้ำซ้อนกันได้ ข้อมูลที่ซ้ำซ้อนนี้จะก่อให้เกิดปัญหาในโลกของการแข่งขันในยุคปัจจุบันนี้ การเข้าถึงข้อมูลอย่างรวดเร็วและมีประสิทธิภาพ ซึ่งจะนำไปสู่การตัดสินใจนั้นถือเป็นสิ่งจำเป็น ดังนั้นซอฟต์แวร์ประยุกต์ที่เหมาะสมกับลักษณะของงานจะทำให้เกิดประโยชน์ได้สูงสุด ส่วนนี้เป็นหน้าที่ของนักออกแบบซอฟต์แวร์ (Software Designer) ที่จะต้องศึกษาและเข้าใจความต้องการที่แท้จริงของกลุ่มผู้ใช้ให้ได้ ในอดีตนั้น การจัดการเพิ่มข้อมูลจะใช้มนุษย์ทำการจัดเรียงข้อมูลและประมวลผล ซึ่งช้าและมักเกิดความผิดพลาดได้ง่าย จึงก่อให้เกิดการนำระบบการประมวลผลเพิ่มข้อมูล (File Processing) ด้วยระบบคอมพิวเตอร์มาใช้แทน ซึ่งมีประสิทธิภาพดีกว่าการทำด้วยมนุษย์ แต่อย่างไรก็ตามความผิดพลาดและความซ้ำซ้อนของข้อมูลก็ยังคงเกิดขึ้น จึงส่งผลให้ระบบฐานข้อมูลถูกพัฒนาขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

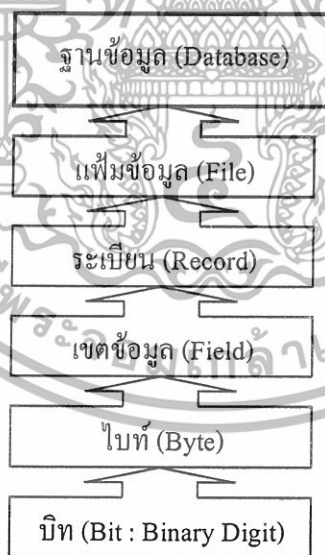
เพื่อแก้ปัญหาเหล่านี้ และเพื่อแยกข้อมูลและโปรแกรมให้เป็นอิสระจากกัน (Data and Program Independence) ให้มากที่สุด

### 2.2.1 ความหมายของระบบฐานข้อมูล

ฐานข้อมูล หมายถึง กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน โดยไม่ได้บังคับว่าข้อมูลทั้งหมดนี้จะต้องเก็บไว้ในแฟ้มข้อมูลเดียวกันหรือแยกเก็บหลายๆ แฟ้มข้อมูล คือการเก็บข้อมูลในฐานข้อมูลนั้นอาจจะเก็บทั้งฐานข้อมูล ซึ่งใช้แฟ้มข้อมูลเพียงแฟ้มข้อมูลเดียวกันได้ หรือจะเก็บไว้ในหลายๆ แฟ้มข้อมูล ที่สำคัญคือจะต้องสร้างความสัมพันธ์ระหว่างระเบียบและเรียกใช้ความสัมพันธ์นั้นได้ มีการกำจัดความซ้ำซ้อนของข้อมูลออกและเก็บแฟ้มข้อมูลเหล่านี้ไว้ที่ศูนย์กลาง เพื่อที่จะนำข้อมูลเหล่านี้มาใช้ร่วมกัน ความคุ้มค่าและรักษาเมื่อผู้ต้องการใช้งานและผู้มีสิทธิ์จะใช้ข้อมูลนั้นสามารถดึงข้อมูลที่ต้องการออกไปใช้ได้ ข้อมูลบางส่วนอาจใช้ร่วมกับผู้อื่นได้ แต่บางส่วนผู้มีสิทธิ์เท่านั้นจึงจะสามารถใช้ได้

### 2.2.2 โครงสร้างของฐานข้อมูล

โครงสร้างของฐานข้อมูลแสดงถึงการรวมกลุ่มของข้อมูลต่างๆ ซึ่งมีลำดับของหน่วยข้อมูลดังรูปที่ 2.3



รูปที่ 2.3 ลำดับชั้นของหน่วยข้อมูลในระบบฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับชั้นของหน่วยข้อมูลประกอบด้วย

- 1) บิต (Bit) หมายถึง หน่วยของข้อมูลที่มีขนาดเล็กที่สุด
- 2) ไบต์ (Byte) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำข้อมูลชนิดบิตมารวมกันจนกลายเป็นตัวอักษร (Character)
- 3) เขตข้อมูล (Field) หมายถึง หน่วยของข้อมูลประกอบขึ้นจากตัวอักขระตั้งแต่หนึ่งตัวขึ้นไป มารวมกันแล้วได้ความหมายของสิ่งใดสิ่งหนึ่ง เช่น ชื่อ ที่อยู่ เป็นต้น
- 4) การจัดเก็บเพิ่มข้อมูลหลายๆ เพิ่มข้อมูลไว้ภายใต้ระบบเดียวกัน เรียกว่า ฐานข้อมูล เช่น เก็บเพิ่มข้อมูล นักเรียน อาจารย์ และวิชาที่เปิดสอน เป็นต้น
- 5) ระเบียบ (Record) หมายถึง หน่วยของข้อมูลที่เกิดจากการเอาเขตข้อมูลหลายๆ เขตข้อมูลมารวมกัน เพื่อเกิดเป็นข้อมูลเรื่องใดเรื่องหนึ่ง เช่น ข้อมูลของนักศึกษา 1 ระเบียบ (1 คน) ประกอบด้วย

รหัสประจำตัวนักศึกษา 1 เขตข้อมูล

ชื่อนักศึกษา 1 เขตข้อมูล

ที่อยู่ 1 เขตข้อมูล

- 6) เพิ่มข้อมูล (File) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำข้อมูลหลายๆ ระเบียบที่เป็นเรื่องเดียวกันมารวมกัน เช่น เพิ่มข้อมูลนักศึกษา เพิ่มข้อมูลลูกค้า เพิ่มข้อมูลพนักงาน
- 7) เอนทิตี (Entity) หมายถึง ชื่อของสิ่งใดสิ่งหนึ่ง ได้แก่ คน สถานที่ สิ่งของ การกระทำ ซึ่งต้องการจัดเก็บข้อมูลไว้ เช่น เอนทิตีลูกค้า เอนทิตีพนักงาน
- 8) แอททริบิวต์ (Attribute) หมายถึง รายละเอียดข้อมูลที่แสดงลักษณะและคุณสมบัติของเอนทิตีหนึ่งๆ เช่น เอนทิตีนักศึกษา ประกอบด้วย แอททริบิวต์รหัสนักศึกษา แอททริบิวต์ชื่อนักศึกษา และแอททริบิวต์ที่อยู่นักศึกษา

### 2.2.3 รูปแบบของระบบฐานข้อมูล

2.2.3.1 ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เป็นการเก็บข้อมูลในรูปแบบที่เป็นตาราง (Table) หรือเรียกว่าความสัมพันธ์ (Relation) มีลักษณะเป็น 2 มิติ คือแถว (Row) และคอลัมน์หรือหลัก (Column) การเชื่อมโยงข้อมูลระหว่างตาราง จะเชื่อมโยงโดยใช้แอททริบิวต์ (Attribute) หรือคอลัมน์ที่เหมือนกันทั้งสองตารางเป็นตัวเชื่อมโยงข้อมูล ฐานข้อมูลเชิงสัมพันธ์นี้จะเป็นรูปแบบของฐานข้อมูลที่นิยมใช้ในปัจจุบัน ดังตัวอย่างตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตารางที่ 2.1 ตารางพนักงาน

รหัสพนักงาน	ชื่อพนักงาน	ที่อยู่	เงินเดือน	รหัสแผนก
12501535	นายสมพงศ์	กรุงเทพ	12000	VO
12534568	นายมนตรี	นครปฐม	12500	VN
12503452	นายเอก	กรุงเทพ	13500	VO
12356892	นายบรรทัด	นนทบุรี	11500	VD
15689730	นายราชนัน	สมุทรปราการ	12000	VA

2.2.3.2 **ฐานข้อมูลแบบเครือข่าย (Network Database)** ฐานข้อมูลแบบเครือข่ายจะเป็นการรวมระเบียบต่างๆและความสัมพันธ์ระหว่างระเบียบ แต่ลักษณะข้อมูลแบบนี้จะต่างกับฐานข้อมูลเชิงสัมพันธ์ คือ ในฐานข้อมูลเชิงสัมพันธ์จะแฝงความสัมพันธ์เอาไว้ โดยระเบียบที่มีความสัมพันธ์กันจะต้องมีค่าของข้อมูลในแอททริบิวต์ใดแอททริบิวต์หนึ่งเหมือนกัน แต่ฐานข้อมูลแบบเครือข่าย จะแสดงความสัมพันธ์อย่างชัดเจน

2.2.3.3 **ฐานข้อมูลแบบลำดับชั้น (Hierarchical Database)** ฐานข้อมูลแบบลำดับชั้นเป็นโครงสร้างที่จัดเก็บข้อมูลในลักษณะของความสัมพันธ์แบบพ่อ-ลูก (Parent-Child Relationship Type : PCR Type) หรือเป็นโครงสร้างรูปแบบต้นไม้ (Tree) ข้อมูลที่จัดเก็บในที่นี้ คือ ระเบียบ (Record) ซึ่งประกอบด้วยค่าของเขตข้อมูล (Field) ของเอนทิตีหนึ่งๆ ฐานข้อมูลแบบลำดับชั้นนี้คล้ายคลึงกับฐานข้อมูลแบบเครือข่าย แต่ต่างกันที่ฐานข้อมูลแบบลำดับชั้น มีกฎเพิ่มขึ้นมาอีกหนึ่งประการ คือ ในแต่ละกรอบจะมีลูกสร้งเข้าหาได้ไม่เกิน 1 หัวลูกศร

## 2.2.4 ระบบการจัดการฐานข้อมูล (Database Management System, DBMS)

ระบบจัดการฐานข้อมูลเป็นซอฟต์แวร์ที่ทำหน้าที่ดังต่อไปนี้

2.2.4.1 **ดูแลการใช้งานให้กับผู้ใช้** ในการติดต่อกับตัวจัดการระบบแฟ้มข้อมูลได้ ในระบบฐานข้อมูลนี้ข้อมูลจะมีขนาดใหญ่ ซึ่งจะถูกจัดเก็บไว้ในหน่วยความจำสำรองเมื่อผู้ใช้ต้องการจะใช้ฐานข้อมูล ระบบการจัดการฐานข้อมูลจะทำหน้าที่ติดต่อกับระบบแฟ้มข้อมูลซึ่งเสมือนเป็นผู้จัดการแฟ้มข้อมูล (File Manager) นำข้อมูลจากหน่วยความจำสำรองเข้าสู่หน่วยความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำหลักเฉพาะส่วนที่ต้องการใช้งาน และทำหน้าที่ประสานกับตัวจัดการระบบเพิ่มข้อมูลในการจัดเก็บ เรียกใช้ และแก้ไขข้อมูล

2.2.4.2 ควบคุมระบบความปลอดภัยของข้อมูลโดยป้องกันไม่ให้ผู้ที่ไม่ได้รับอนุญาตเข้ามาเรียกใช้หรือแก้ไขข้อมูลในส่วนป้องกันเอาไว้ พร้อมทั้งสร้างฟังก์ชันในการจัดทำข้อมูลสำรอง โดยเมื่อเกิดความขัดข้องของระบบเพิ่มข้อมูลหรือของมูลกลับเข้าสู่สภาพที่ถูกต้องสมบูรณ์ได้ เครื่องคอมพิวเตอร์เกิดการเสียหายนั้น ฟังก์ชันนี้จะสามารถทำการฟื้นฟูสภาพของระบบข้อ

2.2.4.3 ควบคุมการใช้ข้อมูลในสภาพที่มีผู้ใช้พร้อมๆ กันหลายคน โดยจัดการเมื่อมีข้อผิดพลาดของข้อมูลเกิดขึ้น

## 2.2.5 ข้อดีข้อเสียของการใช้ฐานข้อมูล

### 2.2.5.1 ข้อดีของการประมวลผลแบบฐานข้อมูล

- 1) หลีกเลี่ยงความขัดแย้งของข้อมูลได้ การจัดการฐานข้อมูลช่วยลดขั้นตอน และความซับซ้อนของข้อมูลได้ สามารถใช้ข้อมูลร่วมกันได้ การจัดการฐานข้อมูลเป็นการเก็บข้อมูลรวมไว้ด้วยกัน เพื่อให้ผู้ใช้จะสามารถใช้ข้อมูลที่ต้องการได้
- 2) สามารถลดความซ้ำซ้อนของข้อมูล ข้อมูลประเภทเดียวกันจะถูกเก็บไว้ในแฟ้มเดียวกัน ลดปัญหาความซ้ำซ้อนของข้อมูล อีกทั้งลดปัญหาการประมวลผลที่ซ้ำได้
- 3) การรักษาความถูกต้องเชื่อถือได้ของข้อมูล ระบบฐานข้อมูลสามารถระบุเกณฑ์เพื่อควบคุมความผิดพลาดได้
- 4) สามารถกำหนดความเป็นมาตรฐานเดียวกันได้ สามารถกำหนดรูปแบบและควบคุมความมีมาตรฐานของข้อมูลได้
- 5) สามารถกำหนดระบบความปลอดภัยของข้อมูลได้ สามารถกำหนดระดับการเข้าใช้ข้อมูลของบุคคลแตกต่างกันได้
- 6) ความเป็นอิสระของข้อมูลและโปรแกรม ซึ่งโปรแกรมเป็นตัวใช้ข้อมูลในฐานข้อมูล โดยความต้องการของผู้ใช้

### 2.2.5.2 ข้อเสียของการประมวลผลแบบฐานข้อมูล

- 1) มีต้นทุนสูง ต้องใช้ทุนด้านต่างๆ เช่น ซอฟต์แวร์ บุคลากร
- 2) ความซับซ้อน การเริ่มใช้ระบบฐานข้อมูลอาจทำให้เกิดความซับซ้อนได้ เช่น การจัดเก็บ การออกแบบ
- 3) มีการเสี่ยงต่อการหยุดชะงักของระบบ เนื่องจากการเก็บข้อมูลเป็นศูนย์กลาง เมื่อเกิดปัญหาขึ้น ทำให้ส่วนอื่นกระทบไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวมาทั้งหมดนั้น มีโปรแกรมฐานข้อมูลที่นิยมใช้คู่ด้วยกันหลายตัว เช่น แอซเซส (Access), ฟ็อกโปร (FoxPro), คลิปปเปอร์ (Clipper), ดีเบส (dBase), ฟ็อกเบส (FoxBase), ออราเคิล (Oracle), เอสคิวเอล (SQL) เป็นต้น โดยแต่ละโปรแกรมจะมีความสามารถต่างกัน บางโปรแกรมใช้งานง่ายแต่จะจำกัดขอบเขตการใช้งาน แต่บางโปรแกรมใช้งานยากกว่ามีความสามารถในการทำงานมากกว่า

โปรแกรมเอสคิวเอลที่เลือกในการพัฒนาครั้งนี้เป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพในการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง โปรแกรมเอสคิวเอลจึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์ และเป็นภาษาหนึ่งที่มีผู้นิยมใช้กันมาก โดยทั่วไปโปรแกรมฐานข้อมูลของบริษัทต่างๆ ที่มีใช้อยู่ในปัจจุบัน เช่น ออราเคิล (Oracle), ดีบีทู (DB2), มาคัยเอสคิวเอลบราวเซอร์ (mySQLBrowser) ในกรณีนี้ได้เลือกการทำงานของมายเอสคิวเอลบราวเซอร์มาใช้ในร่วมกับโปรแกรมสร้างภาพเคลื่อนไหวจากข้อความภาษาไทย

### 2.3 การนำภาษาจาวามาใช้กับฐานข้อมูล

เจดีบีซี (Java Database Connectivity : JDBC) ใช้สำหรับการประมวลผลคำสั่ง เอสคิวเอลซึ่งเป็นคลาสและส่วนติดต่อผู้ใช้ที่เป็นภาษาจาวา เจดีบีซีได้จัดเตรียมเอพีไอ (API) มาตรฐานสำหรับเป็นเครื่องมือในการพัฒนาระบบฐานข้อมูล ในการเขียนโค้ด โปรแกรมภาษาจาวา ซึ่งเจดีบีซีสามารถติดต่อกับฐานข้อมูล ได้เกือบทุกประเภท เช่น ฐานข้อมูลของออราเคิล ฐานข้อมูลของมาคัยเอสคิวเอล เป็นต้น ที่สำคัญการพัฒนาโปรแกรมด้านระบบฐานข้อมูล โดยใช้จาวา สามารถนำไปประมวลผลได้หลายรูปแบบ ซึ่งเจดีบีซีจะเข้ามาช่วยจัดการในขั้นตอนนี้ จะทำให้ทำการพัฒนาโปรแกรมประยุกต์ได้ใช้คุณสมบัติของภาษาจาวาได้อย่างเต็มที่

#### 2.3.1 การทำงานที่สำคัญของเจดีบีซี

- ทำการติดต่อกับฐานข้อมูล
- สั่งคำสั่งเอสคิวเอลไปยังฐานข้อมูล
- การรับผลลัพธ์จากการใช้คำสั่งเอสคิวเอล แล้วทำการประมวลผล

#### 2.3.2 ลักษณะที่สำคัญของเจดีบีซี

เจดีบีซีเป็นส่วนติดต่อกับผู้ใช้ระดับต่ำ (Low-level Interface) ซึ่งสามารถเรียกใช้คำสั่งเอสคิวเอลได้โดยตรง สามารถใช้งานได้อย่างมีประสิทธิภาพ มีขนาดเล็ก และง่ายต่อการใช้กว่าตัวติดต่อฐานข้อมูลชนิดอื่นๆ

### 2.3.3 ประเภทตัวติดตั้งของเจดีบีซี

2.3.3.1 เจดีบีซีโอดีบีซีบิซ (JDBC - ODBC Bridge) บวกกับ ตัวติดตั้งโอดีบีซี เป็นตัวติดตั้งของเจดีบีซี ที่ใช้ในการติดต่อการทำงานผ่านโอดีบีซี ซึ่งจะมีการใช้โอดีบีซีในการติดต่อกับฐานข้อมูลก่อนไม่ว่าจะเป็นในเครื่องเดียวกันหรือติดต่อผ่านผู้ให้บริการ จึงใช้เจดีบีซีในการติดต่อกับโอดีบีซีอีกครั้งหนึ่ง

2.3.3.2 ตัวติดตั้ง หรือเนทีฟเอพีไอพาร์ทลิจาวา (Native API partly-Java Driver) สำหรับตัวติดตั้งประเภทนี้จะเปลี่ยนการเรียกโอดีบีซีไปเรียกเอพีไอของผู้ให้บริการ ของออราเคิลซิสเทม หรือดีบีเอ็มเอสตัวอื่นๆ ซึ่งจะคล้ายสะพานเชื่อมต่อไปยังฐานข้อมูล

2.3.3.3 ตัวติดตั้ง เจดีบีซีเน็ตจาวา (JDBC-Net Java Driver) ตัวติดตั้งประเภทนี้จะเปลี่ยนการเรียกเจดีบีซี ไปยัง ดีบีเอ็มเอส อินดิเพนเดน เน็ต โปรโตคอล ซึ่งหลังจากการเปลี่ยนไปเป็น ดีบีเอ็มเอส โปรโตคอลโดยผู้ให้บริการ ซึ่งผู้บริการตัวนี้สามารถติดต่อเข้ากับผู้ให้บริการไปยังฐานข้อมูลหลายประเภทได้ ซึ่งจะสามารถใช้งานผ่านทางอินเทอร์เน็ต และอินทราเน็ต

2.3.3.4 ตัวติดตั้ง เนทีฟโปรโตคอลเพียวจาวา (Native - Protocol Pure Java Driver) ตัวติดตั้งประเภทนี้จะเปลี่ยนการเรียกเจดีบีซีไปเป็นเนทีฟโปรโตคอล จะถูกใช้โดยดีบีเอ็มเอส ซึ่งจะมีการใช้งานโดยมีการเรียกใช้งานโดยตรงจากผู้ให้บริการ ไปยังดีบีเอ็มเอสของผู้ให้บริการ

### 2.3.4 ข้อดีของเจดีบีซี

- 1) ใช้งานง่าย การตั้งค่าไม่ยุ่งยาก
- 2) สามารถทำงานร่วมกับภาษาจาวา ได้เป็นอย่างดี เพราะตัวติดตั้งที่เขียนขึ้นนั้นส่วนมากจะพัฒนาจากภาษาจาวาและมีการประมวลผลเป็นคลาสไฟล์ทำให้ใช้งานง่าย
- 3) สามารถติดต่อฐานข้อมูลได้เกือบทุกประเภท ซึ่งส่วนใหญ่จะเป็นฐานข้อมูลแบบเชิงสัมพันธ์
- 4) ไฟล์ติดตั้งมีขนาดเล็กกะทัดรัด
- 5) สามารถใช้งานได้ทั้งสถาปัตยกรรมแบบจุดต่อจุด (Two-Tier) และมากกว่า 2 จุด (Three-Tier)
- 6) สามารถใช้งานได้หลายระบบปฏิบัติการ
- 7) ส่วนติดต่อกับผู้ใช้ระดับต่ำและสามารถใช้ส่วนติดต่อกับผู้ใช้ระดับสูงด้วยถ้าต้องการ
- 8) การใช้งานจะเป็นเชิงวัตถุ ทำให้ไม่มีผลกระทบต่อกันถ้ามีการเปลี่ยนแปลงหรือพัฒนาระบบ
- 9) ตัวติดตั้งแต่ละตัวมีมาตรฐานใกล้เคียงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 โครงสร้างของคอมไพเลอร์

คอมไพเลอร์ (Compiler) คือ คอมไพเลอร์เป็นโปรแกรมชนิดหนึ่งที่ได้รับเอาโปรแกรมที่เขียนขึ้นด้วยภาษาหนึ่งเป็นโปรแกรมต้นทาง (Input หรือ Source program) เข้ามาทำการประมวลผล เรียกว่าการคอมไพล์ (Compile) แล้วจะสร้างโปรแกรมที่เขียนด้วยภาษาอีกภาษาหนึ่งเรียกว่า ภาษาปลายทาง (Output หรือ Target program) ซึ่งหากการแปลงภาษานั้นไม่ประสบความสำเร็จเกิดข้อผิดพลาดนั้นจะมีการแจ้งข้อผิดพลาด (Error Message)



รูปที่ 2.4 หลักการทำงานของคอมไพเลอร์

### 2.4.1 ขั้นตอนการทำงานของคอมไพเลอร์ มีดังนี้

2.4.1.1 ใส่โปรแกรมต้นฉบับ (Source Program) เข้าไปในพรีโปรเซสเซอร์ (Preprocessor) เพื่อแปลภาษาในส่วนขยาย (Extension) ให้เป็นโปรแกรมย่อยๆ (Subprogram) ของภาษาปลายทางที่ต้องการ หรือเป็นส่วนในการเตรียมโปรแกรมต้นทางให้อยู่ในรูปแบบที่สามารถแปลงได้ง่าย

2.4.1.2 นำโปรแกรมที่จัดเตรียมไว้นั้นมาผ่านกระบวนการคอมไพล์ ซึ่งเป็นการแปลงภาษาต้นทางให้เป็นภาษาปลายทางทั้งหมด ทำให้ได้โปรแกรมปลายทางออกมา โดยผลลัพธ์ที่ได้ส่วนใหญ่จะเป็นภาษาที่เป็นกลางเช่น ภาษาแอสเซมบลี (Assembly) เรียกว่าโปรแกรมแอสเซมบลี (Assembly program)

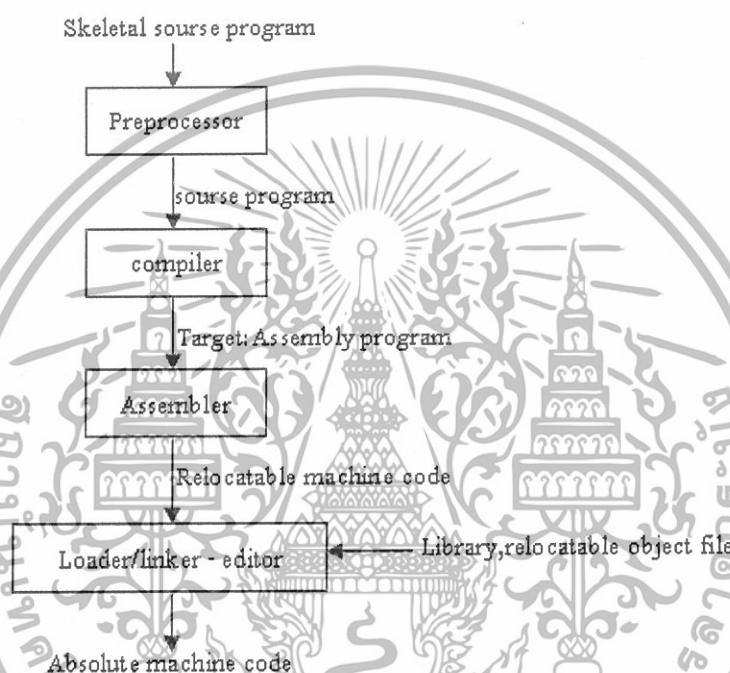
2.4.1.3 เมื่อได้โปรแกรมปลายทางที่ต้องการแล้ว อย่างการอธิบายนี้คือ โปรแกรมแอสเซมบลี แล้วนำมาคอมไพล์อีกครั้ง โดยใช้โปรแกรมตัวแปลภาษาแอสเซมบลี (Assembler) แล้วทำให้ผลลัพธ์นั้นกลายเป็นออบเจ็คโค้ด (Object code) ซึ่งยังไม่สามารถนำมาประมวลผลได้ทันทีเพราะว่าโปรแกรมอ้างอิงถึงตำแหน่งของที่อยู่ (Location address) ซึ่งเป็นตำแหน่งแบบความสัมพันธ์ (Relative address) ข้อดีคือสามารถนำโปรแกรมนี้ไปใส่ไว้ที่ไหนก็ได้ ไม่มีผลต่อตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.4 เป็นการนำโค้ดที่ได้ขึ้นมาแบ่งการทำงานออกเป็น 2 ส่วนได้แก่

1) นำโปรแกรมเข้าสู่หน่วยความจำหลักโดยใช้ตัวแก้ไขการเชื่อมโยง (Linker editor) ซึ่งจะทำหน้าที่เชื่อมโยง (Link) ส่วนต่างเข้าไว้ด้วยกัน

2) อ่านโปรแกรมเข้าไปไว้ในหน่วยความจำ ซึ่งจำทำการแปลแอดเดรสจากความสัมพันธ์ (Relation address) ให้กลายเป็นแอดเดรสที่แท้จริง (Absolute address) และทำให้สามารถประมวลผลโปรแกรมนั้นได้ ทำให้ได้ผลลัพธ์ออกมาเป็นโค้ดที่แท้จริง (Absolute code)



รูปที่ 2.5 ขั้นตอนการทำงานของคอมไพเลอร์

#### 2.4.2 โครงสร้างของคอมไพเลอร์

โครงสร้างของคอมไพเลอร์ คือ โครงสร้างของหลักการทำงานของคอมไพเลอร์ ซึ่งมีการแบ่งโครงสร้างออกเป็นส่วน โดยแต่ละส่วนนั้นจะมีหน้าที่การทำงานที่แตกต่างกันออกไป ดังนี้

2.4.2.1 ส่วนของการวิเคราะห์ (Analysis phase) หรือเรียกว่าฟรอนท์เอนด์ (Front End) ซึ่งเป็นส่วนที่มีความสัมพันธ์กับรูปแบบภาษาของโปรแกรม แต่ไม่สัมพันธ์กับเครื่องที่จะนำมาประมวลผล ประกอบด้วย

1) การทำการวิเคราะห์คำศัพท์ต่างๆ (Lexical analysis) โดยจะเริ่มการทำงานจากการอ่านโปรแกรมต้นฉบับ ถ้าหากโปรแกรมต้นฉบับมีคำสั่งพิเศษ (Non-Standard Syntax) ประกอบอยู่ก็จะต้องใช้ฟรีโปรเซสเซอร์แปลงก่อน โดยจะให้โปรแกรมต้นฉบับนั้นมีแต่คำสั่งพื้นฐานเท่านั้น ต่อมาก็จะทำการกราดข้อมูล (Scan) ซึ่งเล็กซ์ (Lex) จะมองเห็นคำสั่งทั้งหมดเป็นคำสั่งที่อยู่ในบรรทัดเดียวกัน และในเวลาเดียวกันเล็กซ์ จะสร้างตารางสัญลักษณ์ (Symbol table) ขึ้นมาด้วย แล้วจะนำข้อมูลที่กราดนั้นมาใส่ลงในตารางสัญลักษณ์ ซึ่งจะทำให้การตรวจสอบคำสั่งต่างๆ แล้วยังสามารถรายงานข้อผิดพลาดต่างๆ ออกมาได้ด้วย

2) การวิเคราะห์รูปแบบ (Syntax Analysis) ส่วนที่บอกให้รู้ถึงชื่ออ้างอิง (Identifier) ต่างๆ นั้นถูกนำมาใช้งานได้อย่างถูกต้องหรือไม่ ซึ่งเป็นการตรวจสอบรูปแบบทั่วไปของโปรแกรม

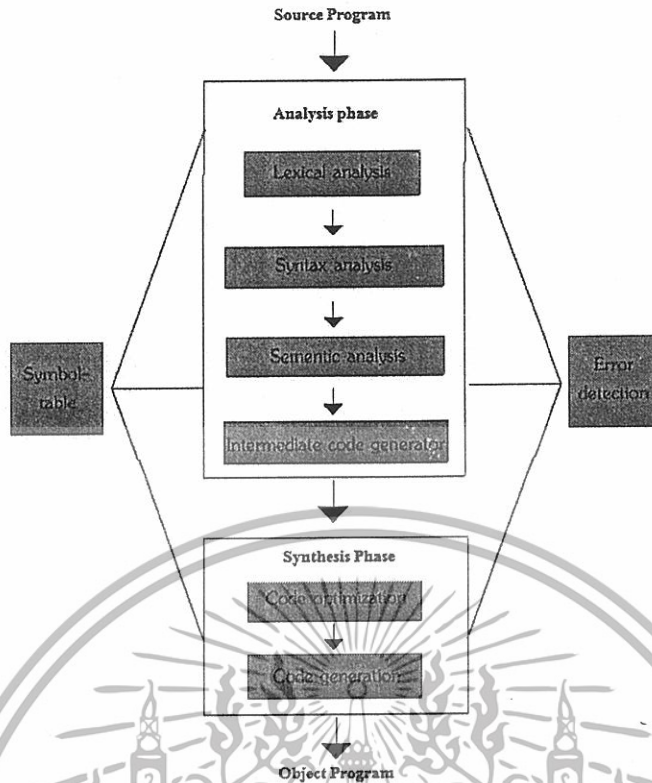
3) การวิเคราะห์ความหมาย (Semantic Analysis) เป็นส่วนที่ใช้ตรวจสอบความหมายของประโยคคำสั่งของโปรแกรม เช่น คำสั่ง  $a = x$  ; ส่วนการจัดการวิเคราะห์คำศัพท์ (lexical analysis) จะทำการกราดข้อมูลได้ "a", "=", "x" และ ";" ต่อมาในส่วนของการวิเคราะห์รูปแบบ (Syntax analysis) จะตรวจสอบความถูกต้องของรูปแบบคำสั่งว่าถูกต้องหรือไม่ ถ้าถูกต้องก็จะส่งค่ามาถึงการวิเคราะห์ความหมาย (Semantic analysis) ซึ่งจะมีการตรวจสอบความหมายว่าชนิดของข้อมูลทั้งสองนั้นต้องเข้ากันได้ (Compatible) เช่นหาก a เป็นข้อมูลประเภทจำนวนเต็ม (Integer) แต่ x เป็นข้อมูลประเภทตัวอักษร (Character) ทำให้เกิดความขัดแย้งกัน

4) การเปลี่ยนแปลง โปรแกรมให้อยู่ในรูปแบบของภาษากลาง เป็นการเปลี่ยนแปลงโปรแกรมต้นฉบับให้อยู่ในอีกรูปแบบหนึ่งเรียกว่าภาษากลาง (Intermediate Code หรือ IR)

2.4.2.2 ส่วนการสังเคราะห์ (Synthesis Phase) หรือเรียกว่าแบล็คเอนด์ (Back End) ซึ่งเป็นส่วนที่เน้นการประมวลผลที่เครื่องที่จะนำมาประมวลผล ประกอบด้วย

1) การรวมโค้ด (Code Optimizer) เป็นการบูรรวมโค้ด โดยพิจารณาคำสั่งที่สามารถบูรรวมกันได้ หรือเทคนิคที่ทำให้โปรแกรมสามารถประมวลผลได้เร็วขึ้น

2) การสร้างออบเจกต์โค้ด (Object Code) เป็นโปรแกรมผลลัพธ์ ซึ่งเป็นโปรแกรมที่อยู่ในลักษณะของความสัมพันธ์ (Relation Code) แต่บางกรณีอาจจะเป็นโค้ดเครื่อง (Absolute Machine Code) ที่สามารถประมวลผลได้ทันที



รูปที่ 2.6 โครงสร้างของคอมไพเลอร์

จากการหลักการ โครงสร้างของคอมไพเลอร์นั้นสามารถนำมาประยุกต์นำมาใช้กับ โครงการงาน ได้ โดยนำหลักการในส่วนของ การวิเคราะห์คำ และการวิเคราะห์รูปแบบประโยค จึงขออธิบายการทำเพียงสองส่วนนี้ต่อไป

## 2.5 ส่วนการวิเคราะห์ของคอมไพเลอร์

วัตถุประสงค์ในการวิเคราะห์ของคอมไพเลอร์ (Lexical analysis) คือการแปลกระแสของข้อมูล (Stream) ไปเป็นกลุ่มของโทเคน (Token) ซึ่งในขั้นสุดท้ายจะนำข้อมูลที่ได้ไปจัดเก็บในตารางสัญลักษณ์ (Symbol table) เพื่อนำไปใช้ในส่วนอื่นของคอมไพเลอร์ต่อไป

### 2.5.1 คำจำกัดความ

- โทเคน (Token) คือการแบ่งกลุ่มของตัวอักษรที่อ่านเข้ามาจากโปรแกรมต้นฉบับ ให้กลายเป็นสัญลักษณ์ที่มีความหมาย และถูกนำไปใช้ในส่วนอื่นของคอมไพเลอร์
- หน่วยคำศัพท์ (Lexeme) คือ กลุ่มของตัวอักษร หรือ กระแสของข้อมูล ที่ตรงกับนิยามของโทเคน
- รูปแบบ (Pattern) คือ กฎที่นำมาใช้จับคู่ หน่วยคำศัพท์ ว่าตรงกับโทเคนตัวใด

## 2.5.2 นิพจน์สามัญ (Regular Expression)

นิพจน์สามัญ เป็นการกำหนดกลุ่มของอักขระและสัญลักษณ์ ที่เรียงกันเป็นรูปแบบเพื่อนำไปใช้เทียบกับข้อมูลที่เข้ามา

2.5.2.1 นิพจน์สามัญ เป็นการกำหนดรูปแบบของคำ (Pattern) ขึ้นมาเพื่อให้ใช้เป็นรูปแบบอ้างอิงในการตรวจสอบ ค้นหาที่อยู่ในกระแสของข้อมูล

ตัวอย่าง การกำหนดนิพจน์สามัญสำหรับชื่ออ้างอิง (Identifier) ตัวหนึ่ง ทำได้ดังนี้

```
id    -->  letter(letter | digits)*
letter -->  A|B|...|Z|a|b|...|z
digits -->  0|1|2|...|9
```

กฎที่ใช้อธิบายนิพจน์สามัญ ที่มีต่อคำสั่งชุดหนึ่งมีดังนี้

1)  $\epsilon$  เป็นนิพจน์สามัญ อย่างหนึ่งที่แสดงถึง  $\{\epsilon\}$  หรือเซตที่ประกอบด้วย เอ็มตี้สตริง (Empty String)

2) ถ้า  $a$  เป็นสัญลักษณ์ ใน  $\Sigma$  ดังนั้น  $a$  ก็จะเป็นนิพจน์สามัญ ที่แสดงถึง  $\{a\}$  หรือเซตที่ประกอบด้วย string  $a$  แม้ว่า  $a$  จะถูกใช้แสดงความหมายเหมือนกันทั้งสามชนิดคือ  $a$  อาจหมายถึงนิพจน์สามัญ ข้อความ หรือสัญลักษณ์ก็ได้ แต่เมื่อนำ  $a$  ไปใช้งานจริงผู้ใช้จะทราบได้เองว่าขณะนั้น  $a$  หมายถึงสิ่งใด

3) สมมติว่า  $r$  และ  $s$  เป็น นิพจน์สามัญ ที่ใช้แทนภาษา  $L(r)$  และ  $L(s)$  แล้ว

3.1)  $(r) | (s)$  เป็นนิพจน์สามัญ ที่แสดงถึง  $L(r) \cup L(s)$

3.2)  $(r)(s)$  เป็นนิพจน์สามัญ ที่แสดงถึง  $L(r)L(s)$

3.3)  $(r)^*$  เป็นนิพจน์สามัญ ที่แสดงถึง  $(L(r))^*$

3.4)  $(r)$  เป็นนิพจน์สามัญ ที่แสดงถึง  $L(r)^2$

2.5.2.2 นิยามของนิพจน์สามัญ (regular definition) หมายถึงลำดับของนิยามที่อยู่ในรูปแบบของ

```
di    -->  ri
d2    -->  r2
...
dn    -->  m
```

โดยที่  $d_i$  นั้นหมายถึงชื่อเฉพาะชื่อหนึ่ง และ  $r_i$  คือนิพจน์สามัญ (Regular Expression) ซึ่งสร้างขึ้นมาจาก  $\Sigma \cup \{d_1, d_2, \dots, d_{i-1}\}$  นั่นคือสัญลักษณ์พื้นฐานต่างๆและชื่อเฉพาะที่ได้สร้างขึ้นก่อนหน้านี้

ตัวอย่าง ตัวแปรที่ประกอบด้วย ตัวอักษรและตัวเลขโดยมีการเริ่มต้นด้วยตัวอักษร

letter --> A | B | ... | Z | a | b | ... | z

digits --> 0 | 1 | 2 | ... | 9

id --> letter(letter|digits)\*

ตัวอย่าง ตัวเลขที่เป็นไม่มีเครื่องหมาย

digit --> 0 | 1 | 2 | ... | 9

digits --> digit digit\*

optional\_fraction --> . digit |  $\epsilon$

optional\_exponent --> (E(+|-|?)digits) |  $\epsilon$

num --> digitsoptional\_fraction

2.5.2.3 การรับรู้โทเคน วัตถุประสงค์ในการรับรู้โทเคนได้ คือ การสร้างส่วนวิเคราะห์คำที่สามารถแยกคำศัพท์ต่างๆสำหรับแต่ละโทเคน และสร้างผลลัพธ์ให้อยู่ในรูปโทเคนที่มีความสัมพันธ์กับชนิดตัวแปร <token, attribute>

## 2.6 การวิเคราะห์รูปแบบประโยค (Syntax analysis)

### 2.6.1 บทบาทของพาร์เซอร์ (Parser)

พาร์เซอร์จะมีหลักการทำงานอยู่ 2 แบบ คือการทำงานแบบบนลงล่าง (Top-Down Parser) และแบบล่างขึ้นบน (Bottom-Up Parser) ในการทำงานแบบบนลงล่างนั้นจะทำการสร้างอัลกอริทึมแบบต้นไม้สำหรับการแจงส่วน (Parse Tree) โดยเริ่มต้นจากรากของต้นไม้ (Root) จนกระทั่งถึงโหนดใบ (Leaves) ส่วนการทำงานแบบล่างขึ้นบนจะทำการสร้างอัลกอริทึมแบบต้นไม้สำหรับการแจงส่วน (Parse Tree) โดยเริ่มจากโหนดใบย้อนกลับไปยังโหนดราก สิ่งที่ได้จากการทำพาร์เซอร์ได้แก่

- จัดการรวบรวมข้อมูลเข้าไปเก็บไว้ในตารางสัญลักษณ์
- ทำการตรวจสอบชนิดของตัวแปร
- ทำการตรวจสอบการวิเคราะห์ความหมายอื่นๆ
- ทำการสร้างไค้ดกลางขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.6.2 วัตถุประสงค์ของส่วนจัดการข้อผิดพลาดในพาร์เซอร์

โดยทั่วไปการตรวจสอบข้อผิดพลาดในโปรแกรมและการฟื้นฟูสภาพในโปรแกรมตัวแปลภาษานั้นจะมารวมกันอยู่ที่ส่วนการวิเคราะห์รูปแบบประโยค (Syntax analysis) เหตุผลหลักที่ยืนยันในเรื่องนี้คือข้อผิดพลาดจำนวนมากมีลักษณะที่เกี่ยวข้องกับโครงสร้างอยู่โดยธรรมชาติหรือข้อผิดพลาดจะถูกเปิดเผยออกมาเมื่อกระแสของโทเคนที่ส่งเข้ามานั้นไม่เป็นไปตามเกณฑ์ของภาษานั้นๆ ดังนี้

- สามารถรายงานข้อผิดพลาดที่ตรวจพบได้อย่างชัดเจน
- สามารถฟื้นฟูสภาพจากข้อผิดพลาดได้อย่างรวดเร็ว
- ไม่ควรที่จะทำให้กระบวนการทำงานของส่วนที่เหลืออยู่ทำงานช้าลง

### 2.6.3 หลักการคอนเท็กซ์ฟรีแกรมม่า (Context-Free Grammar หรือ CFG)

การทำงานของพาร์เซอร์นั้นจะใช้หลักการของคอนเท็กซ์ฟรีแกรมม่าเข้ามาช่วย ซึ่งหลักการของคอนเท็กซ์ฟรีแกรมม่านั้นจะประกอบด้วยตัวแปร 4 ชนิด ได้แก่

- V Variables or non-terminals
- T Terminals
- P Productions or rules
- S Start symbol

ตัวอย่าง ลักษณะการเขียนแกรมม่าแบบ ซีเอฟจี

$$E \rightarrow E+E \mid E * E \mid (E) \mid -E \mid id$$

ดังนั้นจะได้ว่า

$$V = \{E\},$$

$$T = \{+, *, (, ), -, id\},$$

$$S = E$$

$$E + E$$

### 2.6.4 การแจงส่วนชนิดล่างสู่ด้านบน (Bottom-Up Parsing)

การกำหนดว่าประโยคที่เขียนขึ้นนั้นเป็นประโยคของภาษาหนึ่งหรือไม่นั้น ต้องพิจารณาจากไวยากรณ์ของภาษานั้นๆ เช่นประโยค “ฉันกินข้าว” เป็นประโยคภาษาไทยเนื่องจากเขียนตรงตามหลักไวยากรณ์ เช่น ประธาน+กริยา+กรรม เป็นต้น

ซึ่งในโครงการนี้เป็นการแสดงภาพเคลื่อนไหวจากข้อความภาษาไทย ซึ่งโปรแกรมต้องสามารถวิเคราะห์รูปแบบประโยคตามไวยากรณ์ภาษาไทยได้ด้วยหลักการแจงส่วน (Parsing) ชนิดจากด้านล่างสู่ด้านบน มีการพยายามที่จะสร้างอัลกอริทึมแบบต้นไม้สำหรับการแจงส่วน (Parse tree) จากประโยคที่ได้รับเข้ามา โดยเริ่มจากโหนดใบและไล่ขึ้นไปยังโหนดราก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีสายอักขระย่อยที่ตรงกัน กับเทอมด้านขวาของรูปแบบการกำหนดประโยค (Production) ซึ่งจะทำให้การลดรูปอักขระโดยสัญลักษณ์ทางด้านซ้ายของรูปแบบการกำหนดประโยคที่นำมาใช้นั้น

กำหนดให้	S	-->	aABe
	A	-->	Abc  b
	B	-->	d

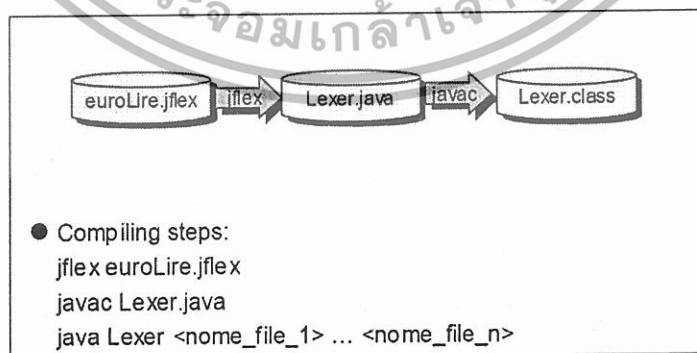
สายอักขระ abcde สามารถลดรูปไปยัง S ได้โดย

abcde --> aAbcde --> aAde --> aABe --> S

ตัวพาร์เซอร์จะทำการอ่านสายอักขระนำเข้ามาทีละตัวเพื่อค้นหาสายอักขระย่อยที่ตรงกันกับสายอักขระทางฝั่งขวาของโปรดักชันบางอัน อย่างเช่นในตอนเริ่มต้นสายอักขระย่อย “b” และ “d” นั้นตรงกับสายอักขระทางฝั่งขวาของโปรดักชัน 2 ข้อ ในที่นี้จะเลือกสายอักขระย่อย “b” และทำการแทนที่ด้วยฝั่งซ้ายของโปรดักชันที่นำมาใช้ (A --> b) ซึ่งก็คือ “A” จึงทำให้ได้เป็น “aAbcde” ต่อไปสายอักขระย่อย “Abc” “b” และ “d” นั้นตรงกับสายอักขระทางฝั่งขวา ในที่นี้จะเลือกแทนที่สายอักขระย่อย “Abc” ด้วย “A” (A --> Abc) ซึ่งก็จะได้เป็น “aAde” ต่อไปเลือกแทนที่ “d” ด้วย “b” (B --> b) จะได้ “aABe” และท้ายที่สุดก็แทนที่สายอักขระนี้ด้วย “S” (S --> aABe) สิ่งที่ได้รับก็คือสายอักขระ “abcde” สามารถถูกลดรูปให้กลายเป็นอนเทอมนิ้นตัวเริ่มต้น “S” ได้ตามต้องการ

## 2.7 การใช้งานไลบรารีเจฟเล็กและไลบรารีคัพ (Library JFlex, LibraryCUP)

ไลบรารีเจฟเล็ก (Library JFlex) ช่วยในส่วนของงานวิเคราะห์คำศัพท์ (Lexical analysis) ซึ่งเป็นการเปลี่ยนประโยคที่กำหนดไว้ให้เป็นโทเคนหรือกลุ่มคำตามที่ได้กำหนดไว้ เพื่อนำไปสร้างโปรแกรมที่ใช้ตรวจสอบรูปแบบประโยค ซึ่งโปรแกรมตรวจสอบรูปแบบประโยคที่ถูกสร้างขึ้นมานี้จะมีตัวกราด (Scanner) ทำหน้าที่รับค่าข้อมูล และนำข้อมูลที่รับเข้ามาไปเปรียบเทียบกับนิพจน์ (Expression) ที่กำหนดไว้ทำให้ได้ผลลัพธ์เป็นข้อมูลที่สอดคล้องกับนิพจน์นั้นๆ



รูปที่ 2.7 การทำงานของไลบรารีเจฟเล็ก (Library JFlex)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.1 โครงสร้างของไฟล์แอสกี สามารถแบ่งได้เป็น 3 ส่วนคือ

2.7.1.1 ส่วนที่ถูกคัดลอก (Code Section) ไปสร้างเป็นไฟล์กราด

2.7.1.2 ส่วนการสร้างกฎ (Declarations Section) สามารถที่จะสร้างกฎซ้อนกันได้ด้วยการสร้างกฎย่อยๆ

2.7.1.3 ส่วนที่ใช้กำหนดความสัมพันธ์ (Rules Section) ระหว่างนิพจน์กับผลลัพธ์ นิพจน์สามัญเป็นส่วนที่ใช้อธิบายลำดับของตัวอักษรโดยใช้เครื่องหมายต่างๆ เช่น “[ ] ^ - ? . \* + | ( ) \$ { } % < >”

กลุ่มของอักขระ (Character Class) คือส่วนที่อยู่ภายในเครื่องหมาย [ ]: และมีการใช้เครื่องหมายช่วยอธิบาย เช่น [0-9]

หมายความว่าตรงกับเลข 0 ถึงเลข 9

เครื่องหมาย ^ เช่น [^0-9] หมายความว่าตรงกัน ทุกตัวอักษร ยกเว้น เลข 0 ถึงเลข 9 สมมุติให้ a และ b เป็นนิพจน์ (Regular Expression)

a | b (union)

ตรวจสอบว่าเป็น a หรือ b

a b (concatenation)

ตรวจสอบว่าข้อมูลที่ได้รับเป็นข้อมูลรูปแบบ a และตามด้วยข้อมูลรูปแบบ b

a\* (kleene closure)

การประกาศว่า สามารถมีหรือ ไม่มีข้อมูลชนิด a

a+ (iteration)

การประกาศว่า จำเป็นต้องมีข้อมูลชนิด a อย่างน้อย 1 ตัว

a? (option)

การตรวจสอบว่ามีค่าข้อมูลชนิด a หรือไม่

!a (negation)

การตรวจสอบว่า เป็นข้อมูลชนิดอื่นทุกชนิด ยกเว้นข้อมูลชนิด a

~a (upto)

การตรวจสอบว่า เป็นข้อมูลชนิดอื่นทุกชนิด ยกเว้นข้อมูลชนิด a

a{n} (repeat)

การตรวจสอบว่าตรงกับข้อมูลชนิด a จำนวน n ครั้ง

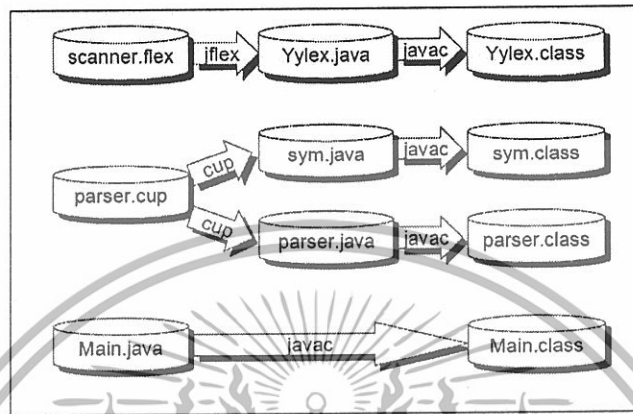
a{n,m} การตรวจสอบว่าตรงกับข้อมูลชนิด a อย่างน้อยที่สุด n ครั้ง และมากที่สุด m ครั้ง

(a) การตรวจสอบว่าตรงกับข้อมูลชนิด a หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไลบรารีคัพ (Library CUP) ที่ใช้ในส่วนของกรวิเคราะห์รูปแบบประโยค ซึ่งใช้หลักการทำคอนเทคฟรีแกรมมา ซึ่งเป็นส่วนที่ใช้ในการวิเคราะห์รูปแบบประโยค โดยจะนำประโยคที่ได้รับเข้าไปนั้นเปรียบเทียบกับกฎ (rule) ต่างๆ ที่กำหนดไว้เป็นภาษาปกติ (Formal Language)

### 2.7.2 ขั้นตอนการใช้งานของไลบรารีเจฟเล็กและไลบรารีคัพ (Library JFlex, Library CUP)



รูปที่ 2.8 การทำงานของ คอนเทคฟรีแกรมมา

- 1) หลักการทำงานของไลบรารีเจฟเล็กเริ่มจากกำหนดรูปแบบนิพจน์ โดยกำหนดความสัมพันธ์ระหว่างนิพจน์กับการกระทำ โดยจะเขียนเป็นไฟล์ชื่อ JFlex แล้วนำไปคอมไพล์ด้วยคำสั่ง `JFlex Lexer.jflex` จะได้ผลลัพธ์เป็นไฟล์ชื่อ `java` แล้วจึงนำไปคอมไพล์อีกครั้ง ด้วยคำสั่ง `javac name.java` และตั้งรัน โปรแกรม
- 2) หลักการทำงานของไลบรารีคัพนั้นเริ่มจากการกำหนดกฎต่างๆ โดยจะแบ่งออกเป็นคำหน่วยสุดท้าย และคำที่สามารถขยายได้ ต่อจากนั้นจะทำการกำหนดกฎต่างๆ และกระบวนการทำงานหากข้อมูลที่ได้รับมานั้นถูกต้องตามกฎที่กำหนด ซึ่งกระบวนการเหล่านี้จะได้ไฟล์ชื่อ `cup` เมื่อคอมไพล์ด้วยคำสั่ง `java java-cup-11a.Main parser.cup` จะทำให้ได้ ไฟล์ 2 ไฟล์ชื่อ `symbol.java` และ `parser.java` ออกมา
- 3) สร้าง `Main.java` เพื่อใช้เป็นตัวเชื่อมการทำงานของไฟล์ทั้งสอง ซึ่งจะทำให้สามารถติดต่อรับส่งข้อมูลระหว่างกันได้
- 4) คอมไพล์ไฟล์ทั้งหมด ได้แก่ `Lexer.jflex`, `parser.java` และ `Main.java` ด้วยคำสั่ง `javac scanner.java Lexer.java parser.java Main.java`
- 5) รัน `Main.class` ด้วยคำสั่ง `java Main`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งในการรัน

```
jflex scanner.jflex
```

```
java java-cup-11a.Main parser.cup
```

```
javac scanner.java Lexer.java parser.java Main.java
```

```
java Main
```

### 2.7.3 ตัวอย่าง โค้ดของการทำคอนเทกฟรีแกรมมา

scanner.jflex

```
import java_cup.runtime.*;
%%
%class Lexer
%standalone
%cup
/*
%{
    private Symbol symbol(int num){
        return new Symbol(num);
    }
    private Symbol symbol(char sign){
        return new Symbol(sign);
    }
%}
*/
num = [1-9][0-9]*
sign = [a-z]
%%
{num} {return new Symbol(sym.EL);}
{sign} {return new Symbol(sym.VIR);}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นขั้นตอนการกำหนด ลักษณะของข้อมูลที่จะรับเข้าไปทำงานซึ่งมี 2 อย่าง คือ

1) num = [1-9][0-9]\* หมายความว่า ข้อมูลต้องเป็นตัวเลข โดยเลขตัวแรกต้องเป็น 1-9 แต่ตัวถัดไปจะมีหรือไม่มีก็ได้ซึ่งจะอยู่ในช่วง 0-9

2) sign = [a-z] หมายความว่าข้อมูลที่เข้ามานั้นต้องเป็นตัวอักษรเพียง 1 ตัวที่อยู่ในช่วง a-z แล้วมีการส่งค่ากลับไปยังตัวแปร EL หรือ VIR โดยเรียกใช้ เมธอด symbol

parser.cup

```
import java_cup.runtime.*;
init with {::Lexer.init();};
scan with {::return Lexer.next_token();};
terminal EL,VIR;
non terminal List,Lista;
start with Lista;
Lista ::= List {::System.out.println("List found");};
List ::= VIR EL;
List ::= EL;
```

เป็นส่วนที่ใช้ในการกำหนดรูปแบบประโยคต่างๆ ซึ่ง Lista มีส่วนประกอบของ List แล้ว List ประกอบด้วย VIR และ EL หรือ EL ซึ่งหากข้อมูลที่รับเข้ามาเป็นไปตามกฎ ผลลัพธ์ที่ได้จากโปรแกรมคือ "List found"

Main.java

```
import java.io.*;
public class Main{
    public static void main (String arg[]){
        try{
            Lexer l = new Lexer(new FileReader(arg[0]));
            parser p = new parser(l);
            Object result = p.parse();
        }catch(Exception e){
            e.printStackTrace();
        }
    }
}
```

เป็นตัวสร้าง วัตถุ จากคลาส Lexer และ Parser โดยนำ วัตถุ ที่ได้จาก Lexer มาเป็น input ของ Parser เพื่อให้ นำข้อมูลนั้นมาวิเคราะห์ว่าเป็นไปตามกฎหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### 2.8.3 การเรียงคำในประโยค

- ประธาน + กริยา
- ประธาน + กริยา + กรรม
- ประธาน + ส่วนขยาย + กริยา
- ประธาน + ส่วนขยาย + กริยา + ส่วนขยาย
- ประธาน + ส่วนขยาย + กริยา + กรรม + ส่วนขยาย
- ประธาน + ส่วนขยาย + กริยา + ส่วนขยาย + กรรม + ส่วนขยาย

ในระดับประโยค นักภาษาศาสตร์ศึกษาถึงโครงสร้างของวลีและประโยค ซึ่งเกิดจากการที่นำคำมาเรียงต่อกันเป็นลำดับ จากซ้ายไปขวา ถ้าพิจารณาประโยคต่อไปนี่ก็จะเห็นได้ว่า การนำคำมาเรียงกันเป็นประโยคนั้นมีกฎเกณฑ์ กล่าวคือ มิใช่คำใดจะอยู่ในตำแหน่งใดก็ได้ และถ้าไม่เรียงตามกฎเกณฑ์ก็จะเป็นประโยค เช่น

- แจ่วใส่เสื้อสีเขียวตัวนี้ทุกวันพุธ (ถูกต้อง)
- ทุกวันพุธ แจ่วใส่เสื้อสีเขียวตัวนี้ (ถูกต้อง)
- เสื้อสีเขียวตัวนี้ แจ่วใส่ทุกวันพุธ (ถูกต้อง)
- แจ่วตัวนี้ เสื้อใส่สีเขียววันพุธ (ไม่ถูกต้อง)
- ทุกแจ่ว เสื้อวันพุธใส่สีเขียวตัวนี้ (ไม่ถูกต้อง)
- วันพุธทุกสีเขียวใส่ตัวนี้แจ่ว (ไม่ถูกต้อง)
- เสื้อแจ่วนี้ทุกวันพุธตัวสีเขียวใส่ (ไม่ถูกต้อง)

ถ้าวิเคราะห์รายละเอียดต่อไปอีกก็จะเห็นว่า ถ้าไม่เรียงคำตามกฎเกณฑ์ จะได้สิ่งที่ไม่เป็นประโยคไม่สามารถสื่อความหมายให้เข้าใจได้เท่าสิ่งที่เรียงตามเกณฑ์จะได้เป็นประโยคซึ่งสื่อความหมายได้เข้าใจมากกว่า และเมื่อพิจารณาจากข้อมูล ประโยคที่ถูกต้อง ก็ได้ตั้งสมมติฐานว่าคำในภาษานั้นแตกต่างกัน แต่ละประเภทมีลักษณะและหน้าที่แตกต่างกัน และคำที่อยู่ในประโยคเดียวกันมีความสัมพันธ์กันน้อยไม่เท่ากัน เป็นต้นเช่น

- "สีเขียว" สัมพันธ์ใกล้ชิดกับ "เสื้อ" มากกว่าสัมพันธ์กับคำว่า "ใส่"
- "ทุก" สัมพันธ์กับ "วันพุธ" มากกว่าคำอื่นๆในประโยค

ซึ่งแสดงว่าคำในประโยคเดียวกันเกาะกันเป็นกลุ่ม ตามความสัมพันธ์หรือที่เรียกกันว่า "วลี" และกลุ่มคำเหล่านี้ก็แยกเป็นประเภทต่างๆ ได้ ในแง่ของความสัมพันธ์ของคำและกลุ่มคำในประโยค เมื่อพิจารณาในรายละเอียด จะเห็นได้ว่าความสัมพันธ์เหล่านี้มีได้อยู่ในระดับเดียวกัน (จากซ้ายขวา) ทั้งหมด แต่สูงต่ำลดหลั่นกัน

นักภาษาศาสตร์ใช้วิธีแยกแยะคำในประโยคภาษาไทยออกมาเป็นกลุ่มในลักษณะนี้ เพื่อแสดงถึงความสัมพันธ์ของกลุ่มคำ และระดับของความสัมพันธ์ว่าอยู่ระดับเดียวกันหรือไม่ ในแง่ของการเรียงคำ มีการศึกษาประโยคในภาษาต่างๆ

#### 2.8.4 การเรียงคำในประโยคของภาษาต่างๆ มี 2 แบบใหญ่ๆ คือ

- 1) การวางคำกริยาไว้หน้ากรรม หรือ กริยา+กรรม
- 2) การวางคำกริยาไว้หลังกรรม หรือ กรรม+กริยา

ตัวอย่างเช่น ภาษาไทย ภาษาเขมร ภาษาเวียดนาม ภาษาอังกฤษ โดยทั่วไปวางคำกริยาไว้หน้ากรรม ส่วนภาษาพม่า ภาษาญี่ปุ่น ภาษาฮินดู โดยทั่วไปวางกรรมไว้หน้ากริยาหรือกริยาไว้ท้ายประโยค เช่น ภาษาพม่า เรียงคำดังนี้ "อุถิ่น-ซองจดหมาย-ซื้อ" "เขา-ตลาด-ไป" และการเรียงคำ 2 แบบนี้ผูกพันกับการเรียงคำอื่นๆ ด้วย เป็นต้นว่า คำวิเศษณ์อยู่หน้าหรือหลังคำนาม คำบุพบทอยู่หน้าหรือหลังคำนาม คำเปรียบเทียบอยู่หน้าหรือหลังคำเปรียบเทียบ เช่น ภาษาไทยวางคำเปรียบเทียบไว้หลังคำวิเศษณ์ เช่น ใหญ่กว่า ดีกว่า แต่ภาษาพม่าเรียงคำเป็น กว่าใหญ่กว่าดี หรือภาษาอังกฤษเรียงคำขยายไว้หน้าคำนาม เช่น ใหญ่ บ้าน ดีเด็ก แต่ภาษาไทยเรียงแบบตรงข้าม เช่น บ้านใหญ่ เด็กดี เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

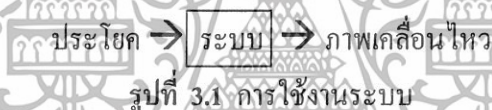
## บทที่ 3

### การวิเคราะห์และออกแบบโครงการ

โครงการนี้เป็นการพัฒนาโปรแกรมที่สามารถอ่านประโยคภาษาไทยแล้วแสดงเป็นภาพเคลื่อนไหวที่สอดคล้องกัน ซึ่งการทำงานเพื่อให้ได้มานี้ต้องมีการแบ่งการทำงานเป็นหลายส่วน เช่นบทนี้จะกล่าวถึงการวิเคราะห์การทำงานต่างๆของระบบ และออกแบบการทำงานของระบบ โดยจะแบ่งการวิเคราะห์และการออกแบบนี้ออกเป็น 2 แบบ คือส่วนการออกแบบระบบ และการออกแบบฐานข้อมูล

#### 3.1 วิเคราะห์ลักษณะการทำงานจากมุมมองผู้ใช้

การออกแบบระบบแบ่งเป็นการวิเคราะห์การทำงานรูปแบบต่างๆตามระดับที่ผู้ใช้งานต้องการจะสื่อสาร ซึ่งการออกแบบระบบนั้นสามารถแสดงผลให้อยู่ในรูปของแผนภาพ (Diagram) โดยที่แต่ละแผนภาพนั้นสามารถอธิบายถึงความละเอียดของการทำงานของระบบที่แตกต่างกันไป โดยวิเคราะห์การทำงานจากมุมมองของผู้ใช้ได้ดังรูปที่ 3.1

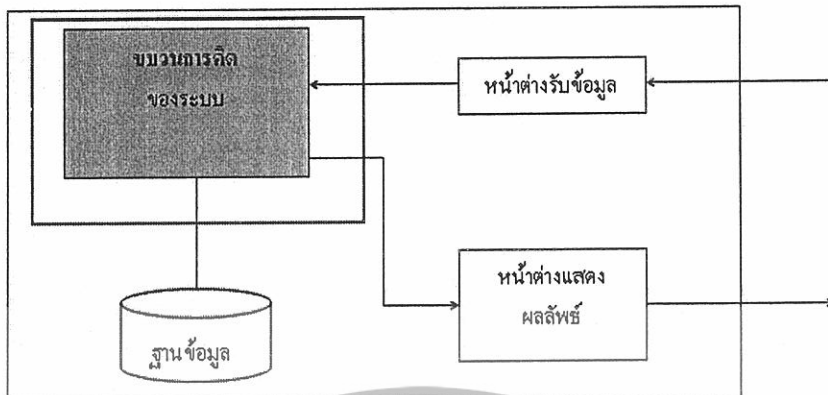


รูปที่ 3.1 การใช้งานระบบ

จากรูปนี้อธิบายถึงการทำงานในรูปแบบของระบบที่เป็นลักษณะของกล่องดำ (Black Box) คือผู้ใช้งานไม่จำเป็นต้องรู้การทำงานของระบบ ผู้ใช้รู้เพียงว่าต้องใส่ข้อมูลอะไรเข้าไปเพื่อให้ได้ข้อมูลอะไรออกมาเท่านั้น ซึ่งในที่นี้ผู้ใช้งานจะต้องใส่ข้อมูลที่เป็นข้อความประโยคภาษาไทยลงไปในระบบ เมื่อระบบรับข้อมูลนั้นเข้าไปแล้วจะประมวลผลแล้วแสดงผลลัพธ์กลับมาเป็นภาพเคลื่อนไหวตามประโยคที่ใช้นั้นกรอกลงไปในระบบในตอนต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 วิเคราะห์ส่วนแนวคิดของการทำงานระบบ



รูปที่ 3.2 กรอบแนวคิด

จากการทำงานของระบบนั้น โดยแบ่งเป็นขั้นตอนแนวคิดเป็นส่วนๆ ได้ดังนี้

#### 3.2.1 ส่วนรับข้อมูล

โดยส่วนรับค่านั้นจะรับประโยชน์จากส่วนเชื่อมต่อกับผู้ใช้ แล้วนำเข้ามาในระบบเพื่อนำข้อมูลเหล่านั้นไปประมวลผลให้ได้มาซึ่งผลลัพธ์ที่เป็นภาพเคลื่อนไหว

#### 3.2.2 ส่วนขบวนการคิดของระบบ

ส่วนขบวนการคิด คือ ส่วนต่างๆที่เป็นกระบวนการคิดและประมวลผลของระบบ ซึ่งเป็นการกระบวนการที่ใช้ทฤษฎีที่ศึกษามาประยุกต์ใช้ โดยถูกแบ่งย่อยออกเป็นส่วนๆดังนี้

- 1) ส่วนตัดคำ คือการนำประโยคที่ได้รับจากผู้ใช้ไปตัดให้เป็นคำย่อยๆ เพื่อนำคำเหล่านี้ไปตรวจสอบกับฐานข้อมูล
- 2) นำข้อมูลของแต่ละคำนั้นมาวิเคราะห์ความถูกต้อง เพื่อหาประธาน กริยา และกรรมของประโยค โดยใช้หลักการคอนเทคฟรีแกรมมา
- 3) ส่วนตีความ คือนำข้อมูลนั้นมาหาความหมายโดยรวมของประโยค เพื่อนำความหมายนั้นไปแปลงให้กลายเป็นรูปแบบของการแสดงภาพเคลื่อนไหว ซึ่งต้องมีการเก็บโครงสร้างของความสัมพันธ์ของวัตถุต่างๆ ที่จะนำมาแสดง
- 4) ส่วนติดต่อกับฐานข้อมูล คือนำคำต่างๆ ที่ได้จากการตัดคำไปตรวจสอบกับฐานข้อมูล แล้วดึงข้อมูลต่างๆ ของคำนั้นออกมา โดยฐานข้อมูลนั้นจะเก็บคำศัพท์ที่ใช้เชื่อมกับภาษาไทยเป็นรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

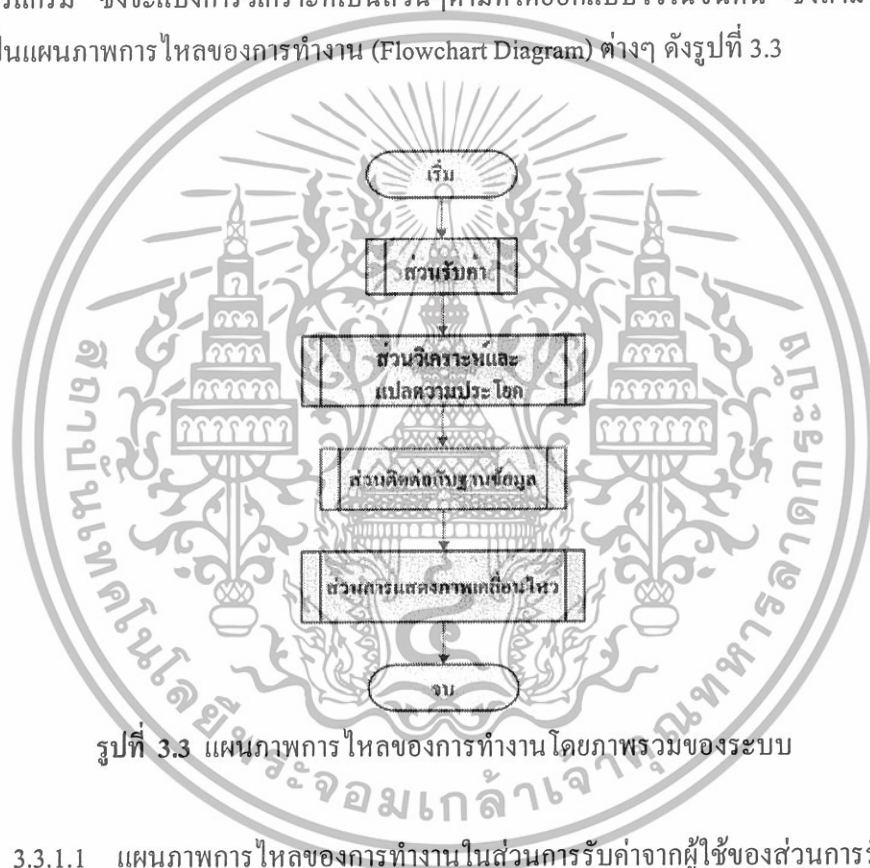
### 3.2.3 ส่วนแสดงภาพ

ส่วนแสดงภาพ คือ ส่วนที่นำรูปแบบของข้อมูลที่ได้นั้น มาสร้างเป็นภาพเคลื่อนไหว แล้วส่งการแสดงผลออกไปยังหน้าต่างแสดงผล

## 3.3 วิเคราะห์ส่วนแนวคิดขั้นตอนการทำงานของระบบ

### 3.3.1 แผนภาพการไหลของการทำงาน ( Flowchart Diagram )

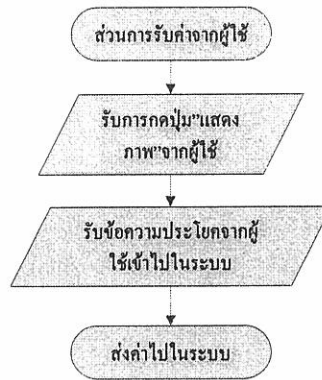
การวิเคราะห์ขั้นตอนการทำงานของโปรแกรม เป็นการวิเคราะห์ถึงลำดับขั้นตอนการทำงานของโปรแกรม ซึ่งจะแบ่งการวิเคราะห์เป็นส่วนๆตามที่ได้ออกแบบไว้ในขั้นต้น ซึ่งสามารถนำมาเขียนเป็นแผนภาพการไหลของการทำงาน (Flowchart Diagram) ต่างๆ ดังรูปที่ 3.3



รูปที่ 3.3 แผนภาพการไหลของการทำงานโดยภาพรวมของระบบ

3.3.1.1 แผนภาพการไหลของการทำงานในส่วนการรับค่าจากผู้ใช้งานของผู้ใช้ของผู้ใช้เป็นส่วนการรับค่าจากผู้ใช้ เป็นขั้นตอนของการรับประโศคจากผู้ใช้เพื่อนำประโศคนั้น ไปประมวลผลในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แผนภาพการไหลของการทำงานในส่วนการรับค่าจากผู้ใช้

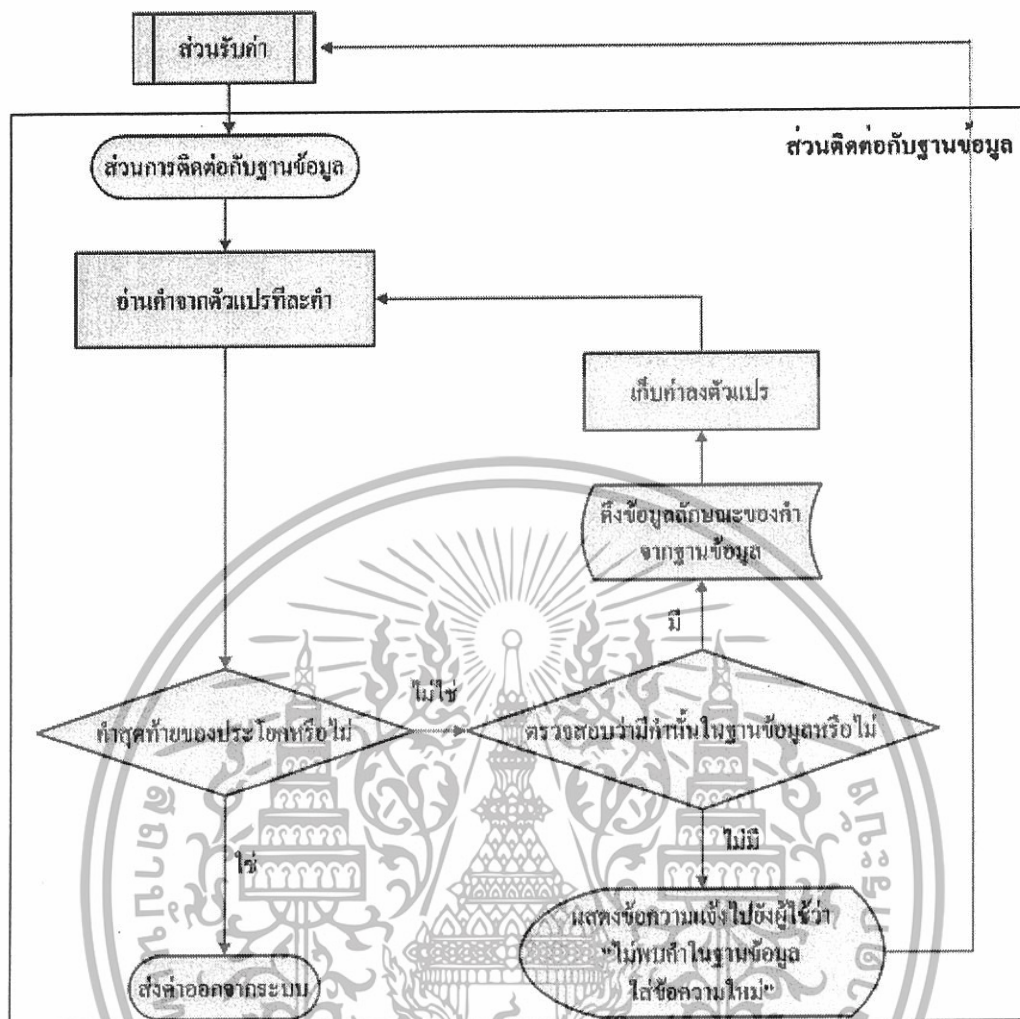
3.3.1.2 แผนภาพการไหลของการทำงานส่วนวิเคราะห์และแปลความประโยค ส่วนนี้เป็นขั้นตอนของการนำประโยคจากผู้ใช้เข้ามาแล้วตัดแบ่งคำออกเป็นคำย่อยๆ จากนั้นจึงวิเคราะห์รูปประโยค เพื่อตรวจสอบว่าแต่ละคำนั้นมีหน้าที่เป็นประธาน กริยา และกรรม หรือพวกส่วนขยายของประโยค ซึ่งหลักการนี้จะพิจารณาถึงตำแหน่งของนาม และกริยาเป็นหลัก ต่อจากนั้นจึงตรวจสอบประโยคว่ามีความถูกต้องและประโยคสามารถหาความหมายได้ เมื่อผ่านการตรวจสอบก็ทำการส่งค่าไปยังตัวติดต่อฐานข้อมูล



รูปที่ 3.5 แผนภาพการไหลของการทำงานของส่วนวิเคราะห์และแปลความประโยค

3.3.1.3 แผนภาพการไหลของการทำงานของส่วนการติดต่อการดึงค่าจากฐานข้อมูล เป็นขั้นตอนที่นำคำต่างๆที่ได้รับเข้ามาไปตรวจสอบคำในฐานข้อมูล แล้วดึงข้อมูลเหล่านั้นออกมาซึ่งได้แก่ คำศัพท์ภาษาไทย คำศัพท์ที่เป็นภาษาอังกฤษ เพื่อนำไปวิเคราะห์หาความหมายและใช้ในการแสดงภาพต่อไป แต่หากไม่พบคำในฐานข้อมูลก็จะส่งข้อความไปบอกเพื่อให้ผู้ใช้ให้กรอกข้อความใหม่ผ่านทางหน้าจอ

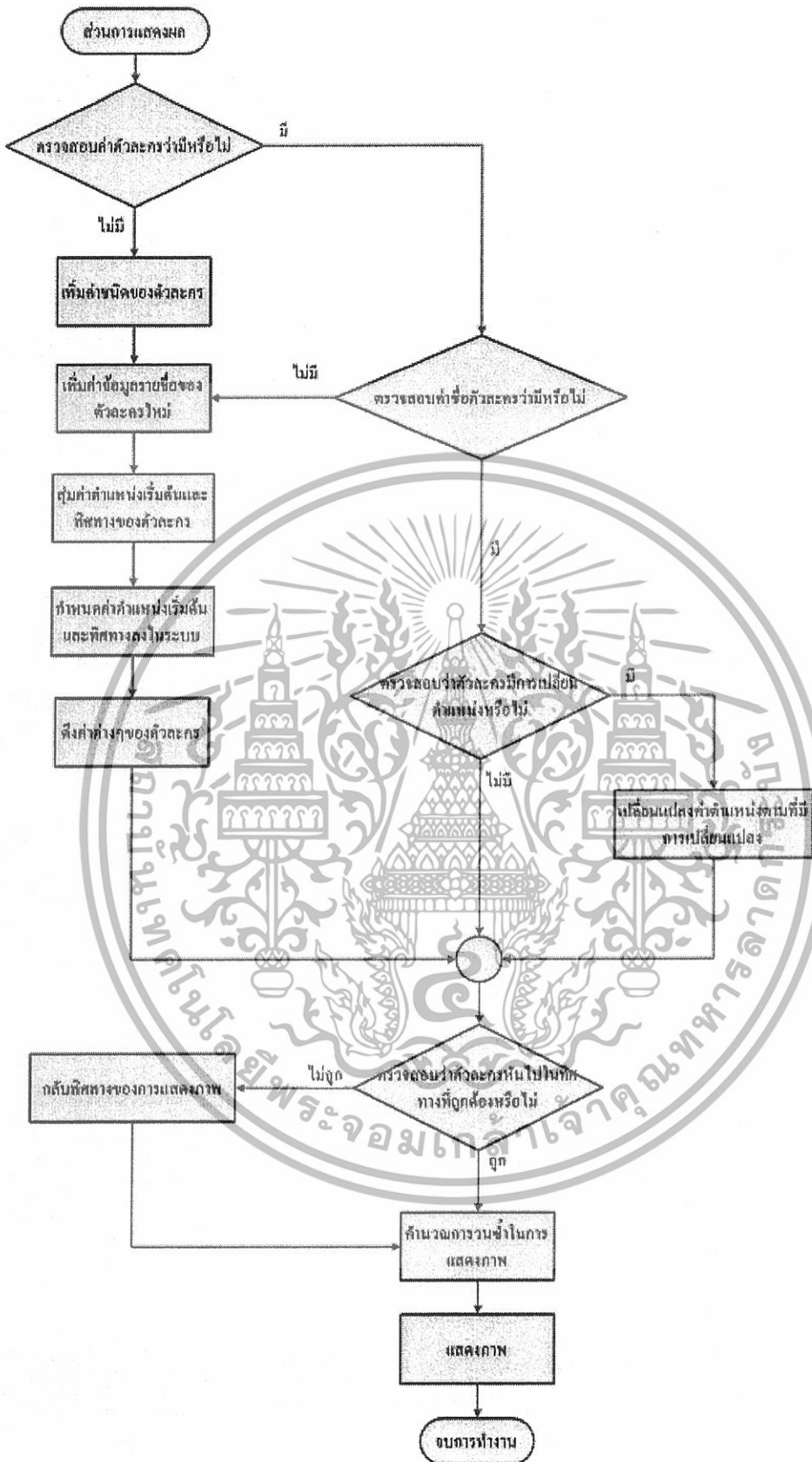
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แผนภาพการไหลของการทำงานของการทำงานของการติดต่อดึงค่าจากฐานข้อมูล

3.3.1.4 แผนภาพการไหลของส่วนการแสดงผล ซึ่งการแสดงผลนี้เป็นการค้นหาการแสดงผลให้มีความสอดคล้องกับความหมายของประโยคที่ได้ตีความมาแล้ว และมีการเก็บข้อมูลความสัมพันธ์ของวัตถุ หรือตัวละครที่แสดงออกทางหน้าจอ เพื่อให้สามารถแสดงผลภาพเคลื่อนไหวที่โดยอ้างอิงจุดปลายทางได้ ซึ่งจะมีการเก็บชนิดของตัวละคร ตำแหน่ง และทิศทาง ซึ่งก่อนที่จะดึงค่าเหล่านั้นออกมาใช้นั้นจะต้องมีการตรวจสอบค่าภายในต่างๆก่อนว่ามีค่าอยู่หรือไม่ หากยังก็ให้เพิ่มค่าเข้าไป โดยการสุ่มค่าขึ้นมาตามความเหมาะสม แล้วนำค่าเหล่านั้นไปคำนวณเพื่อหาค่าอื่นที่ต้องการใช้ได้แก่ คำนวณระยะทาง จำนวนรอบการแสดงผล ตรวจสอบทิศทาง แล้วส่งค่าตัวแปรเหล่านั้นไปแสดงผลเป็นภาพเคลื่อนไหว ดังรูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แผนภาพการไหลของส่วนการแสดงภาพ

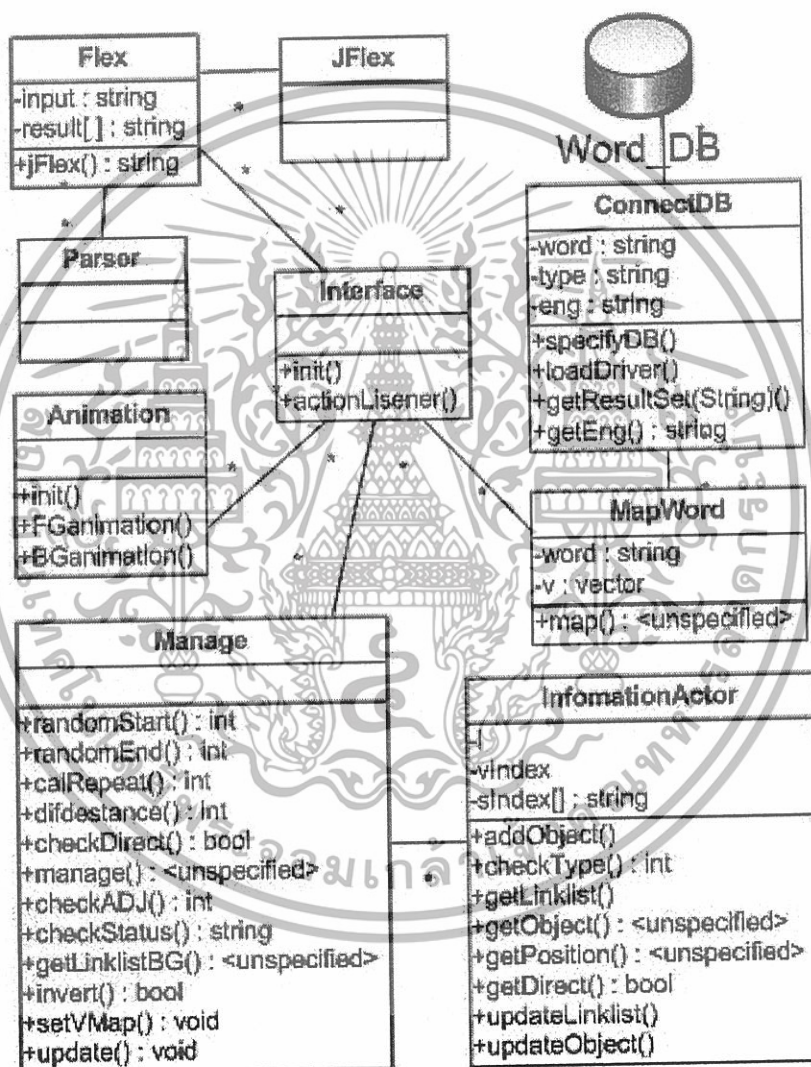
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 วิเคราะห์และออกแบบการพัฒนาโปรแกรม

เมื่อวิเคราะห์การทำงานของระบบได้การออกแบบคลาสแล้ว จึงมาวิเคราะห์การทำงานของ เมธอดของแต่ละคลาส และหาความสัมพันธ์ของแต่ละคลาสได้เป็นคลาสแผนภาพดังนี้

#### 3.4.1 แผนภาพของคลาส (Class Diagram)

แผนภาพของคลาสเป็นแผนภาพที่แสดงถึงความสัมพันธ์ของคลาส และบอกถึงรายละเอียดของส่วนต่างๆของคลาสได้แก่ ตัวแปร (Attribute) เมธอด (Method) ซึ่งคลาสแผนภาพของระบบเป็นดังนี้



รูปที่ 3.8 แผนภาพของคลาสของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## CRC Card

## ตารางที่ 3.1 ตารางแสดงคลาส Interface

ด้านหน้า :

ชื่อคลาส: Interface	ลำดับ: 1	ชนิด: Boundary
คำอธิบาย: คลาสที่ติดต่อกับผู้ใช้ โดยจะรับประโยคเพื่อนำมาใช้ในการทำงานของระบบ , รับคำสั่งจากเมนูต่างๆ และแสดงภาพเคลื่อนไหวที่เป็นผลลัพธ์ของระบบ		
<b>เมธอด</b> <ul style="list-style-type: none"> <li>- init()</li> <li>- ActionListener()</li> </ul>	<b>คลาสที่เกี่ยวข้อง</b> <ul style="list-style-type: none"> <li>- Flex</li> <li>- MapWord</li> <li>- Manage</li> <li>- Animation</li> </ul>	

ด้านหลัง :

ตัวแปร:
<b>ความสัมพันธ์:</b> <b>Generalization (a-kind-of):</b> <b>Aggregation (has-parts):</b> <b>Other Associations:</b> - Flex - MapWord - Manage - Animation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตารางที่ 3.2 ตารางแสดงคลาส Flex

ด้านหน้า :

ชื่อคลาส: Flex	ลำดับ: 2	ชนิด: Controller
คำอธิบาย: คลาสที่ใช้ในการตัดคำและจำแนกประเภทของคำ โดยเรียกใช้ผ่าน Library Flex&CUP จากประโยชน์ที่ได้รับจากผู้ใช้ที่ติดต่อผ่านทางคลาส Interface		
<b>เมธอด</b>	<b>คลาสที่เกี่ยวข้อง</b>	
- JFlex()	<ul style="list-style-type: none"> <li>- Interface</li> <li>- JFlex</li> <li>- Parser</li> </ul>	

ด้านหลัง :

<b>ตัวแปร:</b>	<ul style="list-style-type: none"> <li>- input</li> <li>- result</li> </ul>
<b>ความสัมพันธ์:</b>	<p><b>Generalization (a-kind-of):</b></p> <p><b>Aggregation (has-parts):</b></p> <p><b>Other Associations:</b></p> <ul style="list-style-type: none"> <li>- JFlex</li> <li>- Parser</li> </ul>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตารางที่ 3.3 ตารางแสดงคลาส MapWord

ด้านหน้า :

ชื่อคลาส: MapWord	ลำดับ: 3	ชนิด: Controller
คำอธิบาย: คลาสที่ใช้ในการดึงข้อมูลต่างๆของคำที่ได้รับจากคลาส Swath โดยจะเรียกข้อมูลเหล่านั้นมาโดยผ่านคลาส DBBean		
<b>เมธอด</b>		<b>คลาสที่เกี่ยวข้อง</b>
- map()		- Interface - ConnectDB

ด้านหลัง :

ตัวแปร:	- word[ ] - v
ความสัมพันธ์:	<p><b>Generalization (a-kind-of):</b></p> <p><b>Aggregation (has-parts):</b></p> <p><b>Other Associations:</b> - Interface - ConnectDB</p>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตารางที่ 3.4 ตารางแสดงคลาส ConnectDB

ด้านหน้า:

ชื่อคลาส: ConnectDB	ลำดับ: 4	ชนิด: Controller
คำอธิบาย: คลาสที่ใช้ในการติดต่อกับฐานข้อมูลเพื่อให้สามารถติดต่อกับข้อมูลภายในฐานข้อมูลได้		
<b>เมธอด</b> <ul style="list-style-type: none"> <li>- loadDriver ( )</li> <li>- specifyDB ( )</li> <li>- getResultSet ( )</li> <li>- getEng ( )</li> </ul>		<b>คลาสที่เกี่ยวข้อง</b> <ul style="list-style-type: none"> <li>- MapWord</li> </ul>

ด้านหลัง :

<b>ตัวแปร:</b> <ul style="list-style-type: none"> <li>- word</li> <li>- type</li> <li>- eng</li> </ul>
<b>ความสัมพันธ์:</b> <p><b>Generalization (a-kind-of):</b></p> <p><b>Aggregation (has-parts):</b></p> <p><b>Other Associations:</b> - MapWord</p>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตารางที่ 3.5 ตารางแสดงคลาส Manage

ด้านหน้า :

ชื่อคลาส: Manage	ลำดับ: 5	ชนิด: controller
คำอธิบาย: คลาสที่ใช้ในการจัดการกับการแสดงภาพ เป็นตัวจัดการหาค่าต่างๆของตัวแปรที่จะต้องส่งไปยังคลาส Animation และต้องมีการตรวจสอบเงื่อนไขของการกำหนดค่าต่างๆด้วย		
<b>เมธอด</b>		<b>คลาสที่เกี่ยวข้อง</b>
<ul style="list-style-type: none"> <li>- randomStart ( )</li> <li>- randomEnd ( )</li> <li>- randomDirect ( )</li> <li>- calRepeat ( )</li> <li>- calDistance ( )</li> <li>- difDistance ( )</li> <li>- checkDirect ( )</li> <li>- manage ( )</li> <li>- checkADJ ( )</li> <li>- checkStatus ( )</li> <li>- getLinkedListBG ( )</li> </ul>		<ul style="list-style-type: none"> <li>- update ( )</li> <li>- setVmap ( )</li> <li>- invert ( )</li> <li>- Interface</li> <li>- InformationActor</li> </ul>

ด้านหลัง :

ตัวแปร:	<ul style="list-style-type: none"> <li>- vMap</li> <li>- vManage</li> <li>- sStatus[]</li> <li>- vADJ</li> </ul>	<ul style="list-style-type: none"> <li>- type</li> <li>- directLEFT</li> <li>- posX</li> <li>- posY</li> <li>- invert</li> <li>- width</li> <li>- height</li> </ul>
ความสัมพันธ์:	<p><b>Generalization (a-kind-of):</b></p> <p><b>Aggregation (has-parts):</b></p> <p><b>Other Associations:</b></p> <ul style="list-style-type: none"> <li>- Interface</li> <li>- InformationActor</li> </ul>	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตารางที่ 3.6 ตารางแสดงคลาส InformationActor

ด้านหน้า :

ชื่อคลาส: InformationActor	ลำดับ: 6	ชนิด: Entity
<p>คำอธิบาย: คลาสที่ใช้ในการเก็บข้อมูลของตัวละคร โดยเก็บข้อมูลของตัวละครที่กำลังแสดงอยู่บนหน้าจอ ได้แก่ ชนิด ,ชื่อ ,ตำแหน่ง ,ทิศทางของตัวละคร เพื่อนำข้อมูลเหล่านั้นมาใช้ในการแสดงภาพต่อไป และยังมีการทำงานอื่นๆที่เป็นการสร้างค่าตัวแปรที่จำเป็น เช่น การสุ่มตำแหน่งเริ่มต้นของตัวละคร , จำนวนระยะเวลาทางการเคลื่อนที่ของตัวละคร</p>		
<p><b>เมธอด</b></p> <ul style="list-style-type: none"> <li>- addObject ( )</li> <li>- checkType ( )</li> <li>- getLinkedList ( )</li> <li>- getObject ( )</li> <li>- getPosition ( )</li> <li>- getDirect ( )</li> <li>- updateLinkedList ( )</li> <li>- updateObject ( )</li> </ul>	<p><b>คลาสที่เกี่ยวข้อง</b></p> <ul style="list-style-type: none"> <li>- Manage</li> </ul>	

ด้านหลัง :

<p>ตัวแปร:</p> <ul style="list-style-type: none"> <li>- l</li> <li>- vIndex</li> <li>- sIndex[ ]</li> </ul>
<p>ความสัมพันธ์:</p> <p><b>Generalization (a-kind-of):</b></p> <p><b>Aggregation (has-parts):</b></p> <p><b>Other Associations:</b>           - Manage</p>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตารางที่ 3.7 ตารางแสดงคลาส Animation

ด้านหน้า :

ชื่อคลาส: Animation	ลำดับ: 7	ชนิด: controller
<p>คำอธิบาย: คลาสที่ใช้ในการประมวลการแสดงผลภาพว่าภาพเคลื่อนไหวที่ได้นั้นเคลื่อนที่หรือไม่ และเคลื่อนไปในทิศทางใด ซึ่งจะใช้การแสดงผลภาพเป็นรูปแบบ โดยจะรับค่าต่างๆมาจากคลาส Manage แล้วนำผลลัพธ์ที่ได้นั้นไปแสดงที่คลาส Interface</p>		
<p><b>เมธอด</b></p> <ul style="list-style-type: none"> <li>- init ()</li> <li>- FGanimation ()</li> <li>- BGanimation ()</li> </ul>		<p><b>คลาสที่เกี่ยวข้อง</b></p> <ul style="list-style-type: none"> <li>- Interface</li> </ul>

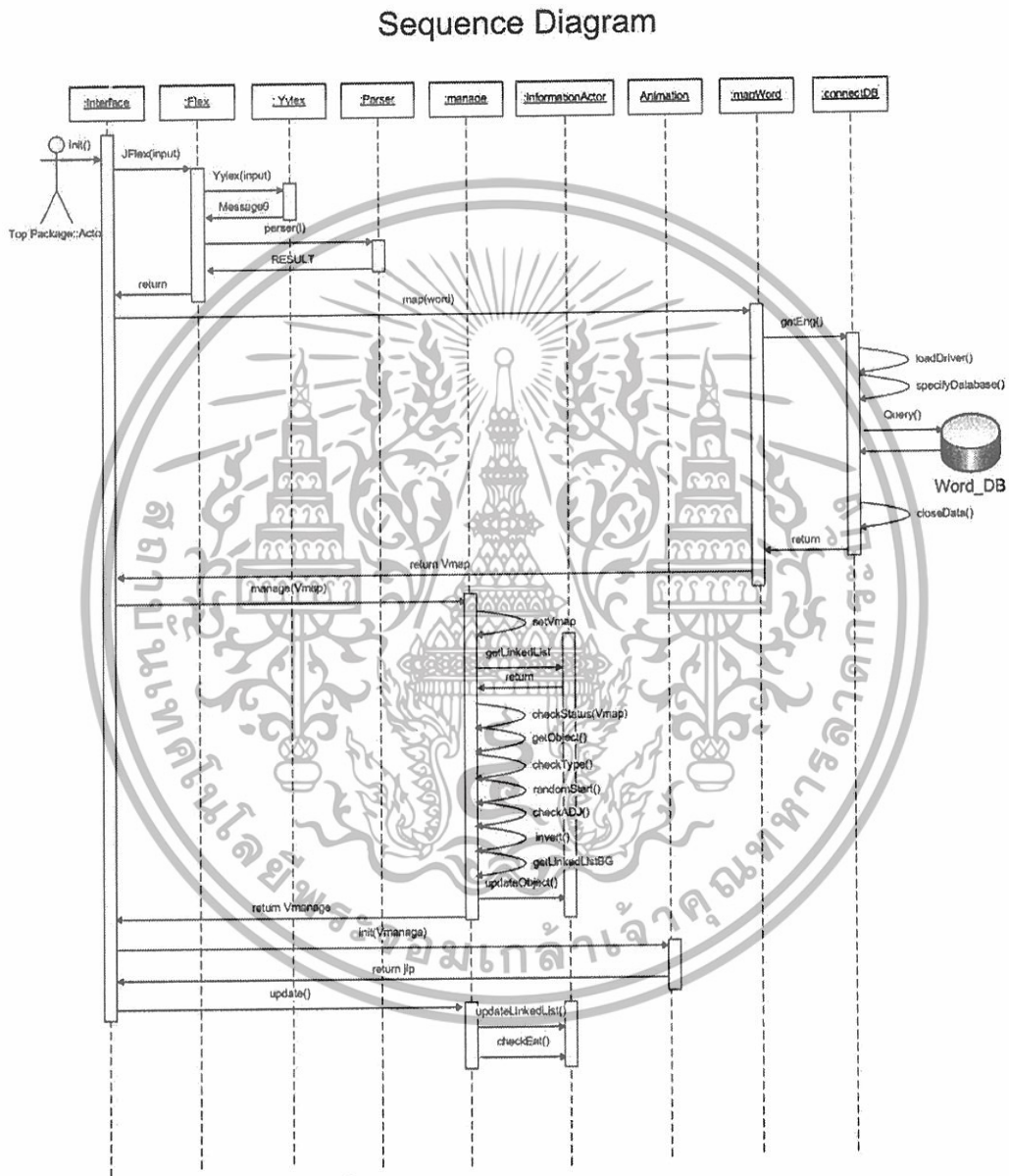
ด้านหลัง :

ตัวแปร:	
ความสัมพันธ์:	<p><b>Generalization (a-kind-of):</b></p> <p><b>Aggregation (has-parts):</b></p> <p><b>Other Associations:</b> - Interface</p>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.2 แผนภาพแสดงลำดับ (Sequence Diagram)

แผนภาพแสดงลำดับ เป็นแผนภาพที่แสดงถึงการติดต่อกันระหว่างวัตถุ โดยที่มีการติดต่อกันผ่านการส่งข้อความไปยังเมธอดของอีกคลาสหนึ่งเป็นการเรียกให้อีกคลาสให้ทำงาน โดยเน้นการติดต่อโดยใช้เวลาเป็นหลัก แผนภาพนี้แสดงลำดับของการทำงานเป็นแผนภาพที่บอกถึงลำดับการทำงานงานของวัตถุของระบบ โดยจะแสดงได้ดังรูปที่ 3.9



รูปที่ 3.9 แผนภาพแสดงลำดับของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบมีขั้นตอนการทำงานดังนี้

1) เริ่มจากผู้ใช้กรอกข้อมูลประโยคเข้าไปในระบบผ่านคลาส Interface แล้วประโยคนั้นจะถูกส่งไปที่คลาส Flex

2) คลาส Flex จะนำประโยคไปเขียนลงไฟล์ก่อนแล้วจึงเรียกไลบรารีของ JFlex เพื่ออ่านข้อความนั้นขึ้นมาตัดคำประโยคให้เป็นคำย่อยๆ แล้วนำคำเหล่านั้นส่งไปยังไลบรารี Parser เพื่อแปลความและหาความหมายโดยรวมของประโยค คำที่ได้จากการตัดคำนั้นไปยังคลาส MapWord

3) คลาส MapWord จะดึงข้อมูลจากฐานข้อมูลของคำที่ส่งเข้ามา โดยเรียกผ่านคลาส ConnectDB

4) คลาส ConnectDB สร้างการเชื่อมต่อกับฐานข้อมูล เพื่อให้สามารถดึงข้อมูลให้กับคลาส MapWord ตามที่ร้องขอได้ แล้วส่งข้อมูลที่ได้กลับไปให้

5) เมื่อคลาส MapWord ได้รับข้อมูลจากคลาส ConnectDB แล้วส่งข้อมูลนั้นคืนให้กับคลาส Interface

6) คลาส Interface ส่งข้อมูลที่ได้รับเหล่านั้นไปยังคลาส Manage โดยผ่านเมธอด manage(input)

7) คลาส Manage เมื่อได้รับข้อมูลเข้าซึ่งจะเป็นข้อมูลของตัวละครที่ต้องการให้แสดง ออกเป็นผลลัพธ์นำมาทำงานดังนี้

7.1) นำข้อมูลเข้ามาแตกเป็นข้อมูลย่อยๆจะได้เป็นข้อมูลของตัวละครต่างๆ แล้วเก็บค่าไว้ในตัวแปร

7.2) นำค่าของตัวละครไปตรวจสอบค่าของตัวละครอยู่ในคลาส InformationActor หรือไม่

- มี โปรแกรมจะดึงค่าต่างๆที่เก็บเอาไว้ในลิงค์ลิส (LinkedList) ที่เก็บค่าของตัวละครนั้นออกมาเก็บไว้

- ไม่มี โปรแกรมจะสับตำแหน่งเริ่มต้น และทิศทางของตัวละครนั้น แล้วเพิ่มค่าชนิดและชื่อของตัวละครและค่าต่างๆเข้าไปในลิงค์ลิส แล้วดึงค่าเหล่านั้นออกมาเก็บไว้

7.3) เมื่อดึงค่าข้อมูลของตัวละครต่างๆแล้วนำมาคำนวณหาระยะทาง และทิศทางของตัวละคร แล้วแก้ไขค่าเหล่านั้นให้ถูกต้อง

7.4) เรียกเมธอด getLinklistBG() เพื่อหาข้อมูลของตัวละครอื่นที่ต้องแสดงเป็นภาพพื้นหลัง

7.5) รวบรวมข้อมูลแล้วส่งคืนค่าให้กับคลาส Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) คลาส Interface เรียกคลาส Animation โดยส่งค่าผลลัพธ์ที่ได้จากคลาส Manage ไปให้

9) คลาส Animation รับค่าเข้าไปแล้วเก็บค่าเหล่านั้นลงตัวแปรต่างๆ โดยมีการทำงานต่อไปดังนี้

9.1) ตรวจสอบว่าตัวครมีการเคลื่อนที่หรือไม่

- มี กำหนดตำแหน่งปลายทางให้กับตัวละครที่เคลื่อนที่
- ไม่มี ออกจากการทำงาน

9.2) เรียกการแสดงภาพตามรูปแบบต่างๆ แล้วส่งภาพเคลื่อนไหวนั้นไปแสดงที่

คลาส Interface



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การพัฒนาโปรแกรม

ขั้นตอนการทำงานเพื่อการแสดงภาพเคลื่อนไหว 2 มิติจากข้อความภาษาไทยนั้น เริ่มต้นตั้งแต่การรับประโยคง่ายๆ ไม่ซับซ้อน แล้วนำไปทำการตัดคำ เมื่อได้คำออกมาแล้วจึงนำไปวิเคราะห์รูปประโยค เพื่อให้ได้ความหมายของคำเหล่านั้นก่อนที่จะนำไปแสดงผลเป็นภาพเคลื่อนไหวตามที่มีความหมายของประโยคผู้ใช้ใส่เข้ามาในระบบ

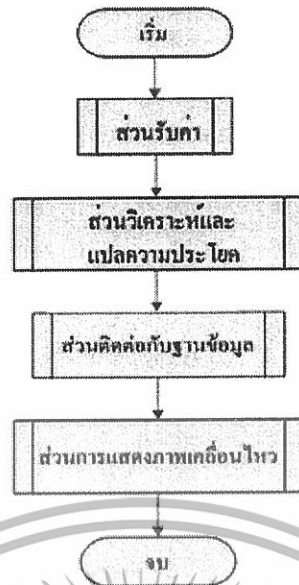
เนื้อหาโดยรวมของบทนี้จะอธิบายแนวคิดและกระบวนการในการพัฒนาโปรแกรมแบ่งเป็น 2 ส่วน คือ การพัฒนาคลาสของส่วนการทำงานต่างๆ และการกำหนดรูปแบบไวยากรณ์ของประโยคภาษาไทยที่โปรแกรมรองรับได้ ในหัวข้อที่ 4.1 และ 4.2 ตามลำดับ

#### 4.1 การพัฒนาคลาสของส่วนการทำงานต่างๆ

จากการวิเคราะห์ลำดับของการทำงานนั้นสามารถแบ่งขั้นตอนการทำงานออกเป็นส่วนๆ ได้ ดังนี้คือ

- 1) ส่วนรับค่า ทำหน้าที่รับข้อความจากผู้ใช้
- 2) ส่วนวิเคราะห์และแปลความประโยค ทำหน้าที่รับข้อความประโยคมาวิเคราะห์หาประธาน กริยา และกรรมของประโยค เพื่อหาความหมายโดยรวมของประโยค
- 3) ส่วนติดต่อกับฐานข้อมูล จะทำหน้าที่เป็นส่วนการนำค่าที่ได้นั้น ไปดึงข้อมูลจากฐานข้อมูล เพื่อจัดเตรียมค่าในการนำมาแสดงผล
- 4) ส่วนการแสดงผล เป็นส่วนที่จัดเตรียมค่าและนำค่าเหล่านั้นมาแสดงเป็นภาพเคลื่อนไหว ซึ่งมีการเรียกใช้งานส่วนติดต่อกับฐานข้อมูลในการจัดเตรียมค่าด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แสดงหลักการทำงานของโปรแกรม

โดยสามารถเขียนการทำงานย่อยๆ ออกเป็นคลาสต่างๆ ได้แก่คลาสที่ได้ออกแบบ ซึ่งแบ่งการพัฒนาออกเป็นส่วนๆ ดังนี้

#### 4.1.1 ส่วนรับค่า

การพัฒนาโปรแกรมนั้นจะอ้างอิงการทำงานจากลำดับการทำงานนี้ เริ่มจากการทำงานในลำดับแรกคือส่วนของการรับค่า



รูปที่ 4.2 แสดงการทำงานของส่วนการรับค่า

การสร้างส่วนรับค่าและแสดงผลนั้นถูกออกแบบโดยให้คลาส Interface เป็นคลาสที่ใช้ในการจัดการการทำงานนี้ ซึ่งจะคำนึงถึงข้อมูลที่รับเข้า และข้อมูลที่ต้องการจะแสดงผล ข้อมูลที่รับเข้ามานั้นจะเป็นข้อความภาษาไทย และแสดงผลลัพธ์เป็นภาพเคลื่อนไหว ซึ่งจะทำงานเมื่อมีตัวกระตุ้น เพราะฉะนั้นส่วนที่ติดต่อกับผู้ใช้จึงสามารถแบ่งออกได้เป็นส่วนการใช้งานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.1 ส่วนการใช้งานหลัก เป็นส่วนที่ถูกการออกแบบมาสำหรับติดต่อกับฟังก์ชันการแปลภาษา ซึ่งต้องมีส่วนในการรับข้อความ และส่วนแสดงภาพ ซึ่งจะทำงานเมื่อมีการกระตุ้นจากการกดปุ่ม “แสดงภาพ” โดย

- เมื่อกดปุ่มแล้วผลลัพธ์นั้นจะถูกนำมาแสดงที่ส่วนแสดงภาพ โดยใช้ ActionListener รับเหตุการณ์การกดปุ่ม เพื่อเรียกการทำงานส่วนการรับค่า
- การรับค่าเข้าสู่โปรแกรมผ่านทางคำสั่ง `tf.getText()` แล้วส่งต่อค่าที่รับเข้ามาจากหน้าจอไปยังคลาส Flex ซึ่งเป็นคลาสที่ใช้ทำการแปลความต่อไป (จะกล่าวถึงรายละเอียดในหัวข้อถัดไป) โดยเรียกการทำงานผ่านตัวแปรที่สร้าง กรณีตัวอย่าง (instance) เอาไว้ แล้วเรียกการทำงานของคลาส Flex ผ่านทาง method `jflex()` ซึ่งจะได้เป็นคำสั่งดังนี้

```
Flex flex = new Flex();
```

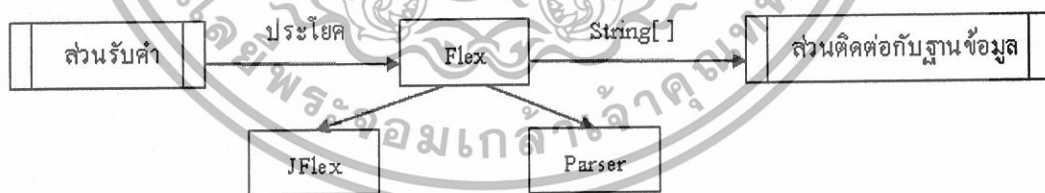
```
flex.jFlex(tf.getText());
```

- การใช้งานโปรแกรมนั้นข้อความที่ถูกพิมพ์เข้าสู่ระบบนั้นจะถูกเก็บลงตัวแปรชนิดสายอักขระ/สตริง (String) เมื่อผู้ใช้ต้องการบันทึกข้อมูลโดยมีการกดปุ่ม “บันทึก” ข้อความเหล่านั้นก็จะถูกเก็บลงไฟล์

4.1.1.2 ส่วนการใช้งานเพิ่มเติม เป็นส่วนอธิบายการทำงานต่างๆของโปรแกรม เมื่อกดปุ่มต่างๆแล้วก็จะเรียกหน้าจอที่ใช้แสดงภาพนั้นขึ้นมาแสดงที่ส่วนการแสดงผล

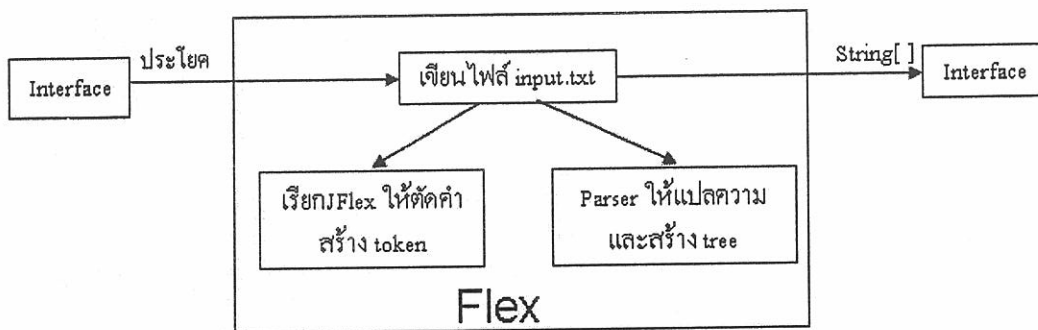
#### 4.1.2 ส่วนวิเคราะห์และแปลความประโยค

ส่วนการทำงานถัดไปเมื่อได้รับข้อความเข้ามาในระบบแล้ว คือส่วนของการแปลความ ซึ่งจะแยกออกมาได้เป็นการทำงานของคลาส Flex คลาส JFlex และคลาส Parser



รูปที่ 4.3 แสดงการทำงานของส่วนแปลความการรับค่า

คลาส Flex เป็นคลาสที่ใช้ในการแปลความ ซึ่งการทำงานในคลาสนี้ส่วนใหญ่เป็นการเรียกโปรแกรมอื่นมาใช้งาน โดยมีข้อมูลเข้าเป็นข้อความที่ได้ผู้ใช้พิมพ์เข้ามาในระบบ แล้วนำมาทำงานตามลำดับดังนี้



รูปที่ 4.4 แสดงการทำงานของส่วนการรับค่าคลาส Flex

- นำข้อความเข้านั้นมาเขียนลงไฟล์ชื่อว่า input.txt ก่อนโดยใช้ FileWriter เข้ามาช่วย
- เรียกคลาส JFlex แล้วใส่ข้อมูลเข้าเป็นชื่อไฟล์ที่ต้องการให้ทำงาน ซึ่งในกรณีนี้ใส่เป็น "input.txt" โดย JFlex จะอ่านค่าที่ได้จากในไฟล์มาเก็บไว้ในโทเกิน แล้วส่งข้อมูลผลลัพธ์กลับคืนออกมาให้ ซึ่งค่าที่ได้นั้นจะถูกจัดแบ่งให้อยู่ในโทเกินต่างๆ (การอธิบายการใช้งานโทเกินนั้นจะกล่าวในหัวข้อ 4.2)

ตัวอย่าง การตัดคำลงในโทเกิน ประโยค "หมากินแมว" คือ

โทเกิน นาม ประกอบด้วย หมา แมว

โทเกิน กริยา ประกอบด้วย กิน

- นำโทเกินที่ได้ข้างต้นมาใช้ในการวิเคราะห์รูปแบบไวยากรณ์ของประโยคนั้น แล้วแปลความ โดยเรียกการทำงานของคลาส Parser แล้วสร้างตัวแปรมารับค่าของผลลัพธ์ที่ได้ ซึ่งค่าที่ได้รับออกมาอยู่ในรูปของต้นไม้ (Tree) ถูกจัดอยู่ในรูปของอาร์เรย์ของสตริง (Array of String) รูปแบบของอาร์เรย์ของสตริง (Array of String) ผลลัพธ์คือ

```
String[] result = [ sub , adjSub , locaSub , adjLocS , pre ,
                    verb , adv ,
                    obj , adjObj , locaObj , adjLocO , pre , conv ]
```

เช่น ประโยค เจ้าวัวอ้วนกินหญ้าอยู่ที่ต้นไม้ใหญ่

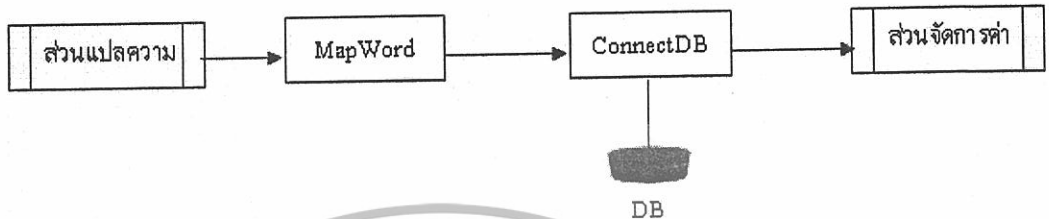
```
String[] result = [ วัว , อ้วน , " , " , " ,
                    กิน , " ,
                    หญ้า , " , ต้นไม้ , ใหญ่ , ได้ , " ]
```

โดยข้อมูลผลลัพธ์ที่ได้นั้นจะเป็นภาษาไทย (รายละเอียดการกำหนดรูปแบบไวยากรณ์ของประโยคจะกล่าวในหัวข้อ 4.2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

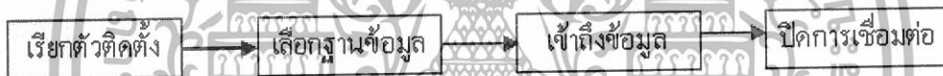
### 4.1.3 ส่วนติดต่อกับฐานข้อมูล

เนื่องจากข้อมูลผลลัพธ์ที่ได้ออกมาจากส่วนแปลความนั้นเป็นข้อมูลภาษาไทยและคำศัพท์หลายคำที่มีความหมายซ้ำกัน จึงต้องมีการจัดการการเก็บข้อมูลของคำศัพท์ทั้งหมด และรวบรวมคำศัพท์ที่มีความหมายเหมือนกันไว้ด้วยกัน ทำให้นำข้อมูลที่ได้ไปเรียกใช้งานในส่วนของการแสดงภาพได้อย่างเหมาะสม ซึ่งการทำงานของคลาสในส่วนนี้แบ่งเป็น 2 คลาสดังนี้



รูปที่ 4.5 แสดงการทำงานของส่วนติดต่อกับฐานข้อมูล

4.1.3.1 คลาส ConnectDB เป็นคลาสที่ใช้ในการติดต่อดึงข้อมูล และเป็นคลาสที่ใช้ติดต่อกับฐานข้อมูล เพื่อดึงค่าของคำที่ต้องการออกมา โดยแบ่งขั้นตอนการติดต่อเป็นขั้นตอนดังรูป 4.6 นี้



รูปที่ 4.6 รูปแสดงการทำงานของ คลาส ConnectDB

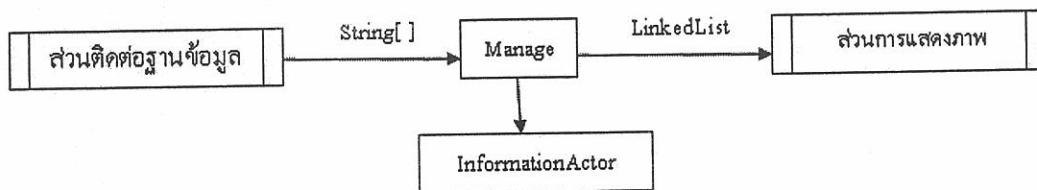
โดยฐานข้อมูลนั้นจะเก็บข้อมูลของคำศัพท์ภาษาไทย และภาษาอังกฤษไว้เช่น

ตารางที่ 4.1 ฐานข้อมูลเก็บข้อมูลของคำศัพท์ภาษาไทย และภาษาอังกฤษ

คำภาษาไทย(Thai word)	คำภาษาอังกฤษ(English word)
หมา	dog
แมว	cat
กิน	eat

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 4.7 แสดงการทำงานของส่วนจัดการค่า

1) คลาส Manage เป็นคลาสที่ทำงานในส่วนของการจัดการกับค่าต่างๆ ที่ได้จากการแปลความแล้ว ซึ่งส่วนใหญ่เป็นการจัดการค่าต่างๆ ของตัวละคร และการเพิ่มค่าในตัวแปรชนิดลิสต์ที่เก็บตัวละครที่อยู่ในคลาส InformationActor มีลำดับการทำงานคือ

1.1) โปรแกรมรับค่าเข้ามาผ่านเมธอด manage(vector v) ซึ่งเมธอดนี้จะทำการแตกค่าภายในเวกเตอร์นั้นออกมาเป็นค่าของตัวแปร และเก็บค่าลงตัวแปรต่างๆ จากการกำหนดค่า หรือจากการเรียกเมธอดต่างๆ

การเก็บค่าลงตัวแปร ได้แก่ ตัวแปร sStatus เป็นตัวแปรชนิดอาร์เรย์ของสตริงจะเก็บค่าของตัวละครที่จะใช้ในการแสดง ได้แก่ [subject, LocaSubject, object, LacaObject] ได้ค่าจากการเรียกเมธอด checkStatus()

เช่น ประโยค เจ้าวัวอ้วนกินหญ้าอยู่ที่ต้นไม้ใหญ่

sStatus[4] = [cow, "", grass, tree]

1.2) เป็นการจัดการเก็บตัวละครลงในฐานข้อมูล โดยตรวจสอบค่าว่ามีตัวละครอยู่ในข้อมูลตัวแปรชนิดลิสต์ของคลาส InformationActor หรือไม่

- มี ออกจากลูป

- ไม่มี จะเรียกเมธอด randomStart( ) โดยระบบจะคำนวณหาตำแหน่งของตัวละครใหม่ โดยมีการรับค่าของตัวแปรชนิดลิสต์ทั้งหมดเข้าเพื่อคำนวณหาตำแหน่งเฉลี่ยทั้งหมดของตัวละคร แล้วหักกลับด้วยขนาดของจอแสดงผล

1.3) ตรวจสอบค่าในเวกเตอร์ vmap ว่ามีค่าของตำแหน่งอ้างอิงตัวละครหรือไม่

- มี ต้องตรวจสอบว่าเป็นตำแหน่งอ้างอิงของตัวละครประเภทใด โดยจะตรวจสอบค่าจากตำแหน่งของค่าในเวกเตอร์ vmap

- ไม่มี ออกจากลูปการทำงาน

2) ส่วนการจัดส่งข้อมูลของตัวละคร คือส่วนที่คำนวณค่าตัวแปรอื่นๆ ที่เหลือและส่งผลลัพธ์ออกไปยังคลาส Animation มีลำดับการทำงานคือ

2.1) ส่วนของการกำหนดค่าตัวแปรอื่นๆ ได้แก่ส่วนขยายของประธาน กริยาและส่วนขยายกริยา โดยตรวจสอบจากค่าเวกเตอร์ vmap ที่เป็นค่าเริ่มต้นที่รับเข้ามาในเมธอด manage()

2.2) การค้นหาข้อมูลของตัวละครที่ใช้แสดงเป็นภาพพื้นหลัง โดยดึงค่าเวกเตอร์ข้อมูลของตัวละครทั้งหมดจากคลาส InformationActor แล้วดึงค่าของตัวละครที่เป็นประธาน หรือกรรมออกมา แล้วเก็บลงตัวแปรข้อมูลชนิดลิสต์ที่ชื่อว่า lBG จากการเรียกเมธอด getLinklistBG()

2.3) ดึงค่าของเวกเตอร์ของตัวละครจากคลาส InformationActor โดยชื่อของตัวละครนั้นจะได้มาจากตัวแปร sStatus ที่สร้างขึ้นในตอนแรกนั้น โดยใช้เมธอด getObject() แล้วนำค่าต่างๆ ที่ได้นั้นมาจัดให้อยู่ในรูปแบบของเวกเตอร์ vManage ที่ต้องการ แล้วส่งไปออกไปยังคลาส Animation

รูปแบบของตัวแปรที่จะส่งไปให้คลาส Animation คือ

```
vManage = [ [subject , invert , posx , posy , width , high ],
             verb , adverb ,
             [object , invert , posx , posy , width , high ],
             [location , invert , posx , posy , width , high ];
             llBackground , conv ]
```

ตัวอย่างประโยค เจ้าวัวอ้วนกินหญ้าอยู่ที่ต้นไม้ใหญ่

```
vManage = [ [cow , false , 100 , 200 , 1 , 0 ],
             eat , “” ,
             [grass , false , 150 , 200 , 0 , 0 ],
             [tree , false , 150 , 200 , 1 , 0 ],
             [ ] , “” ]
```

3) คลาส InformationActor เป็นคลาสที่เก็บข้อมูลต่างๆ ของตัวละคร โดยค่าที่เก็บนั้นจะเป็นข้อมูลลิสต์ ซึ่งรวมกลุ่มของเวกเตอร์ของตัวละครแต่ละตัวเอาไว้ ซึ่งลักษณะของเวกเตอร์คือ

```
[name , invert , posx , posy , width , high ]
```

การทำงานของคลาสนี้จะถูกเรียกใช้โดยคลาส Manage ซึ่งจะเรียกใช้งานเมื่อต้องการดึงค่าตัวละคร แก้ไขเปลี่ยนแปลงข้อมูล เพิ่มค่าตัวละคร โดยข้อมูลจะมีการเพิ่มข้อมูลเข้าไปเรื่อยๆ จากทางด้านหลัง และการเข้าถึงข้อมูลจะต้องเข้าถึงแบบเรียงต่อกัน ไปเป็นลำดับ

$$l = [ \text{cow} , \text{false} , 100 , 200 , 1 , 0 ],$$

$$[ \text{grass} , \text{false} , 150 , 200 , 0 , 0 ],$$

$$[ \text{tree} , \text{flase} , 150 , 200 , 1 , 0 ]]$$

เป็นส่วนที่เมื่อได้รับค่าต่างๆ ของตัวละครแล้วจัดการกับค่าเหล่านั้น เพื่อให้สามารถแสดงออกมาเป็นภาพเคลื่อนไหว โดยให้การแสดงผลนั้นทำโดยคลาส Animation



รูปที่ 4.8 แสดงการทำงานของส่วนการแสดงผล

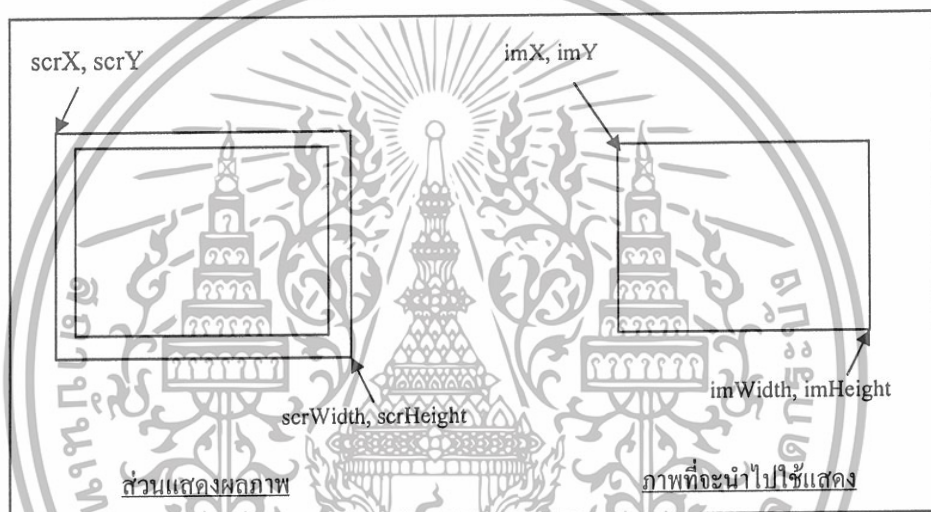
4.1.4.2 ส่วนของการแสดงผล เป็นส่วนที่รับค่าต่างๆ จากส่วนการจัดการค่าแล้วนำค่าเหล่านั้นมาตรวจสอบเพื่อนำไปแสดงผล

คลาส Animation เป็นส่วนการทำภาพเคลื่อนไหว เนื่องจากเลือกใช้หลักการของการมองในรูปแบบภาพรวม ดังนั้นทำให้ไม่ว่าจะเป็นกริยาไหนก็ตาม สามารถที่จะนำมาแสดงได้เลย โดยที่ไม่เกิดปัญหาต่อระบบตามมา และยังสามารถเพิ่มกริยาต่างๆ ที่ผู้ใช้ต้องการเข้าไปโดยไม่มีปัญหาตามมา เนื่องจากโปรแกรมได้สร้างมาเพื่อสนับสนุนการเปลี่ยนแปลงในอนาคตภายในส่วนภาพเคลื่อนไหวประกอบด้วยส่วนต่างๆ ดังนี้

- 1) ส่วนที่ทำการประกาศตัวแปร ประกาศตัวแปรที่ใช้ทั้งหมด รวมทั้งรับค่าตัวแปรจากคลาส Manage เมธอดที่ทำการรับค่าคือเมธอด Action เมธอดนี้อยู่ในคลาสที่ชื่อว่า Animation ซึ่งจะนำค่าทั้งหมดที่รับมาประกาศในส่วนตัวแปรส่วนกลาง (Global) ให้หมดทุกตัว เพื่อที่จะได้ให้เมธอดอื่นมาเรียกใช้ต่อไปได้
- 2) ส่วนการตรวจสอบทิศทางภาพว่าสอดคล้องกับทิศทางที่เคลื่อนที่หรือไม่ การพิจารณาทิศทางจะดูที่ค่าจุดเริ่มต้นและค่าจุดสิ้นสุดของพิกัด x ว่าเท่ากันหรือไม่ ในกรณีที่มีการเคลื่อนที่ในลักษณะแนวนอน และพิกัด Y จะดูเมื่อเคลื่อนที่ในแนวตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การนำทิศทางเคลื่อนที่ของตัวละครว่าไปด้านไหนโดยดูค่าพิกัด  $x$  ว่าลดลงหรือเพิ่มขึ้น ถ้าลดลงคือไปด้านซ้าย ถ้าเพิ่มขึ้นคือไปด้านขวา
- เมื่อทราบการเคลื่อนที่ก็ให้ดูว่าทิศทางของภาพตรงกับเคลื่อนที่หรือไม่ โดยรูปภาพที่สร้างจะหันหน้าไปด้านซ้ายทุกรูป ดังนั้นถ้าค่าพิกัด  $x$  มีค่ามากขึ้นหรือตัวละครเคลื่อนที่ไปทางขวาก็ให้กลับภาพจากซ้ายไปขวา ซึ่งจะมีเมธอด `invert()` คอยจัดการภาพที่แสดงให้กลับด้านจากซ้ายเป็นขวา โดยใช้หลักการของการเทียบค่าแกนของแนวนอน ให้พิกัด  $x$  ที่จุดเริ่มต้นของภาพ มาแสดงบนหน้าจอแสดงผลที่ตำแหน่งสุดท้ายในแนวแกน  $x$  ของภาพ ซึ่งทำให้การแสดงผลออกมาบนหน้าจอจะสลับจากด้านซ้ายเป็นขวา เพียงแค่เปลี่ยนการเทียบค่าจากภาพจริงกับภาพที่นำแสดงบนหน้าจอแสดงผล



รูปที่ 4.9 การจำลองรูปแบบการแสดงผลภาพลงบนหน้าจอ

รูปแบบคำสั่ง

```
g.drawImage(Image,scrX,scrY,scrWidth,scrHeight,imX,imY,imWidth, imHeight);
```

Image                      คือภาพที่จะนำมาแสดง

scrX, scrY                คือจุดพิกัดเริ่มต้นของภาพที่ส่วนแสดงผลบนหน้าจอ

scrWidth, scrHeight    คือจุดพิกัดสุดท้ายของส่วนแสดงผลบนหน้าจอ

imX, imY                 คือจุดพิกัดเริ่มต้นของรูปภาพ

imWidth, imHeight      คือจุดพิกัดสุดท้ายของรูปภาพ

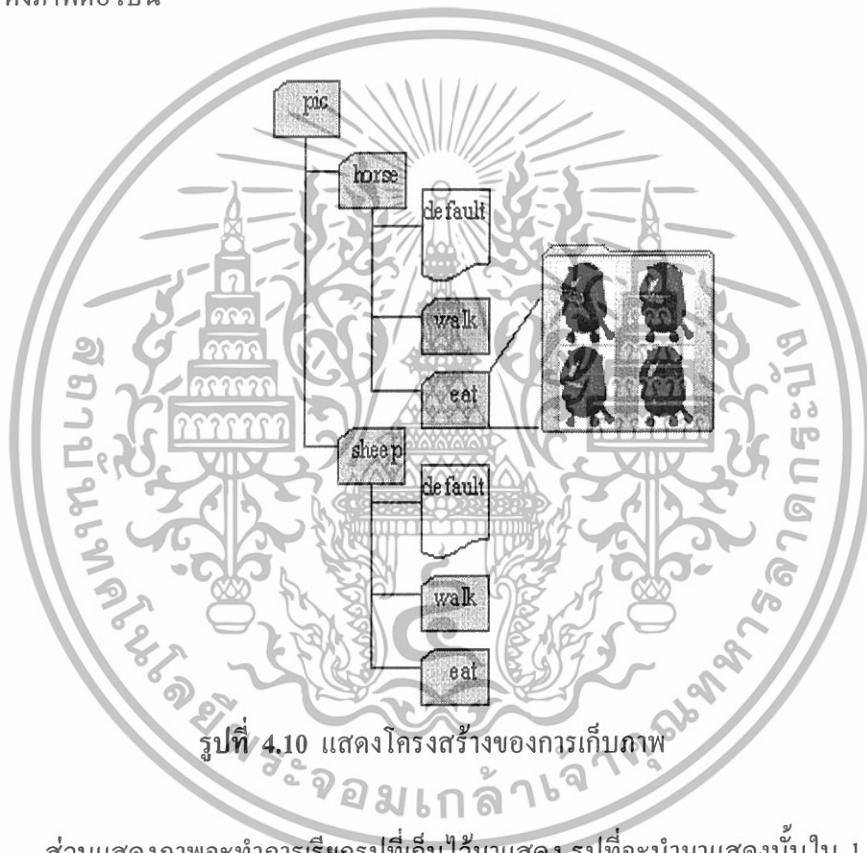
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างคำสั่งที่ทำให้ภาพกลับจากด้านซ้ายเป็นขวา

```
g.drawImage(Image,scrWidth,scrY,scrX,scrHeight,imX,imY,imWidth, imHeight);
```

จะเห็นว่าพิกัดเริ่มต้นของแนวแกน x มีการสลับค่ากันในส่วนแสดงภาพบนหน้าจอ ระหว่างจุดพิกัดเริ่มต้นของแกน x กับจุดพิกัดสุดท้ายของแกน y

ส่วนแสดงภาพเคลื่อนไหวนั้น มีการวางโครงสร้างโดยให้แต่ละภาพที่จะนำมาแสดงมีรูปแบบการแสดงผลเหมือนกัน โดยจะเริ่มที่แฟ้ม (Folder) ชื่อ pic จะบรรจุแฟ้มของตัวแสดงทุกตัวที่ได้สร้างไว้ โดยแฟ้มตัวแสดงจะถูกตั้งชื่อตามชนิดตัวละคร ในแฟ้มตัวแสดงจะบรรจุรูปพื้นฐาน (Default) และแฟ้มที่ตั้งชื่อตามกริยา ซึ่งภายในแฟ้มกริยาจะมีรูปตัวแสดงในรูปแบบที่ต่างกันไปอยู่ 4 ภาพ แต่ละตัวละครจะมีโครงสร้างแฟ้มแบบนี้เหมือนกันหมด ดังภาพต่อไปนี้



รูปที่ 4.10 แสดงโครงสร้างของการเก็บภาพ

ส่วนแสดงภาพจะทำการเรียกรูปที่เก็บไว้มาแสดง รูปที่จะนำมาแสดงนั้นใน 1 รูปแบบการแสดงผล (Pattern) มีการเรียกรูปเพื่อนแสดงทั้งหมด 4 รูป ซึ่งรูปแบบการจัดการการแสดงผลแบ่งออกเป็น 2 ลักษณะคือ ส่วนที่ไม่มีกรเคลื่อนตำแหน่งของรูปภาพ กับส่วนที่มีการเคลื่อนตำแหน่งรูปภาพ

1) ส่วนที่ไม่มีการเคลื่อนที่ มีการแสดงรูปโดยวนแสดงรูป  $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4$  แล้วก็ให้แสดงรูปแรกเป็นรูปจบ ซึ่งการวนเปลี่ยนแสดงรูปจะให้วนแค่ครั้งเดียว รูปนั้นไม่มีการเคลื่อนที่จึงไม่ต้องมีกลไกการเปลี่ยนตำแหน่งมาเกี่ยวข้อง

2) ส่วนที่มีการเคลื่อนตำแหน่งของรูปภาพในส่วนนี้การแสดงผลจะต้องวาดรูปตัวละครเข้าไปเรื่อยๆ เริ่มตั้งแต่จุดเริ่มต้นจนกว่าจะไปถึงยังจุดสิ้นสุด เมื่อถึงจุดสิ้นสุดก็จะทำการเรียกรูปพื้นฐานมาแสดง

ส่วนนี้จะตรวจสอบค่าพิกัดว่ามีค่าเท่ากันหรือไม่โดยจะแยกเป็น 2 ทางคือ

- if(subj\_Start <= subj\_End) ถ้าจุดเริ่มน้อยกว่าจุดสิ้นสุดจะให้เพิ่มค่าพิกัด x ของตัวประธานไปเรื่อยๆ จนกว่าจะเท่ากัน จึงออกจากลูปนี้ได้

- if(subj\_Start >= subj\_End) ถ้าจุดเริ่มมากกว่าจุดสิ้นสุดจะให้ลดค่าพิกัด x ของตัวประธานไปเรื่อยๆ จนกว่าจะเท่ากัน จึงออกจากลูปนี้ได้

การที่ภาพจะมีการแสดงแบบเคลื่อนไหวได้นั้นจำเป็นต้องใช้ ActionListener ในส่วนของ Timer เป็นการกำหนดค่าช่วงเวลา (Delay) ซึ่งจะทำให้รูปนำมาเปลี่ยนแปลงเกิดเป็นภาพเคลื่อนไหววนสลับกัน โดยมีค่าความเร็วตามที่กำหนดลงใน Timer

#### 4.2 การกำหนดรูปแบบไวยากรณ์ของประโยค

จากการทำงาน โปรแกรมนี้เป็นการแปลข้อความภาษาไทยให้กลายเป็นภาพเคลื่อนไหว ดังนั้นส่วนการทำงานที่สำคัญที่สุดก็คือ การแปลความหมายของประโยค จากหลักการที่ได้กล่าวมาแล้ว ใช้การทำงานที่จะแปลความประโยคได้ เริ่มต้นดังนี้

- การวิเคราะห์กลุ่มคำในประโยค (Lexical Analysis) เพื่อตัดโทเเกินในประโยค ซึ่งการกำหนดรูปแบบคำหรือโทเเกินนั้น กำหนดได้ด้วยนิพจน์สามัญ

- การวิเคราะห์ไวยากรณ์ของประโยค เป็นการนำโทเเกินที่ได้มาตรวจสอบว่ามีลำดับสอดคล้องกับไวยากรณ์ที่กำหนด

การสร้างรูปแบบประโยคตามที่ต้องการมีการกำหนดรูปแบบของประโยคตามหลักภาษาไทย โดยการกำหนดรูปแบบของประโยคมีขั้นตอนดังนี้

#### 4.2.1 การกำหนดคำและชนิดของคำ

การกำหนดคำและชนิดของคำเป็นการกำหนดคำในส่วนของไลบรารีเจเฟล็กซ์ (Library JFlex) โดยแบ่งการกำหนดคำดังนี้

4.2.1.1 กำหนดคำต่างๆ ตามลักษณะของคำ โดยมีการจัดหมวดหมู่ไว้เป็นกลุ่มต่างๆ

เช่น

```
import java_cup.runtime.*;
%%
%cup
%unicode
%standalone
noun = แมว|หมา|วัว|ช้าง|ม้า|แกะ|หนู|ต้น|ไม้|หญ้า|ป่า|แม่น้ำ|ต้น|ไม้|ลำธาร|บ้าน
verb = กระโดด|นอน|นั่ง|เดิน|กิน|คุย|พูด|ว่า|บอก|วิ่ง|ไป|มา|หา|รู้|เดิน|แอบ|ดู|มอง
vs = ดี|ใจ|เสีย|ใจ|เศร้า|ใจ|ตก|ใจ|โวยวาย|มีความสุข
adj = อ้วน|ผอม|สูง|เตี้ย|ค้ำ|ยาว|สั้น |ใหญ่
sent = "[^ก-ฮ][^๑-๗][^ไ-๐][๑-๘]"
```

4.2.1.2 กำหนดส่วนคืนค่าตามกลุ่มคำ โดยกำหนดให้คำอยู่ในรูปของตัวแปร เช่น

```
%%
{noun} {return new Symbol(sym.n, new String(yytext()));}
{verb} {return new Symbol(sym.v, new String(yytext()));}
{vs} {return new Symbol(sym.vs, new String(yytext()));}
{adj} {return new Symbol(sym.adj, new String(yytext()));}
{sent} {return new Symbol(sym.sent, new String(yytext()));}
```

ซึ่งเมื่อตรวจสอบค่าของคำแล้วตรงกับกลุ่มคำใด ก็จะสร้างโทเก้นของกลุ่มคำนั้นไว้ แล้วเก็บคำนั้นลงไป โดยเก็บใส่ตัวแปรที่กำหนดในช่วงของการคืนค่า เช่น เมื่อใส่ข้อความคำว่า “แมวกินหนู” นำตัวอักษรแต่ละตัวเข้าไปตรวจสอบค่าจากการประกาศคำได้ชนิดของคำดังนี้

noun = แมว , หนู

verb = กิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำค่าที่ได้นั้นมาคืนค่าออกไปโดยตรวจสอบจากกลุ่มค่าที่ได้ แล้วนำค่าเหล่านั้นมาใส่ใน  
โทเกินได้แก่

$n = [\text{แมว}][\text{หนู}]$

$v = [\text{กิน}]$

เมื่อสร้างเสร็จแล้วบันทึกไฟล์ชื่อ `jflexproject.jflex` แล้วสั่งคอมไพล์ให้ไฟล์เปลี่ยนเป็น  
ไฟล์นามสกุล `java` เพื่อให้สามารถนำไปใช้งานกับภาษาจาวาที่พัฒนาได้ แล้วจะได้ผลลัพธ์  
เป็นไฟล์ `Yylex.java` ด้วยคำสั่ง

`jflex jflexProject.jflex`

#### 4.2.2 การกำหนดรูปแบบของประโยค

เป็นการนำลักษณะของค่าที่ได้กำหนดไว้ขึ้นมาเรียงตามลำดับ เพื่อให้เกิดลักษณะรูปแบบของ  
ประโยคตามที่ต้องการเป็นกฎของประโยค ซึ่งจะสร้างให้เป็นไปตามหลักภาษาไทยตามที่ได้ศึกษามา  
ในขั้นต้น โดยการสร้างประโยคนั้นสามารถสร้างประโยคได้หลายแบบ โดยสร้างรูปแบบข้อมูล  
แบบต้นไม้ การกำหนดรูปแบบของประโยค (Rule) ได้แก่

```
import java_cup.runtime.*;
terminal n,v,vs,aj,sent;
non terminal Lista ;
non terminal String[] List;
non terminal String[] VP1;
non terminal String VP;
non terminal String[] NP;
start with Lista;
Lista ::= List ;
List ::= NP:npp VP1:vpp NP:npp2  {:RESULT = new String[3];
                                RESULT[0] = "+"npp;
                                RESULT[1] = ""+vpp[0];
                                RESULT[2] = ""+vpp[1];
                                RESULT[] = ""; :};

List ::= NP:npp VP1:vpp  {:RESULT = new String[3];
                           RESULT[0] = ""+npp;
                           RESULT[1] = ""+vpp[0];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        RESULT[2] = ""+vpp[1];
        RESULT[3] = ""; :};

NP ::= n:n1 aj:a1      {:RESULT = new String[2];
                        RESULT[0] = ""+n1;
                        RESULT[1] = ""+a1 :};

NP ::= n:n1            {:RESULT = new String[2];
                        RESULT[0] = ""+n1;
                        RESULT[1] = "" :};

VP1 ::= VP:vpp sent:s {:RESULT = new String[3];
                       RESULT[0] = ""+vpp;
                       RESULT[1] = "";
                       RESULT[2] = ""+s;
                       System.out.println("ssss"+s);}; VP sent;

VP1 ::= VP:vpp        {:RESULT = new String[3];
                       RESULT[0] = ""+vpp;
                       RESULT[1] = "";
                       RESULT[2] = "" :};VP;

VP ::= v:v1 v         {:RESULT = ""+v1; :};
VP ::= v:v1          {:RESULT = ""+v1; :};;
VP ::= vs:v1         {:RESULT = ""+v1; :};;

```

เป็นการกำหนดรูปแบบของประโยคอย่างง่ายของโปรแกรม ประโยคที่ได้ได้แก่  
รูปแบบที่ 1 List : NP VP1;

NP : n;

VP1: VP sent;

VP : v v;

เป็นรูปแบบประโยคที่ไม่ต้องการกรรม โดยพิจารณาชนิดของคำแต่ละส่วนคำ แล้วหากมีส่วน  
ใดที่สามารถขยายออกได้นั้น ก็จะต้องแตกผลลงมาเรื่อยๆ เช่น หมาพูดว่า “สวัสดีค่ะ”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบที่ 2 List : NP VP1 NP;

NP : n;

VP1: VP;

VP : v;

เป็นรูปแบบประโยคที่ต้องการกรรม โดยพิจารณาชนิดของคำแต่ละส่วนคำ แล้วหากมีส่วนขยาย ก็จะต้องแตกผลลงมาเรื่อยๆ เช่น แมวกินหนู

เมื่อสร้างไฟล์เสร็จให้บันทึกเป็นไฟล์ชื่อ parserproject.cup แล้วเมื่อคอมไพล์ไฟล์ด้วยคำสั่ง

```
java -jar java-11a-cup.cup parserproject.cup
```

เมื่อคอมไพล์แล้วจะได้ไฟล์ออกมาเป็น 2 ไฟล์ คือ parser.java sym.java ซึ่งเป็นไฟล์จาวาที่ใช้ในการแปลภาษาจากประโยคที่ใส่เข้าไป แล้วตรวจสอบว่าประโยคนั้นถูกต้องตามรูปแบบประโยคที่กำหนดไว้หรือไม่ หากเป็นไปตามรูปแบบแล้วก็จะมีการส่งค่ากลับ (Return) ซึ่งลำดับข้อมูลภายในที่ต้องการนั้นก็สามารรถกำหนดได้จากกรกำหนดค่าการคืนกลับในแต่ละลำดับชั้น

เช่น ประโยค หมาพูดว่า "สวัสดีค่ะ"

RESULT[0] = หมา

RESULT[1] = พูด

RESULT[2] = "ค่ะ"

RESULT[3] = "สวัสดีค่ะ"

เช่น ประโยค แมวกินหนู

RESULT[0] = แมว

RESULT[1] = กิน

RESULT[2] = หนู

RESULT[3] = ""

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### คู่มือและตัวอย่างการทำงานของโปรแกรม

โปรแกรมสร้างภาพเคลื่อนไหวจากข้อความภาษาไทยนั้น เป็นโปรแกรมที่รับข้อมูลประโยคภาษาไทยเข้ามาในระบบแล้วจะทำการเปลี่ยนประโยคนั้นให้กลายเป็นภาพเคลื่อนไหว สามารถแบ่งออกเป็น 2 ส่วนได้แก่

1. ส่วนใช้งานหลัก เป็นส่วนที่อธิบายการใช้งานด้านการสร้างภาพเคลื่อนไหวจากข้อความ
2. ส่วนการใช้งานเพิ่มเติม เป็นส่วนที่มาให้เข้าใจการใช้งานโปรแกรมเพิ่มขึ้น

ดังจะอธิบายดังนี้

#### 5.1 ส่วนใช้งานหลัก

เมื่อผู้ใช้เปิดโปรแกรมขึ้นมาแล้วจะมีหน้าจอปรากฏขึ้นตามรูปที่ 5.1 จากนั้นเมื่อผู้ใช้งานต้องการใช้งานโปรแกรมสามารถทำได้โดย

1) เลือกฉากหลังที่ผู้ใช้งานต้องการจากเมนูทางซ้ายมือของหน้าจอ โดยมีชื่อของภาพพื้นหลังขึ้นมาให้เลือก เช่นหากเลือกภาพพื้นหลังเป็น “ลำธาร” ฉากจะถูกเปลี่ยนภาพพื้นหลังเป็นฉากลำธาร ดังรูปที่ 5.2

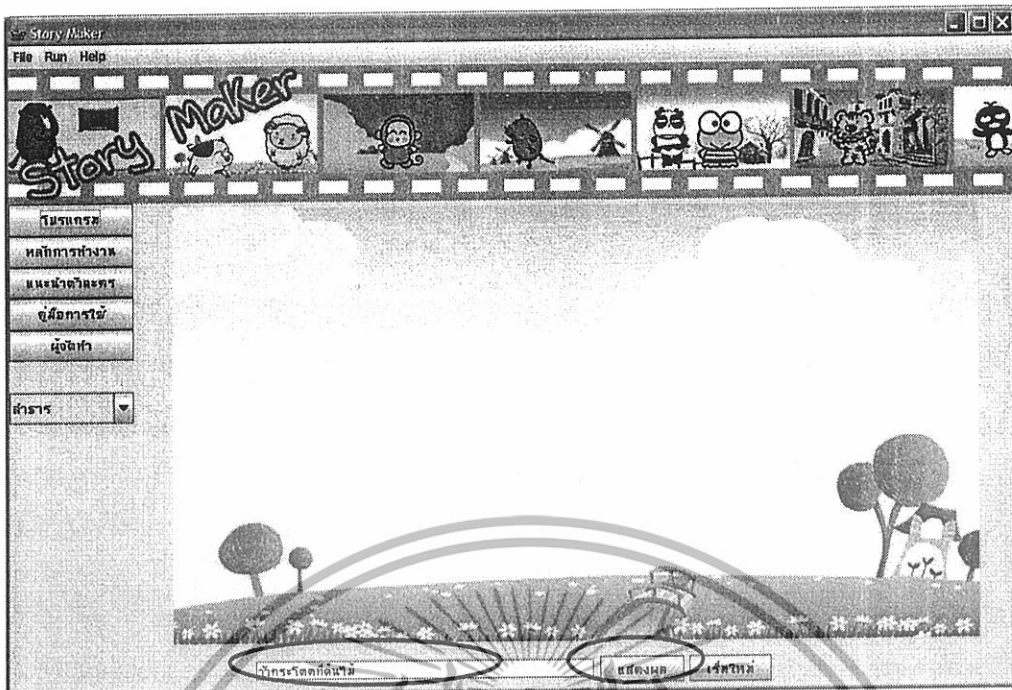
2) เมื่อต้องการเริ่มการสร้างภาพเคลื่อนไหว สามารถทำได้โดยใส่ข้อความประโยคภาษาไทยที่ต้องการลงไปในช่วงรับข้อความแล้วกดปุ่ม “แสดงผล” ที่อยู่ทางด้านล่างของหน้าจอ เช่นใส่ประโยคว่า “วิวกระโดดที่ต้นไม้” ดังรูปที่ 5.3

3) เมื่อทำการกดปุ่มแสดงผล ภาพของประโยคที่ใส่ลงไปก็จะถูกแสดงอยู่บนหน้าจอ เช่นจากประโยคที่ใส่เข้าไปทำให้ได้ภาพเคลื่อนไหวของวิวที่กำลังกระโดดอยู่ที่ต้นไม้ปรากฏขึ้นมาแสดงที่หน้าจอแสดงผล ดังรูปที่ 5.4

4) หากต้องการแสดงภาพต่อเนื่องก็สามารถใส่ข้อความใหม่ลงในช่องรับข้อความต่อไปได้ แล้วกดปุ่ม “แสดงผล” ทำให้ภาพเคลื่อนไหวที่ได้นั้นจะแสดงต่อเนื่องจากภาพเดิม เช่นใส่ประโยคใหม่ลงไปว่า “วิวเดินไปหาแกะ” แล้วกดแสดงผล ดังรูปที่ 5.5 และรูปที่ 5.6

5) หากต้องการเริ่มเรื่องใหม่ก็ให้ทำการกดปุ่ม “เริ่มใหม่” ตัวละครและค่าต่างๆ ก็จะถูกลบออกไปกลายเป็นหน้าภาพพื้นหลัง ดังรูปที่ 5.7



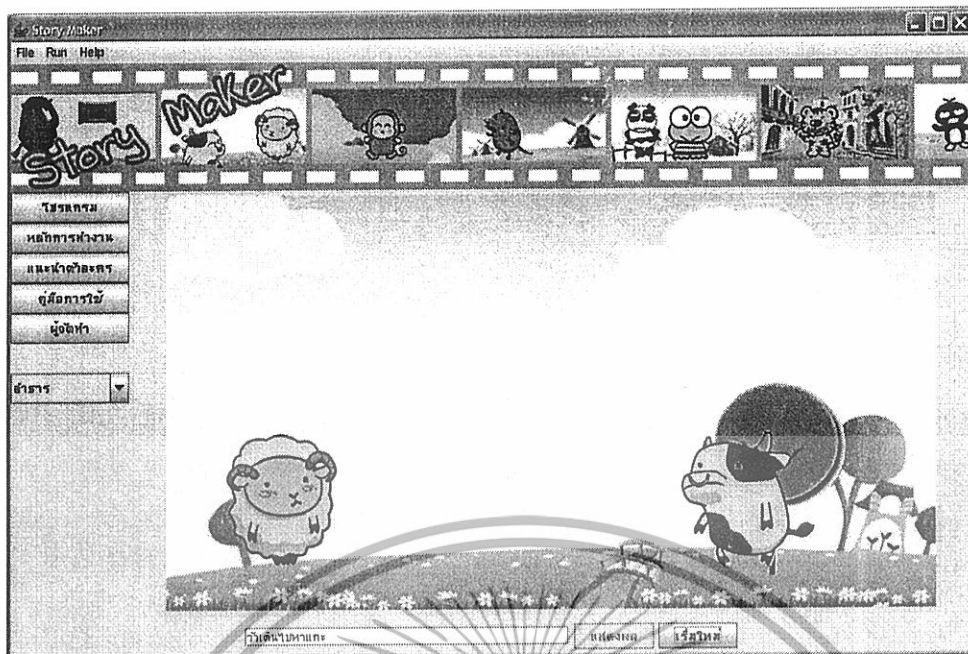


รูปที่ 5.3 แสดงตัวอย่างการใช้งานโปรแกรม

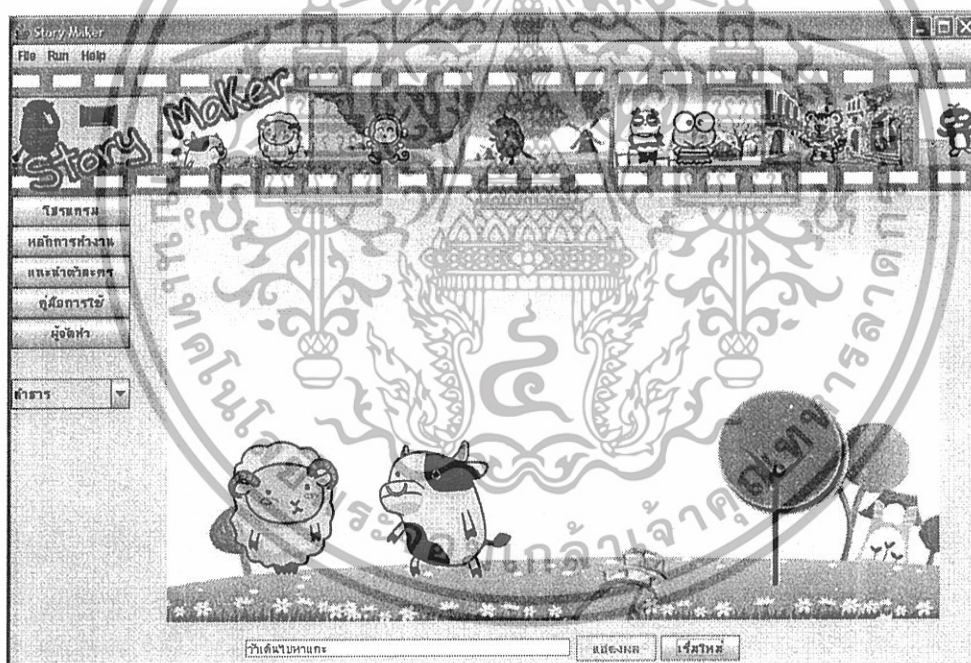


รูปที่ 5.4 แสดงผลลัพธ์ที่เป็นภาพเคลื่อนไหวของประโยค “วีธีการโคลนนิ่ง”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

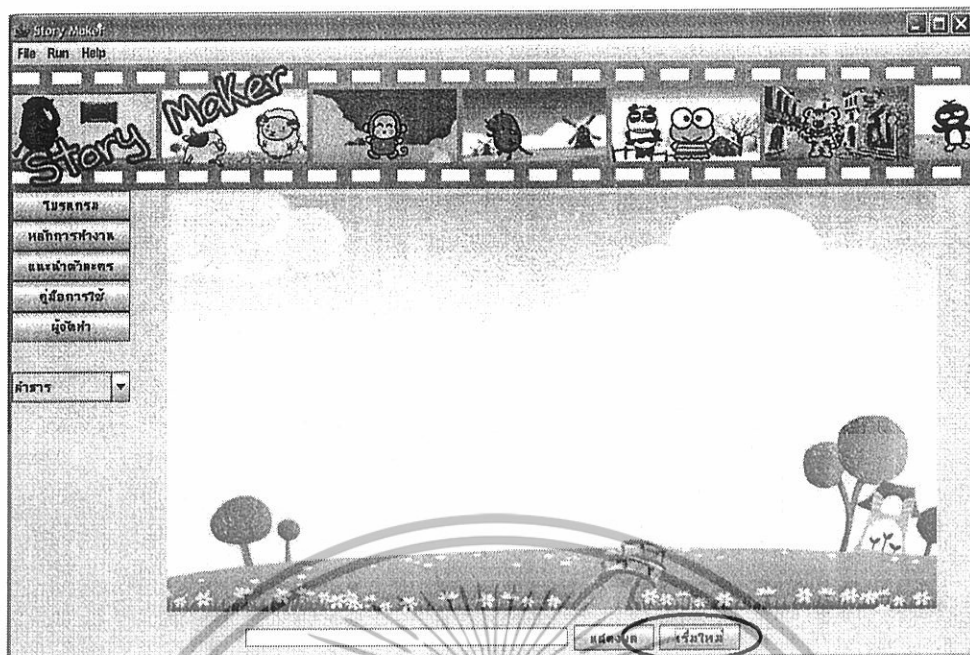


รูปที่ 5.5 แสดงผลลัพธ์ที่เป็นภาพเคลื่อนไหวของประโยค “วัวเดินไปหาแกะ”



รูปที่ 5.6 แสดงผลลัพธ์ที่เป็นภาพเคลื่อนไหวของประโยค “วัวเดินไปหาแกะ”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



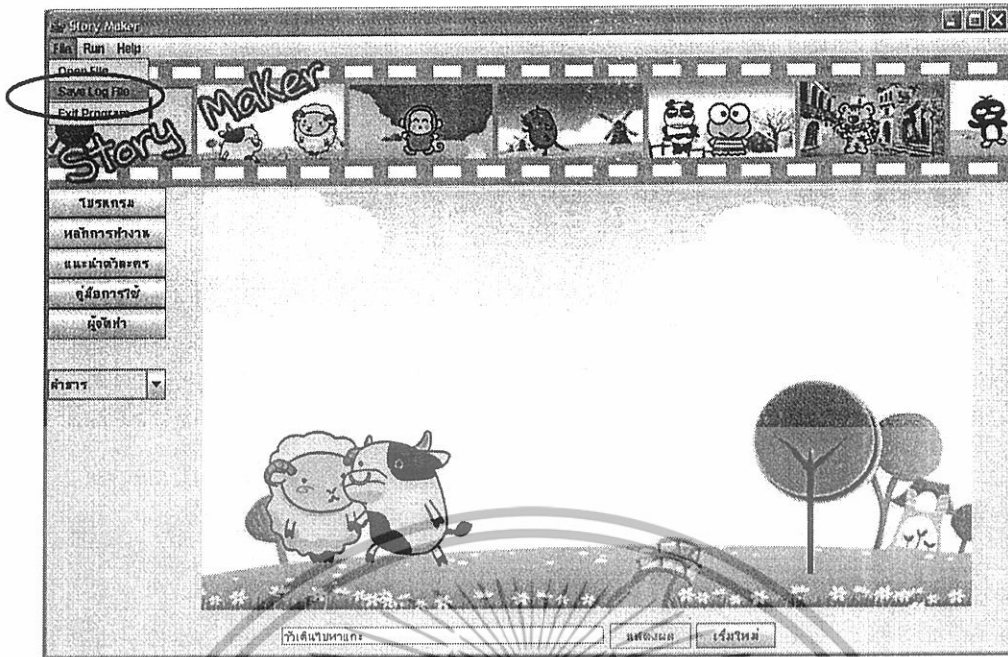
รูปที่ 5.7 แสดงผลลัพธ์เมื่อกดปุ่ม “เริ่มใหม่”

เมื่อใส่ข้อความต่างๆ เข้าไปแล้วข้อความเหล่านั้นจะถูกเก็บเอาไว้ และแสดงให้เห็นในหน้าจอเก็บข้อมูล ดังรูปที่ 5.8 โดยที่สามารถเลือกที่จะบันทึกข้อความทั้งหมดไว้ได้ โดยคลิกที่ปุ่ม “file” แล้วคลิก “save log file” ดังรูปที่ 5..9



รูปที่ 5.8 แสดงข้อมูลที่ผู้ใช้กรอกทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.9 แสดงการบันทึกข้อมูลทั้งหมด

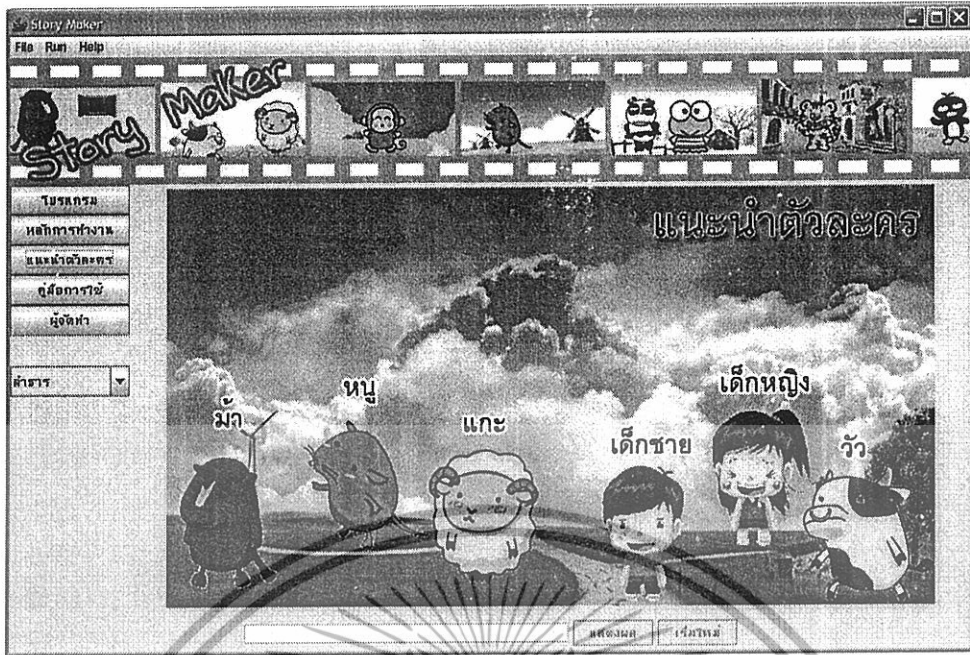
โปรแกรมสามารถเรียกไฟล์ข้อมูลที่บันทึกไว้ขึ้นมาแสดงใหม่ให้มีความต่อเนื่องได้โดยคลิกที่ปุ่ม “file” แล้วเลือก “open file” แล้วข้อความทั้งหมดของไฟล์ที่เปิดนั้นจะปรากฏขึ้นที่หน้าจอแสดงข้อความดังรูปที่ 5.10 แล้วหากต้องการให้แสดงภาพเคลื่อนไหวจากข้อความที่เปิดขึ้นมานั้นก็คลิกที่ปุ่ม “ส่งผ่านค่า” แล้วกดปุ่มแสดงผล ตามปกติภาพเคลื่อนไหวก็จะปรากฏขึ้น



รูปที่ 5.10 แสดงการเปิดไฟล์และข้อมูลที่ได้จากการเปิดไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

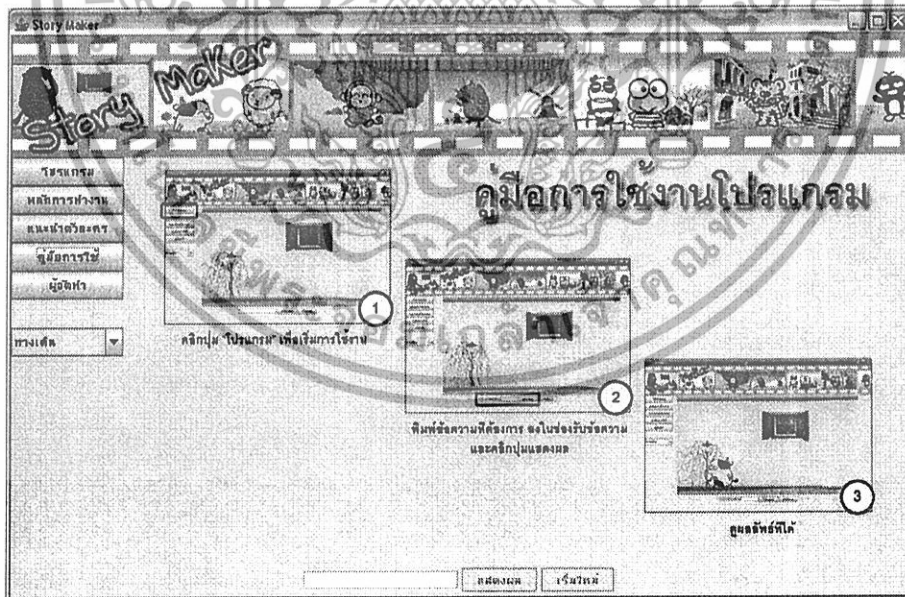




รูปที่ 5.12 แสดงตัวละครในโปรแกรม

### 5.2.3 ส่วนคู่มือการใช้งาน

เป็นส่วนช่วยเหลือผู้ใช้ให้สามารถใช้งานโปรแกรมได้อย่างถูกต้องโดยกลุ่ม “คู่มือการใช้งาน” ที่อยู่ทางซ้ายมือ ดังรูปที่ 5.13



รูปที่ 5.13 แสดงคู่มือการใช้งานโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.2.4 ส่วนแนะนำผู้จัดทำ

เป็นส่วนแนะนำผู้จัดทำและลักษณะของโปรแกรมนี้ ซึ่งจะมีรายละเอียดต่างๆ ของผู้จัดทำอยู่ โดยกลุ่ม “ผู้จัดทำ” ที่อยู่ทางซ้ายมือ ดังรูปที่ 5.14



รูปที่ 5.14 แนะนำผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### สรุปการทำงานของระบบ

โปรแกรมสร้างภาพเคลื่อนไหวจากข้อความภาษาไทย เป็นสื่อที่ใช้สร้างนิทานเพื่อความบันเทิง โดยใช้แนวคิดของการสร้างคอมไพเลอร์มาประยุกต์ใช้ โปรแกรมจะรับประโยคที่ผู้ใช้ใส่เข้ามา ผ่านกระบวนการการตัดคำ (Lexicon Analysis) การวิเคราะห์ประโยคเพื่อหาความหมาย (Syntax Analysis) เชื่อมความสัมพันธ์ระหว่างคำกับภาพผ่านฐานข้อมูล และทำการดึงข้อมูลของแต่ละคำ ผ่านตัวจัดการข้อมูลเพื่อวิเคราะห์คุณสมบัติของแต่ละคำ แล้วส่งคำนั้นไปตัวแสดงผลจนสุดท้ายได้ ภาพเคลื่อนไหวออกมา

รายละเอียดและแนวคิดของโครงการสามารถนำไปประยุกต์ใช้ในการสร้างสื่อแสดงภาพแบบอื่นได้ อาจจะปรับเปลี่ยนการวิเคราะห์ประโยคที่เป็นภาษาอื่นต่อไป หรือจะทำการพัฒนาให้สามารถรับประโยคที่มีความซับซ้อนและขยายขอบเขตการแสดงให้รองรับกับความต้องการของผู้ใช้ให้มากขึ้น

#### 6.1 ประโยชน์ของการสร้างโปรแกรมขึ้นมาใหม่

##### 1) ด้านผู้พัฒนาโปรแกรม

1.1) สามารถนำความรู้ด้านการเขียนโปรแกรม (Programming Language) มาประยุกต์สร้างโปรแกรมขึ้นมาใหม่ ทำให้ผู้พัฒนาทราบถึงรูปแบบการเขียนโปรแกรมที่เหมาะสมและมีประสิทธิภาพมากขึ้น รวมทั้งทราบถึงจุดอ่อนในการเขียนโปรแกรม และแนวทางการแก้ไขข้อบกพร่อง

1.2) เรียนรู้ระบบการทำงานของคอมไพเลอร์นำมาประยุกต์ใช้ในการสร้างตัวจัดการประโยคภาษาไทยที่เข้ามาในโปรแกรมให้เป็นภาพเคลื่อนไหว

##### 2) ด้านผู้ใช้โปรแกรม

2.1) เพิ่มทางเลือกใหม่ในการสร้างนิทานที่มีภาพประกอบ ซึ่งโปรแกรมจะช่วยให้การสร้างนิทานที่มีภาพประกอบง่ายขึ้นและสามารถทำได้ด้วยตนเอง เนื่องจากวิธีการใช้งานไม่ซับซ้อน

2.2) สามารถประยุกต์ใช้เป็นการสอนเพื่อสร้างความบันเทิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.2 ข้อดีของระบบ

- 1) ระบบสามารถแปลงประโยคภาษาไทยอย่างง่ายให้เป็นภาพเคลื่อนไหวได้
- 2) ระบบสร้างภาพเคลื่อนไหวทำให้สามารถสื่อสารกับบุคคลอื่นได้ง่ายขึ้น เช่นเด็กเล็ก หรือคนที่อ่านหนังสือไม่ออก
- 3) ระบบสามารถนำไปประยุกต์เพื่อใช้ในการพัฒนาเป็นสื่อส่งเสริมในการเล่านิทานได้

## 6.3 ข้อจำกัดของระบบ

- 1) ระบบไม่สามารถจัดการกับประโยคที่มีความซับซ้อนได้
- 2) ระบบไม่สามารถทำงานกับประโยคภาษาอื่นนอกจากภาษาไทยได้ เช่น ภาษาอังกฤษ ไม่สามารถทำงานกับตัวเลขได้
- 3) ระบบยังแสดงภาพเคลื่อนไหวได้ไม่สมจริง เป็นภาพเคลื่อนไหวอย่างง่ายเท่านั้น
- 4) คำศัพท์ที่ใช้ยังเป็นคำที่ง่าย ไม่หลากหลาย
- 5) มีตัวละครที่จำกัด รวมทั้งฉากและสถานที่
- 6) ไม่สามารถรับข้อความที่เป็นชื่อเฉพาะได้

## 6.4 ข้อเสนอแนะ

โปรแกรมที่ได้ทำสร้างนับได้ว่าเสร็จสมบูรณ์ในระดับที่น่าพอใจ มีความสามารถที่รองรับการทำงานพื้นฐานได้ และสามารถนำมาใช้งานได้จริง แต่ยังขอบเขตจำกัดการทำงานอยู่พอสมควร ซึ่งสามารถพัฒนาต่อเนืองไปได้ เพื่อให้โปรแกรมมีประสิทธิภาพมากขึ้น ได้แก่

- 1) สามารถรองรับข้อมูล และตัวละครที่จะเพิ่มขึ้นมาจากเดิมได้
- 2) สามารถเพิ่มความสามารถในการจัดการประโยคที่รับเข้ามา ให้สามารถรับประโยคที่มีความซ้ำซ้อนขึ้น โดยใช้การสร้างเงื่อนไขการพิจารณาประโยคเพิ่มลงไปโปรแกรม
- 3) สามารถปรับเปลี่ยนรูปตัวละครให้สวยงาม โดยใช้ภาพใหม่ลงทับกับภาพเดิม และเพิ่มความสามารถพิเศษ เช่น ภาพเอฟเฟกพิเศษ เสียง ให้กับตัวละคร
- 4) เพิ่มส่วนตั้งชื่อให้กับตัวละคร เพื่อสร้างความแตกต่างให้ตัวละครที่มีหน้าตาเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- กองบริการการศึกษามหาวิทยาลัยราชภัฏยะลา. 2549. ภาษาธรรมชาติ(Natural Language).  
[Online].Available :  
[http://yalor.yru.ac.th/~nipon/Archi\\_STD43/chapter9/group\\_20/machine.html](http://yalor.yru.ac.th/~nipon/Archi_STD43/chapter9/group_20/machine.html).
- กิตติ ภัคดีวัฒนกุล. 2545. Java โปรแกรมเมอร์. กรุงเทพฯ : บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด.
- ธีรพงศ์ โหมดหิรัญ. 2548. “แก้ไขปัญหาความกำวมของคำในภาษาไทยโดยใช้ซอฟต์แวร์แมชชีน”. วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- วีระศักดิ์ ชิงถาวร. 2543. Java Programming Volume I. กรุงเทพฯ : บริษัท ซีเอ็ดดูเคชั่น จำกัด.
- วีระศักดิ์ ชิงถาวร. 2545. Java Programming Volume II. กรุงเทพฯ : บริษัท ซีเอ็ดดูเคชั่น จำกัด.
- สมจิตร อัจฉินทร์ และงามนิญ อัจฉินทร์. 2543. หนังสือหลักการวิเคราะห์และออกแบบระบบฐานข้อมูล. กรุงเทพฯ : ห้างหุ้นส่วนจำกัด ขอนแก่นการพิมพ์.
- สลยุทธ์ สว่างวรรณ. 2549. หนังสือคอมไพเลอร์. กรุงเทพฯ : บริษัท วิรัตน์ เอ็ดดูเคชั่น จำกัด.
- Henk van Leeuwen. 2006. การสร้าง Library JFlex และ CUP. [Online].Available :  
<http://www.hio.hen.nl/~vanleeuw/pse/spanje/tutorial.html>,  
<http://www.jflex.de/manual.html>.
- Sun Microsystems, Inc. 2004. เนื้อหาและวิธีการใช้Timer. [Online].Available :  
<http://java.sun.com/j2se/1.5.0/docs/api/java/util/Timer.html>.
- Sun Microsystems, Inc. 1991. Java Programming. [Online].Available : [Java.sun.com](http://java.sun.com).
- Wikimedia Foundation, Inc. 2006. การวิเคราะห์และสร้างUML. [Online].Available :  
[http://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language), <http://www.rational.com>,  
<http://www.thaiall.com>.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก. การติดตั้งโปรแกรม

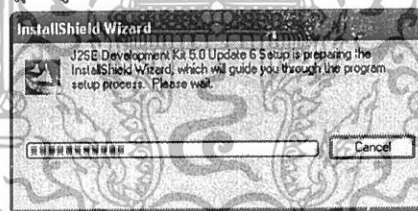
เนื่องจากโปรแกรมนั้นพัฒนามาจากภาษาจาวา ดังนั้นหากต้องการจะใช้งานโปรแกรมจะต้องมีตัวประมวลผลภาษาจาวาก่อน ซึ่งในกรณีนี้จะใช้เครื่องมือ (Tools) Netbeans ช่วยในการทำงานของโปรแกรม โดยที่ผู้ใช้ต้องทำการติดตั้ง Netbeans ก่อนจึงจะสามารถใช้งานโปรแกรมได้ แต่ผู้ใช้อาจจะใช้เครื่องมืออื่นที่สามารถทำงานกับภาษาจาวามาใช้งานแทนได้ และยังต้องมีการเชื่อมต่อกับฐานข้อมูลอีก เพื่อเตรียมความพร้อมก่อนที่จะติดตั้งโปรแกรม ซึ่งจะแบ่งขั้นตอนการติดตั้งเป็น 3 ขั้นตอนคือ

- การติดตั้ง Netbeans
- การติดตั้งฐานข้อมูลชนิด MySQL
- การติดตั้งโปรแกรม

### 1. การติดตั้งโปรแกรม Netbeans

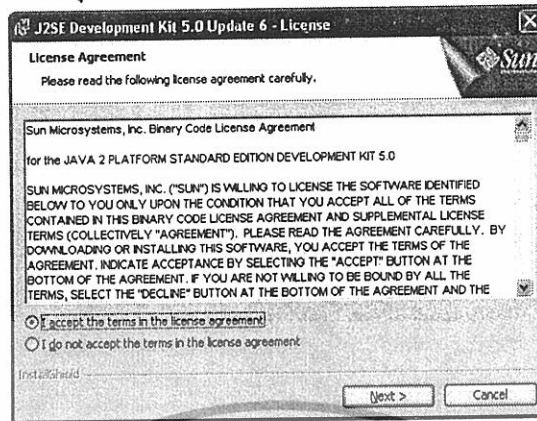
การติดตั้งโปรแกรม Netbeans ซึ่งมีกระบวนการ แบ่งการติดตั้งเป็น 2 ส่วนคือการติดตั้งตัวที่ใช้ทำงานพื้นฐานของจาวา และเครื่องมือ Netbeans

1.1 ติดตั้งตัวช่วยเหลือก่อน โดยคลิกที่ชื่อไฟล์ชื่อ jdk-1\_5\_0\_06-windows-i586-p.exe จากมีหน้าจอลงโปรแกรมปรากฏดังรูป



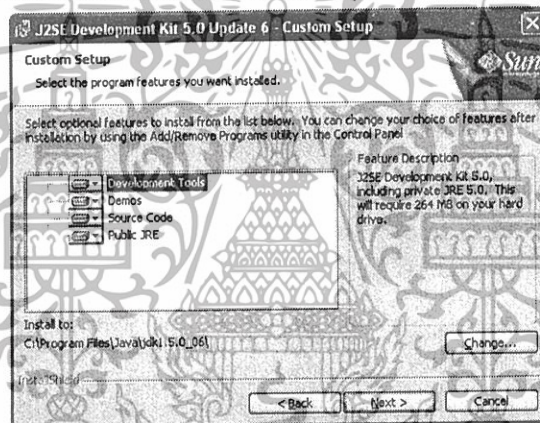
รูปที่ ก.1 การติดตั้งโปรแกรม Netbeans (แสดงภาพต่อเนื่อง 1)

เมื่อจัดเตรียมการลงโปรแกรมเสร็จแล้ว ขั้นตอนต่อไปจะถามว่ายอมรับกับเงื่อนไขหรือไม่ ให้คลิกที่ปุ่มยอมรับ แล้วคลิกปุ่ม “Next”



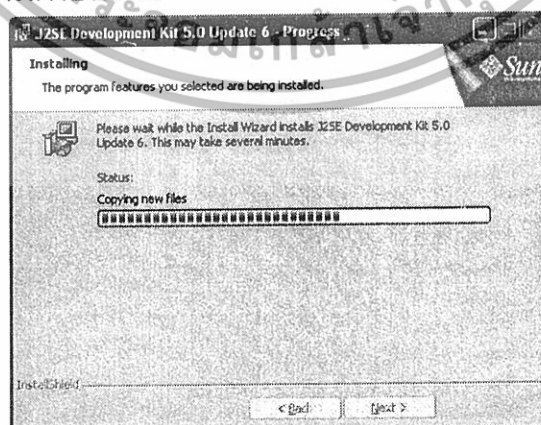
รูปที่ ก.2 การติดตั้งโปรแกรม Netbeans (แสดงภาพต่อเนื่อง 2)

ต่อไปเลือกตำแหน่งที่จะลงโปรแกรม แล้วคลิกปุ่ม “Next”



รูปที่ ก.3 การติดตั้งโปรแกรม (แสดงภาพต่อเนื่อง 3)

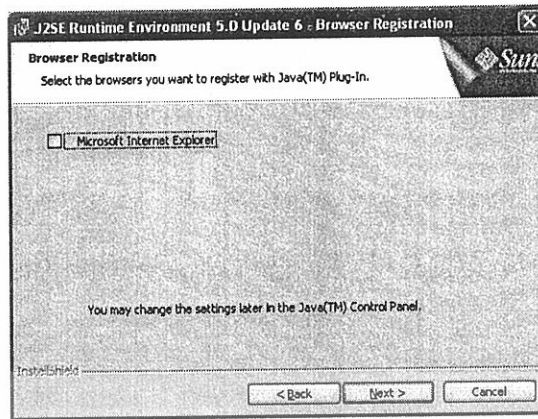
โปรแกรมทำการลงโปรแกรม



รูปที่ ก.4 การติดตั้งโปรแกรม Netbeans (แสดงภาพต่อเนื่อง 4)

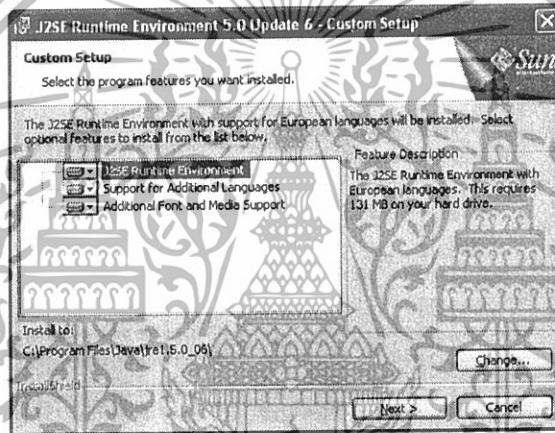
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลิกปุ่ม “Next” เพื่อลงโปรแกรมต่อไป



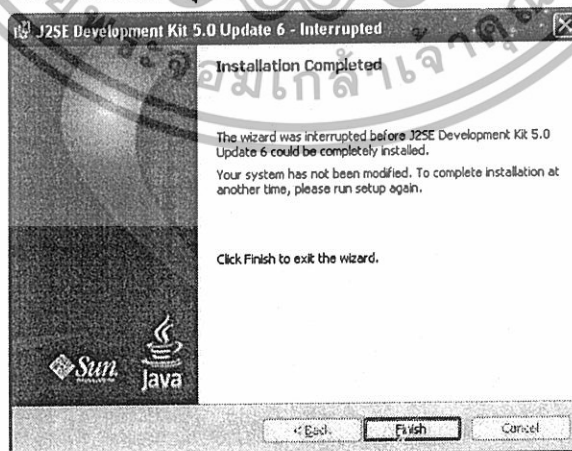
รูปที่ ก.5 การติดตั้งโปรแกรม Netbeans (แสดงภาพต่อเนื่อง 5)

ต่อไปเลือกตำแหน่งที่จะลงโปรแกรม แล้วคลิกปุ่ม “Next”



รูปที่ ก.6 การติดตั้งโปรแกรม Netbeans (แสดงภาพต่อเนื่อง 6)

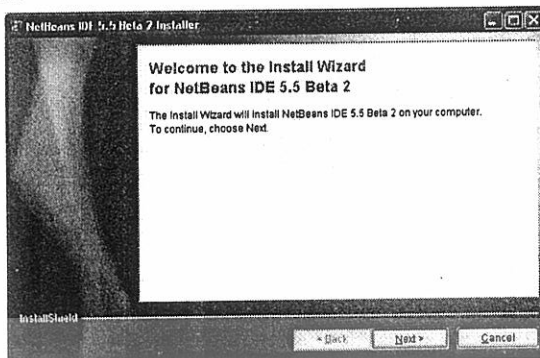
เสร็จสิ้นการลงโปรแกรมคลิกที่ปุ่ม “Finish”



รูปที่ ก.7 การติดตั้งโปรแกรม Netbeans (แสดงภาพต่อเนื่อง 7)

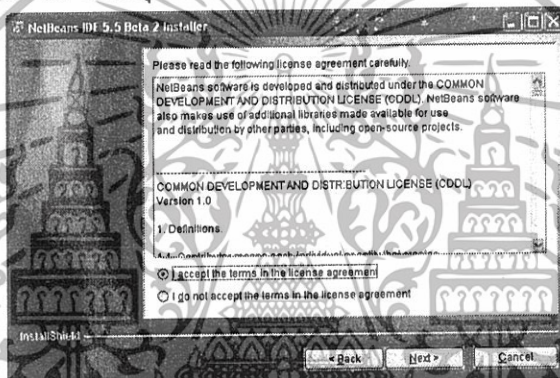
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.2 ติดตั้งตัวเครื่องมือ Netbeans ซึ่งเป็นตัวพัฒนาโปรแกรมภาษาจาวา โดยคลิกที่ไฟล์ชื่อ netbeans.exe เมื่อหน้าจอติดตั้งขึ้นให้คลิก “Next”



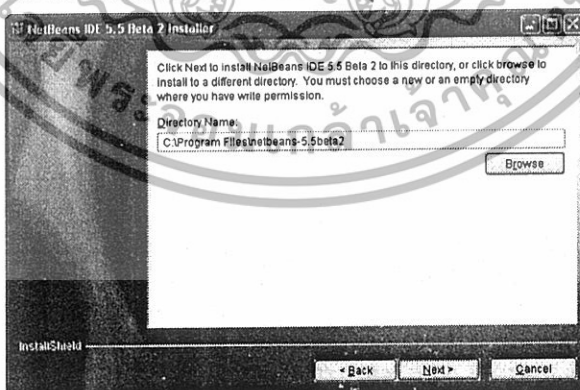
รูปที่ ก.8 การติดตั้งโปรแกรม Netbeans (แสดงภาพต่อเนื่อง 8)

คลิกที่ปุ่มยอมรับ แล้วคลิกปุ่ม “Next”



รูปที่ ก.9 การติดตั้งโปรแกรม Netbeans (แสดงภาพต่อเนื่อง 9)

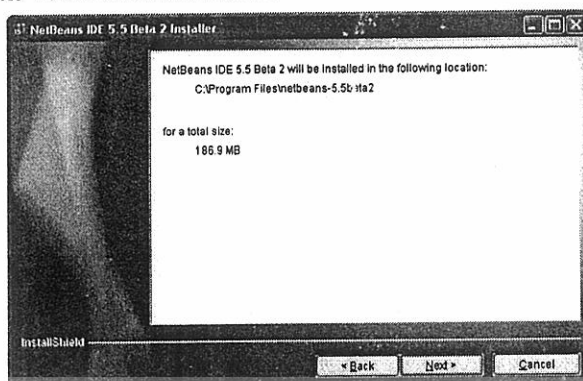
ต่อไปเลือกตำแหน่งที่จะลงโปรแกรม แล้วคลิกปุ่ม “Next”



รูปที่ ก.10 การติดตั้งโปรแกรม Netbeans (แสดงภาพต่อเนื่อง 10)

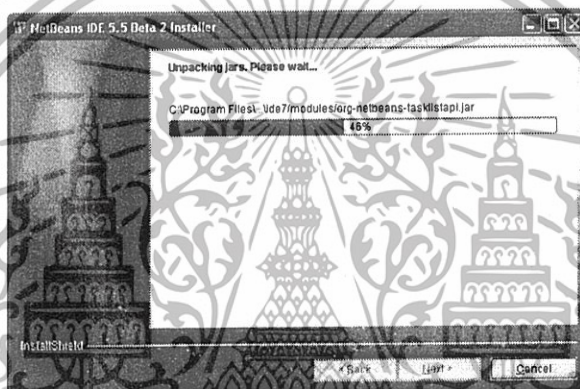
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลิกปุ่ม “Next” เพื่อลงโปรแกรมต่อไป



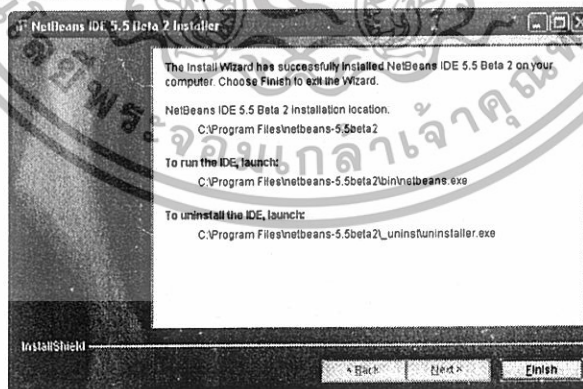
รูปที่ ก.11 การติดตั้งโปรแกรม Netbeans (แสดงภาพต่อเนื่อง 11)

โปรแกรมทำการติดตั้ง



รูปที่ ก.12 การติดตั้งโปรแกรม Netbeans (แสดงภาพต่อเนื่อง12)

เสร็จสิ้นการลงโปรแกรมคลิกที่ปุ่ม “Finish”



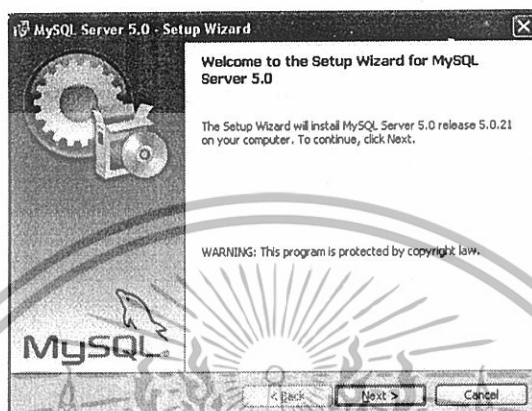
รูปที่ ก.13 การติดตั้งโปรแกรม Netbeans (แสดงภาพต่อเนื่อง13)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. การติดตั้งในส่วนของฐานข้อมูล

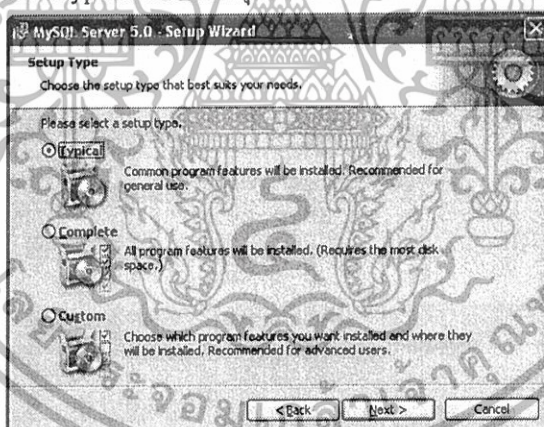
การติดตั้งในส่วนของฐานข้อมูล โดยใช้เครื่องมือ MySQL Browser ซึ่งมีกระบวนการ แบ่งการติดตั้งเป็น 2 ส่วนคือการติดตั้งตัวที่ใช้ทำงานพื้นฐานของฐานข้อมูลของ MySQL และตัว Browser ที่เป็นหน้าจอส่วนเชื่อมต่อกับผู้ใช้ มีขั้นตอนดังนี้

1.1 ติดตั้งตัวช่วยเหลือก่อน โดยคลิกที่ชื่อไฟล์ชื่อ SetupMySQL.exe จากมีหน้าจอการลงโปรแกรมปรากฏดังรูป



รูปที่ ก.14 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 1)

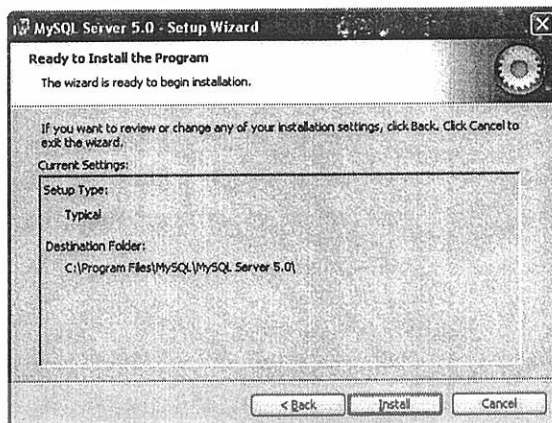
เลือกการติดตั้งแบบ typical แล้วคลิกปุ่ม "Next"



รูปที่ ก.15 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 2)

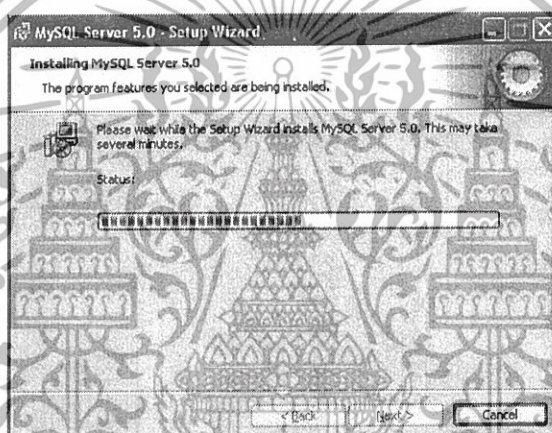
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลิกปุ่ม “Install” เพื่อลงโปรแกรม



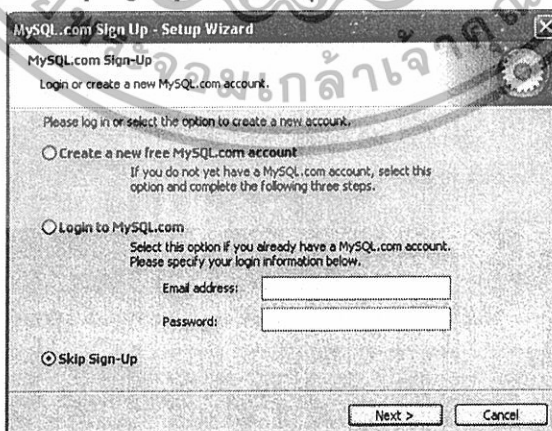
รูปที่ ก.16 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 3)

โปรแกรมกำลังทำการติดตั้ง



รูปที่ ก.17 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 4)

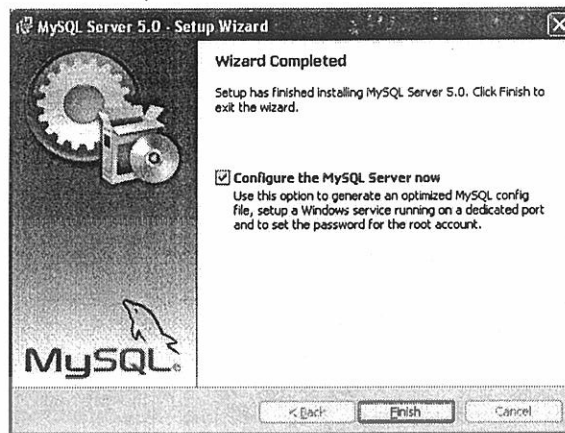
เลือกการลงแบบ “Skip Sign-up” แล้วคลิกปุ่ม “Next”



รูปที่ ก.18 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมติดตั้งเสร็จคลิกปุ่ม “Finish”



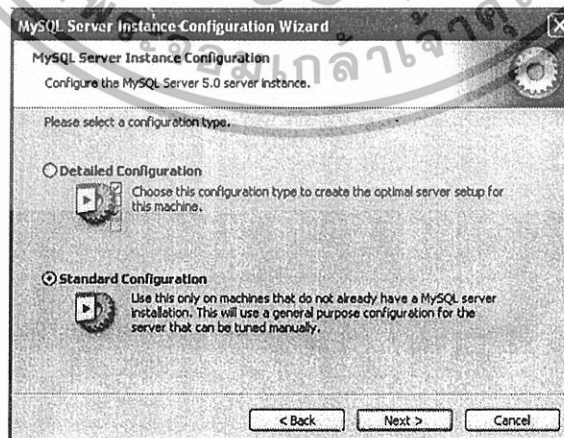
รูปที่ ก.19 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 6)

ทำการติดตั้ง server ต่อโดยคลิกปุ่ม “Next”



รูปที่ ก.20 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 7)

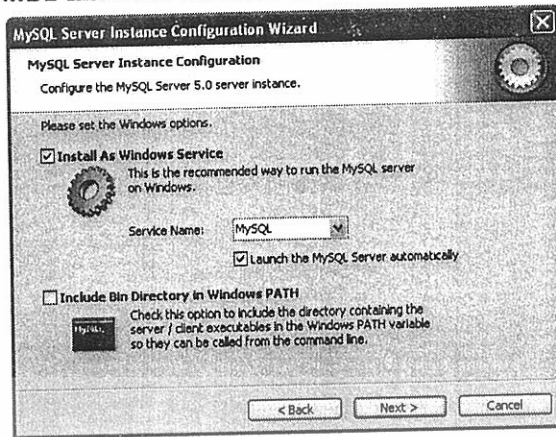
เลือกการลงแบบ Standard Configuration แล้วคลิกปุ่ม “Next”



รูปที่ ก.21 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 8)

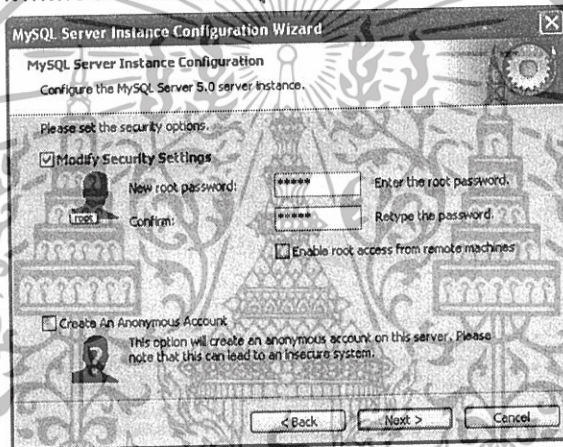
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกการติดตั้งแบบ Install As Windows Service แล้วคลิกปุ่ม “Next”



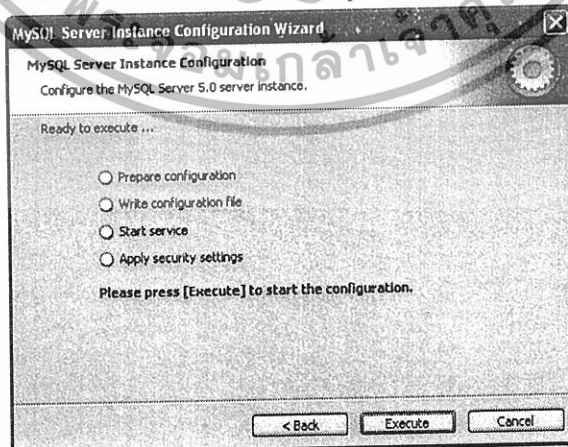
รูปที่ ก.22 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 9)

ใส่รหัสผ่าน ในที่นี้คือ kmitl แล้วคลิกปุ่ม “Next”



รูปที่ ก.23 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 10)

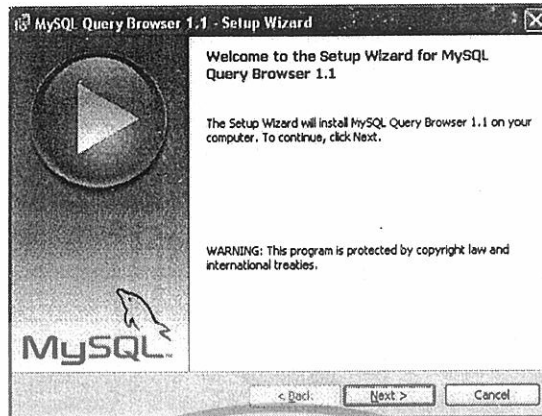
คลิกปุ่ม “Execute” เพื่อติดตั้งโปรแกรมให้สมบูรณ์ เป็นการเสร็จสิ้น



รูปที่ ก.24 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 11)

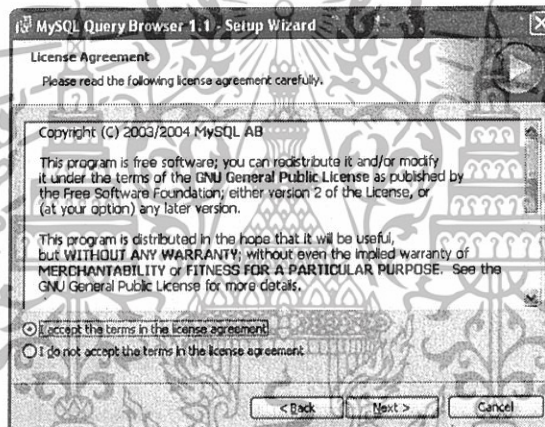
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ติดตั้ง Browser โดยคลิกที่ไฟล์ชื่อ mysql-query-browser-1[1].1.20-win.msi คลิกปุ่ม “Next”



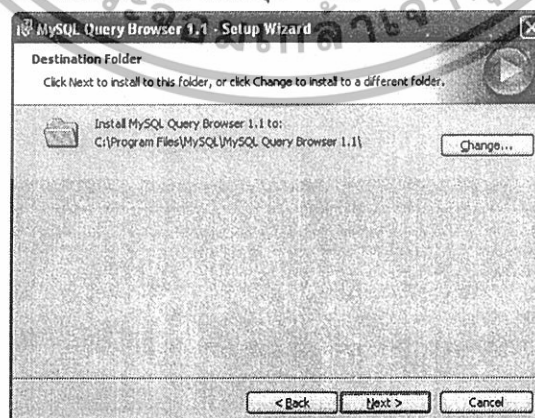
รูปที่ ก.25 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 12)

เลือกตกลงเพื่อขึ้นขังการลงโปรแกรม แล้วคลิกปุ่ม “Next”



รูปที่ ก.26 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 13)

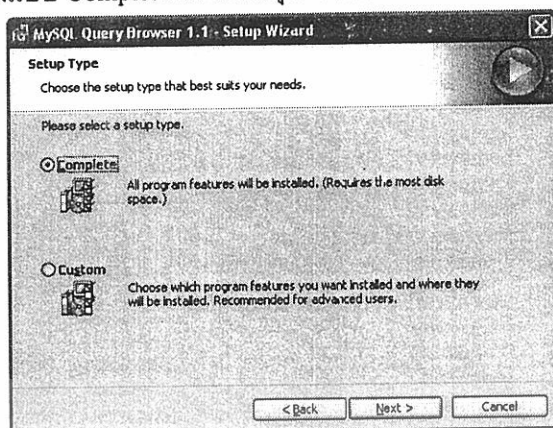
เลือกตำแหน่งที่ต้องการติดตั้งแล้วคลิกปุ่ม “Next”



รูปที่ ก.27 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 14)

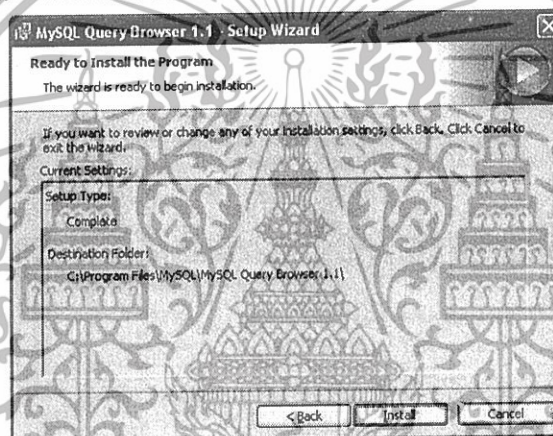
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกการติดตั้งแบบ Complete แล้วคลิกปุ่ม “Next”



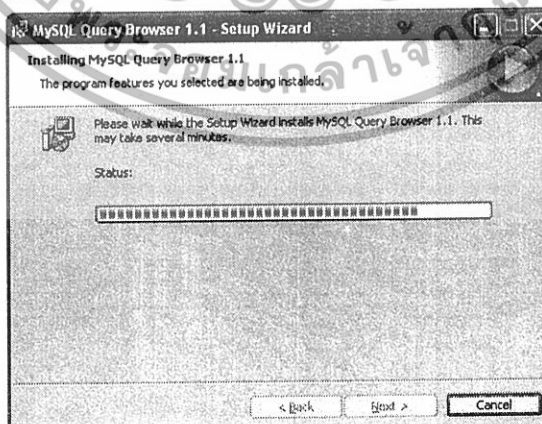
รูปที่ ก.28 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 15)

คลิกปุ่ม “Next>” เพื่อติดตั้งโปรแกรมต่อไป



รูปที่ ก.29 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 16)

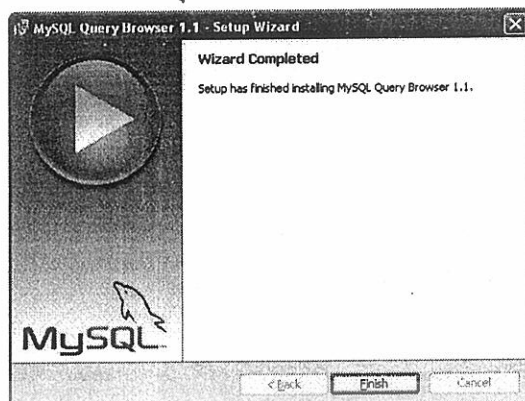
โปรแกรมทำการติดตั้ง



รูปที่ ก.30 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 17)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมติดตั้งเสร็จสิ้น คลิกที่ปุ่ม “Finish”



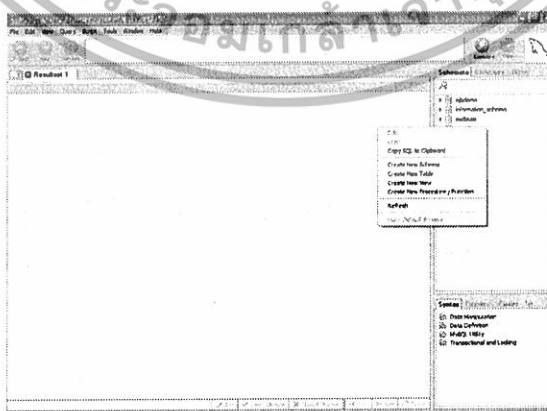
รูปที่ ก.31 การติดตั้งโปรแกรมฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 18)

เมื่อติดตั้งฐานข้อมูลแล้วให้เปิดโปรแกรมขึ้นมาโดยคลิกที่ชื่อ โปรแกรม แล้วใส่ชื่อและรหัสผ่าน ในที่นี้ชื่อก็คือ root และรหัสผ่านคือ kmitl แล้วคลิกปุ่ม “OK” ดังรูป



รูปที่ ก.32 การใช้งานฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 1)

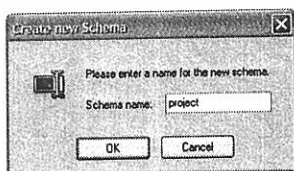
เมื่อหน้า โปรแกรมเปิดขึ้นมาแล้วให้สร้างฐานข้อมูลโดยคลิกขวาที่ส่วน Schemata แล้วเลือก create new schema ดังรูป



รูปที่ ก.33 การใช้งานฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วใส่ชื่อ schema ว่า project เพื่อเป็นการตั้งชื่อฐานข้อมูล แล้วคลิกที่ปุ่ม “OK”



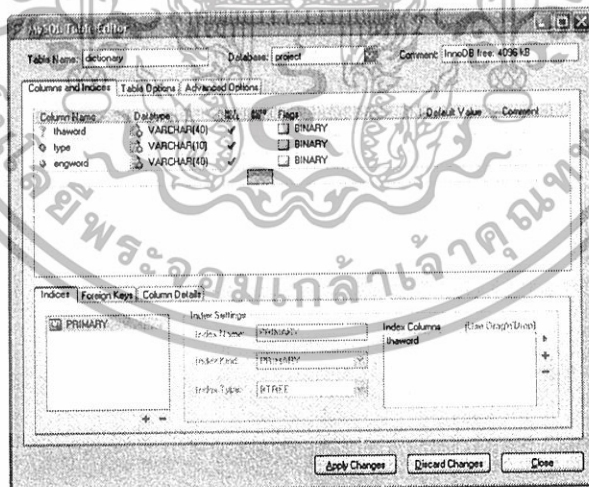
รูปที่ ก.34 การใช้งานฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 3)

เมื่อถึงขั้นตอนนี้จะได้ฐานข้อมูลที่ชื่อว่า project ปรากฏขึ้นในส่วน schemata ให้ทำการเพิ่มตาราง โดยคลิกขวาที่ฐานข้อมูล แล้วเลือก create new table ดังรูป



รูปที่ ก.35 การใช้งานฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 4)

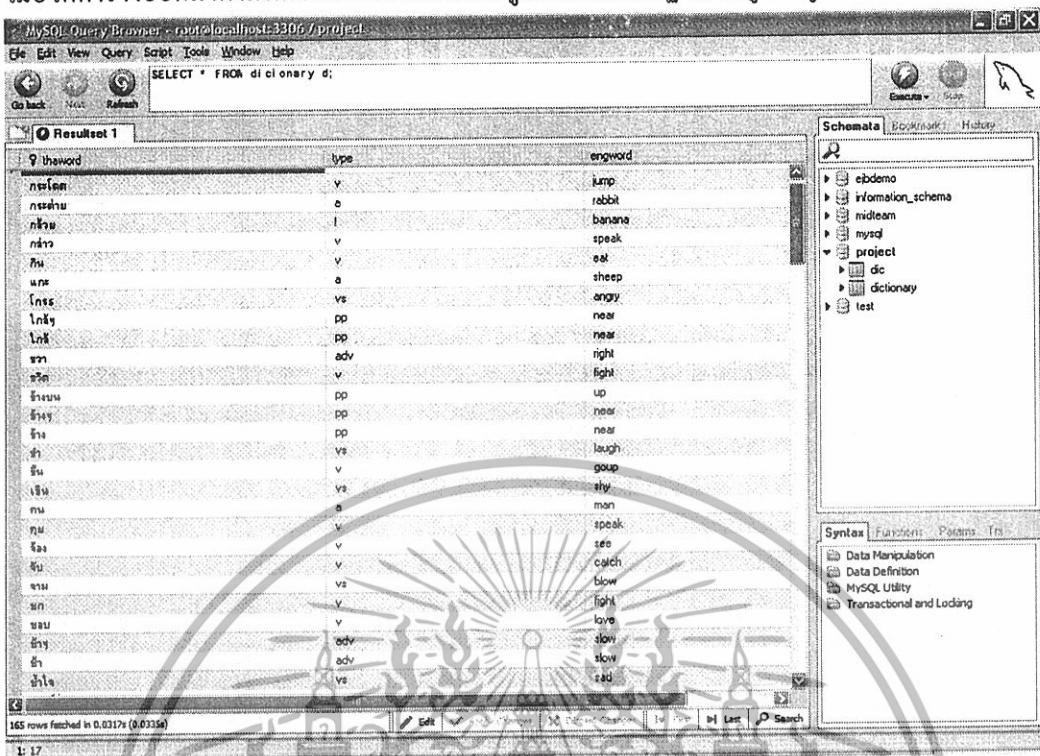
แล้วใส่ชื่อตารางว่า dictionary และกำหนดชนิดของข้อมูลต่างๆ ดังรูป แล้วคลิกที่ปุ่ม Apply Changes



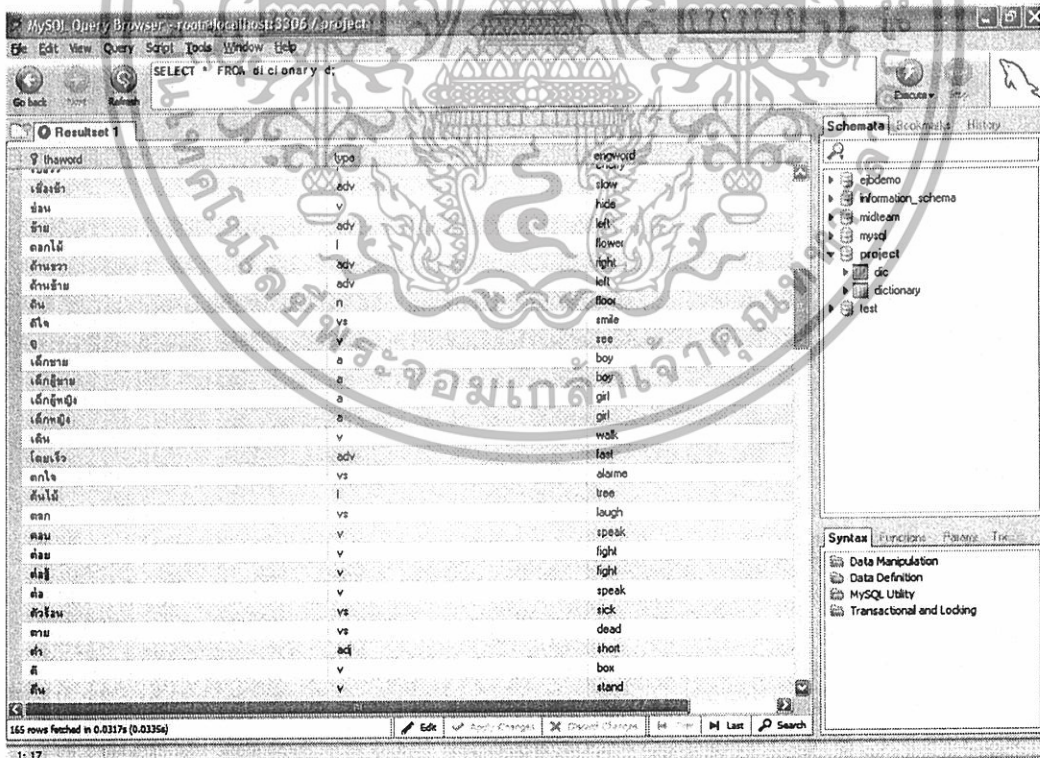
รูปที่ ก.36 การใช้งานฐานข้อมูล MySQL (แสดงภาพต่อเนื่อง 5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ตารางออกมาตามที่ต้องการแล้วให้ใส่ข้อมูลเหล่านี้ลงในฐานข้อมูลดังรูป



รูปที่ ก.37 การเพิ่มข้อมูลลงฐานข้อมูล (แสดงภาพต่อเนื่อง 1)



รูปที่ ก.38 การเพิ่มข้อมูลลงฐานข้อมูล (แสดงภาพต่อเนื่อง 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

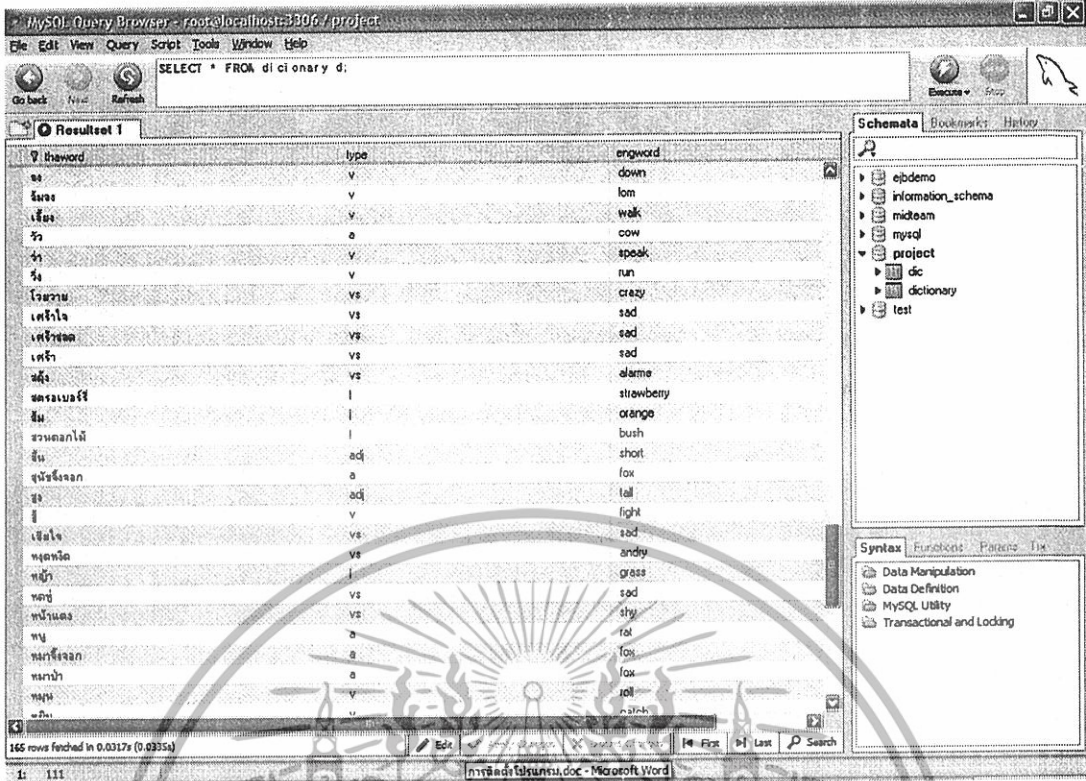
thaword	type	engword
เต้น	v	dance
เตะ	v	kick
เค็ม	adj	short
แตงโม	n	watermelon
ใต้	pp	under
ใต้ดิน	v	goup
ไล่	v	down
ถาม	v	speak
ถือ	v	catch
คว้า	pp	near
แคว	pp	near
ขวา	v	fight
ขวา	adv	right
ซ้าย	adv	left
ทาน	v	eat
ที่	pp	at
เธอ	pron	it
นอน	v	sleep
นั่ง	v	sit
โน	pp	at
บน	pp	up
บอก	v	speak
บ้าน	n	home
ป่วย	vs	sick
ป่วย	v	goup
ป่วย	v	down
ป่วย	vs	sick
ป่วย	vs	sick

รูปที่ ก.39 การเพิ่มข้อมูลลงฐานข้อมูล (แสดงภาพต่อเนื่อง 3)

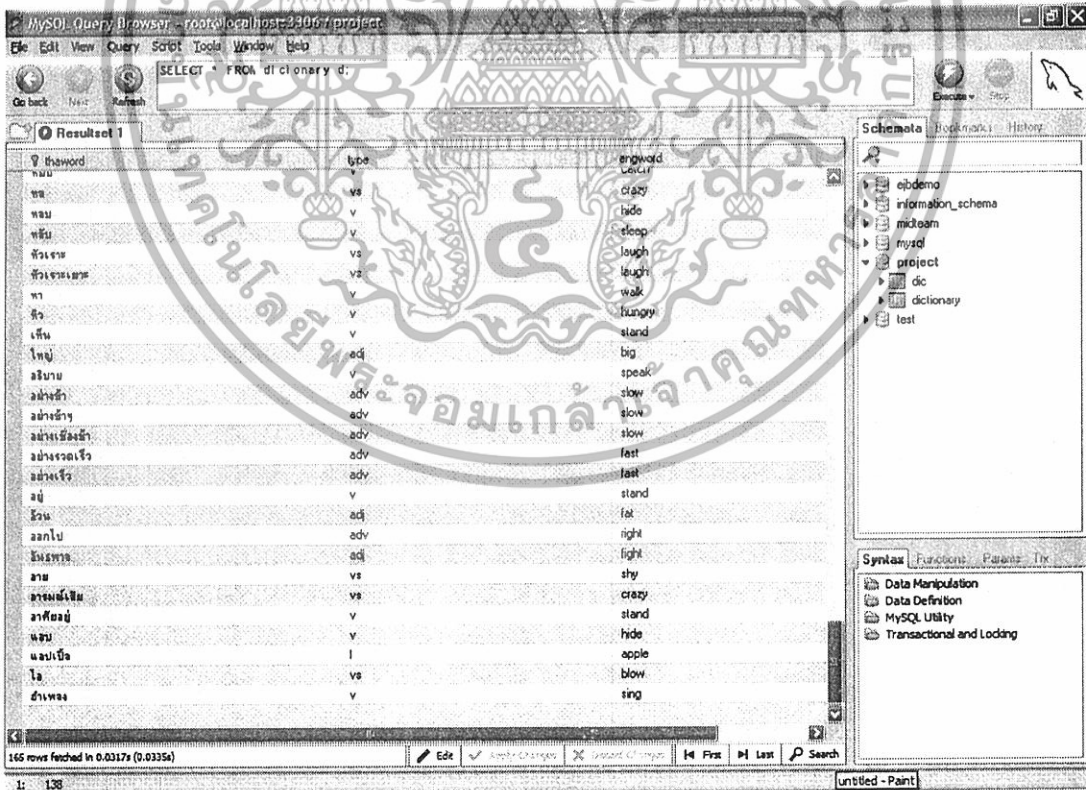
thaword	type	engword
เป็นพิษ	vs	sick
ไม่	v	walk
หม	adj	thin
ผู้ชาย	a	man
ชี้อา	n	pizza
พื้น	n	floor
ทุ่งไม้	n	garden
พูด	v	speak
มอง	v	see
สนทนา	n	lamon
อัน	pron	it
มา	v	walk
ม้า	n	horse
ยิ้ม	vs	smile
ไม่สบาย	vs	sick
ยาว	adj	tall
ยิ้ม	vs	smile
ยืน	v	stand
โยก	v	swing
รวดเร็ว	adv	fast
ร้องเพลง	v	sing
รัก	vs	love
รัก	v	love
รับประทาน	v	eat
รำ	v	dance
เร็ว	adv	fast
เร็ว	adv	fast

รูปที่ ก.40 การเพิ่มข้อมูลลงฐานข้อมูล (แสดงภาพต่อเนื่อง 4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.41 การเพิ่มข้อมูลลงฐานข้อมูล (แสดงภาพต่อเนื่อง 5)

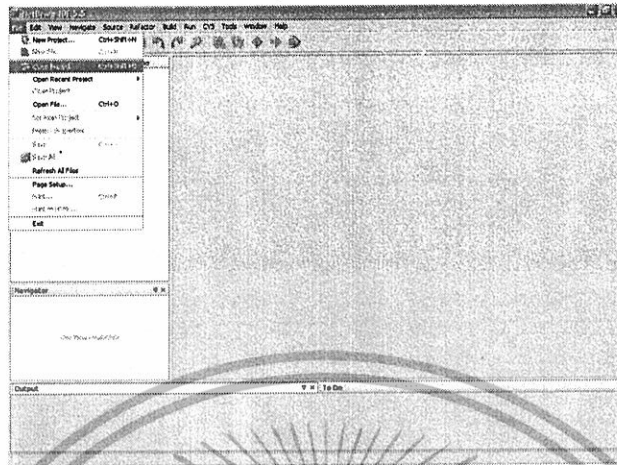


รูปที่ ก.42 การเพิ่มข้อมูลลงฐานข้อมูล (แสดงภาพต่อเนื่อง 6)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

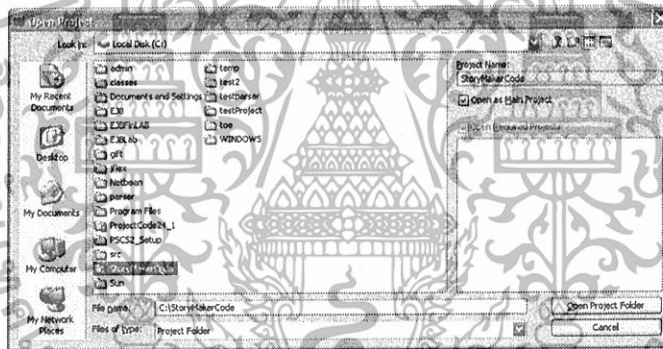
### 3. การติดตั้งโปรแกรม

เมื่อจัดเตรียมโปรแกรมเสร็จแล้ว จึงจะเริ่มใช้งานโปรแกรม โดยคลิกเปิดโปรแกรม Netbeans จะได้น้ำจอของโปรแกรมขึ้นมา แล้วคลิกที่ปุ่ม file เลือก Open Project ดังรูป



รูปที่ ก.43 การติดตั้งโปรแกรม (แสดงภาพต่อเนื่อง 1)

แล้วเลือกตำแหน่งของโปรเจกต์แล้วคลิก Open Project Folder



รูปที่ ก.44 การติดตั้งโปรแกรม (แสดงภาพต่อเนื่อง 2)

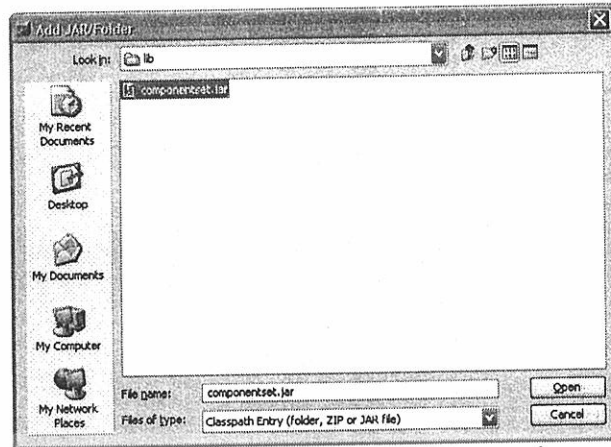
เนื่องจากโปรแกรมนั้นมีการใช้ไลบรารีภายนอกด้วย จึงต้องทำการเพิ่มไลบรารีก่อน โดยคลิกขวาที่ Libraries แล้วคลิกที่ Add JAR/Floder



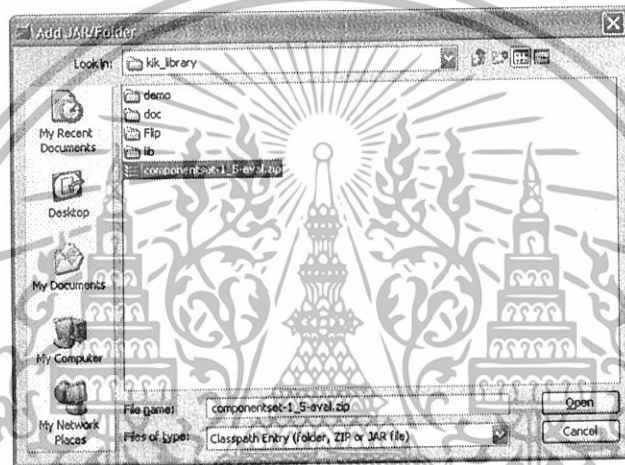
รูปที่ ก.45 การติดตั้งโปรแกรม (แสดงภาพต่อเนื่อง 3)

โดยติดตั้งไลบรารี 2 อันได้แก่ componenrset.jar และcomponentset-1\_5-eval.zip ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

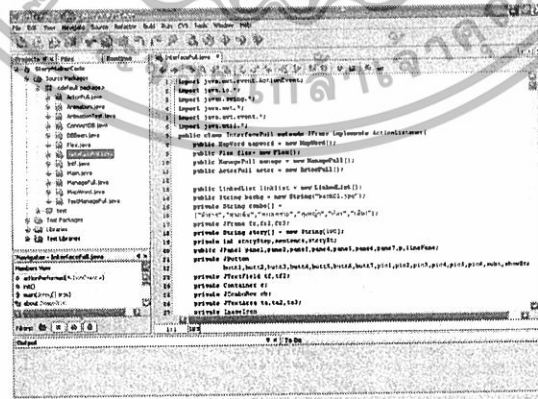


รูปที่ ก.46 การติดตั้งโปรแกรม (แสดงภาพต่อเนื่อง 4)



รูปที่ ก.47 การติดตั้งโปรแกรม (แสดงภาพต่อเนื่อง 5)

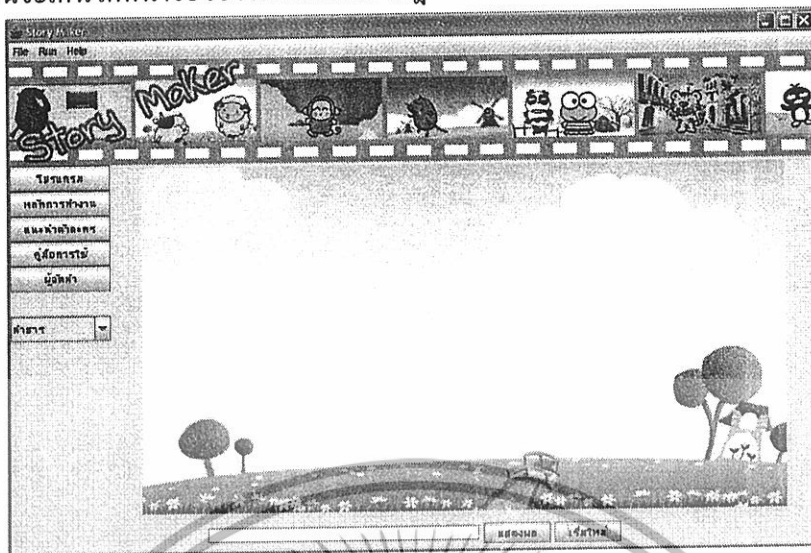
หลังจากนั้นให้เปิดไฟล์จาวาที่ชื่อ InterfaceFull.java แล้วกดปุ่ม Shift + f6 เพื่อทำการ run โปรแกรม



รูปที่ ก.48 การติดตั้งโปรแกรม (แสดงภาพต่อเนื่อง 6)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักจากนั้นจะเห็นได้หน้าจอของโปรแกรมปรากฏขึ้นและสามารถใช้งานโปรแกรมได้เลย



รูปที่ ก.49 หน้าจอโปรแกรม

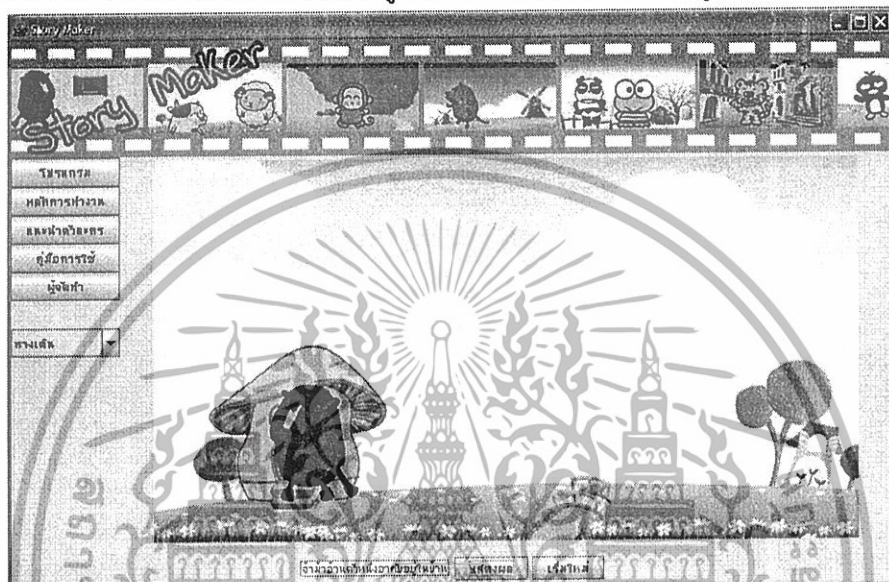


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ข.

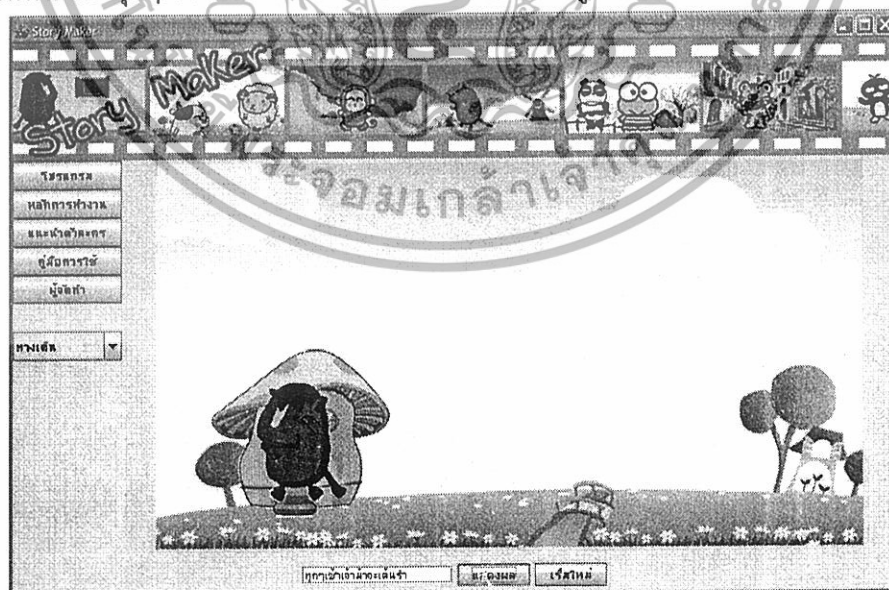
## ตัวอย่างการใช้งานโปรแกรม

การยกตัวอย่างการทำงานของโปรแกรมโดยประยุกต์ให้เป็นการเล่าเรื่องด้วยประโยค  
ดังต่อไปนี้  
ประโยคที่ 1 คือ มีเจ้าม้าอ้วนตัวหนึ่งอาศัยอยู่ในบ้าน ได้ผลลัพธ์ออกมาดังรูป



รูปที่ ข.1 แสดงผลของประโยค “มีเจ้าม้าอ้วนตัวหนึ่งอาศัยอยู่ในบ้าน”

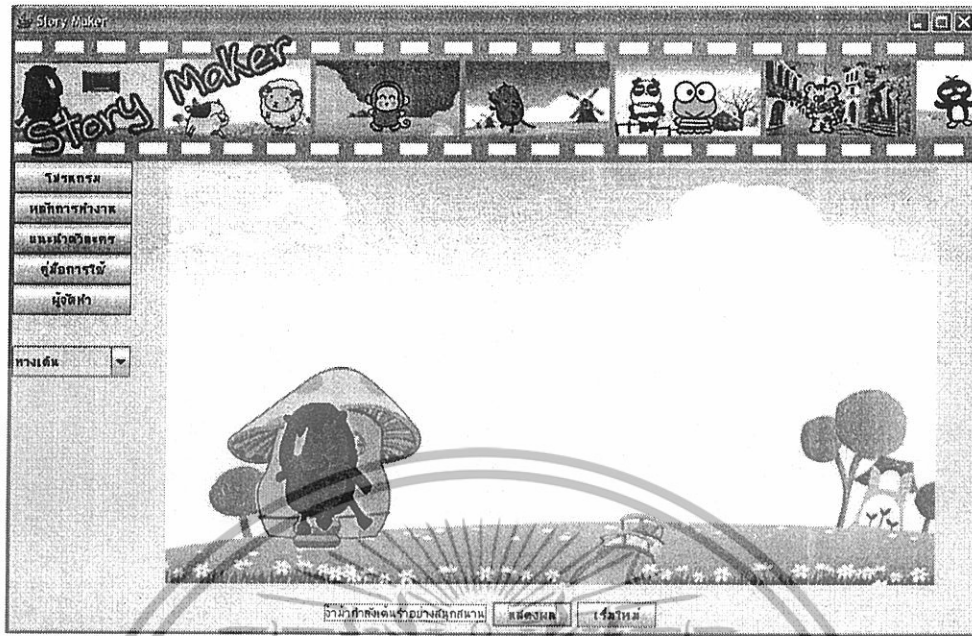
ประโยคที่ 2 คือ ทุกๆเจ้าเจ้าม้าจะตื่นเช้า ได้ผลลัพธ์ออกมาดังรูป



รูปที่ ข.2 แสดงผลของประโยค “ทุกๆเจ้าเจ้าม้าจะตื่นเช้า”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยคที่ 3 คือ วันหนึ่งขณะที่เจ้าม้ากำลังเดินร่าอย่างสนุกสนาน ได้ผลลัพธ์ออกมาดังรูป



รูปที่ ข.3 แสดงผลของประโยค “วันหนึ่งขณะที่เจ้าม้ากำลังเดินร่าอย่างสนุกสนาน”

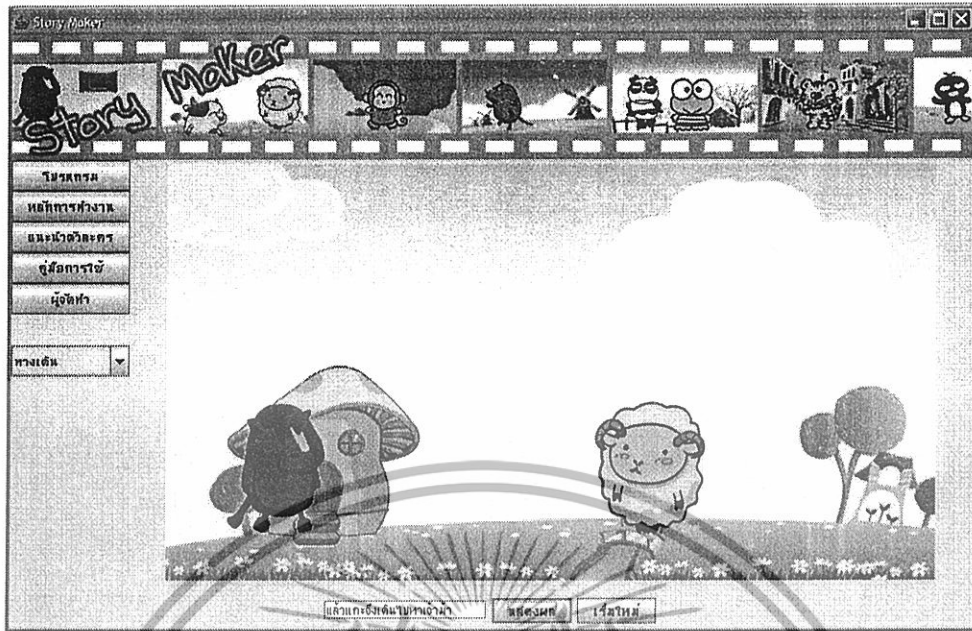
ประโยคที่ 4 คือ มีเจ้าแกะตัวหนึ่งเดินผ่านมา ได้ผลลัพธ์ออกมาดังรูป



รูปที่ ข.4 แสดงผลของประโยค “มีเจ้าแกะตัวหนึ่งเดินผ่านมา”

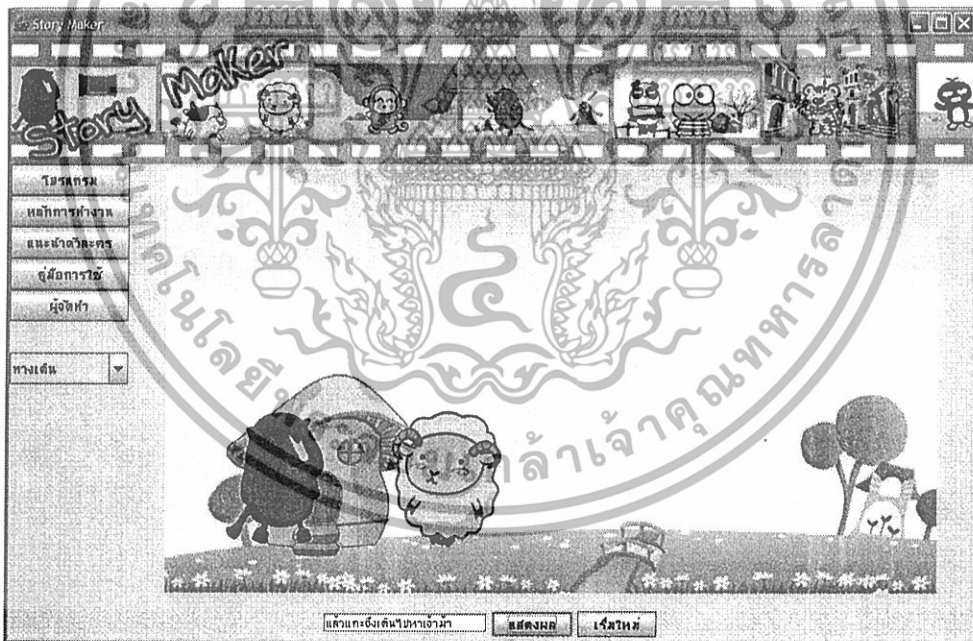
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยคที่ 5 คือ แล้วแกะจึงเดินไปหาเจ้าม้า ได้ผลลัพธ์ออกมาดังรูป



รูปที่ ข.5 แสดงผลของประโยค “แล้วแกะจึงเดินไปหาเจ้าม้า” (แสดงภาพต่อเนื่อง 1)

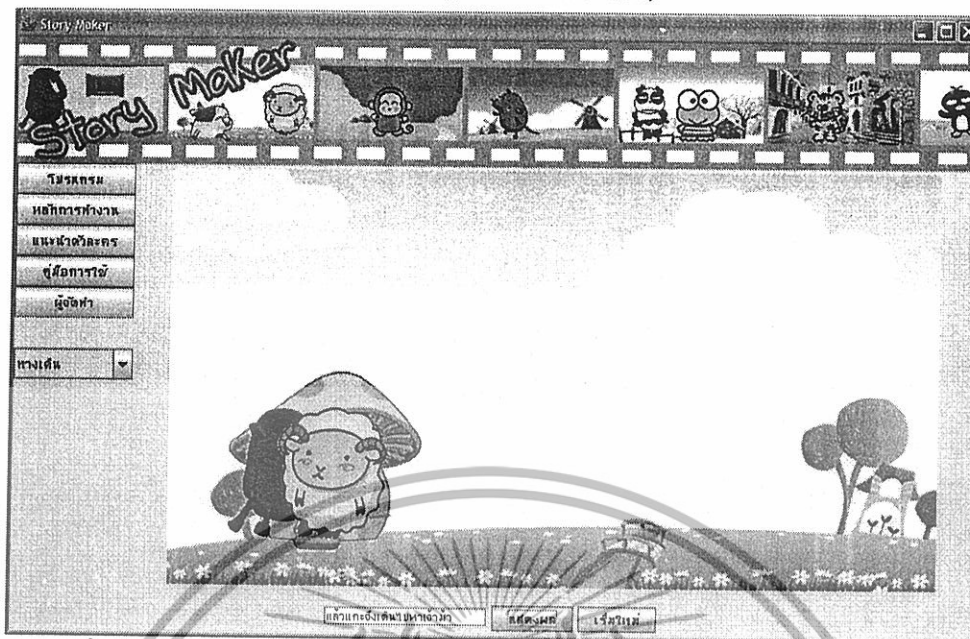
ประโยคที่ 5 คือ แล้วแกะจึงเดินไปหาเจ้าม้า (แสดงภาพต่อเนื่อง)



รูปที่ ข.6 แสดงผลของประโยค “แล้วแกะจึงเดินไปหาเจ้าม้า” (แสดงภาพต่อเนื่อง 2)

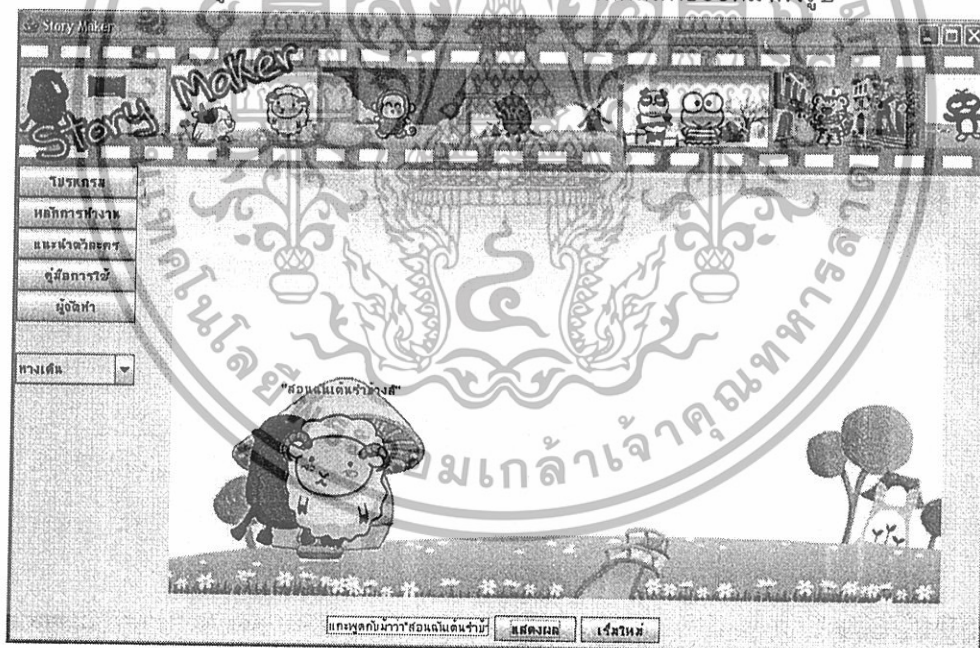
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยคที่ 5 คือ แล้วแกะจึงเดินไปหาเจ้าม้า (แสดงภาพต่อเนื่อง)



รูปที่ ข.7 แสดงผลของประโยค "แล้วแกะจึงเดินไปหาเจ้าม้า" (แสดงภาพต่อเนื่อง 3)

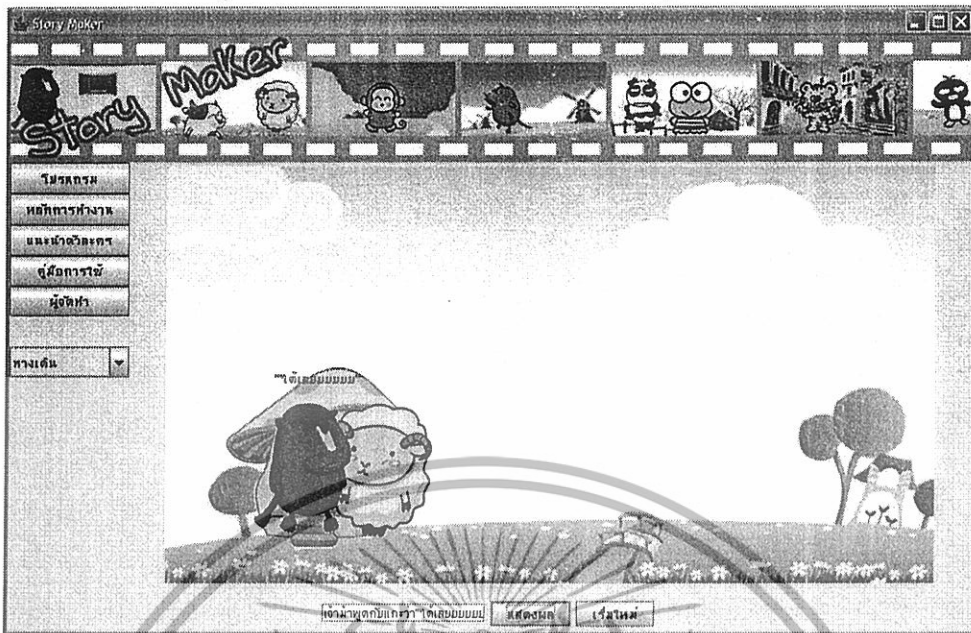
ประโยคที่ 6 คือ แกะพูดกับแกะว่า "สอนฉันเดินรำบั้งสิ" ได้ผลลัพธ์ออกมาดังรูป



รูปที่ ข.8 แสดงผลของประโยค "สอนฉันเดินรำบั้งสิ"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยคที่ 7 คือ เจ้าม้าพูดกับแกะว่า"ได้เลย" ได้ผลลัพธ์ออกมาดังรูป



รูปที่ ข.9 แสดงผลของประโยค “เจ้าม้าพูดกับแกะว่า"ได้เลย" ”

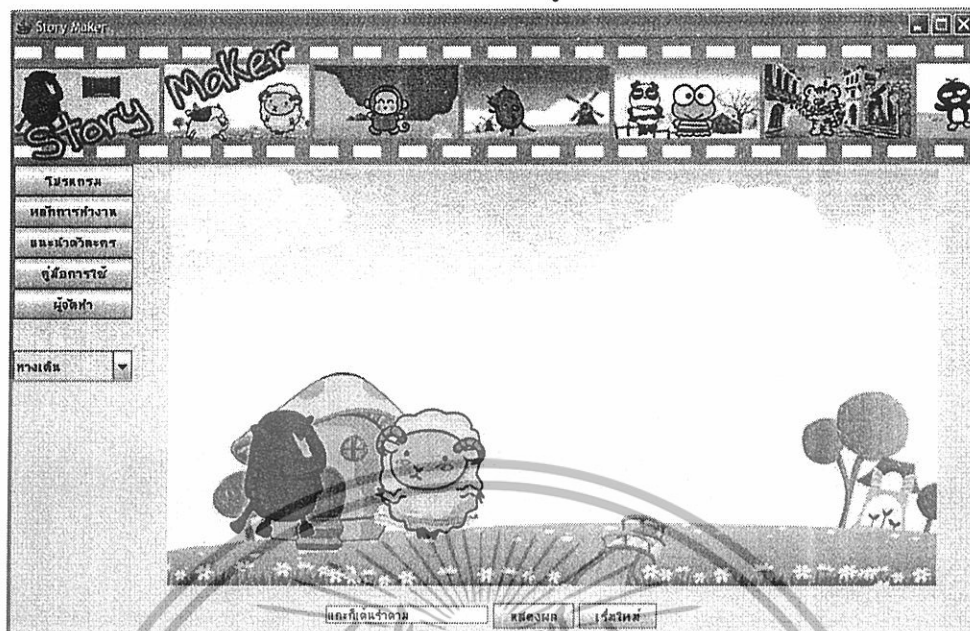
ประโยคที่ 8 คือ และม้าจึงสอนเดินรำให้แกะ ได้ผลลัพธ์ออกมาดังรูป



รูปที่ ข.10 แสดงผลของประโยค “และม้าจึงสอนเดินรำให้แกะ”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยคที่ 9 คือ แกะก็เดินรำตาม ได้ผลลัพธ์ออกมาดังรูป



รูปที่ ข.11 แสดงผลของประโยค “แกะก็เดินรำตาม”



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้จัดทำ

ชื่อผู้เขียน

นางสาววรรณพร เตียเจริญกิจ

สถานที่เกิด

จังหวัดกรุงเทพมหานคร

ระดับมัธยมศึกษา

โรงเรียนบดินทรเดชา(สิงห์ สิงหเสนี)2

ชื่อผู้เขียน

นายวีรธร อัครธิตานนท์

สถานที่เกิด

จังหวัดกรุงเทพมหานคร

ระดับมัธยมศึกษา

โรงเรียนบดินทรเดชา(สิงห์ สิงหเสนี)2

ชื่อผู้เขียน

นางสาววราภรณ์ พันโท

สถานที่เกิด

จังหวัดกรุงเทพมหานคร

ระดับมัธยมศึกษา

โรงเรียนเตรียมอุดมศึกษาน้อมเกล้า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้