เว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ ด้วยไอบีเอ็มบลูมิกซ์

WEB APPLICATION FOR MANAGING BIG DATA FROM TWITTER USING IBM BLUEMIX

ชาโชติ สมันตรมกุล
กฤษฎ์ เสมพืช
จิราวรรณ เงียมใจ

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

เว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์

WEB APPLICATION FOR MANAGING BIG DATA FROM TWITTER USING IBM BLUEMIX

ธนโชติ สมันตธนกุล
คณิศร เสมพืช
จีราวรรณ เจียมใจ

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

WEB APPLICATION FOR MANAGING BIG DATA FROM TWITTER USING IBM BLUEMIX

TANACHOT SAMANTATHANAKUL
KANISON SEMPHUECH
JEERAWAN JIAMJAI

A SPECIAL PROBLEM SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR
THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2015

หัวข้อปัญหาพิเศษ	เว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์ Web Application for Managing Big Data from Twitter Using IBM Bluemix		
ชื่อนักศึกษา	นายธนโชติ	สมันตธนกุล	รหัสนักศึกษา 55050315
	นางสาวคณิศร	เสมพืช	รหัสนักศึกษา 55050229
	นางสาวจีราวรรณ	เจียมใจ	รหัสนักศึกษา 55050250
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)		
ภาควิชา	วิทยาการคอมพิวเตอร์		
ปีการศึกษา	2558		
อาจารย์ที่ปรึกษา	ผศ.ดร.วรางคณา กิ้มปาน		

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2558

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.กฤษฎา บุศรา ประธานกรรมการ	
ดร.สันทิฏฐ์ นรบิน กรรมการ	
ผศ.ดร.วรางคณา กิ้มปาน กรรมการและอาจารย์ที่ปรึกษา	

หัวข้อปัญหาพิเศษ	เว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์
ชื่อนักศึกษา	นายธนโชติ สมันตธนกุล รหัสนักศึกษา 55050315
	นางสาวคณิศร เสมพืช รหัสนักศึกษา 55050229
	นางสาวจีราวรรณ เจียมใจ รหัสนักศึกษา 55050250
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2558
อาจารย์ที่ปรึกษา	ผศ.ดร.วรางคณา กิ้มปาน

บทคัดย่อ

บทความนี้นำเสนอเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์ โดยผู้ใช้สามารถกรองข้อมูลมาจากทวิตเตอร์แล้วนำข้อมูลเหล่านี้ไปเก็บในฮาดูปซึ่งสามารถเก็บข้อมูลได้จำนวนมากและมีกระบวนการแมปรีดิวซ์ฟังก์ชันสำหรับนับจำนวนคำที่พบในประโยค จากนั้นเว็บแอปพลิเคชันจะนำข้อมูลไปแสดงเป็นกราฟวงกลมเปรียบเทียบและกราฟวิเคราะห์อารมณ์และความรู้สึกของข้อความ โดยจำแนกประเภทอารมณ์และความรู้สึกของข้อความเป็นเชิงบวก เชิงลบ เป็นกลาง และคลุมเครือ เพื่อช่วยให้ผู้ใช้สามารถนำไปประกอบการตัดสินใจในทางธุรกิจและสามารถนำไปวิเคราะห์ข้อมูลตามที่ต้องการหรืองานวิจัยทางวิทยาศาสตร์ข้อมูล ดังนั้นจึงทำให้เกิดฟังก์ชันหลักในการทำงาน 4 ฟังก์ชันคือ ฟังก์ชันกรองคำจากทวิตเตอร์ ฟังก์ชันโหลดข้อมูลไปยังฮาดูป ฟังก์ชันแมปรีดิวซ์ และฟังก์ชันการแสดงผล โดยการออกแบบเว็บแอปพลิเคชันจะทำให้ผู้ใช้สามารถใช้งานได้ง่ายขึ้นและเพิ่มประสิทธิภาพในการวิเคราะห์ข้อมูล

คำสำคัญ : เว็บแอปพลิเคชัน ทวิตเตอร์ บิ๊กดาต้า ฮาดูป แมปรีดิวซ์ กราฟ

Title	Web Application for Managing Big Data from Twitter Using IBM Bluemix
Students	Mr. Tanachot Samantathanakul Student ID 55050315 Miss Kanison Semphuech Student ID 55050229 Miss Jeerawan Jiamjai Student ID 55050250
Degree	Bachelor of Science (Computer Science)
Department	Computer Science
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2558
Advisor	Asst.Prof.Dr.Warangkhana Kimpan

Abstract

This paper presents web application for managing big data from Twitter using IBM Bluemix. Users are able to filter Hashtag data from Twitter and then the data are stored into Hadoop. Hadoop then enables the users to store large volumes of data. Moreover, it provides the process map and reduces the functions for counting the number of interesting keywords. At last, the results will be shown in comparing pie graphs and sentiment analysis graphs. The sentiment classifications of text into positive, negative, neutral and ambivalent groups will help the users make a decision for the business and analyze the data or data science research. Therefore, this web application has 4 functions consist of filter data from Twitter, load data into HDFS, MapReduce and Output. Web application design will easily make the users understand and improve analyzing data performance.

Keywords : Web application, Twitter, Big data, Hadoop, MapReduce, Graph

กิตติกรรมประกาศ

ปัญหาพิเศษเรื่องเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์สำเร็จลุล่วงด้วยความกรุณาและความช่วยเหลืออย่างยิ่งจาก ผศ.ดร.วรางคณา กิ้มปาน อาจารย์ที่ปรึกษาที่ได้กรุณาให้คำปรึกษาแนะนำและตรวจสอบแก้ไขข้อบกพร่องทุกขั้นตอนของการจัดทำปัญหาพิเศษ และขอขอบพระคุณ ผศ.กฤษฎา บุศรา และดร.สันทิฏฐ์ นรบิน ที่ได้ให้ข้อคิดเห็นและคำแนะนำช่วยเหลือในการทำโครงงานพิเศษให้สำเร็จลุล่วงไปด้วยดี

ท้ายสุดนี้ขอขอบพระคุณบิดา มารดา เพื่อนนักศึกษา ตลอดจนผู้ที่มีส่วนเกี่ยวข้องทุกท่านที่ไม่ได้กล่าวนามไว้ ณ ที่นี้ ที่ได้ให้กำลังใจและมีส่วนช่วยเหลือให้ปัญหาพิเศษฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

ธนโชติ สมันตธนกุล
คณิศร เสมพืช
จีราวรรณ เจียมใจ

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญรูป ..... ฉ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหาพิเศษ 1

1.2 วัตถุประสงค์ของปัญหาพิเศษ 1

1.3 ขอบเขตของปัญหาพิเศษ 1

1.4 ประโยชน์ที่คาดว่าจะได้รับ 2

1.5 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ 2

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 3

2.1 บิ๊กดาต้า (Big Data) 3

2.1.1 คุณลักษณะของบิ๊กดาต้า 3

2.1.2 การจัดการบิ๊กดาต้าโดยรูปแบบการกระจายข้อมูล 4

2.2 ระบบฮาดูป 5

2.3 ระบบการจัดการไฟล์แบบกระจายของฮาดูป 5

2.3.1 โครงสร้างของฮาดูป 6

2.3.2 ส่วนประกอบในการทำงานของฮาดูป 7

2.4 แมปรีดิวซ์ 9

2.4.1 หลักการทำงานของแมปรีดิวซ์ 10

2.4.2 ขั้นตอนการแมป 11

2.4.3 ขั้นตอนการรีดิวซ์ 12

2.4.4 ตัวอย่างการทำงานของแมปรีดิวซ์ 13

2.5 IBM Bluemix 14

2.6 Servlet 15

2.7 JSP (Java Server Pages) 15

2.8 ภาษากรูวี (Groovy) 16

สารบัญ (ต่อ)

หน้า

บทที่ 3 วิธีการดำเนินงาน 17

3.1 สถาปัตยกรรมของระบบ 17

3.2 ความสามารถของระบบ 19

3.2.1 แผนภาพยูสเคส 19

3.2.2 แผนภาพแอคทิวิตี้ไดอะแกรม 20

3.2.3 แผนภาพซีเควนซ์ไดอะแกรม 27

3.3 การออกแบบส่วนติดต่อกับผู้ใช้ 28

บทที่ 4 ผลการดำเนินงานและการอภิปรายผล 32

4.1 การแสดงผลของเว็บไซต์ 32

4.1.1 หน้าจอแรกของเว็บไซต์ 32

4.1.2 หน้าจอแสดงการเข้าระบบ 33

4.1.3 หน้าจอการลงทะเบียนเพื่อเข้าใช้งานระบบ 33

4.1.4 หน้าจอเมนูหลักของเว็บไซต์ 34

4.1.5 หน้าจอฟังก์ชันกรองคำจากทวิตเตอร์ 34

4.1.6 หน้าจอฟังก์ชันโหลดข้อมูลไปยังฮาดูป 35

4.1.7 หน้าจอฟังก์ชันแมปรีดิวซ์ 37

4.1.8 หน้าจอฟังก์ชันการแสดงผล 41

4.2 การดำเนินการของเว็บไซต์ 43

4.2.1 ขั้นตอนการโหลด 43

4.2.2 ขั้นตอนการทำแมปรีดิวซ์ 43

บทที่ 5 สรุปผลและข้อเสนอแนะ 47

5.1 สรุปผล 47

5.2 ปัญหาที่พบและข้อจำกัด 47

5.3 แนวทางในการพัฒนาต่อ 48

เอกสารอ้างอิง 49

ภาคผนวก 50

ภาคผนวก ก คู่มือการใช้งาน 51

ภาคผนวก ข การติดตั้งซอฟต์แวร์ 62

ภาคผนวก ค ผลงานที่ได้รับรางวัล 79

สารบัญรูป

รูปที่		หน้า
2.1	คุณลักษณะของบิ๊กดาต้า	3
2.2	รูปแบบการแสดงการจัดการข้อมูลที่เป็นบิ๊กดาต้า	4
2.3	การแบ่งไฟล์เป็นบล็อกของระบบจัดการไฟล์แบบกระจายของฮาดูป	5
2.4	ไฟล์บล็อกของระบบจัดการไฟล์แบบกระจายของฮาดูป	6
2.5	โครงสร้างของฮาดูป	6
2.6	ส่วนประกอบต่างๆใน Master Node และ Slave Node	7
2.7	ส่วนประกอบในการทำงานของฮาดูป	8
2.8	ตัวอย่างฟังก์ชัน Map และ Reduce	9
2.9	รูปแบบการทำงานของ Map และ Reduce	10
2.10	หลักการทำงานของแมปรีดิวซ์	10
2.11	กระบวนการ Map	12
2.12	กระบวนการ Reduce	12
2.13	แผนภาพการทำงานของแมปรีดิวซ์	13
2.14	ตัวอย่างการนับคำของข้อมูลด้วยวิธีแมปรีดิวซ์	14
2.15	ตัวอย่างโปรแกรมภาษาจาวา	16
2.16	ตัวอย่างโปรแกรมภาษากรูวี	16
3.1	กระบวนการทำงานของระบบ	17
3.2	Use Case Diagram ของเว็บแอปพลิชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์	19
3.3	แผนภาพ Activity Diagram ของการสมัครสมาชิก	20
3.4	แผนภาพ Activity Diagram ของการเข้าสู่ระบบ	21
3.5	แผนภาพ Activity Diagram ของฟังก์ชันกรองคำจากทวิตเตอร์	22
3.6	แผนภาพ Activity Diagram ของฟังก์ชันการโหลดข้อมูลไปยังฮาดูป	23
3.7	แผนภาพ Activity Diagram ของฟังก์ชันการแมปรีดิวซ์	24
3.8	แผนภาพ Activity Diagram ของฟังก์ชันการแสดงผลกราฟเปรียบเทียบ	25
3.9	แผนภาพ Activity Diagram ของฟังก์ชันการแสดงผลกราฟอารมณ์และความรู้สึก	26
3.10	แผนภาพ Sequence Diagram ภาพรวมของระบบและฟังก์ชันต่างๆ	27
3.11	ตัวอย่างหน้าจอแสดงการเข้าสู่ระบบ	28
3.12	ตัวอย่างหน้าจอลงทะเบียนเข้าสู่ระบบ	29

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.13	ตัวอย่างหน้าจอเมนูหลักของเว็บไซต์	29
3.14	ตัวอย่างหน้าจอฟังก์ชันกรองคำจากทวิตเตอร์	30
3.15	ตัวอย่างหน้าจอฟังก์ชันโหลดข้อมูลไปยังฮาดูป	30
3.16	ตัวอย่างหน้าจอฟังก์ชันแมปรีดิวซ์	31
3.17	ตัวอย่างหน้าจอฟังก์ชันการแสดงผล	31
4.1	หน้าจอแรกของเว็บไซต์	32
4.2	หน้าจอการเข้าสู่ระบบ	33
4.3	หน้าจอการลงทะเบียน	33
4.4	หน้าจอเมนูหลักของเว็บไซต์	34
4.5	ขั้นตอนการกรองคำ	34
4.6	รายละเอียดของข้อมูลที่กรองมาจากทวิตเตอร์	35
4.7	ขั้นตอนการโหลดข้อมูลไปเก็บไว้บนฮาดูป	35
4.8	หน้าจอ webhdfs สำหรับผู้ดูแลระบบ (1)	36
4.9	หน้าจอ webhdfs สำหรับผู้ดูแลระบบ (2)	36
4.10	รายละเอียดเกี่ยวกับไฟล์สำหรับผู้ดูแลระบบ	37
4.11	หน้าจอฟังก์ชันแมปรีดิวซ์	37
4.12	หน้าจอ webhdfs สำหรับผู้ดูแลระบบ (3)	38
4.13	หน้าจอ webhdfs สำหรับผู้ดูแลระบบ (4)	38
4.14	ไฟล์ต้นฉบับในกระบวนการแมปรีดิวซ์	39
4.15	รายละเอียดไฟล์ SourceFile.txt	39
4.16	หน้าจอ webhdfs สำหรับผู้ดูแลระบบ (5)	40
4.17	ไฟล์ Output ที่ผ่านกระบวนการแมปรีดิวซ์	40
4.18	รายละเอียดของไฟล์ part-r-00000	41
4.19	หน้าจอแสดงผลลัพธ์กราฟเปรียบเทียบ	41
4.20	ขั้นตอนการแสดงกราฟเปรียบเทียบ	42
4.21	หน้าจอแสดงผลลัพธ์อารมณ์และความรู้สึก	42
4.22	โปรแกรมขั้นตอนการโหลดข้อมูล	43
4.23	โปรแกรมการกำหนดค่าเส้นทางต่างๆ	43
4.24	โปรแกรมการกำหนดค่าให้กับ Job Tracker (1)	44

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.25	โปรแกรมการกำหนดค่าให้กับ Job Tracker (2)	44
4.26	โปรแกรมการเริ่มทำงานของฮาดูป	45
4.27	โปรแกรมการนำไฟล์ที่ถูกโยนขึ้นไปบนฮาดูปมาตัดคำ	46
ก.1	หน้าแรกของเว็บไซต์	51
ก.2	หน้าจอเข้าสู่ระบบของเว็บไซต์	52
ก.3	หน้าจอลงทะเบียนของเว็บไซต์	52
ก.4	หน้าจอเมนูหลักของเว็บไซต์	53
ก.5	หน้าจอฟังก์ชันการกรองคำ	53
ก.6	ขั้นตอนการกรองคำ	54
ก.7	การสร้างตารางที่สามารถใช้งานได้	54
ก.8	การสร้างตารางที่ไม่สามารถใช้งานได้	55
ก.9	รายละเอียดของข้อมูลที่ได้กรองมาจากทวิตเตอร์	55
ก.10	ขั้นตอนการโหลดข้อมูลไปเก็บไว้บนฮาดูป (1)	56
ก.11	ขั้นตอนการโหลดข้อมูลไปเก็บไว้บนฮาดูป (2)	56
ก.12	หน้าจอฟังก์ชันแมปรีดิวซ์	57
ก.13	ขั้นตอนการทำแมปรีดิวซ์	57
ก.14	หน้าจอแสดงผลลัพธ์กราฟเปรียบเทียบแบรนด์เครื่องสำอาง	58
ก.15	ขั้นตอนการแสดงกราฟเปรียบเทียบ	58
ก.16	หน้าจอแสดงกราฟเปรียบเทียบ	59
ก.17	หน้าจอแสดงผลลัพธ์อารมณ์และความรู้สึก	59
ก.18	ขั้นตอนการแสดงกราฟอารมณ์และความรู้สึก (1)	60
ก.19	ขั้นตอนการแสดงกราฟอารมณ์และความรู้สึก (2)	60
ก.20	กราฟอารมณ์และความรู้สึก	61
ข.1	ขั้นตอนการติดตั้งกรูวี (1)	62
ข.2	ขั้นตอนการติดตั้งกรูวี (2)	63
ข.3	ขั้นตอนการติดตั้งกรูวี (3)	63
ข.4	ขั้นตอนการติดตั้งกรูวี (4)	64
ข.5	ขั้นตอนการติดตั้งกรูวี (5)	64
ข.6	ขั้นตอนการติดตั้งกรูวี (6)	65

สารบัญรูป (ต่อ)

รูปที่		หน้า
ข.7	ขั้นตอนการติดตั้งกรูวี (7)	65
ข.8	ขั้นตอนการติดตั้งกรูวี (8)	66
ข.9	ขั้นตอนการติดตั้งกรูวี (9)	66
ข.10	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (1)	67
ข.11	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (2)	67
ข.12	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (3)	68
ข.13	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (4)	68
ข.14	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (5)	69
ข.15	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (6)	69
ข.16	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (7)	70
ข.17	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (8)	70
ข.18	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (9)	71
ข.19	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (10)	71
ข.20	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (11)	72
ข.21	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (12)	72
ข.22	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (13)	73
ข.23	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (14)	73
ข.24	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (15)	74
ข.25	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (16)	74
ข.26	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (17)	75
ข.27	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (18)	75
ข.28	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (19)	76
ข.29	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (20)	76
ข.30	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (21)	77
ข.31	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (22)	77
ข.32	ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (23)	78
ค.1	ภาพถ่ายคณะผู้จัดทำปัญหาพิเศษขณะเข้าร่วมการนำเสนอผลงาน	79
ค.2	ประกาศนียบัตรรางวัล “Very Good Paper Award”	80

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหาพิเศษ

ในยุคปัจจุบันมีการใช้อุปกรณ์ประเภทสมาร์ทโฟนและแท็บเล็ตกันอย่างแพร่หลาย มีแอปพลิเคชันถูกพัฒนามาเพื่อสนับสนุนแพลตฟอร์มดังกล่าวมากมาย รวมถึงความนิยมในการใช้โซเชียลเน็ตเวิร์คไม่ว่าจะเป็น ทวิตเตอร์ (Twitter) เฟซบุ๊ก (Facebook) หรือไฟล์จำพวกมีเดีย (Media) ที่มีขนาดใหญ่ เป็นต้น และการทำธุรกิจหรือธุรกรรมออนไลน์ต่างๆ ทำให้มีข้อมูลเกิดขึ้นในระบบเป็นจำนวนมากทั้งข้อมูลที่มีโครงสร้างและข้อมูลที่ไม่มีโครงสร้าง การจัดการกับข้อมูลจำนวนมากและเกิดขึ้นตลอดเวลาซึ่งไม่สามารถทำได้ด้วยวิธีการจัดเก็บไว้ในดาต้าเบสรูปแบบเดิมๆได้ดี ดังนั้นจึงมีการจัดการข้อมูลขนาดใหญ่หรือบิ๊กดาต้า ซึ่งวิธีการจัดการกับบิ๊กดาต้านั้น ณ ปัจจุบันมีเครื่องมือที่ได้รับความนิยมอย่างแพร่หลายและมีชื่อเสียงตัวหนึ่งเข้ามาช่วยจัดการ ได้แก่ ฮาดูป (Hadoop) ที่พัฒนามาจากอาปาเช่โอเพนซอร์สเทคโนโลยี (Apache Open Source Technology) ให้ทำหน้าที่เป็นแหล่งจัดเก็บข้อมูลแบบกระจาย (Distributed Storage) ที่สามารถเก็บข้อมูลขนาดใหญ่และนำมาประมวลผลได้ ข้อมูลต่างๆที่เกิดขึ้นเหล่านี้ก่อให้เกิดโอกาสมากมายโดยการนำข้อมูลมาประมวลผลและวิเคราะห์ให้เกิดมูลค่าทางธุรกิจ ข้อได้เปรียบทางการแข่งขัน ช่วยสนับสนุนการตัดสินใจ และช่วยในกำหนดการวางแผนเชิงรุกของการทำงานในอนาคตได้อีกด้วย

1.2 วัตถุประสงค์ของปัญหาพิเศษ

1) เพื่อให้ผู้ใช้มีเครื่องมือในการช่วยวิเคราะห์ข้อมูลจากทวิตเตอร์
2) เพื่อช่วยให้ผู้ใช้มีเครื่องมือในการเปรียบเทียบและใช้ประกอบในการตัดสินใจ
3) เพื่อรองรับการเก็บข้อมูลขนาดใหญ่ (Big Data)
4) เพื่อศึกษาการทำงานของฮาดูป (Hadoop) ไปใช้ประมวลผลข้อมูลขนาดใหญ่ได้
5) เพื่อศึกษาเครื่องมือที่ใช้ในการวิเคราะห์ข้อมูลขนาดใหญ่

1.3 ขอบเขตของปัญหาพิเศษ

1) พัฒนาเว็บแอปพลิเคชันเพื่อใช้วิเคราะห์ข้อมูลจากทวิตเตอร์ด้วยซอฟต์แวร์ IBM Bluemix
2) สามารถใช้งานฟังก์ชันกรองคำจากทวิตเตอร์ซึ่งสามารถกรองข้อมูลได้สูงสุด 50,000 ทวิตเตอร์โดยจะสุ่มตัวอย่างข้อมูลจากทวิตเตอร์มา 10% และสามารถค้นหาคำที่เป็นภาษาอังกฤษได้เท่านั้นเพื่อนำข้อมูลไปวิเคราะห์ต่อไป
3) ใช้งานฟังก์ชันการโหลดข้อมูลไปฮาดูปซึ่งสามารถนำตารางที่ได้จากฟังก์ชันกรองคำจากทวิตเตอร์ไปเก็บไว้บนฮาดูป

4) สามารถใช้งานฟังก์ชันแมปรีดิวซ์ในการนับคำที่กรองมาจากทวิตเตอร์ได้
5) สามารถใช้งานฟังก์ชันการแสดงผลซึ่งจะแสดงออกมาในรูปแบบของกราฟวงกลม
6) รองรับผู้ใช้งานได้ไม่เกิน 100 คน

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ช่วยให้ผู้ใช้สามารถนำข้อมูลไปวิเคราะห์เพื่อให้เกิดประโยชน์ต่างๆได้อีกมากมายทั้งในเชิงธุรกิจหรือด้านงานวิจัยทางด้าน Data Science
2) ได้รับความรู้ในการใช้เครื่องมือต่างๆที่นำมาใช้ในการวิเคราะห์ข้อมูลขนาดใหญ่
3) สามารถรองรับการเก็บข้อมูลขนาดใหญ่ได้ (Big data)

1.5 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

1) ฮาร์ดแวร์
   - โน้ตบุ๊ก 1 เครื่อง
2) ซอฟต์แวร์
   - IBM Bluemix
   - Eclipse
   - DashDB
   - BigSQL
   - Hadoop
   - Websphere

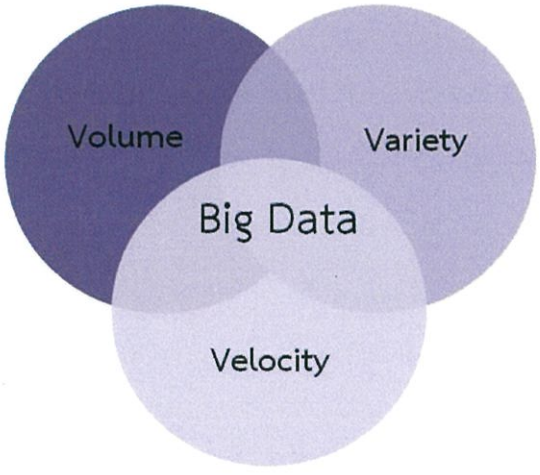
บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 บิ๊กดาต้า (Big Data)

บิ๊กดาต้า [1] หมายถึง ปริมาณข้อมูลที่มีขนาดใหญ่มหาศาลเกินกว่าขีดความสามารถในการประมวลผลของระบบฐานข้อมูลธรรมดาที่จะรองรับได้ ปริมาณข้อมูลที่มีขนาดใหญ่มากๆจะมีอัตราการเพิ่มข้อมูลได้อย่างรวดเร็วมากและจะมีรูปแบบที่ไม่มีโครงสร้างหรือกึ่งโครงสร้างซึ่งไม่สามารถอยู่ในระบบฐานข้อมูลที่จะจัดเก็บข้อมูลได้

2.1.1 คุณลักษณะของบิ๊กดาต้า

คุณลักษณะของบิ๊กดาต้าได้มีการจัดแบ่งออกเป็น 3 ลักษณะ [1] แสดงดังรูปที่ 2.1



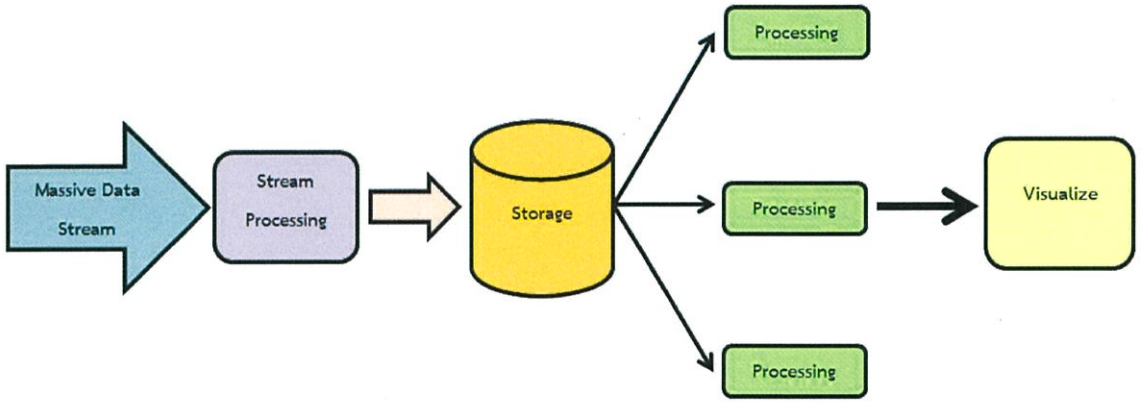
รูปที่ 2.1 คุณลักษณะของบิ๊กดาต้า

1) **ปริมาตร (Volume)** หมายถึง ข้อมูลที่มีปริมาณมหาศาล ซึ่งโครงสร้างข้อมูลของระบบฐานข้อมูลไม่สามารถจัดเก็บข้อมูลได้ ปริมาณข้อมูลมหาศาลมีประโยชน์เพื่อเป็นข้อมูลที่ใช้ในการตัดสินใจ หรือทำนายอนาคต หรือเพื่อเตรียมการวางแผนการทำงานเชิงรุกทางธุรกิจ
2) **ความเร็ว (Velocity)** หมายถึง อัตราการเพิ่มขึ้นของข้อมูลซึ่งข้อมูลที่เข้าสู่ระบบฐานข้อมูลจะมีอัตราการเพิ่มขึ้นอย่างรวดเร็ว เช่น ข้อมูลที่เกิดขึ้นจากโทรศัพท์เคลื่อนที่ที่ผู้ใช้นำขึ้นเก็บเป็นข้อมูลภาพถ่าย ข้อมูลการพิมพ์การสนทนา ข้อมูลการอัดภาพ วิดีโอ หรือข้อมูลการอัดเสียง หรือแม้กระทั่งข้อมูลการสั่งซื้อสินค้า การขนส่ง และการบริการต่างๆ ก็สามารถนำข้อมูลเหล่านั้นเข้าสู่ระบบฐานข้อมูลได้อย่างรวดเร็ว

3) **รูปแบบที่หลากหลาย (Variety)** หมายถึง รูปแบบมีความหลากหลายของรูปแบบข้อมูล ซึ่งอาจจะเป็นรูปแบบที่มีโครงสร้าง ไม่มีโครงสร้าง และกึ่งมีโครงสร้าง เป็นต้น รูปแบบที่ไม่มีโครงสร้างหรือกึ่งโครงสร้างจะไม่เหมือนข้อมูลที่จัดเก็บไว้ในระบบฐานข้อมูล เช่น ข้อความ อีเมล รูปภาพ วิดีโอและเสียง เป็นต้น ซึ่งข้อมูลเหล่านี้มีความซับซ้อน และเชื่อมโยงกัน

2.1.2 การจัดการบิ๊กดาต้าโดยรูปแบบการกระจายข้อมูล

เมื่อขนาดของข้อมูลมีปริมาณที่ใหญ่โตมหาศาลจึงต้องหาวิธีที่สามารถประมวลผลข้อมูลที่มีปริมาณมหาศาลและข้อมูลนั้นไม่สามารถเคลื่อนย้ายได้ซึ่งมีรูปแบบการจัดการข้อมูลที่เป็นบิ๊กดาต้า ซึ่งแสดงดังรูปที่ 2.2 [1]



รูปที่ 2.2 รูปแบบการแสดงการจัดการข้อมูลที่เป็นบิ๊กดาต้า

การจัดการข้อมูลที่เป็นบิ๊กดาต้ามีองค์ประกอบดังต่อไปนี้

1. การจัดเก็บ (Storage)
2. การประมวลผล (Processing)
3. การวิเคราะห์ (Analysis Algorithm)
4. การทำรายงานสรุป (Visualization)

จะมีการเคลื่อนย้ายกลุ่มของข้อมูลเข้าสู่ระบบฐานข้อมูลเพื่อการประมวลผลและนำไปเก็บในส่วนจัดเก็บข้อมูล ซึ่งจะเก็บปริมาณข้อมูลมหาศาลที่ไม่สามารถเคลื่อนย้ายได้ การประมวลผลจะต้องนำข้อมูลมาประมวลผลให้อยู่ในมุมมองต่างๆที่ผู้ใช้ต้องการ ในปัจจุบันได้มีการพัฒนาซอฟต์แวร์ฮาดูปเข้ามาช่วยจัดการปริมาณข้อมูลมหาศาลนี้

2.2 ระบบฮาดูป

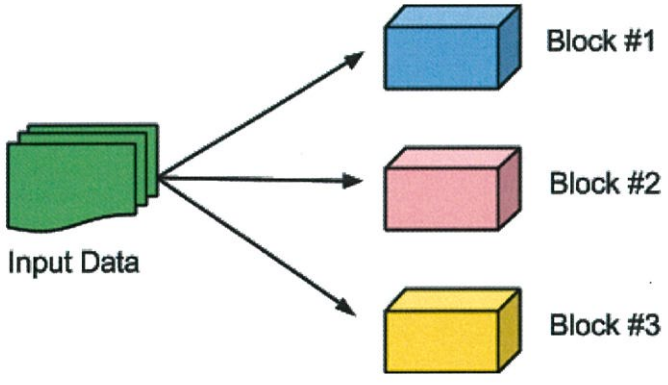
ระบบฮาดูปเป็นซอฟต์แวร์แบบโอเพ่นซอร์ส (Open Source) สำหรับการสร้างระบบการประมวลผลแบบกระจาย (Distributed Computing) โดยอาศัยแนวคิดของ Google File System ใช้สำหรับรันแอปพลิเคชันบนระบบคลัสเตอร์ขนาดใหญ่และสนับสนุนการทำงานแบบขนาน

ฮาดูปสามารถแบ่งออกเป็น 2 ส่วนคือ HDFS (Hadoop Distributed File System) และแมปรีดิวซ์ (MapReduce) ซึ่ง HDFS มีความสามารถในการจัดเก็บข้อมูลขนาดใหญ่แบบกระจาย ส่วนแมปรีดิวซ์ จะเป็นส่วนประมวลผลและวิเคราะห์ข้อมูล โดยทั้ง HDFS และแมปรีดิวซ์ได้ถูกออกแบบมาเพื่อให้เฟรมเวิร์คสามารถจัดการกับโหนดที่ทำงานผิดพลาดให้สามารถทำงานต่อได้โดยอัตโนมัติ

2.3 ระบบจัดการไฟล์แบบกระจายของฮาดูป (Hadoop Distributed File System)

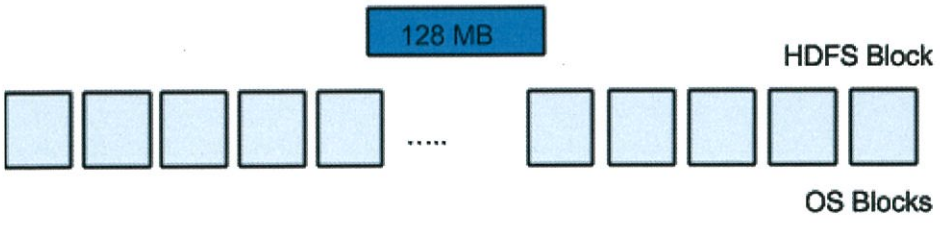
ระบบจัดการไฟล์แบบกระจายของฮาดูป (HDFS: Hadoop Distributed File System) [2] คือส่วนจัดเก็บข้อมูลหลักเป็นระบบแฟ้มข้อมูลแบบกระจายซึ่งระบบจัดการไฟล์แบบกระจายของฮาดูปจะสร้างแบบจำลองเป็นบล็อกข้อมูลบนคลัสเตอร์กระจายข้อมูลที่ต้องการเก็บไว้ในไฟล์บนคอมพิวเตอร์หลายๆเครื่อง (คอมพิวเตอร์แต่ละเครื่องเรียกว่าโหนด) โดยบล็อกไฟล์หนึ่งไฟล์อาจถูกแยกออกเป็นหลายส่วนแต่ละส่วนมีขนาดเท่าๆกัน แล้วเก็บไว้ต่างโหนดกันทำให้ได้กลุ่มโหนดที่ถูกมองเป็นฮาร์ดดิสก์หนึ่งลูกที่มีขนาดใหญ่มาก โดยขนาดของฮาร์ดิสก์สามารถเพิ่มได้ไม่จำกัด การเพิ่มขนาดฮาร์ดดิสก์ไม่กระทบข้อมูลเดิม และเมื่อโหนดใดโหนดหนึ่งเสียระบบไฟล์ทั้งหมดยังสามารถทำงานได้ จึงทำให้มีความน่าเชื่อถือของระบบสูง

โดยระบบจัดการไฟล์แบบกระจายของฮาดูปจะทำการสำรองข้อมูลบนโหนดและเตรียมแบนด์วิดท์ไว้สำหรับการส่งข้อมูลข้ามคลัสเตอร์และเก็บไฟล์ต่างๆเป็นบล็อก (Block) [3] ดังรูปที่ 2.3



รูปที่ 2.3 การแบ่งไฟล์เป็นบล็อกของระบบจัดการไฟล์แบบกระจายของฮาดูป

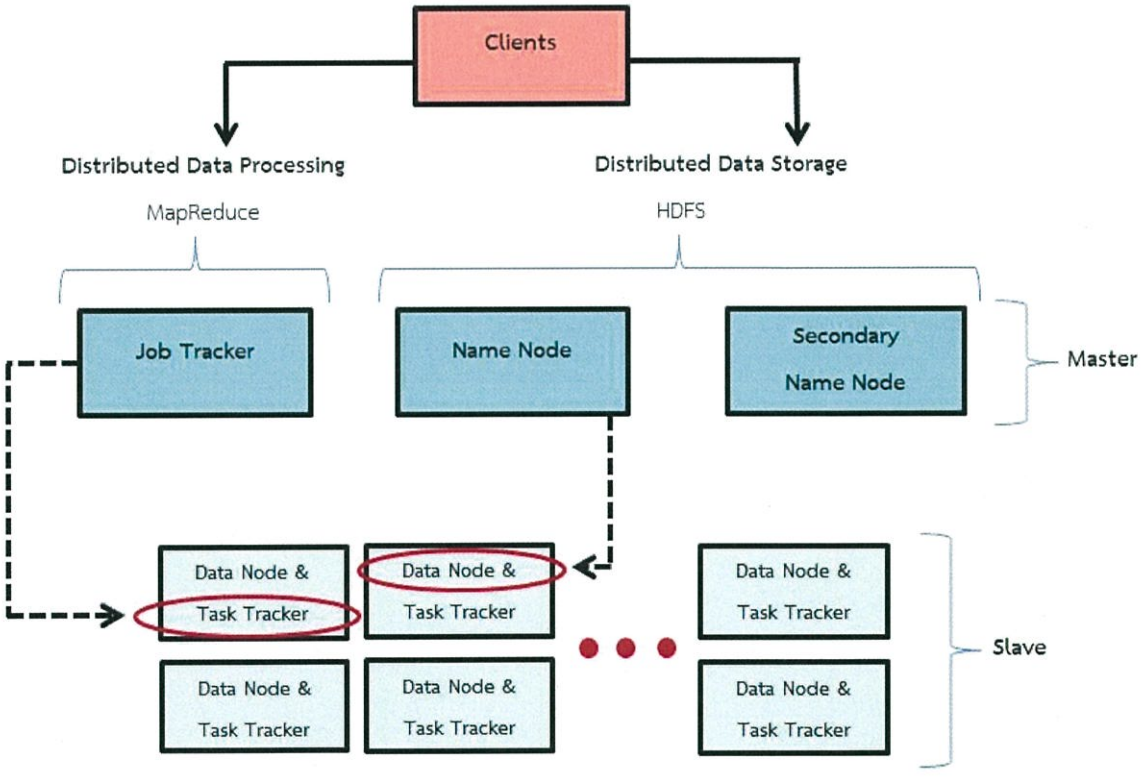
ขนาดบล็อกไฟล์ของระบบจัดการไฟล์แบบกระจายของฮาดูปโดยปกติแล้วจะมีขนาด 64 MB แต่ในระบบไฟล์ของลีนุกซ์ (Linux) จะมีขนาดบล็อก 4KB แต่อาจจะใช้บล็อกไฟล์ของ HDFS ขนาด 128MB จะได้ดังรูปที่ 2.4



รูปที่ 2.4 ไฟล์บล็อกของระบบจัดการไฟล์แบบกระจายของฮาดูป

2.3.1 โครงสร้างของฮาดูป

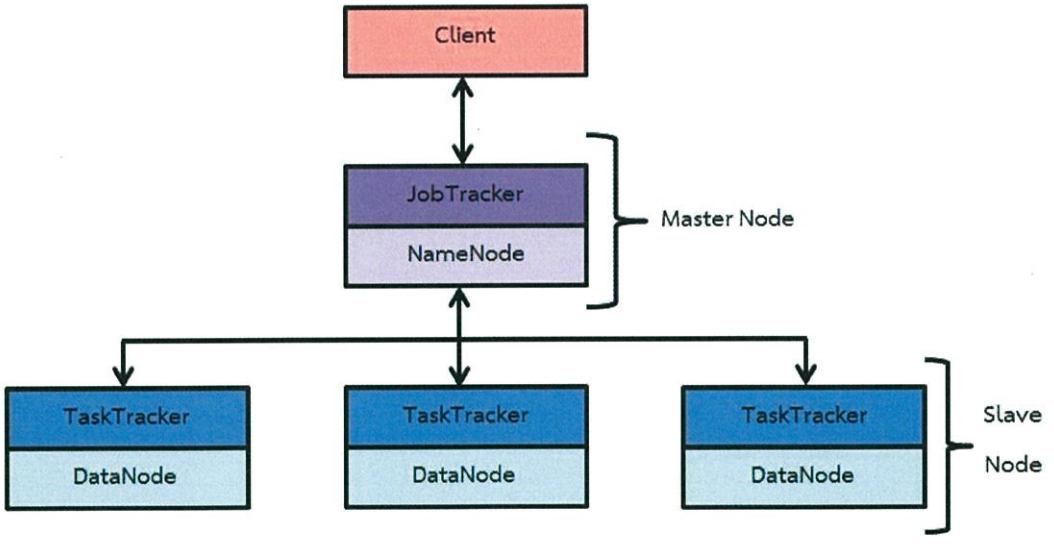
ระบบฮาดูปมีการแบ่งประเภทการทำงานของเครื่องในแต่ละโหลดไว้ 3 ประเภท [3] ได้แก่ เครื่อง Clients, Master Node และ Slave Node ดังรูปที่ 2.5



รูปที่ 2.5 โครงสร้างของฮาดูป

จากรูปที่ 2.5 สามารถอธิบายโครงสร้างของแต่ละส่วนได้ดังนี้

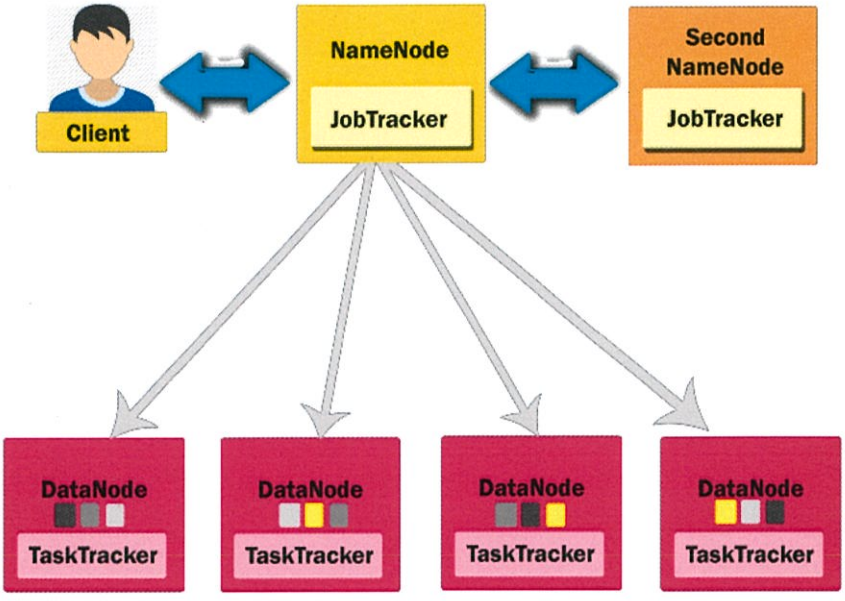
1) **Client Node** จะถูกติดตั้งฮาดูปและตั้งค่าคลัสเตอร์ทั้งหมดไม่ว่าจะเป็นเครื่อง Master หรือ Slaver บทบาทของของเครื่อง Client คือเป็นที่โหลดข้อมูลของคลัสเตอร์และส่งไปให้แมปรีดิวซ์ประมวลผลข้อมูล
2) **Master Node** ดูแลส่วนสำคัญ 2 ส่วนได้แก่ HDFS (Hadoop Distributed File System) เป็นส่วนที่ใช้เก็บข้อมูลขนาดใหญ่ และแมปรีดิวซ์เป็นส่วนประมวลผลข้อมูลทั้งหมดแบบขนาน Master Node ประกอบด้วย Name Node และ Job Tracker แสดงดังรูปที่ 2.6
3) **Slave Node** ทำหน้าที่เก็บข้อมูล คำนวณ และประมวลผล โดย Slave Node แต่ละตัวจะประกอบด้วย Data Node และ Task Tracker แสดงดังรูปที่ 2.6 ซึ่งจะรับคำสั่งมาจาก Master Node โดยที่ Task Tracker เป็น Slave ของ Job Tracker และ Data Node เป็น Slave ของ Name Node



รูปที่ 2.6 ส่วนประกอบต่างๆใน Master Node และ Slave Node

2.3.2 ส่วนประกอบในการทำงานของฮาดูป

ฮาดูปสามารถวิเคราะห์และประมวลผลข้อมูลขนาดใหญ่ได้เร็ว เนื่องจากมีการแบ่งข้อมูลแล้วกระจายงานไปให้เครื่องหลายๆเครื่องช่วยกันประมวลผลข้อมูล [4] ซึ่งส่วนประกอบในการทำงานของฮาดูปจะแสดงในรูปที่ 2.7



รูปที่ 2.7 ส่วนประกอบในการทำงานของฮาดูป

จากรูปที่ 2.7 สามารถอธิบายหน้าที่ในแต่ละส่วนได้ดังนี้

1) **เนมโหนด (Name Node)** ใน HDFS จะประกอบไปด้วย 1 เนมโหนด (Name Node) หรือ 1 มาสเตอร์โหนด (Master Node) ซึ่งมีหน้าที่ดังนี้
   - บริหารจัดการระบบไฟล์ของคลัสเตอร์
   - ควบคุมการเข้าถึงไฟล์จากผู้ใช้หรือ Client
   - บริหารจัดการดาต้าโหนด (Data Node) ต่างๆ
   - ทำหน้าที่ในการทำสำเนาข้อมูลไปยังดาต้าโหนดตัวอื่นๆในคลัสเตอร์
2) **เนมโหนดที่สอง (Secondary Name Node)** เอาไว้สำหรับเป็นจุดเช็กพอยต์ (Check Point) ให้กับเนมโหนดเป็นระยะๆแต่ไม่ใช่ตัวสำรองข้อมูล (Back Up) ของเนมโหนด ซึ่งเนมโหนดที่สองมีไว้แก้ปัญหาในกรณีที่เนมโหนดต้องไปค้นหาข้อมูลโดยตรงจากไฟล์รูปภาพที่มีขนาดใหญ่ (Edit Log) ฮาดูปจึงนำมาแก้ปัญหานี้เพื่อให้เนมโหนดสามารถอ่านเอาข้อมูลล่าสุดไปใช้ได้โดยตรงและรวดเร็ว
3) **ดาต้าโหนด (Data Node)** HDFS จะประกอบไปด้วยหลายดาต้าโหนดหรือหลาย Slaves Node มีหน้าที่ในการจัดเก็บข้อมูลของคลัสเตอร์ซึ่งจะคอยอ่านเขียนข้อมูลตามที่ผู้ใช้ต้องการโดยรับคำสั่งผ่านทางเนมโหนด
4) **Job Tracker** ทำหน้าที่ในการสั่งงาน ควบคุมงาน หรือกระจายงานไปยัง Task Tracker หรือโหนดอื่นๆในคลัสเตอร์ เมื่อได้รับคำสั่งมาจากผู้ใช้หรือ Client โดย Job Tracker จะมีอยู่เฉพาะที่เนมโหนดหรือมาสเตอร์โหนดเท่านั้น

5) **Task Tracker** ทำหน้าที่รันหรือประมวลผลงานตามคำสั่งที่ได้รับมาจาก Job Tracker การประมวลผลที่กล่าวถึงคือ การประมวลผล Map, Combine, Shuffle, Sort, Reduce ของแต่ละโหนด นอกจากนี้ยังทำหน้าที่ในการคอยติดตามงานของแต่ละโหนดและส่งผลลัพธ์กลับไปยัง Job Tracker

2.4 แมปรีดิวซ์ (MapReduce)

แมปรีดิวซ์ [5] เป็นเฟรมเวิร์ก (Framework) ในการเขียนโปรแกรมแบบหนึ่งที่ช่วยในงานประมวลผลที่มีชุดของข้อมูลจำนวนมาก เป็นการทำงานแบบขนานและเหมาะกับการทำงานแบบกระจาย ซึ่งจะอาศัยเครื่องคอมพิวเตอร์หลายๆเครื่องช่วยกันประมวลผลข้อมูล

โปรแกรมแมปรีดิวซ์ประกอบด้วยสองส่วนหลักๆคือ Map เป็นขั้นตอนการประมวลผลข้อมูลที่รับเข้ามา (Input Data) และ Reduce เป็นขั้นตอนการรวบรวมผลเพื่อนำไปเป็นผลลัพธ์สุดท้าย โดยทั่วไปสามารถเขียนโปรแกรมในแมปรีดิวซ์เพื่อคำนวณผลลัพธ์ได้ซึ่งจะแสดงในรูปที่ 2.8

```
Map (String key, String value) :
        // key: document name
        // value: document contents
for each word w in value:
        EmitIntermediate (w, "1");
```

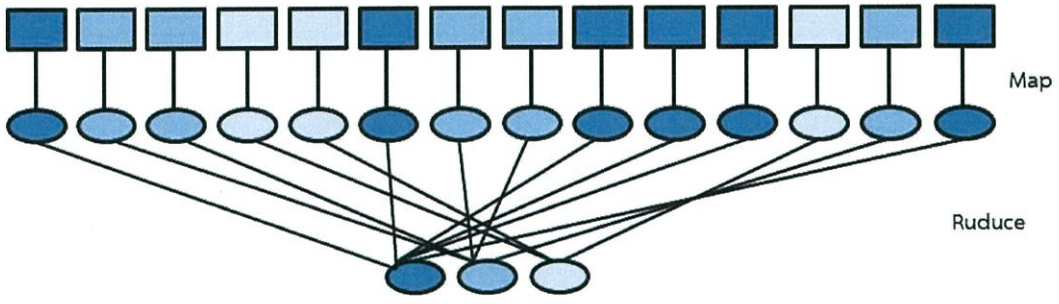
```
Reduce (String key, Iterator values) :
        // key: a word
        // value: a list of counts
        int result = 0;
        for each v in values:
          result += ParseInt(v);
        Emit (AsString (result));
```

รูปที่ 2.8 ตัวอย่างฟังก์ชัน Map และ Reduce

จากรูปที่ 2.8 เป็นตัวอย่างอัลกอริทึมของฟังก์ชัน Map ซึ่งทำหน้าที่รับข้อมูลอินพุตที่เป็นชื่อเอกสาร (Key) และเนื้อหาเอกสาร (Value) เพื่อจัดกลุ่มคำ ส่วนฟังก์ชัน Reduce ทำหน้าที่รับข้อมูลเข้าที่เป็นค่า (Key, Word) และรายการของกลุ่มคำ (List of Counts) เพื่อลดจำนวนผลลัพธ์ ลักษณะการทำงานของฟังก์ชัน Map/Reduce มีรูปแบบการเขียนโปรแกรมให้มีการประมวลผลแบบขนาน (Parallel Programming Model) คือ การนำการประมวลผลมาแบ่งย่อยออกเป็นส่วนๆ ซึ่งแต่ละส่วนสามารถทำงานหรือประมวลผลได้ในเวลาเดียวกัน ดังนั้นหลักการของการเขียนโปรแกรมแบบขนาน (Parallel Programming) จึงสามารถทำงานได้เร็วกว่าและในการทำแมปรีดิวซ์กับข้อมูลก็ใช้รูปแบบการประมวลผลแบบขนานนี้ จึงสามารถจัดการกับความซับซ้อนในการกระจายการประมวลผลให้เกิดความสมดุล

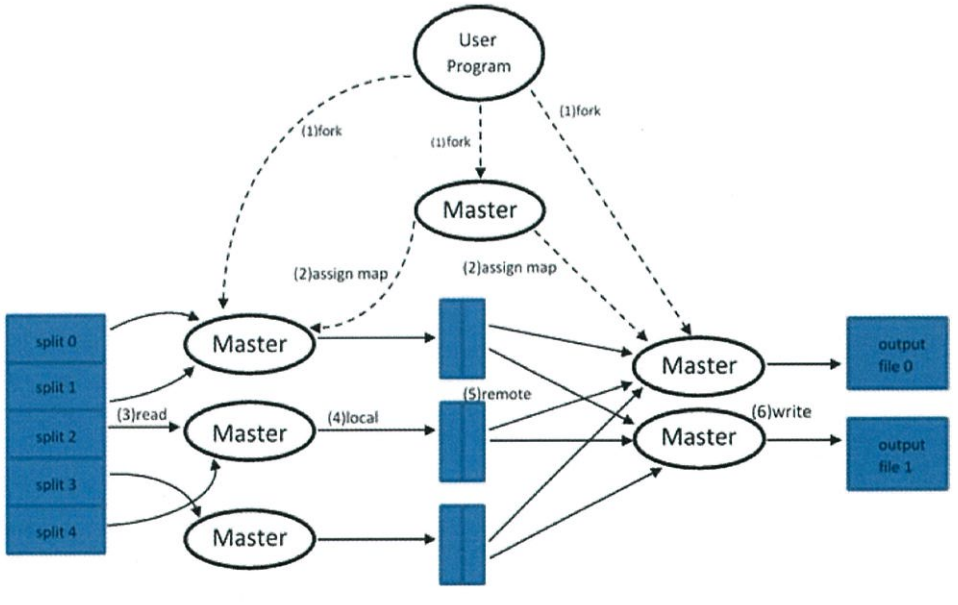
2.4.1 หลักการทำงานของแมปรีดิวซ์

รูปแบบทำงานของแมปรีดิวซ์ [6] คือ จะกระจายงานต่างๆไปให้ Map-Worker ที่อยู่บนแต่ละเครื่องทำงาน ซึ่งผู้ที่ควบคุมการกระจายงานก็คือ Master โดยหลังจากที่ Worker ทำงานเสร็จแล้วก็จะแจ้งให้ Master เพื่อที่ Master จะส่งต่อผลของการ Map ให้กับ Reduce-Worker เพื่อทำงานให้ได้ผลลัพธ์ต่อไปซึ่งมีรูปแบบการทำงานดังรูปที่ 2.9



รูปที่ 2.9 รูปแบบการทำงานของ Map และ Reduce

สามารถอธิบายหลักการทำงานของแมปรีดิวซ์ ได้ดังรูปที่ 2.10



รูปที่ 2.10 หลักการทำงานของแมปรีดิวซ์

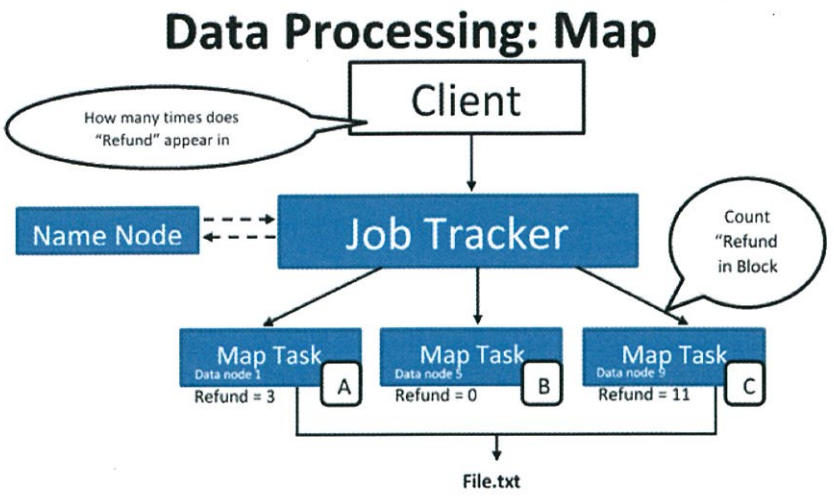
1) เริ่มจากการ Map เมื่อมีการใช้งานตัวไลบรารี่จะแบ่งไฟล์ที่เป็นอินพุตเป็นส่วนย่อยประมาณ 16 MB ถึง 64 MB
2) Master จะดูว่ามี Worker ตัวใดว่างอยู่บ้างก็จะกำหนด Task ให้ ซึ่งจะมีทั้ง Map และ Reduce Task

3) สำหรับ Worker ที่ถูกกำหนด Map Task ก็จะอ่านข้อมูลอินพุตไฟล์ที่ถูกแบ่ง แล้ว Map Key/Value ตามที่ผู้ใช้งานเขียนไว้
4) ผลของการ Map จะถูกเก็บไว้ที่เครื่องที่ทำการ Map ซึ่งจะมีการแบ่งเป็นส่วนๆตามคีย์ที่กำหนด โดยฟังก์ชันการแบ่งของแมปรีดิวซ์ไลบรารี่จะมีการส่งข้อความไปยัง Master เพื่อให้ Master ส่งงานให้ Reduce Worker ต่อไป
5) เมื่อ Reduce Worker ได้รับข้อความจาก Master ก็จะทำอ่านให้ไฟล์จากตำแหน่งที่ Master ให้มา แล้วทำการจัดเรียงข้อมูลตามคีย์
6) เมื่อเรียงข้อมูลตามคีย์เสร็จแล้ว ก็จะทำตาม Reduce Function ตามที่ผู้ใช้งานกำหนดไว้ โดยเอาต์พุตที่ออกมาจะขึ้นอยู่กับจำนวนของ Reduce Worker แล้วสุดท้ายจะทำการรวมไฟล์เอาต์พุตเป็นไฟล์เดียว
7) เมื่อแมปรีดิวซ์เสร็จแล้วก็จะแจ้งให้ Master รู้ เพื่อแจ้งต่อไปยังโปรแกรมว่าทำแมปรีดิวซ์เสร็จแล้ว

2.4.2 ขั้นตอนการแมป

ในขณะนี้ File.txt ถูกกระจายอยู่ในคลัสเตอร์ซึ่งการประมวลผลแบบขนานที่ถูกรวมอยู่ในฮาดูป เรียกว่า แมปรีดิวซ์ เป็นสองขั้นตอนที่สำคัญคือ Map และ Reduce ในขั้นตอนแรก คือ กระบวนการ Map โดยจะให้เครื่องทั้งหมดคำนวณข้อมูลในบล็อก (Local Block Data) โดยในกรณีนี้ เช่น จะให้นับคำว่า Refund ในบล็อกข้อมูลของ File.txt เพื่อเริ่มต้นกระบวนการเครื่องลูกข่าย (Client) จะส่ง MapReduce Job ไปที่ Job Tracker โดยถามคำว่า Refund ได้ปรากฏกี่ครั้งใน File.txt โดยใช้ Java Code, Job Tracker จะทำงานร่วมกับ Name Node เพื่อถามว่า Data Node ไหนที่มี Block ของ File.txt

จากนั้น Job Tracker จะให้ Task Tracker ดำเนินการกับโหนดที่มีโปรแกรมภาษาจาวาประมวลผล Map บนข้อมูลระดับท้องถิ่น (Local Data) ต่อมา Task Tracker เริ่ม Map Task และติดตามการประมวลผล Task Tracker จะส่งติดตามสัญญาณและสถานะกลับไปให้ Job Tracker เพื่อจะได้รู้ว่า Task ยังทำงานอยู่หรือไม่ เมื่อแต่ละ Map Task ทำงานเสร็จ แต่ละโหนดจะเก็บผลลัพธ์จากการประมวลผลข้อมูลในที่เก็บข้อมูลระดับท้องถิ่น (Local) ซึ่งเรียกว่า ข้อมูลระดับกลาง (Intermediate Data) ต่อไปจะทำการส่งข้อมูลระดับกลาง (Intermediate Data) ผ่านไปทางเน็ตเวิร์กเพื่อไปประมวลผล Reduce Task สำหรับการประมวลผลครั้งสุดท้ายดังรูปที่ 2.11

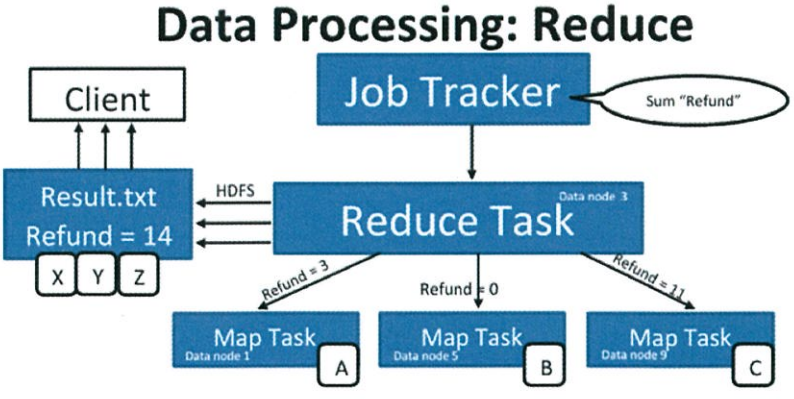


รูปที่ 2.11 กระบวนการ Map

2.4.3 ขั้นตอนการรีดิวซ์

ขั้นที่สองของแมปรีดิวซ์เรียกว่า Reduce เมื่อ Map Task ทำงานเสร็จและเก็บข้อมูลเป็นข้อมูลระดับกลาง (Intermediate Data) แล้ว ดังนั้นขั้นตอนถัดมาคือการรวบรวมข้อมูลระดับกลาง (Intermediate Data) เพื่อสกัดข้อมูลสำหรับการประมวลผลต่อเพื่อให้ได้ผลลัพธ์สุดท้าย

Job Tracker เริ่ม Reduce Task บนโหนดใดโหนดหนึ่งในคลัสเตอร์และคำสั่ง Reduce Task ไปนำข้อมูลจากข้อมูลระดับกลาง (Intermediate Data) ซึ่งถูกประมวลผลโดย Map Task และ Map Task จะตอบสนองต่อ Reducer เกือบจะพร้อมกัน Reduce Task จะได้รวบรวมข้อมูลระดับกลาง (Intermediate Data) ทั้งหมดจาก Map Task และสามารถเริ่มการประมวลผลครั้งสุดท้ายได้ เช่น การหาผลรวมของคำว่า Refund แล้วเขียนผลลัพธ์ลง Result.txt เอาต์พุตจาก Job ที่เรียกว่า Result.txt จะถูกเขียนไปที่ HDFS และเครื่อง Client ก็จะสามารถอ่าน Result.txt จาก HDFS และ Job ก็จะได้ผลลัพธ์ดังรูปที่ 2.12



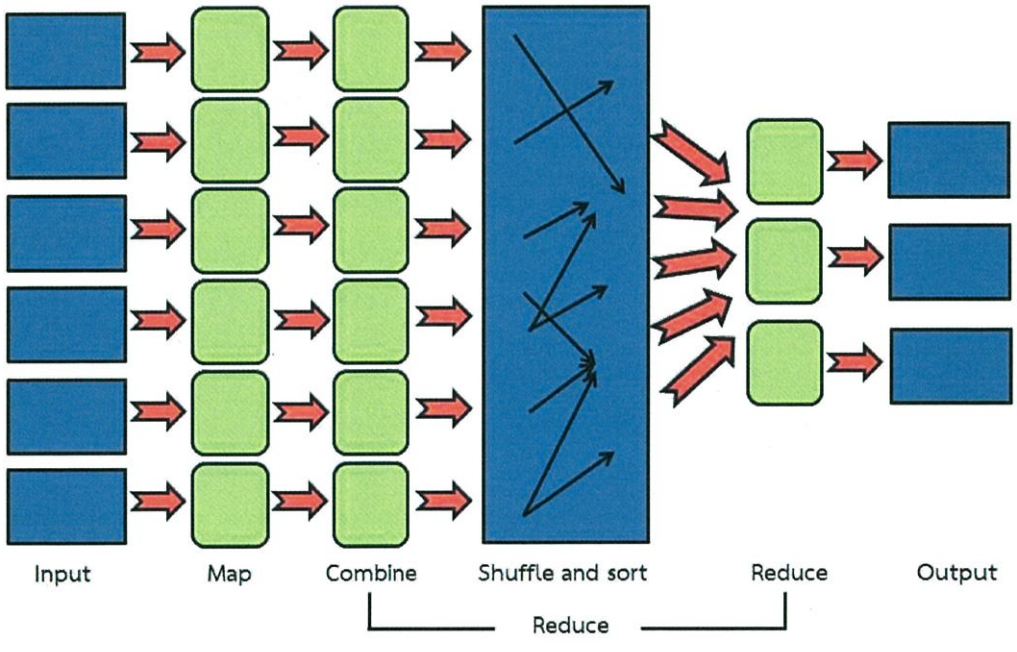
รูปที่ 2.12 กระบวนการ Reduce

2.4.4 ตัวอย่างการทำงานของแมปรีดิวซ์

แมปรีดิวซ์เป็นกระบวนการที่ใช้สำหรับการแบ่งข้อมูลที่นำเข้า (Input Data) ให้มีขนาดเล็กลง แล้วส่งไปประมวลผลยังโหนดอื่นๆที่อยู่ในคลัสเตอร์เมื่อประมวลเสร็จแล้วจึงนำผลลัพธ์ที่ได้กลับมาลดขนาดได้เป็นผลลัพธ์ (Output Data) แล้วค่อยส่งข้อมูลนั้นกลับออกไป โดยสามารถสรุปการทำงานของแมปรีดิวซ์ ได้ดังนี้ [5]

1) แต่ละโหนดจะทำการแปลงข้อมูลนำเข้า (Input Data) ที่ถูกแบ่งแล้วให้อยู่ในรูปของ Key-Value (Map) ตาม Mapper Class ที่ได้เขียนเอาไว้
2) จากนั้นจึงทำการยุบรวม (ผสม) Key Value ใน Combiner Class
3) เมื่อยุบรวม Key Value ที่ Combiner เสร็จแล้ว ก็จะทำการจัดเรียง หรือ Sort Key Value ใหม่
4) แล้วนำผลลัพธ์ที่ได้กลับมาลดขนาด (Reduce) ตามคำสั่งที่ถูกเขียนไว้ใน Reducer Class
5) จากนั้นจึงค่อยส่งผลลัพธ์ (Output Data) ที่ได้กลับออกไป

แผนภาพการทำงานของแมปรีดิวซ์แสดงดังรูปที่ 2.13

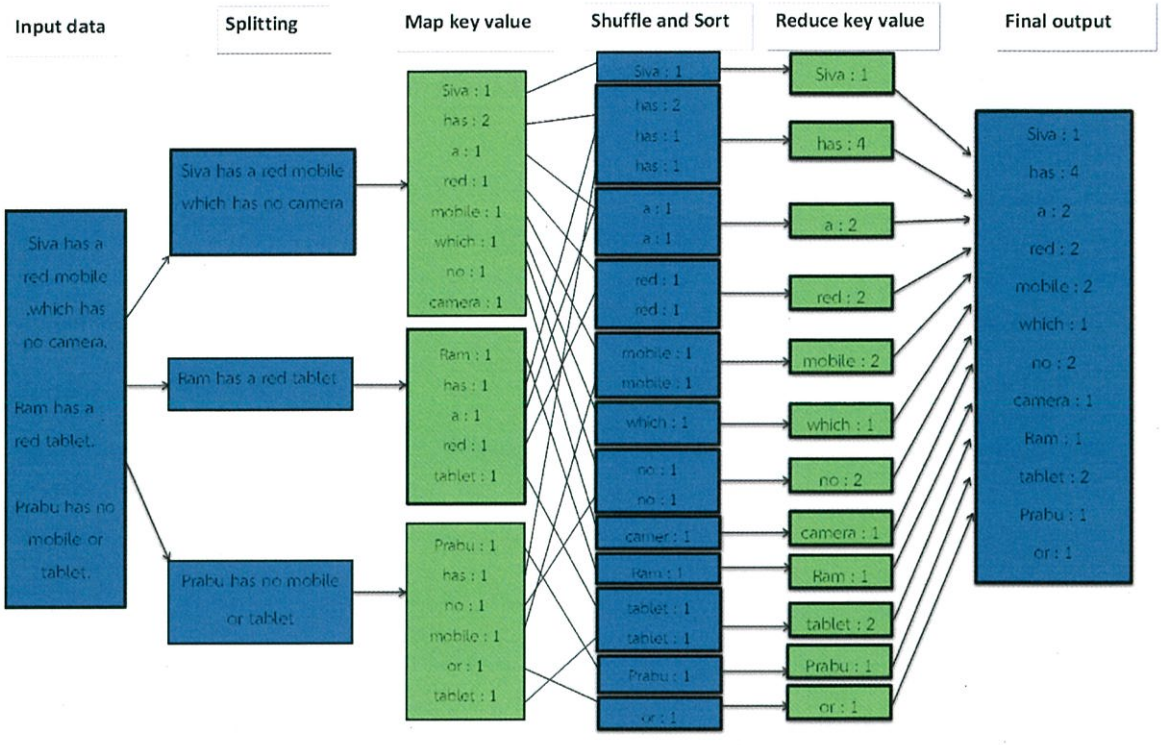


รูปที่ 2.13 แผนภาพการทำงานของแมปรีดิวซ์

จากรูปที่ 2.13 สามารถสรุปแผนภาพการทำงานของแมปรีดิวซ์ได้ดังนี้

Input Data --> Mapper --> Combiner --> Shuffle and Sort --> Reducer --> Output Data

ตัวอย่างการนับจำนวนครั้งในแต่ละคำที่ปรากฏในข้อมูล เช่น Siva has a red mobile, which has no camera. Ram has a red tablet. Prabu has no mobile or tablet. มีขั้นตอนการทำงานของแมปรีดิวซ์ ดังรูปที่ 2.14



รูปที่ 2.14 ตัวอย่างการนับคำของข้อมูลด้วยวิธีแมปรีดิวซ์

2.5 IBM Bluemix

IBM Bluemix [7] เป็นระบบที่ให้บริการประมวลผลบนกลุ่มเมฆของ IBM ซึ่งเน้นการให้บริการแพลตฟอร์มสำหรับการพัฒนาแอปพลิเคชัน (Platform as a Service : PaaS) โดยใช้แพลตฟอร์มสำหรับการประมวลผลแบบกลุ่มเมฆซึ่งให้บริการโดย VMware Bluemix สามารถใช้งานได้ทั้งในรูปแบบส่วนต่อประสานแบบชุดคำสั่ง (Command Line Interfaces or CLI) ผ่านส่วนต่อประสานแบบชุดคำสั่งสำหรับการประมวลผลแบบกลุ่มเมฆ (Cloud Foundry Cli) ปกติ และการทำงานผ่านหน้าเว็บไซต์ Bluemix โดยตรงแพลตฟอร์มที่ให้บริการใน Bluemix มี 4 แพลตฟอร์มหลักคือ Java, Node.js, Ruby on Rails และ Ruby Sinatra นอกจากนี้ยังเป็นเซอร์วิสซึ่งสามารถเพิ่มเข้ามาได้ในภายหลังได้ และยังมีเทมเพลตสำหรับการเริ่มต้นทำโปรเจคเว็บไซต์หรือเว็บแอปพลิเคชัน ที่มีการเตรียมไฟล์ต่างๆให้อย่างครบครันซึ่งจะทำให้มีการทำงานที่เร็วขึ้น แข็งแกร่ง ยืดหยุ่น และสมบูรณ์แบบมากขึ้น (Boilerplates) โดยรวมเอารันไทม์และเซอร์วิสเข้ามาให้บริการร่วมกันสร้างเป็นบริการเฉพาะทาง เช่น Internet of Things Platform, Mobile Cloud, Node Cached Starter และ Big Data เป็นต้น

2.6 Servlet

Servlet [8] เป็นแอปพลิเคชันที่ทำงานอยู่บนเครื่องแม่ข่าย (Server Side Application) ซึ่งมีแนวคิดมาจาก CGI (Common Gateway Interfaces) ข้อดีของ Servlet คือภาษาที่ใช้เขียนด้วยภาษาจาวาซึ่งใช้แนวคิดของการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ซึ่งสามารถลดความซับซ้อนของโครงสร้างโปรแกรม รวมไปถึงการอำนวยความสะดวกในการนำกลับมาใช้ใหม่ นอกจากนี้จาวายังเป็นภาษาที่ไม่ขึ้นกับแพลตฟอร์ม (Platform Independent) ซึ่งจะช่วยให้สามารถพัฒนาระบบโดยใช้สภาวะแวดล้อมใดก็ได้ ซึ่งโดยทั่วไปมักนิยมใช้กับระบบปฏิบัติการวินโดว์ โดยจะนำโปรแกรมที่เขียนเสร็จแล้วมารันบนระบบปฏิบัติการยูนิกซ์ เพื่อเพิ่มความเสถียรภาพของโปรแกรมแทน

นอกจากนี้ Servlet ยังมีความเร็วที่สูงกว่า CGI เพราะใช้หลักการของเทรด (Thread) โดยจะทำการสร้าง 1 เทรดต่อหนึ่งคำสั่ง (Request) ที่มาจากเครื่องลูกข่าย ซึ่งในทางกลับกัน CGI จะทำการสร้าง 1 Process ต่อหนึ่ง Request ซึ่งจะทำให้เปลืองทรัพยากรมากกว่าและในการรันก็จะช้ากว่าด้วย จุดเด่นที่สำคัญของ Servlet คือ API (Application Programming Interface) โดยระบบที่ทำการพัฒนาโดยใช้แนวคิดของ Servlet จะสามารถเรียกใช้ API ที่ทางจาวามีให้ (javax.servlet.*, javax.servlet.http.*) ซึ่งจะช่วยทำให้การพัฒนาระบบดังกล่าวง่ายและเร็วยิ่งขึ้น

2.7 JSP (Java Server Pages)

JSP [9] คือ ภาษาสคริปต์ (Script Language) ที่ทำงานบนเครื่องแม่ข่ายเช่นเดียวกับ Perl, Php, Asp หรือ Cold Fusion เป็นต้น โดยมีโครงสร้างภาษาแบบจาวาหรือเป็นจาวาประเภทหนึ่ง แต่มาเขียนให้อยู่ในรูปของ HTML แต่ผลสุดท้ายเมื่อจะใช้งานจริงตัวไฟล์ JSP จะถูกแปลงให้เป็นไฟล์ของจาวา

JSP ย่อมาจาก Java Server Pages เทคโนโลยีที่คิดค้นโดยบริษัท Sun Microsystems (ผู้ผลิตคอมพิวเตอร์ซันและผู้พัฒนาจาวา) โดยพัฒนาบนพื้นฐานของภาษาจาวาเพื่อเพิ่มประสิทธิภาพให้หน้าเว็บเพจมีความยืดหยุ่นสูงขึ้น โครงสร้างของ JSP นั้นเป็นลักษณะของแท็ก (Tag) ชนิดพิเศษที่แทรกเข้าไปในเอกสาร HTML และเปลี่ยนนามสกุลของเอกสารเป็น .JSP แทนที่จะเป็น .HTM หรือ .HTML โดยแท็กเหล่านี้เว็บเบราว์เซอร์จะไม่สามารถตีความหมายได้ จะต้องนำไปประมวลผลก่อนที่เว็บเซิร์ฟเวอร์เท่านั้น แล้วนำผลลัพธ์ทั้งหมดส่งกลับมายังเว็บเบราว์เซอร์ในลักษณะของเอกสาร HTML ซึ่งเว็บเบราว์เซอร์สามารถตีความหมายและนำมาแสดงผลได้ การทำงานโดยรวมของ JSP จะเริ่มจากเบราว์เซอร์ร้องขอ (HTTP Request) เอกสารที่มีนามสกุลเป็น JSP ไปยังเว็บเซิร์ฟเวอร์ผ่านทางโปรโตคอล HTTP เว็บเซิร์ฟเวอร์ก็จะนำเอกสาร JSP ที่ได้รับมานั้นส่งต่อไปให้ JSP Engine ซึ่งเป็นแอปพลิเคชันที่ถูกโหลดสู่หน่วยความจำและทำงานอยู่บนเว็บเซิร์ฟเวอร์ ทำหน้าที่หลักคือแปลความหมายและประมวลผลเอกสาร JSP จากนั้น JSP Engine ก็จะประมวลผล และส่งผลลัพธ์กลับมา

ยังเว็บเซิร์ฟเวอร์ แล้วเว็บเซิร์ฟเวอร์ก็จะส่งผลลัพธ์กลับมายังเว็บเบราว์เซอร์ (HTTP Response) อีกครั้งในลักษณะของเอกสาร HTML

2.8 ภาษากรูวี (Groovy)

ภาษากรูวี (Groovy) [10] เป็นภาษากึ่งสคริปต์ที่ต่อยอดมาจากภาษาจาวา คือ เป็นภาษาที่เขียนแบบเต็มรูปเหมือนจาวาหรือเขียนแบบสคริปต์ ส่วนที่ต่อยอดมาจากภาษาจาวาเพราะภาษากรูวีสามารถใช้ API Library ของภาษาจาวาได้เต็มที่ 100% ซึ่งเป็นจุดแข็งของภาษาจาวาเพราะไลบรารี่ของภาษาจาวามีความหลากหลายมากและสามารถคอมไพล์ภาษากรูวีไปเป็นไบต์โค้ด (Byte Code) ซึ่งเป็นผลลัพธ์ที่ได้จากการแปลงโค้ดต้นแบบของภาษาจาวาให้เป็นภาษากลางที่เรียกว่า Binary File หรือ Byte Code ของภาษาจาวาได้ นอกจากนี้ก็ยังสามารถใช้จาวาประมวลผลคำสั่งกรูวีที่คอมไพล์แล้วได้โดยต้องติดตั้งกรูวีเข้ามาในระบบ ตัวอย่างภาษาจาวากับภาษากรูวีแสดงดังรูปที่ 2.15 และ 2.16 [11]

ภาษาจาวามาตรฐาน (Java 5 และสูงกว่า)

```
class Filter {
   public static void main (String[] args) {
      List<String> list = Arrays.asList ("Rod", "Carlos", "Chris") ;
      List<String> shorts = new ArrayList<String> () ;
      for (String item : list) {
         if (item.length () <= 4) { shorts.add (item) ; }
      }
      for (String item : shorts) { System.out.println (item) ; }
   }
}
```

รูปที่ 2.15 ตัวอย่างโปรแกรมภาษาจาวา

ภาษากรูวี

```
list = ["Rod", "Carlos", "Chris"]
shorts = list.findAll { it.size () <= 4 }
shorts.each { println it }
```

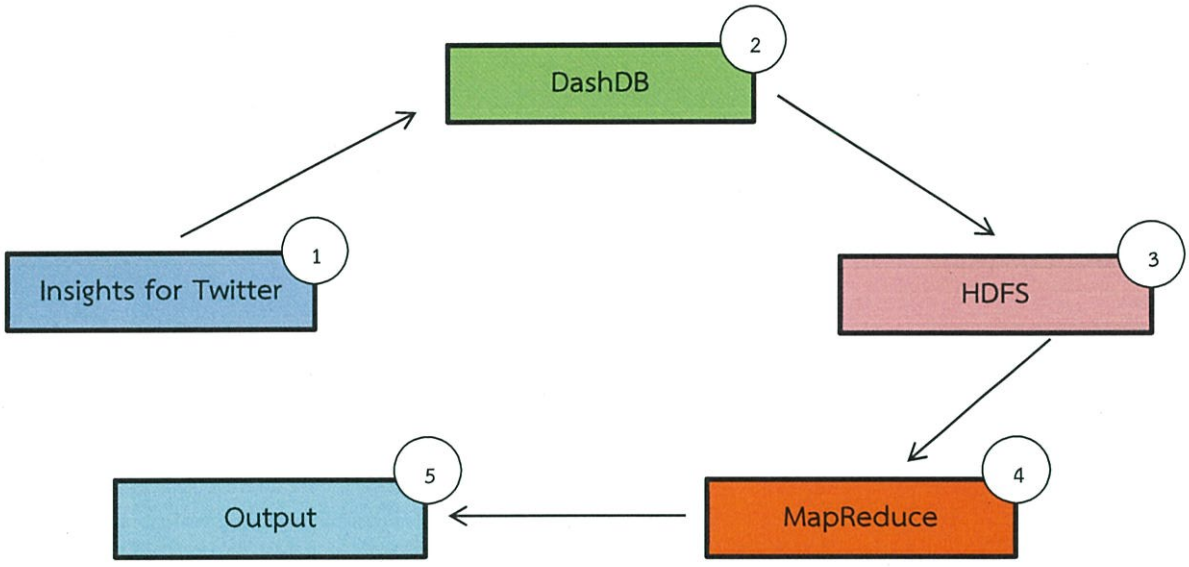
รูปที่ 2.16 ตัวอย่างโปรแกรมภาษากรูวี

บทที่ 3
วิธีการดำเนินงาน

3.1 สถาปัตยกรรมของระบบ

3.1.1 กระบวนการทำงานของระบบ

กระบวนการทำงานของเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์แสดงดังรูปที่ 3.1



รูปที่ 3.1 กระบวนการทำงานของระบบ

จากรูปที่ 3.1 สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

1) Insights for Twitter

ใช้กรองคำที่ต้องการจากทวิตเตอร์ ซึ่งคำที่กรองมาจะสุ่มตัวอย่างจากทวิตเตอร์ทั้งหมด 10% และสามารถเข้าถึงข้อความจากทวิตเตอร์ได้ 100% ข้อมูลที่กรองมาจากทวิตเตอร์จะถูกจัดเก็บและทำดัชนีข้อมูลอยู่ตลอดเวลา ซึ่งช่วยให้การค้นหามีประสิทธิภาพและมีความรวดเร็ว นอกจากนี้ Insights for Twitter ยังมีบริการในส่วนของการวิเคราะห์อารมณ์และความรู้สึกจากข้อความ (Sentiment Analytics) เช่น ความรู้สึกด้านบวก ด้านลบ เป็นกลาง หรือคลุมเครือ

2) DashDB

เป็นระบบฐานข้อมูลของ IBM DB2 ที่มีประสิทธิภาพสูงในการให้บริการเกี่ยวกับคลังข้อมูล (Data Warehouse) ในระบบการประมวลผลแบบกลุ่มเมฆ (Cloud Computing) โดยจะจัดเก็บเป็นข้อมูลเชิงสัมพันธ์ (Relational Data) ซึ่งทำให้การวิเคราะห์ข้อมูลมีประสิทธิภาพและมีความรวดเร็วมากยิ่งขึ้น

3) Hadoop Distributed File System (HDFS)

ทำหน้าเป็นระบบจัดเก็บข้อมูลหลักที่ใช้ในซอฟต์แวร์ฮาดูป ซึ่ง HDFS จะสร้างแบบจำลองเป็นบล็อกข้อมูลบนคลัสเตอร์เพื่อให้เกิดความน่าเชื่อถือและการคำนวณผลที่รวดเร็ว

4) MapReduce

เป็นการเขียนโปรแกรมแบบหนึ่งที่ทำหน้าที่ประมวลผลที่มีชุดของข้อมูลจำนวนมาก เป็นการทำงานแบบขนาน ซึ่งจะอาศัยเครื่องคอมพิวเตอร์หลายๆเครื่องทำงานร่วมกันแบ่งส่วนการทำงานเป็น 2 ขั้นตอน คือ ขั้นตอน Map และ ขั้นตอน Reduce โดยแต่ละขั้นตอนจะมีคู่ Key-Value เป็นข้อมูลเข้า (Input) และข้อมูลออก (Output) ซึ่งผู้ใช้จะกำหนดเอง

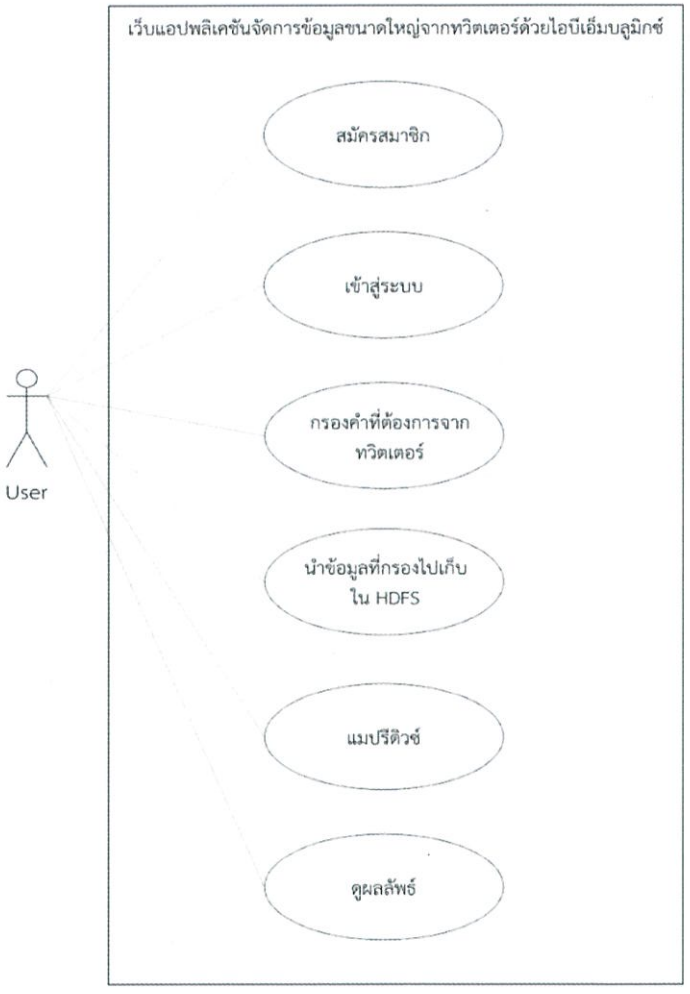
5) Output

แสดงผลลัพธ์ออกมาในรูปแบบของกราฟซึ่งสามารถนำข้อมูลเหล่านี้มาวิเคราะห์และเปรียบเทียบในเชิงธุรกิจหรือด้านงานวิจัยทางด้าน Data Science

3.2 ความสามารถของระบบ

3.2.1 แผนภาพยูสเคส (Use Case Diagram)

แผนภาพยูสเคสแสดงความสามารถของเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์แสดงดังรูปที่ 3.2



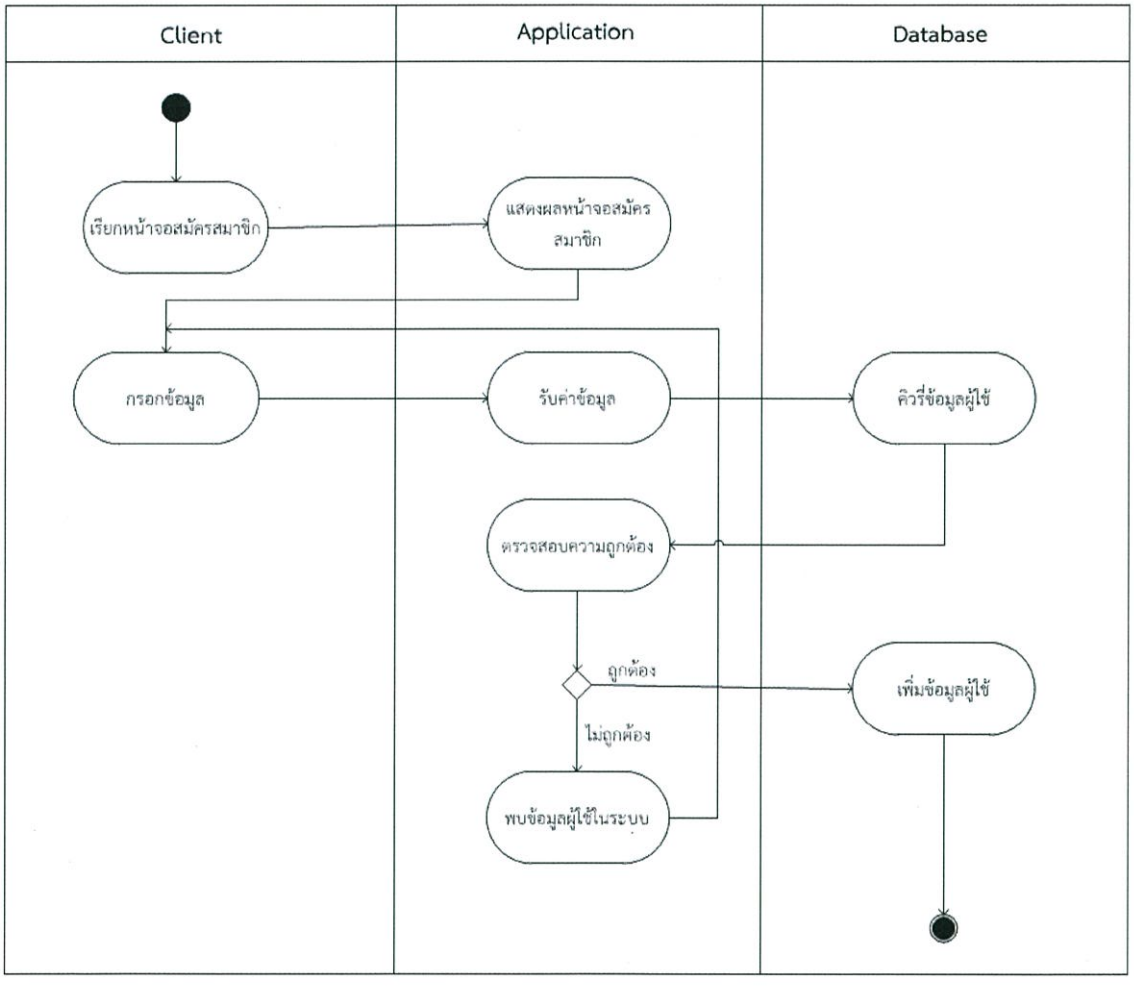
รูปที่ 3.2 Use Case Diagram ของเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์

3.2.2 แผนภาพแอคทิวิตี้ไดอะแกรม (Activity Diagram)

เป็นแผนภาพแสดงขั้นตอนการทำงาน Use Case แต่ละขั้นตอนของเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์มีรายละเอียดดังนี้

1) การสมัครสมาชิก

แผนภาพ Activity Diagram แสดงขั้นตอนการทำงานของเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์ในขั้นตอนการสมัครสมาชิกแสดงดังรูปที่ 3.3

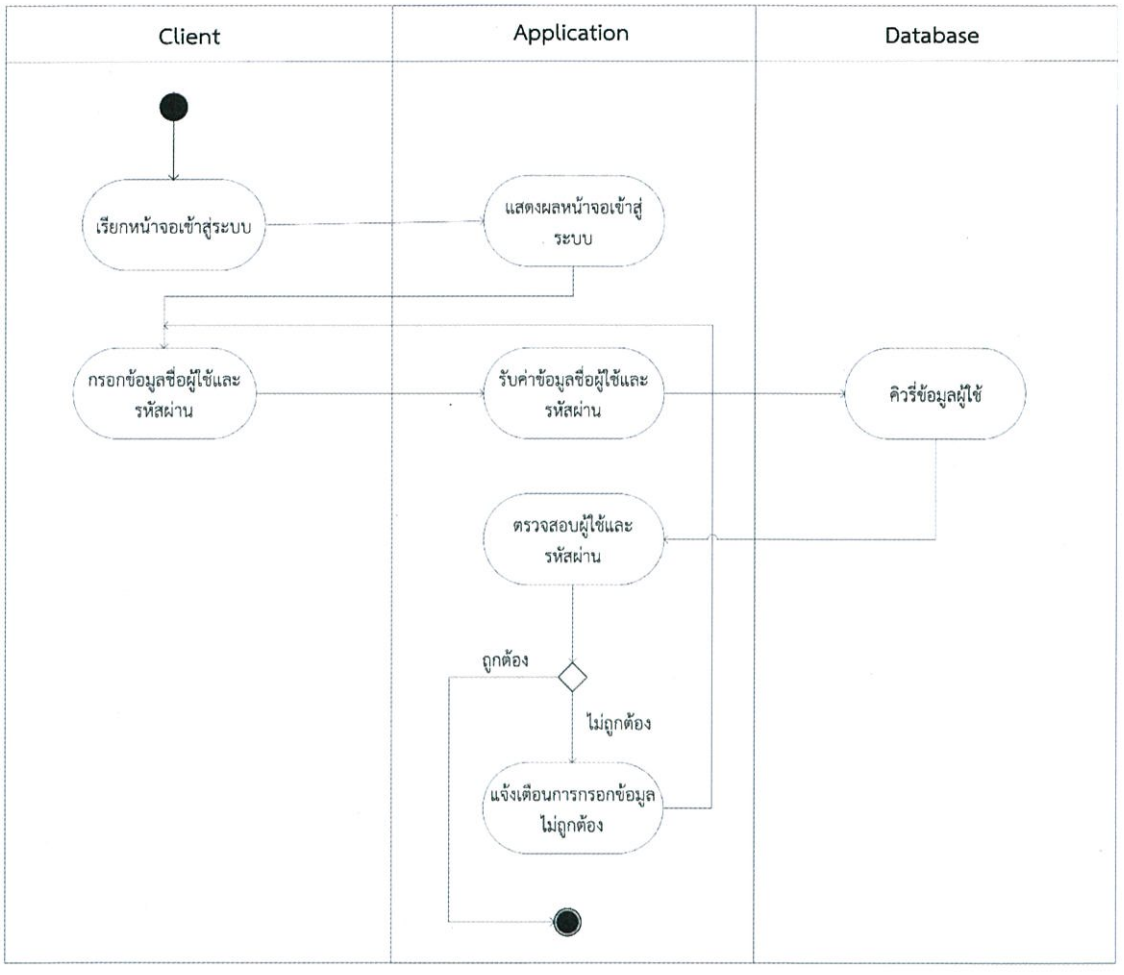


รูปที่ 3.3 แผนภาพ Activity Diagram ของการสมัครสมาชิก

จากรูปที่ 3.3 แสดงถึงการสมัครสมาชิกเพื่อเข้าสู่ระบบ โดยผู้ใช้จะต้องกรอกข้อมูลในหน้าเว็บไซต์ให้ถูกต้อง จากนั้นระบบจะค้นหาข้อมูลผู้ใช้ที่มีอยู่เดิมเพื่อไม่ให้ชื่อผู้ใช้ซ้ำกัน หากระบบตรวจสอบเรียบร้อยแล้วก็จะเพิ่มข้อมูลผู้ใช้เข้าสู่ฐานข้อมูล

2) การเข้าสู่ระบบ

แผนภาพ Activity Diagram แสดงขั้นตอนการทำงานของเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์ในขั้นตอนการเข้าสู่ระบบแสดงดังรูปที่ 3.4

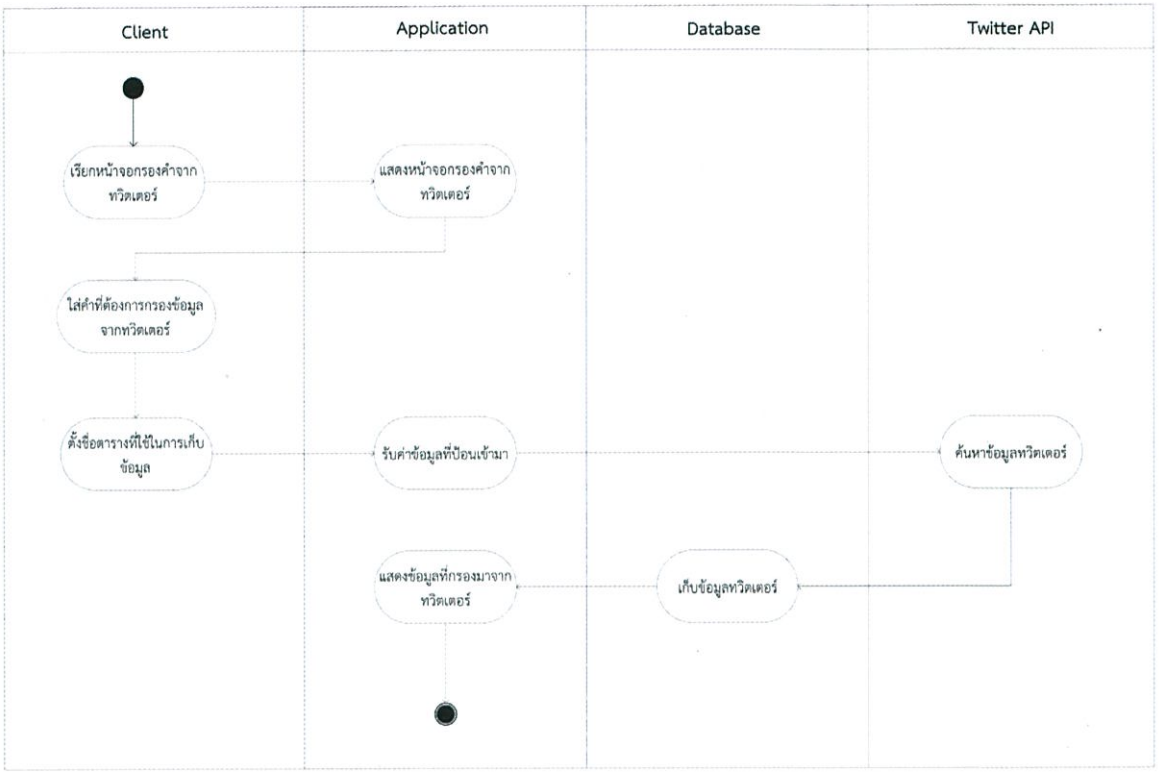


รูปที่ 3.4 แผนภาพ Activity Diagram ของการเข้าสู่ระบบ

จากรูปที่ 3.4 แสดงถึงการเข้าสู่ระบบของผู้ใช้ โดยผู้ใช้จะต้องกรอกข้อมูลชื่อผู้ใช้และรหัสผ่านในหน้าเว็บไซต์ จากนั้นระบบจะค้นหาข้อมูลผู้ใช้จากฐานข้อมูลเพื่อตรวจสอบชื่อผู้ใช้และรหัสผ่าน หากผู้ใช้กรอกข้อมูลไม่ถูกต้อง ระบบจะให้ผู้ใช้งานกรอกข้อมูลใหม่อีกครั้ง

3) ฟังก์ชันกรองคำจากทวิตเตอร์

แผนภาพ Activity Diagram แสดงขั้นตอนการทำงานของเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์ในฟังก์ชันของการกรองคำจากทวิตเตอร์แสดงดังรูปที่ 3.5

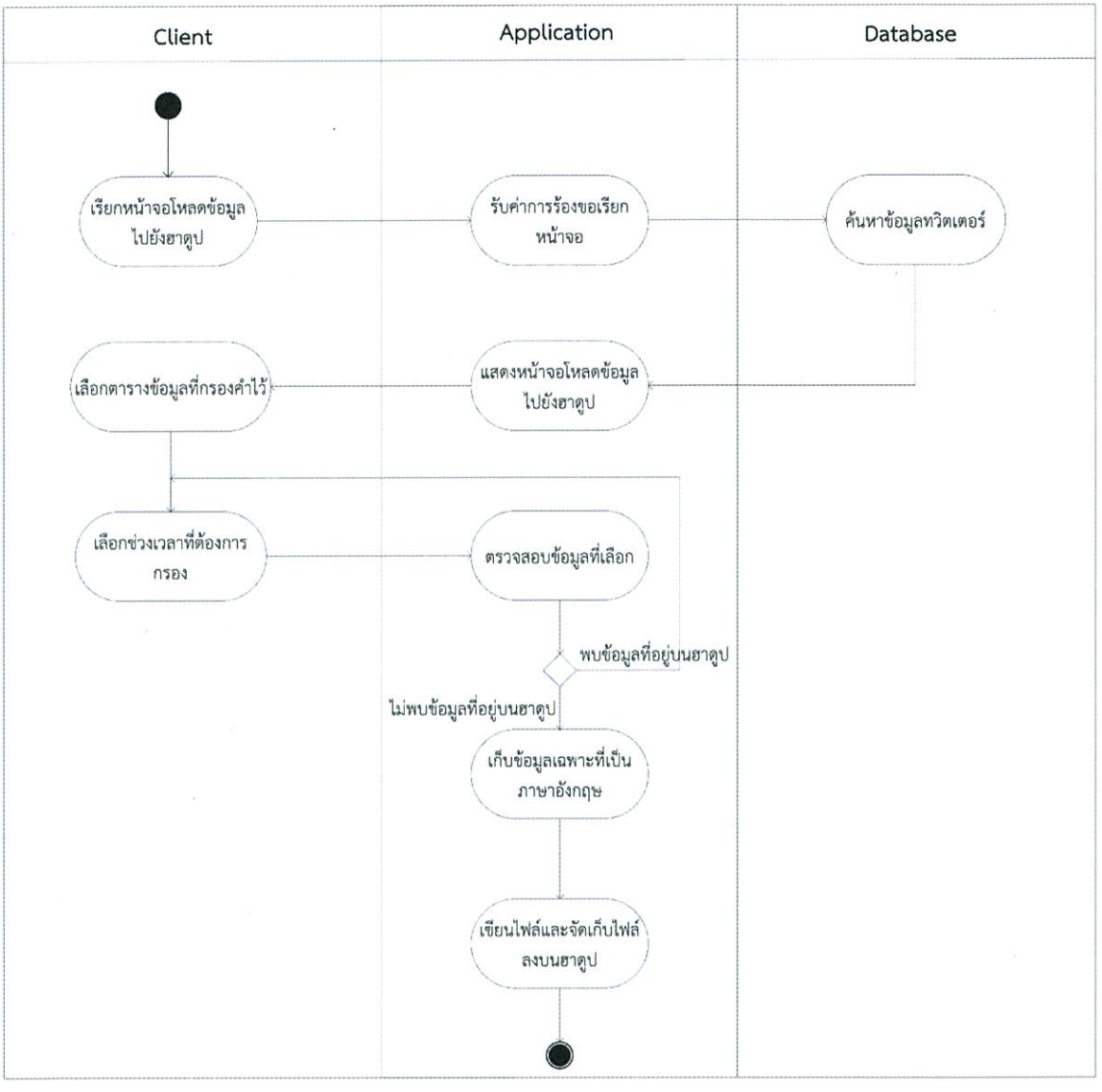


รูปที่ 3.5 แผนภาพ Activity Diagram ของฟังก์ชันกรองคำจากทวิตเตอร์

จากรูปที่ 3.5 แสดงขั้นตอนการทำงานของฟังก์ชันกรองคำจากทวิตเตอร์ โดยผู้ใช้สามารถใส่คำที่ต้องการกรองข้อมูลจากทวิตเตอร์และสามารถตั้งชื่อตารางที่ใช้ในการเก็บข้อมูลได้ จากนั้นระบบจะค้นหาข้อมูลจากทวิตเตอร์ผ่านทาง Twitter API แล้วเก็บข้อมูลลงในฐานข้อมูล

4) ฟังก์ชันโหลดข้อมูลไปยังฮาดูป

แผนภาพ Activity Diagram แสดงขั้นตอนการทำงานของเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์ในฟังก์ชันการโหลดข้อมูลไปยังฮาดูปแสดงดังรูปที่ 3.6

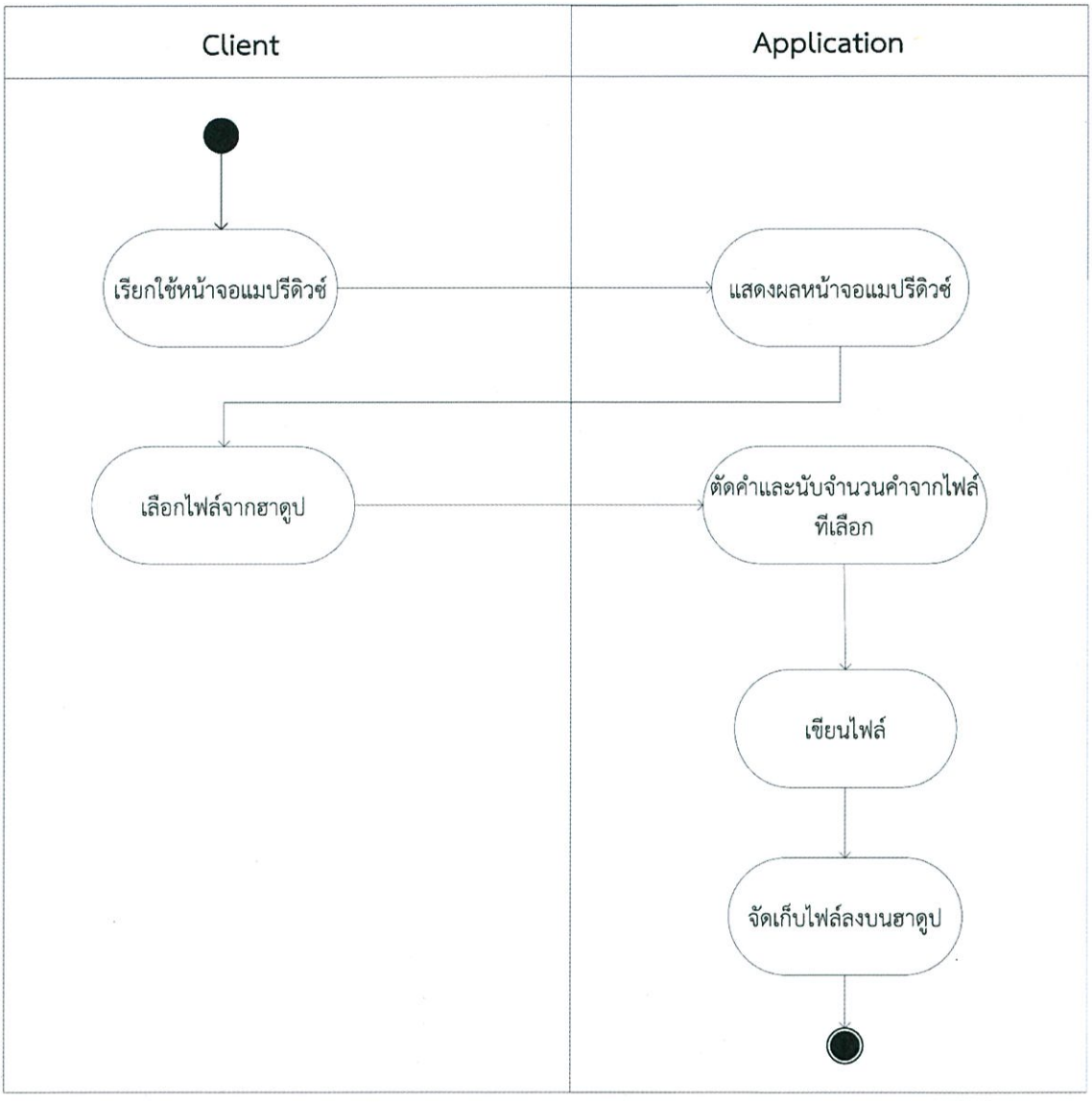


รูปที่ 3.6 แผนภาพ Activity Diagram ของฟังก์ชันการโหลดข้อมูลไปยังฮาดูป

จากรูปที่ 3.6 แสดงขั้นตอนการทำงานของฟังก์ชันการโหลดข้อมูลไปยังฮาดูป โดยหลังจากที่ผู้ใช้กรองคำมาจากทวิตเตอร์แล้ว ผู้ใช้สามารถเลือกตารางข้อมูล ช่วงเวลาที่ต้องการกรองข้อมูลได้ จากนั้นระบบจะตรวจสอบข้อมูลที่อยู่บนฮาดูปแล้วจัดเก็บไฟล์ไปยังฮาดูป

5) ฟังก์ชันแมปรีดิวซ์

แผนภาพ Activity Diagram แสดงขั้นตอนการทำงานของเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์ในฟังก์ชันการแมปรีดิวซ์แสดงดังรูปที่ 3.7

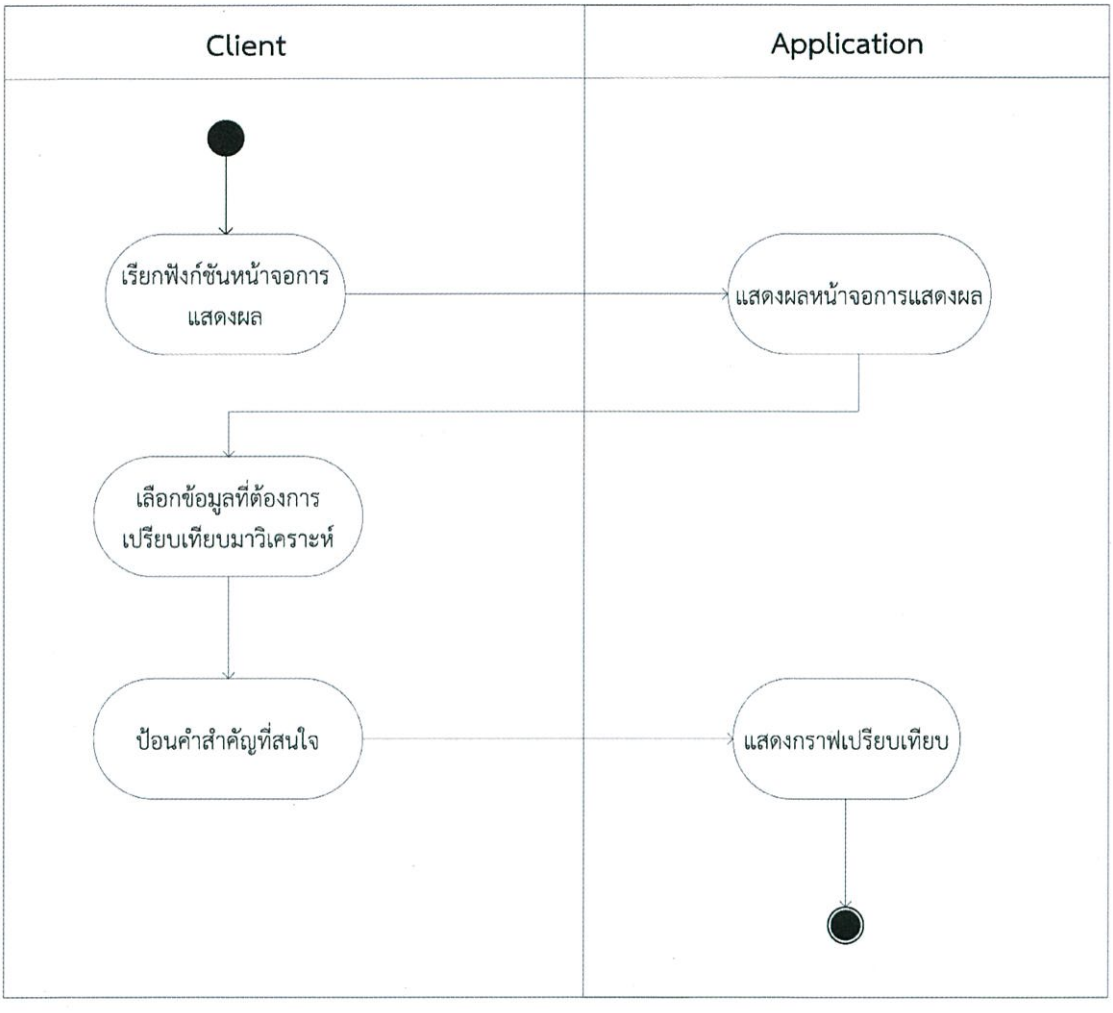


รูปที่ 3.7 แผนภาพ Activity Diagram ของฟังก์ชันการแมปรีดิวซ์

จากรูปที่ 3.7 แสดงขั้นตอนการทำงานของฟังก์ชันการแมปรีดิวซ์ โดยหลังจากที่ผู้ใช้โหลดข้อมูลไปยังฮาดูปเรียบร้อยแล้ว ผู้ใช้สามารถเลือกไฟล์ที่ต้องการเพื่อเข้าสู่กระบวนการนับคำในฟังก์ชันแมปรีดิวซ์

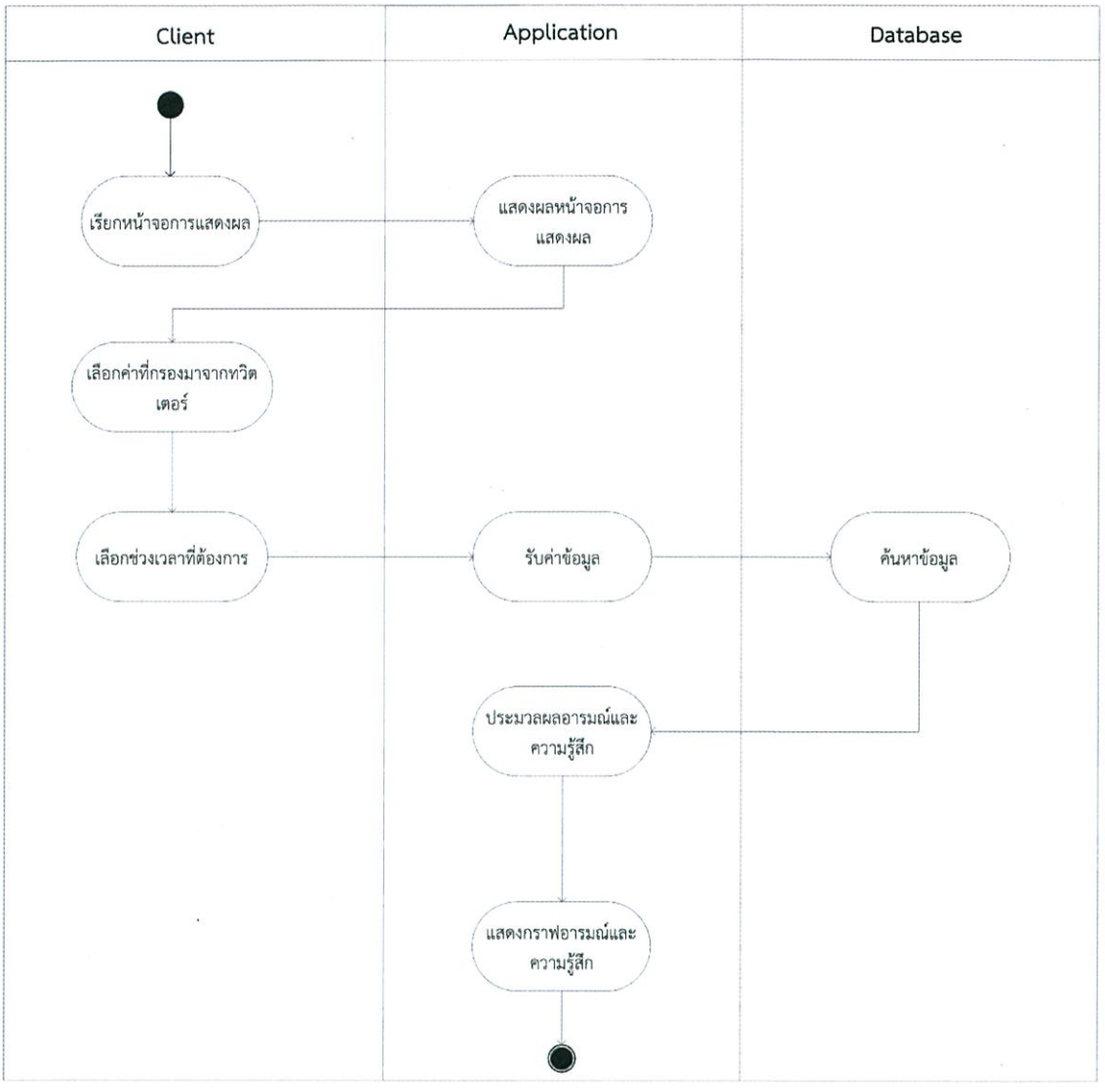
6) ฟังก์ชันการแสดงผล

แผนภาพ Activity Diagram แสดงขั้นตอนการทำงานของเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์ในฟังก์ชันการแสดงผลกราฟเปรียบเทียบผลิตภัณฑ์แสดงดังรูปที่ 3.8 และกราฟแสดงอารมณ์และความรู้สึกแสดงรูปที่ 3.9



รูปที่ 3.8 แผนภาพ Activity Diagram ของฟังก์ชันการแสดงผลกราฟเปรียบเทียบ

จากรูปที่ 3.8 แสดงฟังก์ชันการแสดงผลกราฟเปรียบเทียบผลิตภัณฑ์ โดยผู้ใช้สามารถเลือกข้อมูลและคำที่สนใจในการเปรียบเทียบได้ จากนั้นระบบจะแสดงกราฟตามที่ผู้ใช้ต้องการ

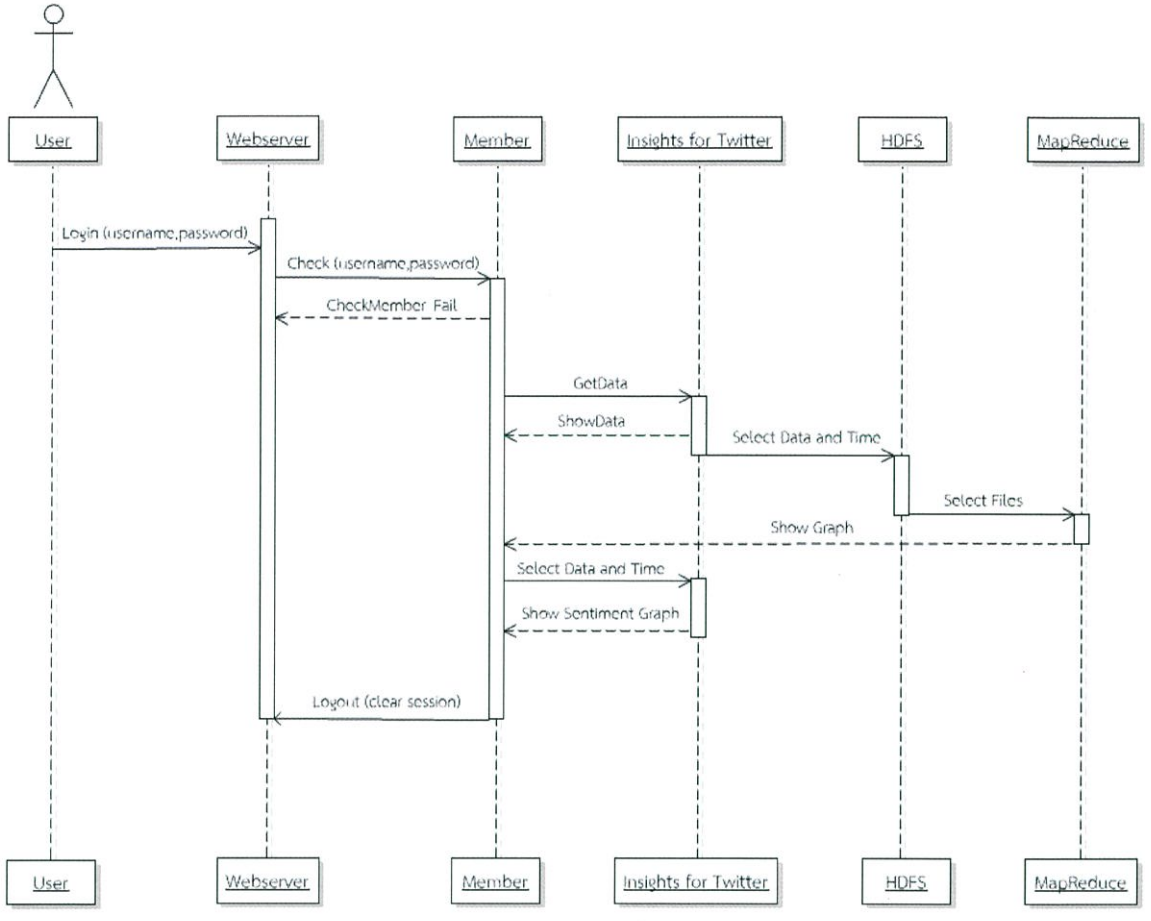


รูปที่ 3.9 แผนภาพ Activity Diagram ของฟังก์ชันการแสดงผลกราฟอารมณ์และความรู้สึก

จากรูปที่ 3.9 แสดงฟังก์ชันการแสดงผลกราฟอารมณ์และความรู้สึก โดยผู้ใช้สามารถเลือกข้อมูลและช่วงเวลาที่กรองมาจากทวิตเตอร์ จากนั้นระบบจะค้นหาข้อมูลจากฐานข้อมูลแล้วแสดงกราฟอารมณ์และความรู้สึก

3.2.3 แผนภาพซีเควนซ์ไดอะแกรม (Sequence Diagram)

เป็นแผนภาพ Sequence Diagram ของระบบทั้งหมดและฟังก์ชันต่างๆแสดงดังรูปที่ 3.10



รูปที่ 3.10 แผนภาพ Sequence Diagram ภาพรวมของระบบและฟังก์ชันต่างๆ

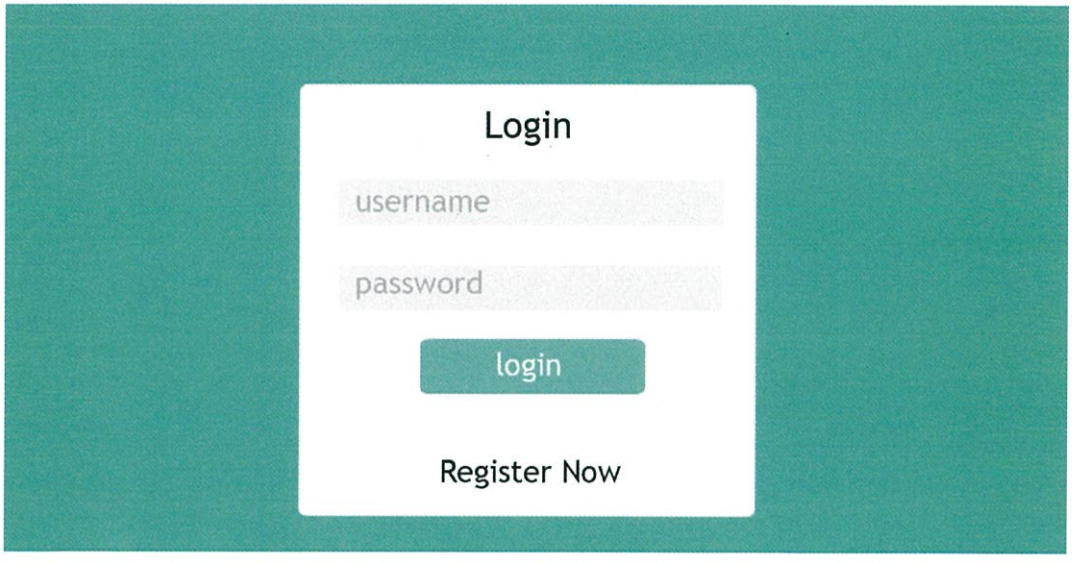
จากรูปที่ 3.10 แสดงถึงภาพรวมของระบบและฟังก์ชันต่างๆของเว็บแอปพลิเคชัน โดยที่ผู้ใช้จะต้องล็อกอินเข้าระบบด้วย Username และ Password แล้วระบบจะตรวจสอบข้อมูลที่ผู้ใช้กรอกว่าถูกต้องหรือไม่ หากไม่ถูกต้องผู้ใช้จะต้องกรอกข้อมูลใหม่อีกครั้ง หลังจากที่เข้าสู่ระบบแล้วผู้ใช้สามารถดึงข้อมูลที่ต้องการจากทวิตเตอร์ได้ ซึ่งข้อมูลทั้งหมดจะแสดงในหน้าจอของเว็บแอปพลิเคชัน จากนั้นผู้ใช้จะเข้าสู่ขั้นตอนของการโหลดข้อมูลเข้าสู่ HDFS โดยผู้ใช้สามารถเลือกข้อมูลและช่วงเวลาของข้อมูลได้ เมื่อโหลดข้อมูลเรียบร้อยแล้วผู้ใช้สามารถเลือกไฟล์เพื่อนำข้อมูลเข้าสู่กระบวนการแมปรีดิวซ์ในขั้นตอนถัดไป โดยระบบจะแสดงกราฟที่ผ่านกระบวนการแมปรีดิวซ์ และสุดท้ายผู้ใช้สามารถเลือกข้อมูลและเวลาเพื่อเข้าสู่กระบวนการ Sentiment จากนั้นผลลัพธ์จะแสดงออกมาทางหน้าเว็บแอปพลิชัน

3.3 การออกแบบส่วนติดต่อกับผู้ใช้

ส่วนติดต่อกับผู้ใช้ของหน้าเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์มีส่วนประกอบดังนี้

1) หน้าจอแสดงการเข้าระบบ

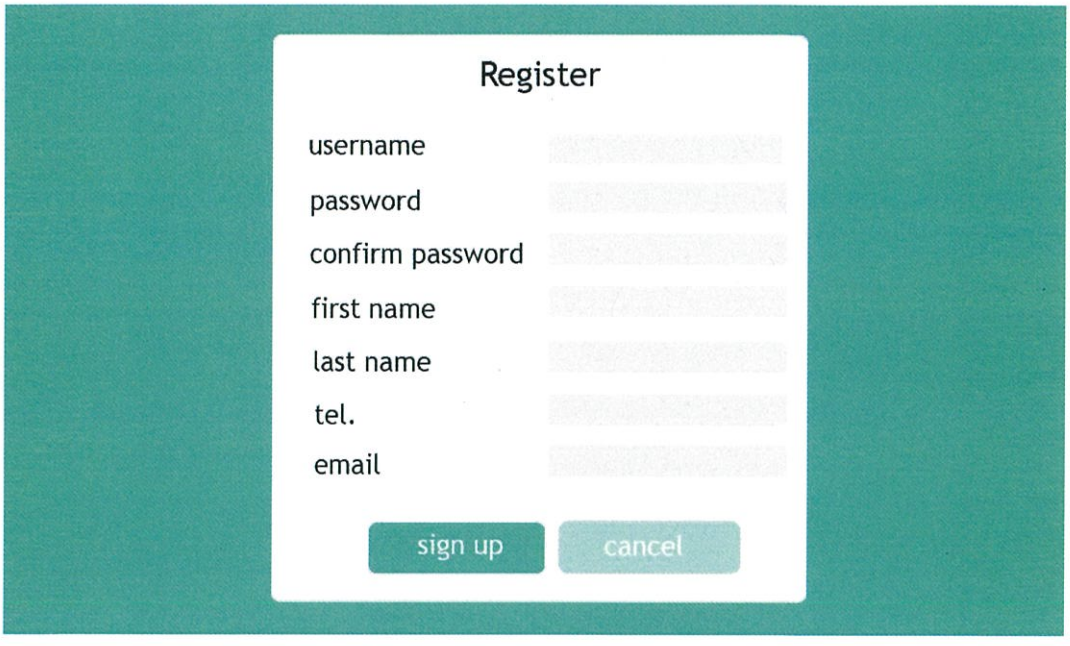
เป็นหน้าจอสำหรับให้ผู้ใช้เข้าสู่ระบบโดยการกรอกชื่อผู้ใช้ (Username) และรหัสผ่าน (Password) แสดงดังรูปที่ 3.11



รูปที่ 3.11 ตัวอย่างหน้าจอแสดงการเข้าสู่ระบบ

2) หน้าจอการลงทะเบียนเข้าสู่ระบบ

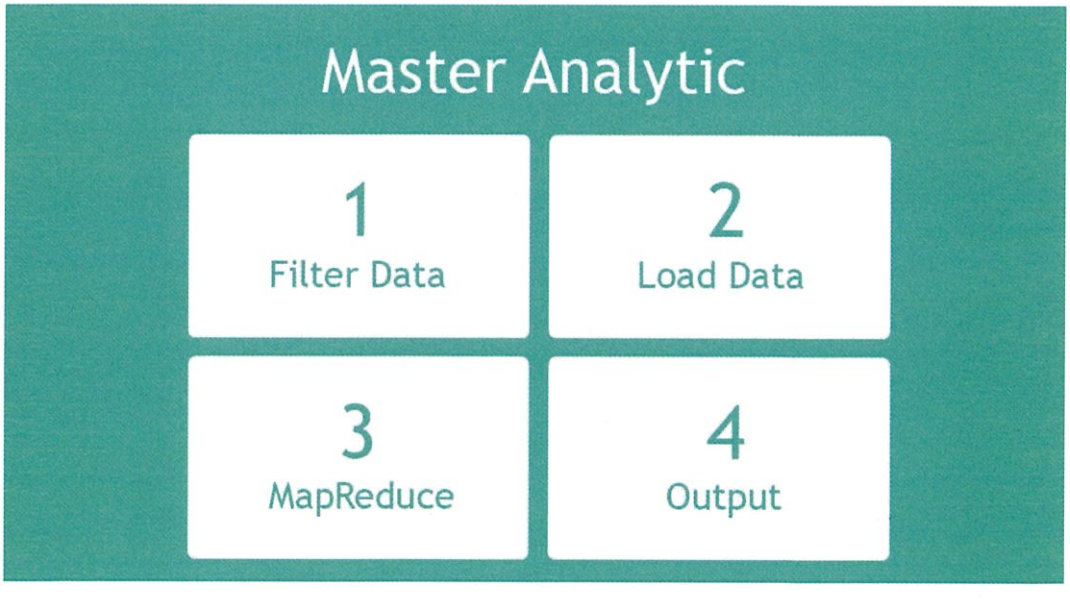
ถ้าผู้ใช้งานระบบยังไม่ได้เป็นสมาชิกจะต้องทำการลงทะเบียนเพื่อเข้าสู่เว็บไซต์โดยต้องกรอกข้อมูลเพื่อล็อกอินเข้าสู่ระบบซึ่งแสดงดังรูปที่ 3.12



รูปที่ 3.12 ตัวอย่างหน้าจอลงทะเบียนเข้าสู่ระบบ

3) **หน้าจอเมนูหลักของเว็บไซต์**

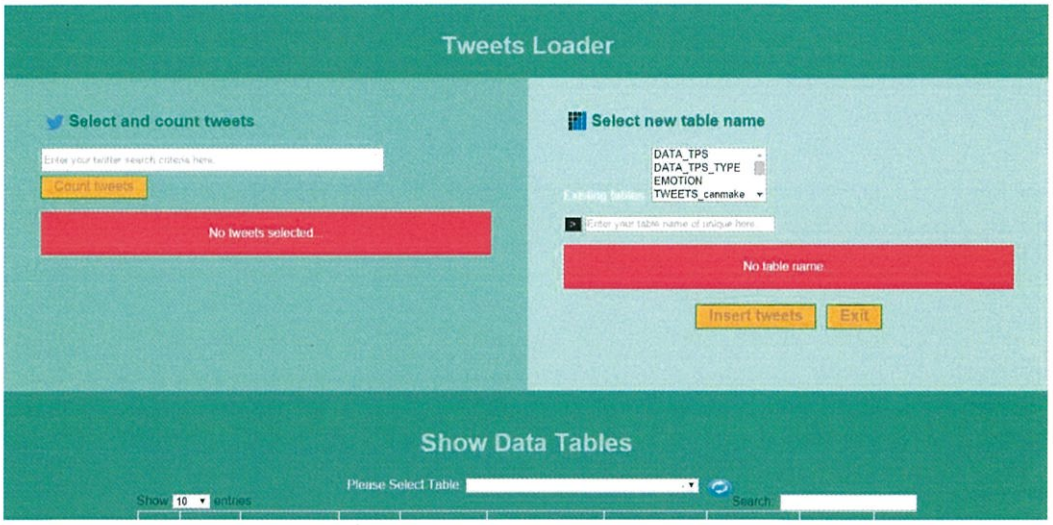
ในหน้าจอนี้จะแสดงฟังก์ชันหลักในการทำงานของระบบซึ่งประกอบด้วย ฟังก์ชันกรองคำจากทวิตเตอร์ (Filter Data) ฟังก์ชันโหลดข้อมูลไปยังฮาดูป (Load Data) ฟังก์ชันแมปรีดิวซ์ (MapReduce) และฟังก์ชันการแสดงผล (Output) แสดงดังรูปที่ 3.13



รูปที่ 3.13 ตัวอย่างหน้าจอเมนูหลักของเว็บไซต์

4) หน้าจอฟังก์ชันกรองคำจากทวิตเตอร์

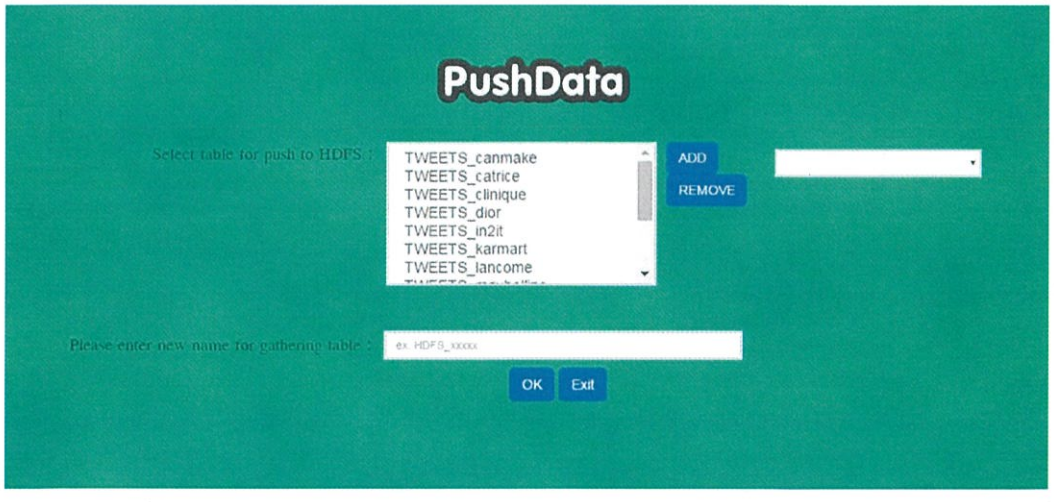
ฟังก์ชันนี้ผู้ใช้สามารถกรองคำจากทวิตเตอร์ได้โดยการพิมพ์คำที่ต้องการลงไปในช่องที่กำหนดแล้วตั้งชื่อตารางที่จะนำข้อมูลไปเก็บซึ่งแสดงดังรูปที่ 3.14



รูปที่ 3.14 ตัวอย่างหน้าจอฟังก์ชันกรองคำจากทวิตเตอร์

5) หน้าจอฟังก์ชันโหลดข้อมูลไปยังฮาดูป

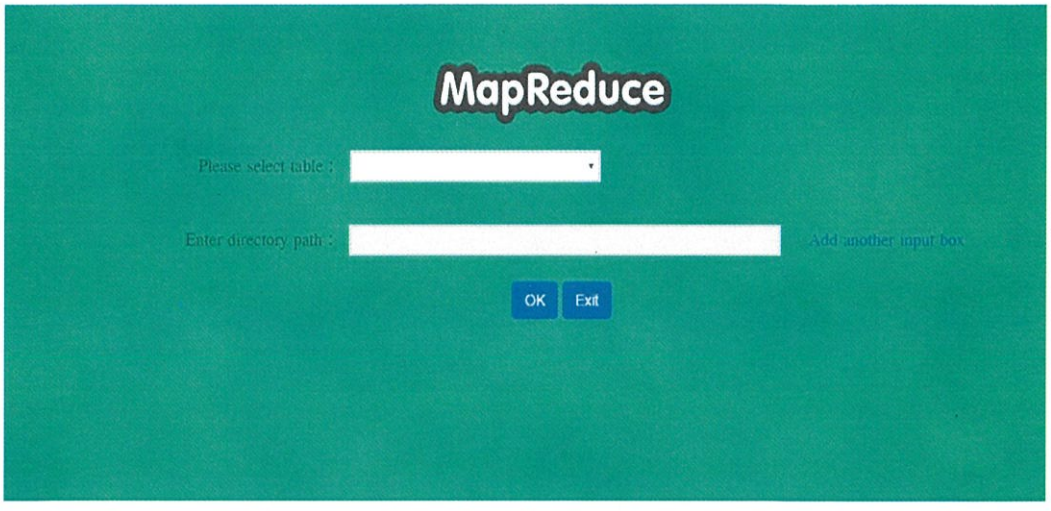
ฟังก์ชันนี้ผู้ใช้สามารถนำตารางที่เก็บข้อมูลที่ได้สร้างไว้ในฟังก์ชันกรองคำจากทวิตเตอร์ไปเก็บไว้บนฮาดูปแสดงดังรูปที่ 3.15



รูปที่ 3.15 ตัวอย่างหน้าจอฟังก์ชันโหลดข้อมูลไปยังฮาดูป

6) หน้าจอฟังก์ชันแมปรีดิวซ์

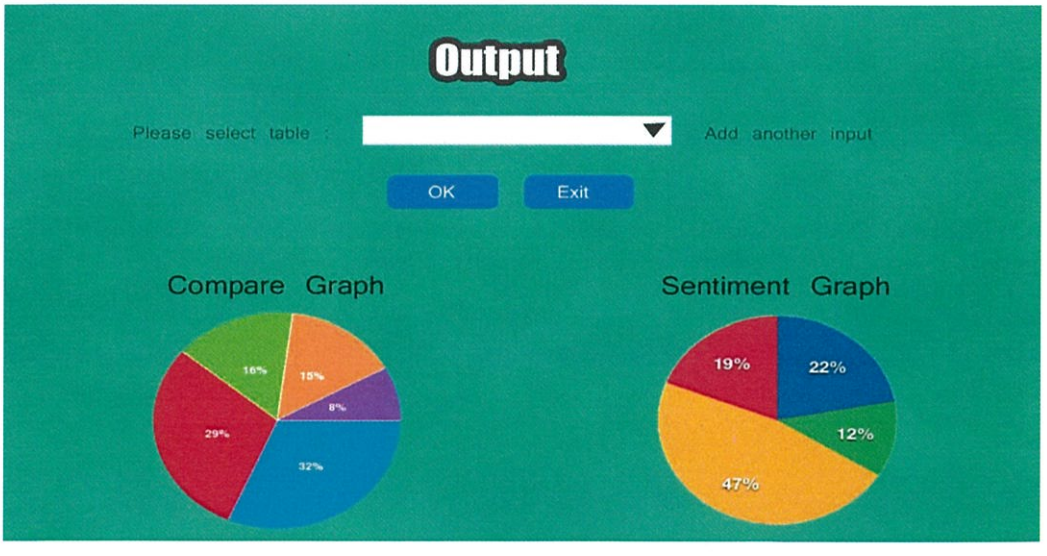
ฟังก์ชันนี้ผู้ใช้จะต้องเลือกตารางที่เก็บไว้บนฮาดูปมาทำแมปรีดิวซ์แสดงดังรูปที่ 3.16 ซึ่งกระบวนการแมปรีดิวซ์จะเป็นการนับจำนวนคำพี่พบ (Word Count) โดยผลลัพธ์ของการแมปรีดิวซ์จะออกมาเป็นกราฟในฟังก์ชันการแสดงผล



รูปที่ 3.16 ตัวอย่างหน้าจอฟังก์ชันแมปรีดิวซ์

7) หน้าจอฟังก์ชันการแสดงผล

ฟังก์ชันนี้จะแสดงผลลัพธ์ออกมาในรูปแบบของกราฟเปรียบเทียบและกราฟแสดงอารมณ์และความรู้สึกเป็นความรู้สึกด้านบวก ด้านลบ เป็นกลาง หรือคลุมเครือ แสดงดังรูปที่ 3.17



รูปที่ 3.17 ตัวอย่างหน้าจอฟังก์ชันการแสดงผล

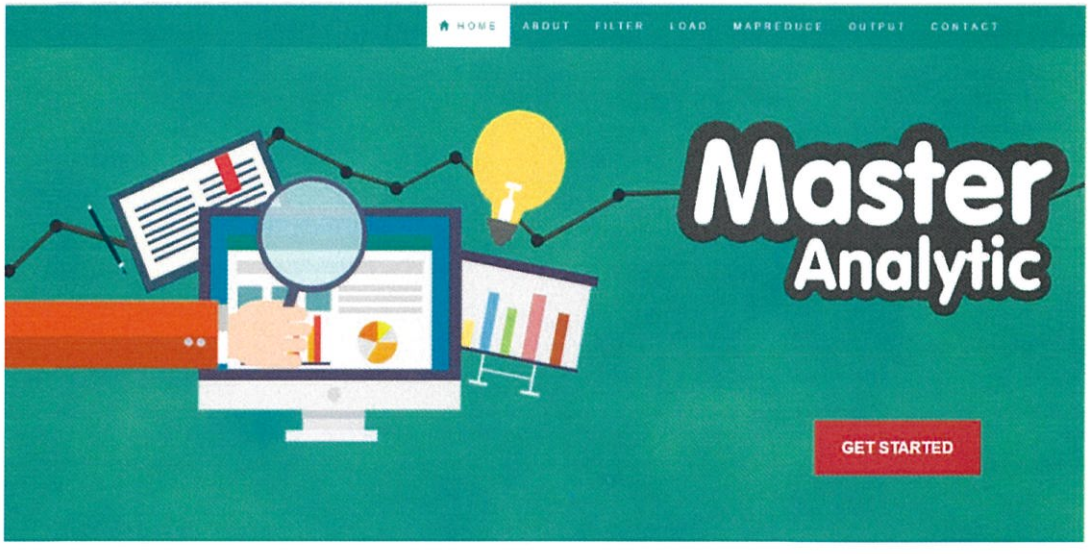
บทที่ 4
ผลการดำเนินงานและการอภิปรายผล

4.1 การแสดงผลของเว็บไซต์

Master Analytic เป็นเว็บไซต์วิเคราะห์ข้อมูลขนาดใหญ่ โดยมีแหล่งข้อมูลจากทวิตเตอร์ ซึ่งช่วยให้ผู้ใช้เข้าถึงการประมวลผลข้อมูลขนาดใหญ่ (Big Data) ได้สะดวกมากยิ่งขึ้น และผลลัพธ์ที่ได้จากการประมวลผลในกระบวนการต่างๆสามารถช่วยเป็นส่วนหนึ่งในการตัดสินใจในเชิงธุรกิจของผู้ใช้ได้ โดยเว็บไซต์ Master Analytic มีหน้าจอหลักๆ ดังนี้

4.1.1 หน้าจอแรกของเว็บไซต์

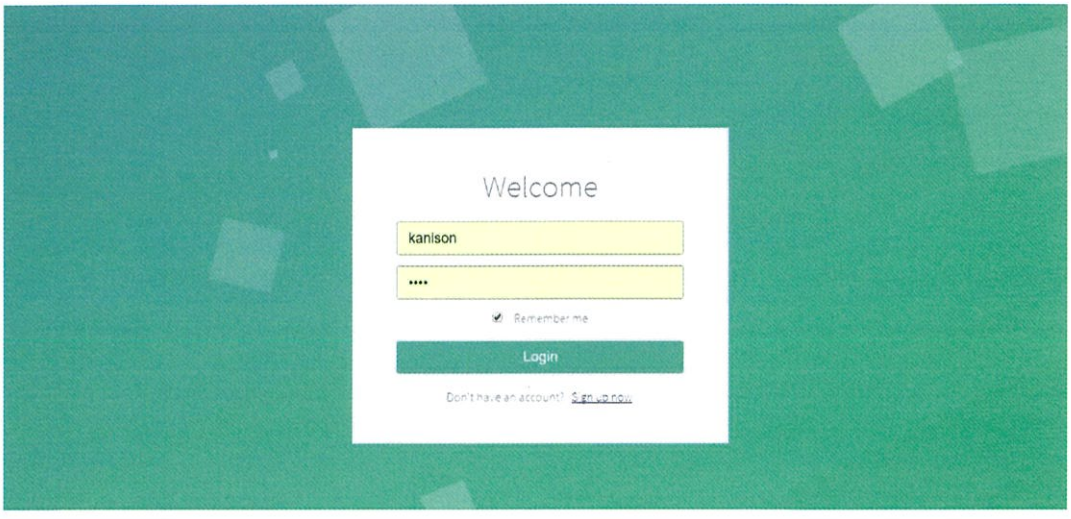
หน้าแรกของเว็บไซต์จะบอกให้ทราบเกี่ยวกับรายละเอียดของเว็บไซต์ รายละเอียดของผู้จัดทำและช่องทางการติดต่อสื่อสาร เมื่อทำการอ่านรายละเอียดเรียบร้อยแล้วสามารถเข้าสู่เว็บไซต์เพื่อวิเคราะห์ข้อมูลได้ดังรูปที่ 4.1



รูปที่ 4.1 หน้าจอแรกของเว็บไซต์

4.1.2 หน้าจอแสดงการเข้าระบบ

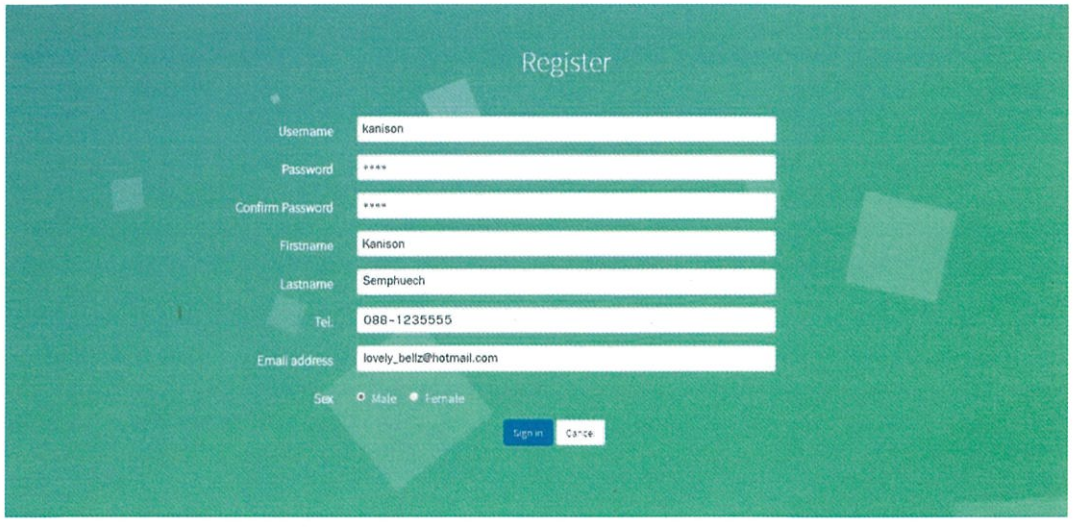
ผู้ใช้สามารถล็อกอินเข้าสู่เว็บไซต์ได้โดยกรอก Username และ Password หากใช้งานเว็บไซต์เป็นครั้งแรก ผู้ใช้จำเป็นต้องลงทะเบียนโดยกดปุ่ม Sign up now เพื่อสมัครสมาชิกก่อนดังรูปที่ 4.2



รูปที่ 4.2 หน้าจอการเข้าสู่ระบบ

4.1.3 หน้าจอการลงทะเบียนเพื่อเข้าใช้งานระบบ

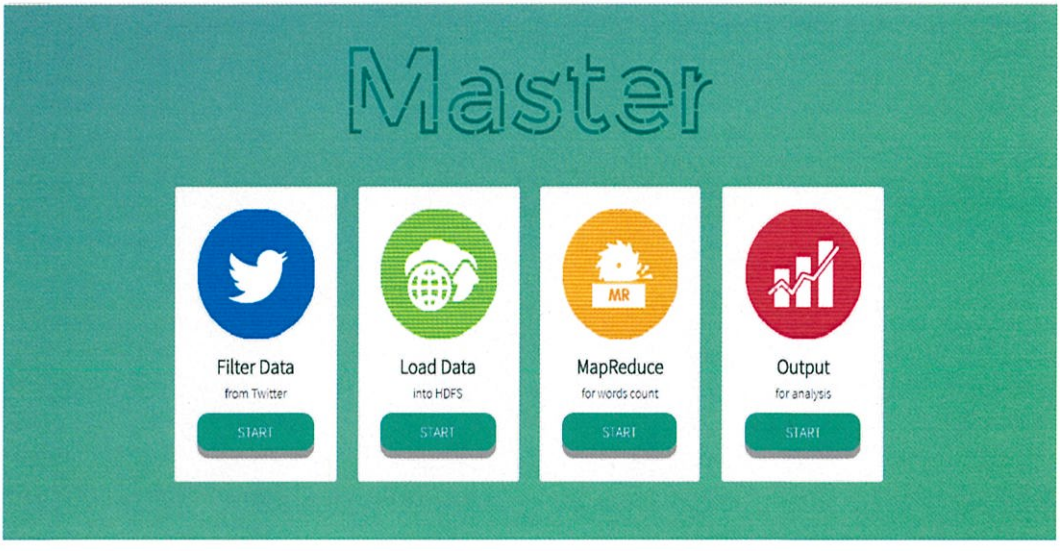
ผู้ใช้ต้องกรอกข้อมูลให้ครบถ้วนเพื่อล็อกอินเข้าสู่เว็บไซต์ แสดงดังรูปที่ 4.3



รูปที่ 4.3 หน้าจอการลงทะเบียน

4.1.4 หน้าจอเมนูหลักของเว็บไซต์

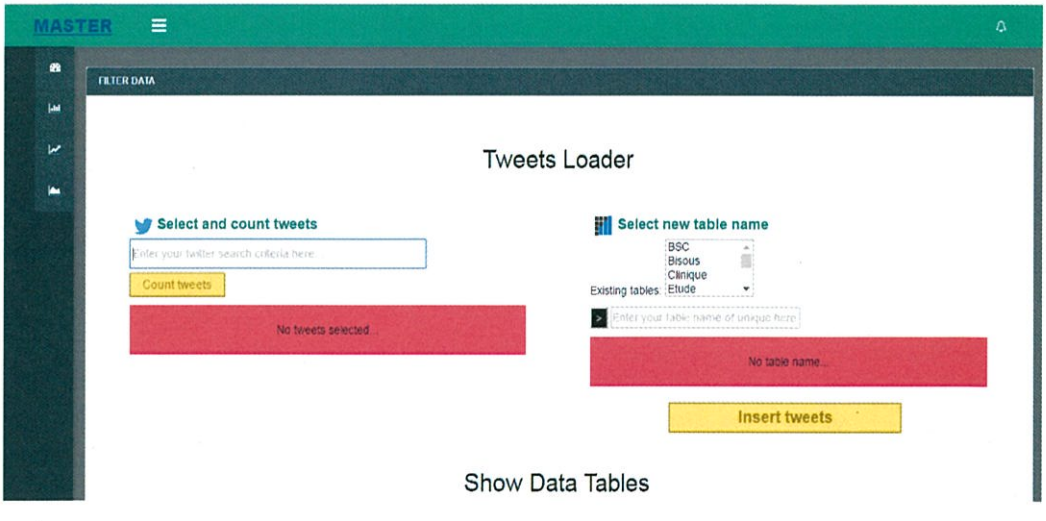
เว็บไซต์ประกอบด้วยฟังก์ชันหลักทั้งหมด 4 ฟังก์ชัน คือ Filter Data, Load Data, MapReduce และ Output แสดงดังรูปที่ 4.4



รูปที่ 4.4 หน้าจอเมนูหลักของเว็บไซต์

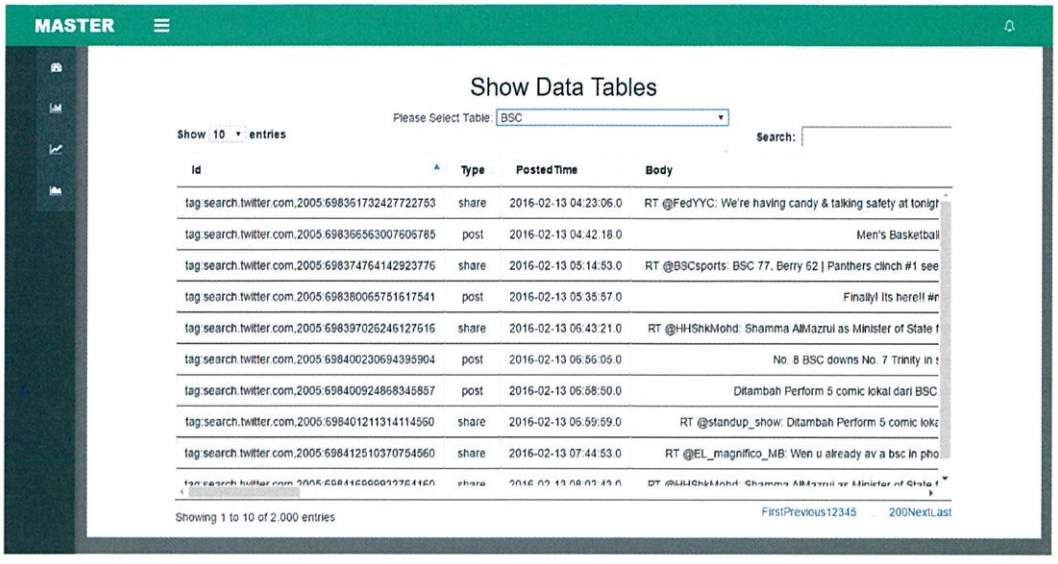
4.1.5 หน้าจอฟังก์ชันกรองคำจากทวิตเตอร์

ฟังก์ชัน Filter Data หรือขั้นตอนการกรองคำจากทวิตเตอร์ เป็นฟังก์ชันที่ให้ผู้ใช้สามารถระบุคำที่ต้องการค้นหาในทวิตเตอร์แล้วตั้งชื่อตารางที่จะจัดเก็บข้อมูลแสดงดังรูปที่ 4.5



รูปที่ 4.5 ขั้นตอนการกรองคำ

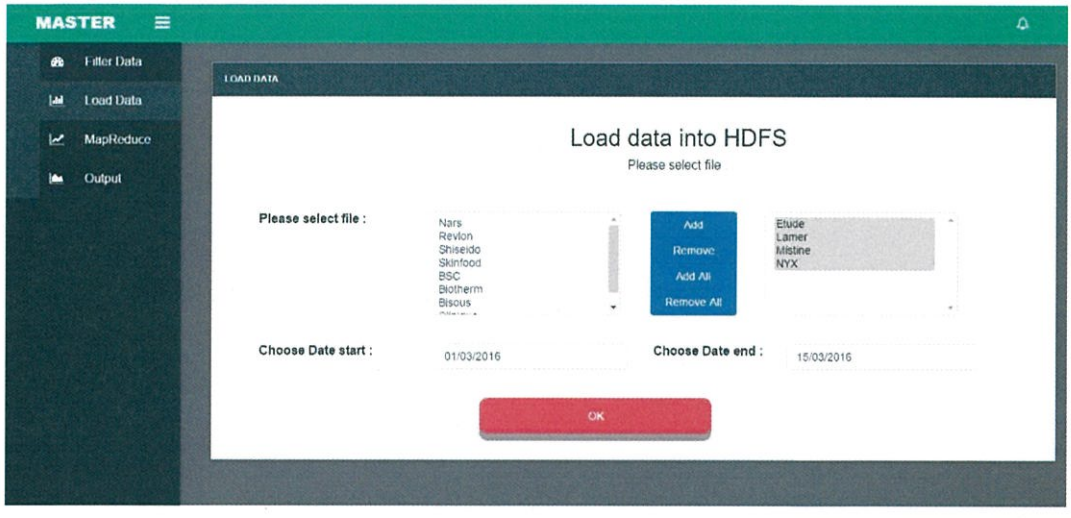
เมื่อทำการกรองข้อมูลเสร็จเรียบร้อย แล้วผู้ใช้สามารถเลือกดูรายละเอียดของตารางได้ซึ่งแสดงดังรูปที่ 4.6



รูปที่ 4.6 รายละเอียดของข้อมูลที่กรองมาจากทวิตเตอร์

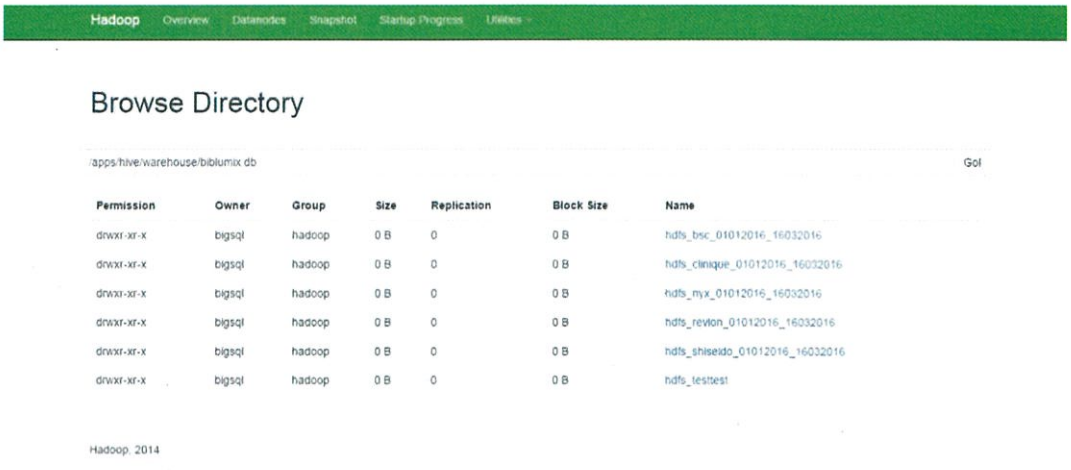
4.1.6 หน้าจอฟังก์ชันโหลดข้อมูลไปยังฮาดูป

ฟังก์ชัน Load Data หรือขั้นตอนการโหลดข้อมูล เป็นขั้นตอนการนำข้อมูลที่กรองมาจากฟังก์ชัน Filter Data แล้วเก็บเป็นตารางในฐานข้อมูลของ DashDB นำไปเก็บไว้บนฮาดูป (HDFS) โดยผู้ใช้สามารถเลือกได้หลายๆตารางและสามารถเลือกช่วงเวลาของการโหลดข้อมูลได้ แสดงดังรูปที่ 4.7



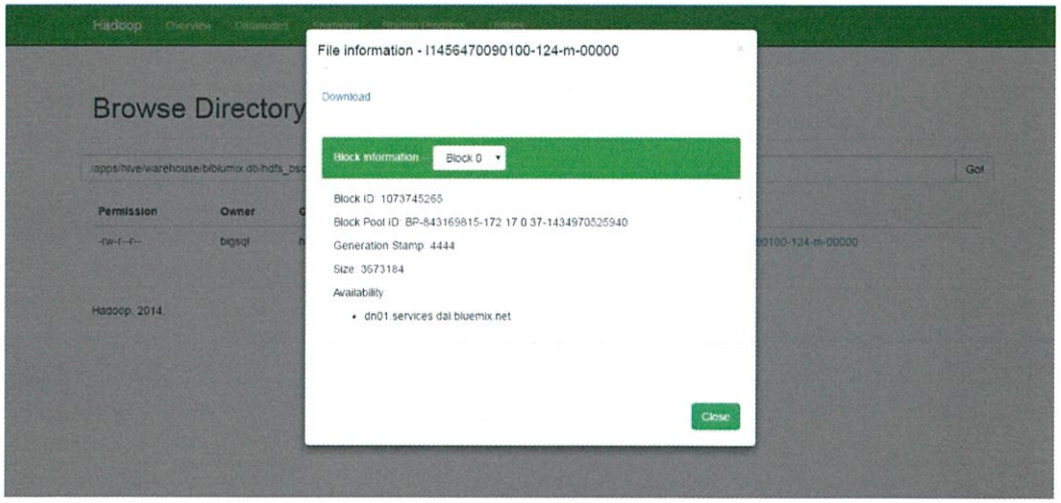
รูปที่ 4.7 ขั้นตอนการโหลดข้อมูลไปเก็บไว้บนฮาดูป

เมื่อผู้ใช้นำข้อมูลลงบนฮาดูปเรียบร้อยแล้ว ผู้ดูแลระบบจะสามารถเข้าไปดูที่ webhdfs เพื่อตรวจสอบดูว่าข้อมูลที่นำลงไป มีรายละเอียดเป็นอย่างไร โดยหน้าจอนี้สำหรับผู้ดูแลระบบเท่านั้นที่สามารถเข้ามาดูได้ แสดงดังรูปที่ 4.8



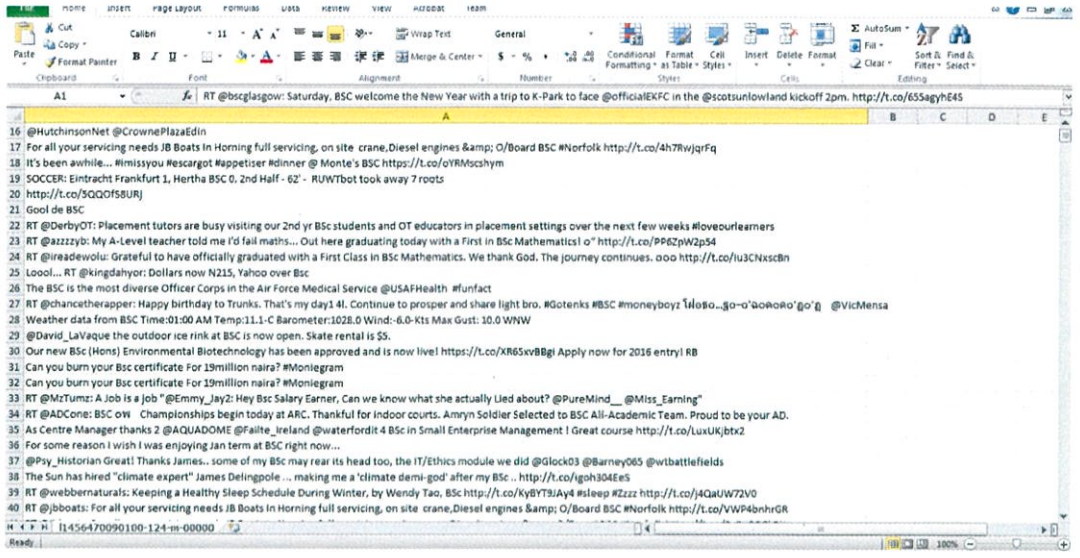
รูปที่ 4.8 หน้าจอ webhdfs สำหรับผู้ดูแลระบบ (1)

ผู้ดูแลระบบทำการโหลดไฟล์ที่อยู่บนฮาดูปเพื่อตรวจสอบรายละเอียดของไฟล์ แสดงดังรูปที่ 4.9



รูปที่ 4.9 หน้าจอ webhdfs สำหรับผู้ดูแลระบบ (2)

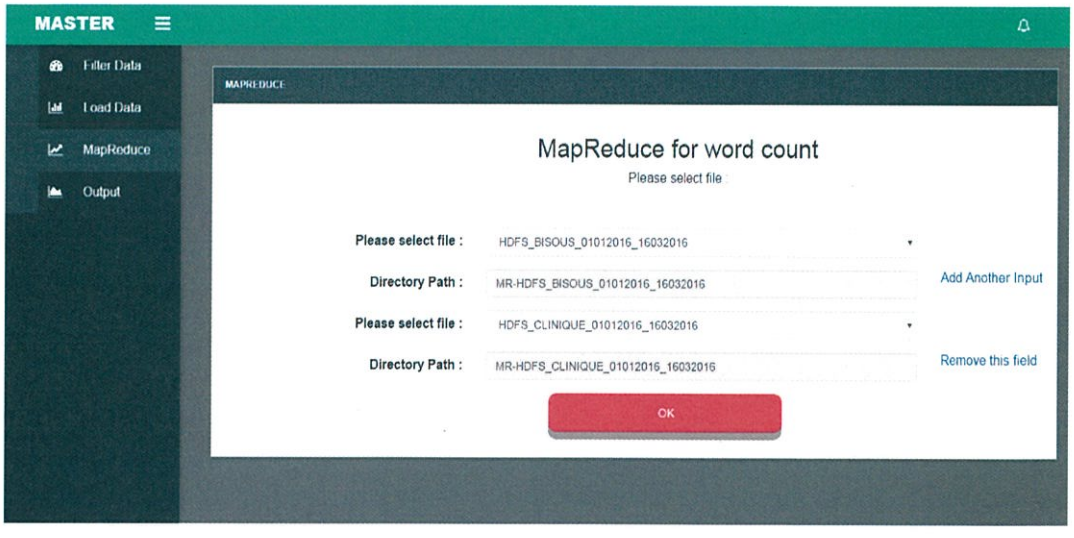
เมื่อผู้ดูแลระบบนำไฟล์ที่โหลดมาไปเปิดในโปรแกรม Excel จะแสดงรายละเอียดภายในของไฟล์ดังรูปที่ แสดงดังรูปที่ 4.10



รูปที่ 4.10 รายละเอียดเกี่ยวกับไฟล์สำหรับผู้ดูแลระบบ

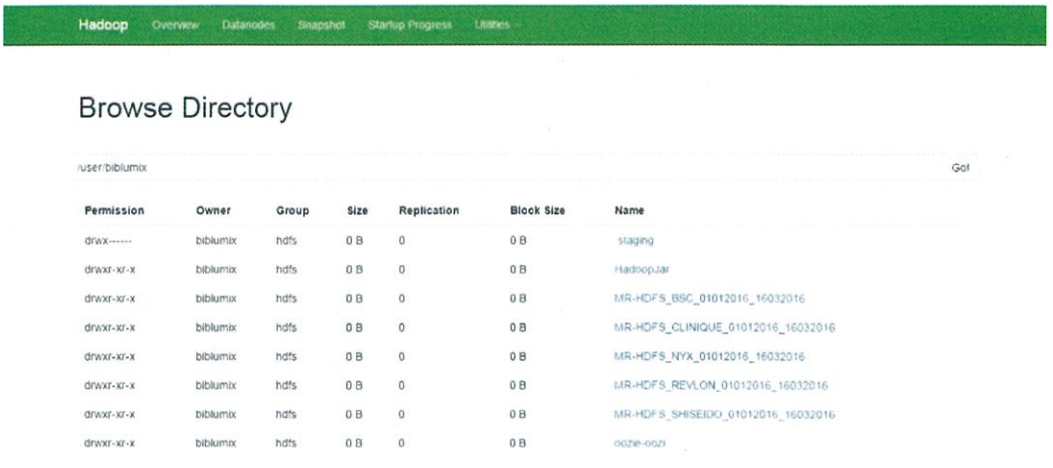
4.1.7 หน้าจอฟังก์ชันแมปรีดิวซ์

ฟังก์ชันแมปรีดิวซ์หรือขั้นตอนการตัดคำ โดยขั้นตอนนี้จะนำไฟล์ที่เก็บอยู่บนฮาดูปมาทำแมปรีดิวซ์เพื่อนับจำนวนคำ ซึ่งสามารถเพิ่มได้หลายๆตาราง แสดงดังรูปที่ 4.11



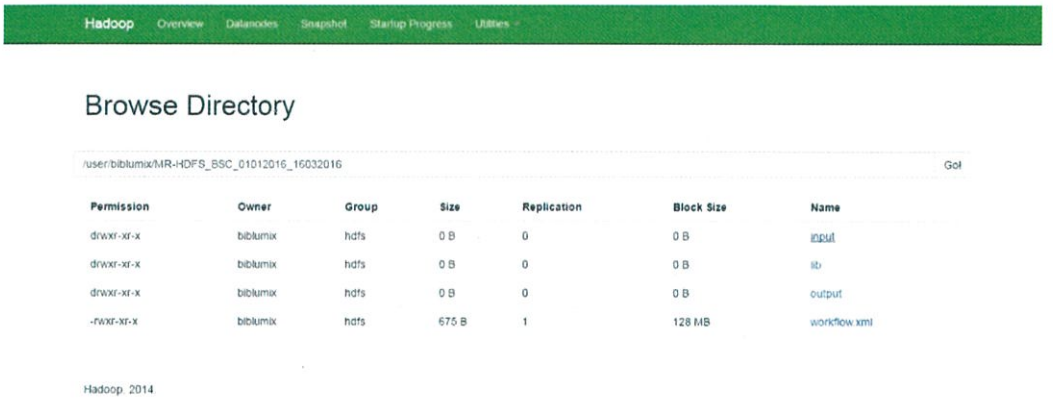
รูปที่ 4.11 หน้าจอฟังก์ชันแมปรีดิวซ์

เมื่อผู้ใช้ทำขั้นตอนแมปรีดิวซ์เสร็จเรียบร้อยแล้ว ผู้ดูแลระบบจะสามารถเข้าไปดูได้ที่ webhdfs เพื่อดูรายละเอียดของไฟล์ โดยจะเห็นว่าไฟล์ที่ผ่านการทำแมปรีดิวซ์แล้วจะถูกนำไปเก็บไว้บนฮาดูปเช่นกัน แสดงดังรูปที่ 4.12



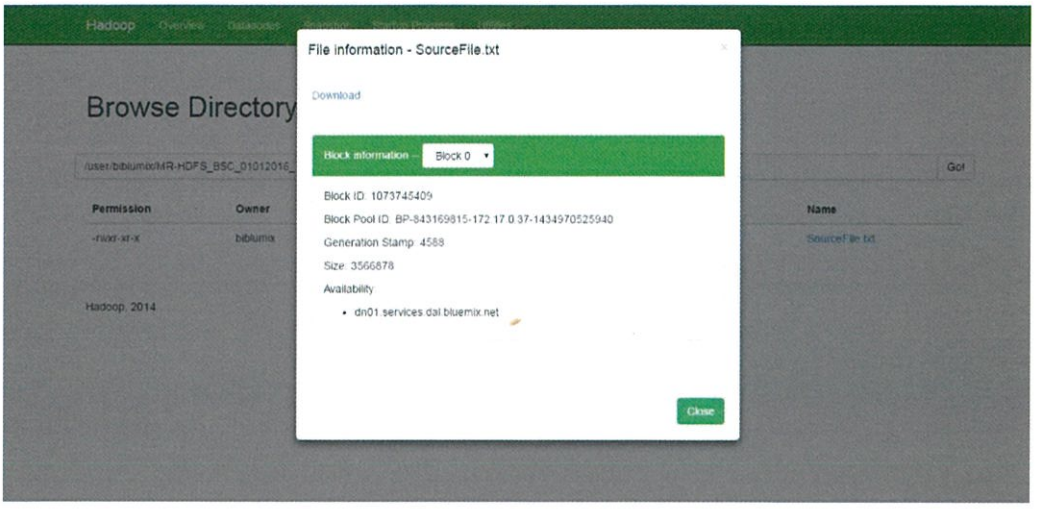
รูปที่ 4.12 หน้าจอ webhdfs สำหรับผู้ดูแลระบบ (3)

เมื่อเข้าไปยัง Path (พาธ) ดังกล่าวจะเห็นว่าภายในจะมีไฟล์ input, lib, output และ workflow.xml แสดงดังรูปที่ 4.13



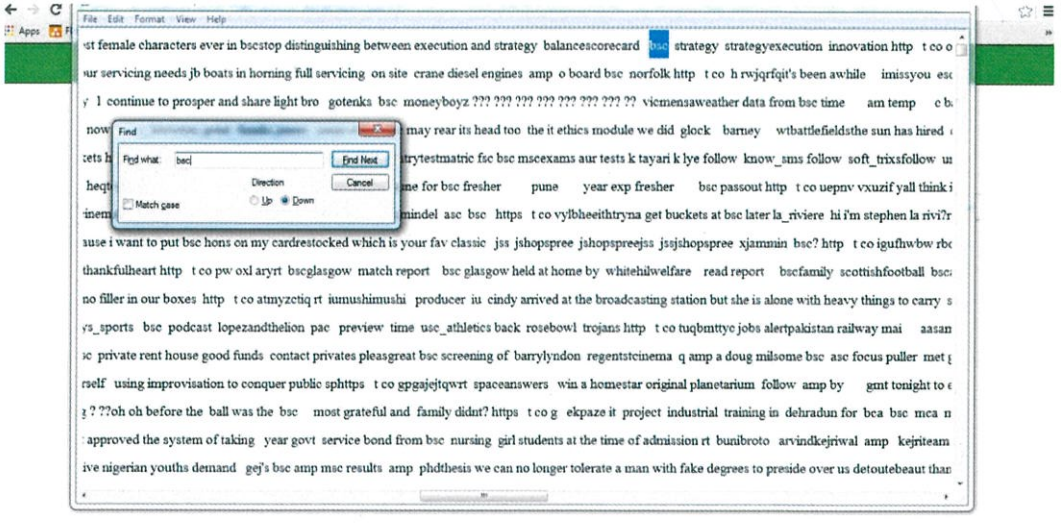
รูปที่ 4.13 หน้าจอ webhdfs สำหรับผู้ดูแลระบบ (4)

โดยภายในไฟล์ input จะเก็บข้อมูลที่เป็น SourceFile.txt คือเป็นไฟล์ต้นฉบับก่อนที่จะผ่านกระบวนการแมปรีดิวซ์ แสดงดังรูปที่ 4.14



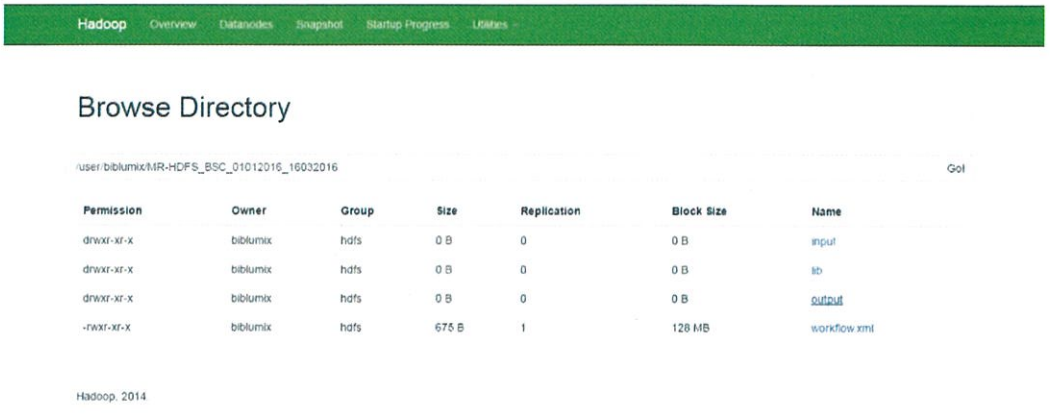
รูปที่ 4.14 ไฟล์ต้นฉบับในกระบวนการแมปรีดิวซ์

เมื่อดาวน์โหลด SourceFile.txt มาดูจะสังเกตได้ว่ารายละเอียดภายในไฟล์มีความกระจัดกระจาย เนื่องจากเป็นไฟล์ต้นฉบับก่อนจะผ่านขั้นตอนการแมปรีดิวซ์ แสดงดังรูปที่ 4.15



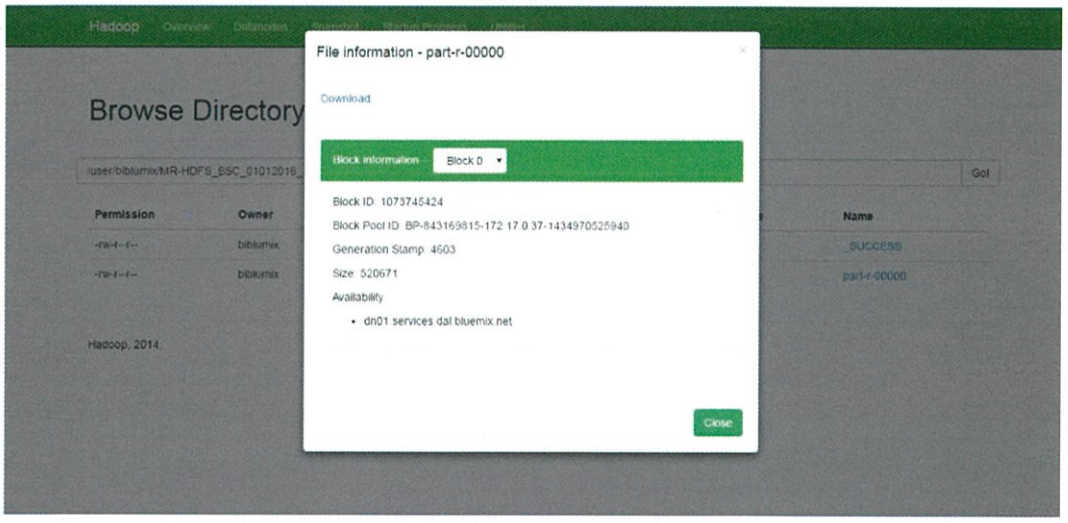
รูปที่ 4.15 รายละเอียดของไฟล์ SourceFile.txt

จากนั้นเข้าไปที่ไฟล์ output โดยไฟล์ที่ผ่านขั้นตอนการทำแมปรีดิวซ์จะถูกนำมาเก็บไว้ที่นี่ แสดงดังรูปที่ 4.16



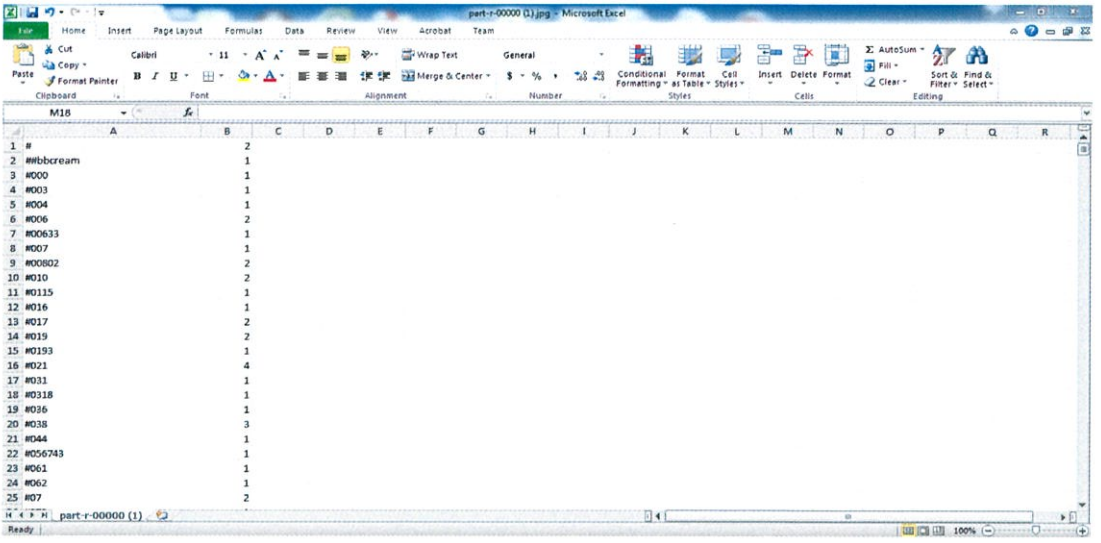
รูปที่ 4.16 หน้าจอ webhdfs สำหรับผู้ดูแลระบบ (5)

ภายในไฟล์ Output จะประกอบไปด้วยไฟล์ _success และ part-r-00000 ซึ่งไฟล์ที่ผ่านกระบวนการแมปรีดิวซ์จะถูกเก็บอยู่ในไฟล์ที่ชื่อ part-r-00000 จากนั้นกดดาวน์โหลด เพื่อดูรายละเอียดภายในไฟล์ แสดงดังรูปที่ 4.17



รูปที่ 4.17 ไฟล์ Output ที่ผ่านกระบวนการแมปรีดิวซ์

เมื่อนำไฟล์ part-r-00000 เปิดในโปรแกรม Excel จะสังเกตได้ว่าภายในไฟล์มีการแบ่งเป็น 2 คอลัมน์ คือ ส่วนของคำและส่วนของจำนวนนับ แสดงดังรูปที่ 4.18

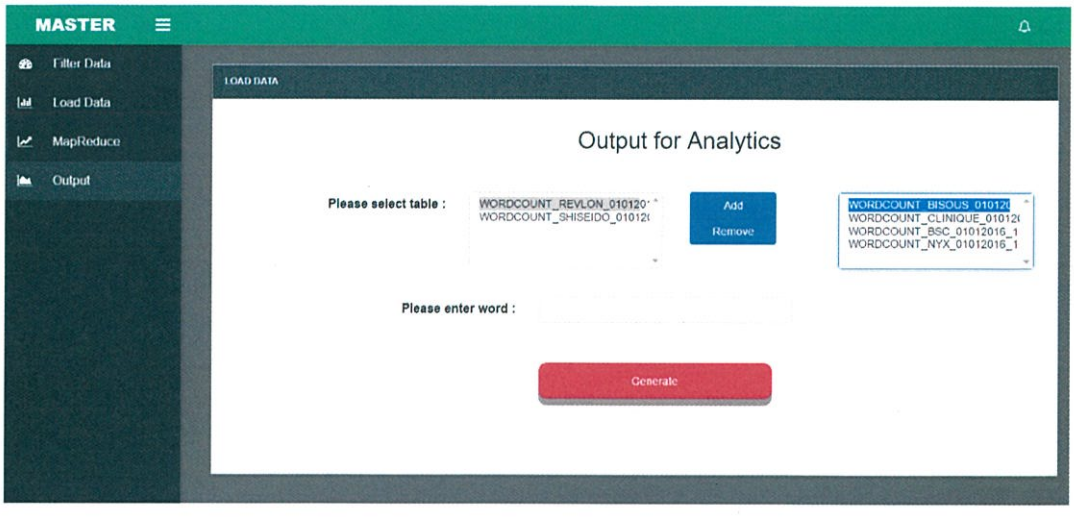


	A	B
1	#	2
2	##bbcream	1
3	#000	1
4	#003	1
5	#004	1
6	#006	2
7	#00633	1
8	#007	1
9	#00802	2
10	#010	2
11	#0115	1
12	#016	1
13	#017	2
14	#019	2
15	#0193	1
16	#021	4
17	#031	1
18	#0318	1
19	#036	1
20	#038	3
21	#044	1
22	#056743	1
23	#061	1
24	#062	1
25	#07	2

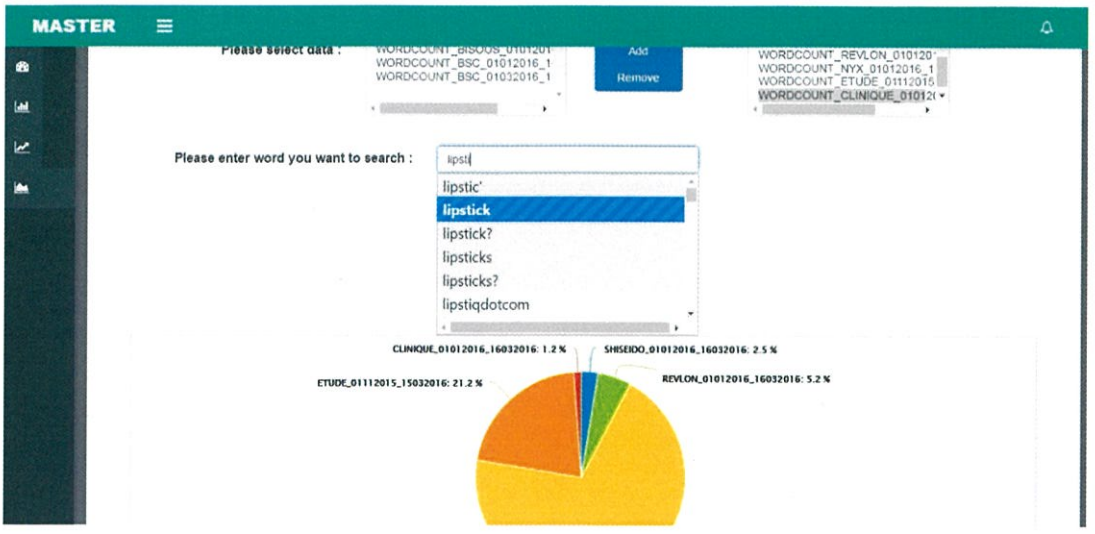
รูปที่ 4.18 รายละเอียดของไฟล์ part-r-00000

4.1.8 หน้าจอฟังก์ชันการแสดงผล

ฟังก์ชัน Output หรือหน้าจอแสดงผลลัพธ์ จะแสดงกราฟเปรียบเทียบและกราฟวิเคราะห์อารมณ์และความรู้สึก โดยในส่วนของกราฟเปรียบเทียบสามารถเลือกข้อมูลได้หลายๆตารางแล้วค้นหาคำที่ผู้ใช้สนใจ แสดงดังรูปที่ 4.19 และ 4.20

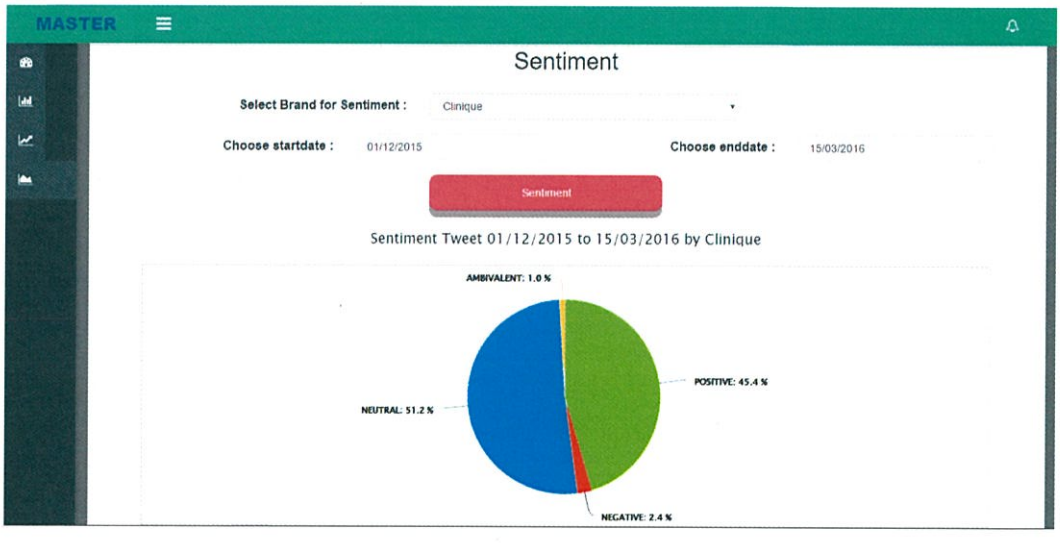


รูปที่ 4.19 หน้าจอแสดงผลลัพธ์กราฟเปรียบเทียบ



รูปที่ 4.20 ขั้นตอนการแสดงกราฟเปรียบเทียบ

ส่วนต่อมาคือส่วนของกราฟวิเคราะห์ความรู้สึกของข้อความ โดยจะเป็นการวิเคราะห์ความรู้สึกของข้อความว่าเป็นเชิงบวก เชิงลบ เป็นกลาง และคลุมเครือ ซึ่งจะแสดงผลลัพธ์ในรูปแบบของกราฟวงกลมแสดงดังรูปที่ 4.21



รูปที่ 4.21 หน้าจอแสดงผลลัพธ์อารมณ์และความรู้สึก

4.2 การดำเนินการของเว็บไซต์

โดยจะกล่าวถึงวิธีการดำเนินงานในขั้นตอนการโหลดข้อมูลและขั้นตอนการทำแมปรีดิวซ์

4.2.1 ขั้นตอนการโหลดข้อมูล

เป็นการโยนตารางไปเก็บในฐานข้อมูลของ BigSQL โดยมีการสร้างตารางสำหรับฮาดูป โดยเฉพาะเพื่อใช้ในการเก็บข้อมูล ซึ่งจะกำหนดให้เก็บข้อมูลเฉพาะภาษาอังกฤษและกำหนดช่วงเวลาของทวีตที่สนใจ ผลลัพธ์ที่ได้จะเป็นไฟล์ข้อมูลที่ถูกคัดกรองแล้วถูกเก็บไว้บนฮาดูปเพื่อรอการแมปรีดิวซ์แสดงดังรูปที่ 4.22

```
public static void pushDataTableHive(String table,String name,String start,String end) throws SQLException{
    String nameU = name.toUpperCase();
    Connection conn = null;
    Statement statement = null;
    Statement statement2 = null;
    String createTH = " CREATE HADOOP TABLE "+nameU+" ( msgBody varchar(2048) ) STORED AS TEXTFILE; ";
    String loadHadoop = " LOAD HADOOP USING JDBC CONNECTION URL '"+jdbcDb2+"' "
        + " WITH PARAMETERS (user='"+userDb2+"',password='"+passwordDb2+"') FROM SQL QUERY ' ";
    String query = "";
    query = query+"SELECT \"msgBody\" FROM DASH110966.\""+table+"\"";
    String condition = " where \"userLanguage\" = ''[en]'' OR \"userLanguage\" = ''[es]'' OR \"userLanguage\" = ''[pl]'' OR \"userLanguage\" = ''[fr]'' "
                + " AND \"msgPostedTime\" >=''"+start+"'' AND \"msgPostedTime\" <=''"+end+"'' AND $CONDITIONS' "
                + " SPLIT COLUMN msgBody INTO TABLE "+nameU+" overwrite "
                + " WITH LOAD PROPERTIES  ('num.map.tasks' = '1', 'max.rejected.records'='100') ";

    try{
        Class.forName(driverDb2);
        conn = DriverManager.getConnection(jdbcBigsql,userBigsql,passwordBigsql);
        //Create
        statement = conn.createStatement();
        statement.execute(createTH);
        //Load
        String sql = loadHadoop+query+condition;
        statement2 = conn.createStatement();
        statement2.execute(sql);
        System.out.println("SuccessToPushHadoop");
    }
     catch (Exception e) {
          e.printStackTrace();
        }
      finally{
          if(conn != null){
                conn.close();
            }
          if(statement != null)
            {
                statement.close();
            }
      }
}
```

รูปที่ 4.22 โปรแกรมขั้นตอนการโหลดข้อมูล

4.2.2 ขั้นตอนการทำแมปรีดิวซ์

การกำหนดค่าเส้นทาง ชื่อผู้ใช้ รหัสและไดเรกทอรี่ที่จะใช้ในการแมปรีดิวซ์บนฮาดูป แสดงดังรูปที่ 4.23

```
class Mapreduce{
    def count(String fromdir,String dir){
        // Host, userid & password taken from VCAP_Services
        String gateway = "https://bi-hadoop-prod-2286.services.dal.bluemix.net:8443/gateway/default"
        String username = "biblumix"
        String password = "aO4RQo@0Nh7~"

        String jobDir = "/user/" + username + "/" + dir
```

รูปที่ 4.23 โปรแกรมการกำหนดค่าเส้นทางต่างๆ

การกำหนดค่าให้กับ Job Tracker ซึ่งทำหน้าที่เป็นตัวกำหนดงานให้กับแต่ละ Task Tracker โดยพัฒนาด้วยภาษากรูวี (Groovy Language) แสดงดังรูปที่ 4.24 และ 4.25

```
// i) Identify and define the core parameters for workflow

String definition = """\
<workflow-app xmlns="uri:oozie:workflow:0.2" name="wordcount-workflow">
    <start to="root-node"/>
    <action name="root-node">
        <java>
            <job-tracker>\${jobTracker}</job-tracker>
            <name-node>\${nameNode}</name-node>
            <main-class>org.apache.hadoop.examples.WordCount</main-class>
            <arg>\${inputDir}</arg>
            <arg>\${outputDir}</arg>
        </java>
        <ok to="end"/>
        <error to="fail"/>
    </action>
    <kill name="fail">
        <message>Java failed, error message[\${wf:errorMessage(wf:lastErrorNode())}]</message>
    </kill>
    <end name="end"/>
</workflow-app>
"""
```

รูปที่ 4.24 โปรแกรมการกำหนดค่าให้กับ Job Tracker (1)

```
// ii) Provide suitable values to these parameters
String configuration = """\
<configuration>
    <property>
        <name>user.name</name>
        <value>default</value>
    </property>
    <property>
        <name>nameNode</name>
        <value>default</value>
    </property>
    <property>
        <name>jobTracker</name>
        <value>default</value>
    </property>
    <property>
        <name>inputDir</name>
        <value>$jobDir/input</value>
    </property>
    <property>
        <name>outputDir</name>
        <value>$jobDir/output</value>
    </property>
    <property>
        <name>oozie.wf.application.path</name>
        <value>$jobDir</value>
    </property>
</configuration>
"""
```

รูปที่ 4.25 โปรแกรมการกำหนดค่าให้กับ Job Tracker (2)

การเริ่มการทำงานของฮาดูปเริ่มจากการล็อกอินเข้าสู่ฮาดูป จากนั้นทำการลบโฟลเดอร์แล้วสร้างขึ้นใหม่ และทำการอ่านรายละเอียดของไฟล์ที่ถูกโยนขึ้นมาในไดเรกทอรี่ /app/hive/warehouse/biblumix.db/ ชื่อไฟล์ที่ถูกโยนไปยังฮาดูป แล้วนำรายละเอียดที่ได้เป็นไฟล์ JSON มาค้นหาชื่อไฟล์ ซึ่งไฟล์นี้จะถูกสร้างขึ้นใหม่แบบไม่ซ้ำ แล้วเก็บค่าในตัวแปร pathFilename แสดงดังรูปที่ 4.26

```
// create session
Hadoop session = Hadoop.login( gateway, username, password )
println "\nSession opened. "

// iii) Create folders and upload necessary files in HDFS thru WebHDFS API

// Cleanup and recreate /user/biblumix/testdir folder
println "\nDrop folder - " + jobDir + ": " + Hdfs.rm( session ).file( jobDir ).recursive().now().statusCode
println "\nCreate folder - " + jobDir + ": " + Hdfs.mkdir( session ).dir( jobDir ).now().statusCode

String fromdirLo = fromdir.toLowerCase();
String findFile = Hdfs.ls( session ).dir( "/apps/hive/warehouse/biblumix.db/"+ fromdirLo ).now().string

JSONObject json1 = new JSONObject(findFile);
String json2 = json1.getJSONObject("FileStatuses");
JSONObject json3 = new JSONObject(json2);
String json4 = json3.getJSONArray("FileStatus");
JSONArray json5 = new JSONArray(json4);
JSONObject json6 = json5.getJSONObject(0);
String pathFilename = json6.getString("pathSuffix");
```

รูปที่ 4.26 โปรแกรมการเริ่มทำงานของฮาดูป

การนำไฟล์ที่ถูกโยนขึ้นไปบนฮาดูปมาตัดคำ ตัดตัวอักษรที่ไม่จำเป็นต่อการแมปรีดิวซ์แสดงดังรูปที่ 4.26 และทำการอัพโหลดไฟล์ที่พร้อมจะเริ่มทำกระบวนการแมปรีดิวซ์ขึ้นไปบนฮาดูป ซึ่งจะมี 3 ไฟล์ คือ ไฟล์ทวีตที่เป็นไฟล์ต้นฉบับยังไม่ผ่านกระบวนการต่างๆ ไฟล์ไลบรารี่และไฟล์ XML ที่ใช้เป็นการกำหนด Job Tracker และ Name Node จากนั้นจะเป็นการเริ่มกระบวนการแมปรีดิวซ์และไฟล์ผลลัพธ์ที่ได้จะถูกนำขึ้นไปบนฮาดูปเช่นกัน แสดงดังรูปที่ 4.27

```
String inputStringAll = Hdfs.get( session ).from( "/apps/hive/warehouse/biblumix.db/" + fromdirLo + "/" + pathFilename ).now().string
String a = inputStringAll.toLowerCase();
String b = a.replaceAll("htt.*? ", "");
String c = b.replaceAll("htt.*?\$", "");
String d = c.replace("[__]"," ");
String e = d.replaceAll("[!.,;?\"()+-=€~@'‘~&•%0123456789#@{};“”#]"," ");
String f = e.replace("[","");
String g = f.replace("]","");
String h = g.replace("|","");
String i = h.replace("*","");
String j = i.replace("\$","");
String k = j.replace(":","");
String l = k.replace(" rt ","");
String m = l.replace("/","");
String n = new String(m.getBytes("ISO_8859_1"), "UTF-8");
String cleanString = n.replace("?"," ");

// Uploading TweetText.txt with different name - SampleFile.txt
Hdfs.put(session).text( cleanString ).to( jobDir + "/input/SourceFile.txt" ).now()
// Uploading jar file
String jarFile = Hdfs.get( session ).from( "/user/biblumix/HadoopJar/hadoop-examples.jar" ).now().string
Hdfs.put(session).text( jarFile ).to( jobDir + "/lib/hadoop-examples.jar" ).now()
// Uploading workflow.xml file
Hdfs.put(session).text( definition ).to( jobDir + "/workflow.xml" ).now()

// iv) Run Oozie workflow
String jobId = Workflow.submit(session).text( configuration ).now().jobId
String status = "RUNNING";
int count = 0;
while( status == "RUNNING" && count++ < 60 ) {
  sleep( 1000 )
  String json = Workflow.status(session).jobId( jobId ).now().string
  status = JsonPath.read( json, "\$.status" )
  print "."; System.out.flush();
}
session.shutdown()
```

รูปที่ 4.27 โปรแกรมการนำไฟล์ที่ถูกนำขึ้นไปบนฮาดูปมาตัดคำ

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

เนื่องมาจากการเพิ่มขึ้นของปริมาณข้อมูลในปัจจุบันจากโลกออนไลน์ต่างๆ ไม่ว่าจะเป็น Facebook, Twitter เป็นต้น คณะผู้จัดทำจึงเล็งเห็นความสำคัญของข้อมูลบนโลกออนไลน์เหล่านั้นว่าสามารถทำให้เกิดเป็นข้อมูลที่มีมูลค่าเพิ่มมากขึ้นได้ จึงเกิดเป็นเว็บไซต์นี้ขึ้นมา โดยในการทำปัญหาพิเศษครั้งนี้ได้ใช้ Cloud Storage ชื่อว่า IBM Bluemix ในการศึกษาทดลองและใช้ฐานข้อมูลที่ชื่อว่า DashDB, BigSQL และ Hive เป็นแหล่งเก็บข้อมูลที่อยู่บนฮาดูป ซึ่งคณะผู้จัดทำได้ศึกษาเกี่ยวกับหลักการเขียนภาษาของฐานข้อมูลชนิดต่างๆข้างต้น ว่ามีหลักการเขียนภาษาอย่างไรและได้ศึกษาการทำงานของฮาดูปว่าภายในทำงานอย่างไร รวมไปถึงการศึกษาการนำข้อมูลขึ้นไปเก็บไว้บนฮาดูปและการทำแมปรีดิวซ์ (กระบวนการนับคำบนฮาดูป) โดยปกติถ้าต้องการนำข้อมูลต่างๆไปเก็บไว้ในฮาดูป รวมไปถึงการทำแมปรีดิวซ์ จะต้องใช้คำสั่งที่เป็นคำสั่งเฉพาะในการเข้าถึงฮาดูปโดยผ่านหน้าจอ Command Line ซึ่งเป็นวิธีที่ยุ่งยากซับซ้อนเกินไปสำหรับผู้ใช้ปกติ โดยคณะผู้จัดทำได้ทำการศึกษาและออกแบบเว็บไซต์เพื่อให้เกิดความสะดวกต่อการใช้งาน จึงเกิดเป็นฟังก์ชันหลักทั้งหมด 4 ฟังก์ชัน ได้แก่ ฟังก์ชันกรองคำจากทวิตเตอร์ (Filter Data), ฟังก์ชันโหลดข้อมูลไปยังฮาดูป (Load Data), ฟังก์ชันแมปรีดิวซ์ (MapReduce) และฟังก์ชันการแสดงผล (Output) ทำให้ผู้ใช้สามารถเข้าถึงฮาดูปได้มากขึ้น โดยหัวใจหลักของเว็บไซต์คือจะสามารถทำอย่างไรให้ข้อมูลที่อยู่บนโลกออนไลน์สามารถเปลี่ยนเป็นข้อมูลที่มีประโยชน์ได้ ซึ่งช่วยให้ผู้ใช้สามารถนำไปประกอบการตัดสินใจในทางธุรกิจหรือในทางต่างๆได้

5.2 ปัญหาที่พบและข้อจำกัด

1) บริการที่ใช้ไม่มีความเสถียรเพราะมีการปรับปรุงแก้ไขบ่อยครั้ง จึงทำให้เกิดความล่าช้าในการทำงาน
2) ซอฟต์แวร์ IBM Bluemix ที่ใช้มีระยะเวลาในการใช้งานเพียง 1 เดือนซึ่งเป็นเพียงตัวทดลองและสามารถใช้งานได้เพียงโหนดเดียวเท่านั้น
3) ปัญหาเรื่อง Proxy เนื่องจากใช้อินเทอร์เน็ตของทางมหาวิทยาลัยทำให้ไม่สามารถเข้าถึงบริการบางอย่างได้
4) เนื่องจากบริการที่ใช้เป็นตัวฟรีซึ่งจำลองการใช้งานฮาดูปเพียง 1 โหนด ทำให้กระบวนการต่างๆบนฮาดูปประมวลผลได้ค่อนข้างช้า
5) บริการที่ใช้ถูกยกเลิกในระหว่างการพัฒนาโปรแกรม

6) คณะผู้จัดทำใช้เวลาศึกษาข้อมูลเรื่องต่างๆเป็นเวลานาน เพราะเป็นความรู้ใหม่เกือบทั้งหมด

5.3 แนวทางในการพัฒนาต่อ

1) ฟังก์ชันกรองคำจากทวิตเตอร์สามารถนำไปพัฒนาต่อให้มีการประมวลแบบเรียลไทม์
2) เพิ่มแหล่งข้อมูลที่นำมาใช้วิเคราะห์ โดยจากเดิมใช้เพียงแค่ทวิตเตอร์
3) ติดตั้งฮาดูปที่เป็นตัวฟรีเพื่อใช้ในการพัฒนา จะสามารถทำให้การประมวลผลมีความรวดเร็วขึ้น
4) พัฒนาคลาสวิเคราะห์อารมณ์และความรู้สึก (Sentiment Analysis) ขึ้นด้วยตนเองโดยจากเดิมใช้คลาสใน IBM Bluemix

เอกสารอ้างอิง

[1] สุกิจ คูชัยสิทธิ์. 2556. **การเข้าสู่โลกยุคใหม่ของข้อมูล "บิ๊กดาต้า".** วารสารนักบริหาร ปีที่ 33, ฉบับที่ 1 (มกราคม-มีนาคม 2556), หน้า 22-28

[2] ximplesoft.com. **Is Hadoop Suitable for BigData?.** [Online]. Available: http://www.ximplesoft.com/blog/219. เข้าถึงเมื่อวันที่ 8 ต.ค. 58.

[3] รวีรัตน์ จตุราพิศพรชัย และลลนาวัลย์ มนตรีธนสาร. 2555. **การพัฒนาฐานข้อมูลขนาดใหญ่ด้วยฮาดูป.** ปริญญานิพนธ์วิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

[4] จิตกร พิทักษ์เมธากุล. **การติดตั้ง apache hadoop 1.2.1 (multi-node cluster) บน Ubuntu 12.04 server.** [Online]. Available: http://na5cent.blogspot.com/2013/11/apache-hadoop-121-multi-node-cluster.html. เข้าถึงเมื่อวันที่ 9 ต.ค. 58.

[5] สุวรรณี ธูปจีน. **เทคนิคการเพิ่มประสิทธิภาพบนกรอบการทำงานของ Map/Reduce.** [Online]. Available: http://it.kmutnb.ac.th/teacher/DrSucha215255811392.pdf. เข้าถึงเมื่อวันที่ 10 ต.ค. 58.

[6] cyberthai.com. **การบริหารจัดการ Big Data.** [Online]. Available: http://www.cyberthai.com/index.php/knowledge-center/96-5-big-data. เข้าถึงเมื่อวันที่ 8 ต.ค. 58.

[7] Thaiopensource.org. **มาเล่น Bluemix บริการ PaaS จาก IBM กัน.** [Online]. Available: http://thaiopensource.org/% เข้าถึงเมื่อวันที่ 15 ก.พ. 59.

[8] ประดับเก่ง. **Introduction to Java Servlet (in depth).** [Online]. Available: http://www.jarticles.com/tutorials/servlet/intro_servlet.html. เข้าถึงเมื่อวันที่ 20 ม.ค. 59.

[9] Suchada. **ภาษา JSP.** [Online]. Available: http://suchada51122470136.blogspot.com/. เข้าถึงเมื่อวันที่ 20 ม.ค. 59.

[10] Aun Thanongchai. **มารู้จัก ภาษา Groovy กันดีกว่า.** [Online]. Available: http://aun-tt.blogspot.com/2012/07/groovy.html. เข้าถึงเมื่อวันที่ 31 ม.ค. 59.

[11] วิกิพีเดีย สารานุกรมเสรี. **ภาษากรูวี.** [Online]. Available: https://th.wikipedia.org/wiki/%. เข้าถึงเมื่อวันที่ 30 มี.ค. 59.

ภาคผนวก

ภาคผนวก ก.
คู่มือการใช้งาน

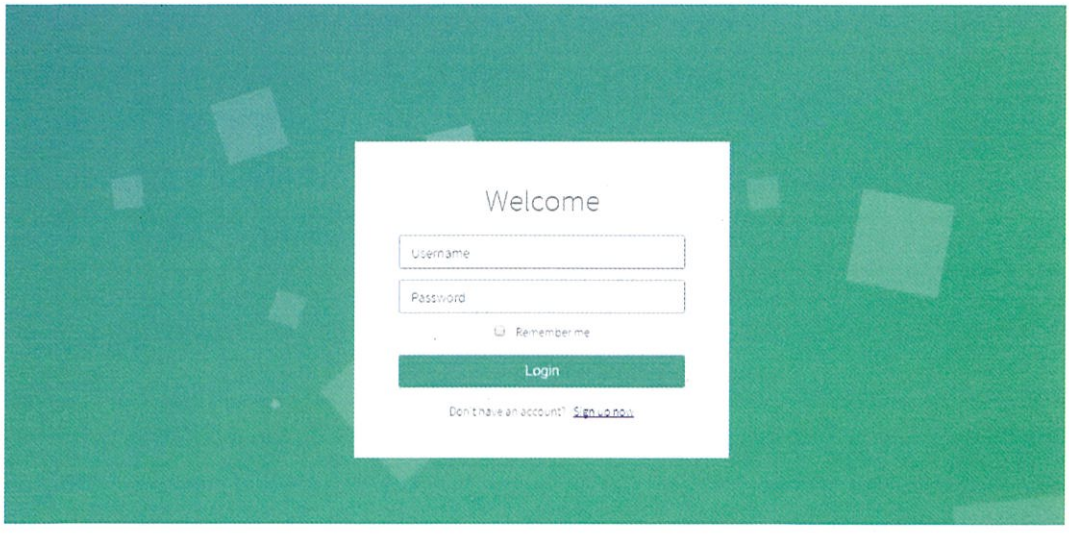
ก.1 หน้าจอแรกของเว็บไซต์

หน้าจอแรกของเว็บไซต์จะแสดงรายละเอียดของเว็บไซต์ โดยจะมีทั้งหมด 7 เมนู คือ Home, About, Filter Load, MapReduce, Output และ Contact ผู้ใช้สามารถคลิกเลือกไปยังเมนูต่างๆ เพื่ออ่านรายละเอียดของเว็บไซต์ และเมื่อผู้ใช้อ่านรายละเอียดบนหน้าเว็บไซต์เรียบร้อยแล้ว ให้ผู้ใช้กดปุ่ม GET STARTED เพื่อเป็นการเข้าสู่เว็บไซต์ แสดงดังรูปที่ ก.1

รูปที่ ก.1 หน้าแรกของเว็บไซต์

ก.2 หน้าจอเข้าสู่ระบบของเว็บไซต์

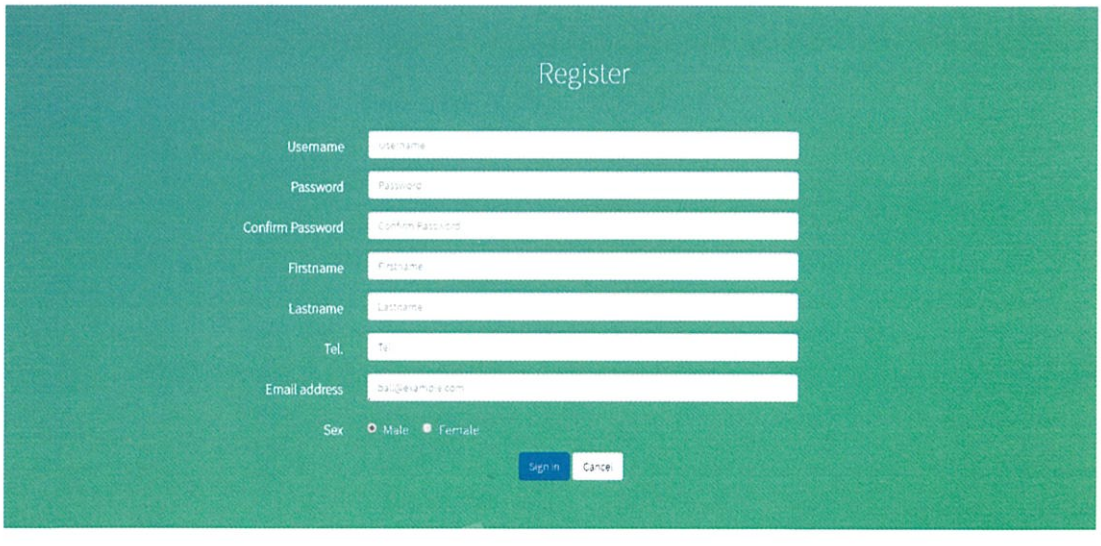
ก่อนที่ผู้ใช้ที่เข้าสู่เว็บไซต์ได้ ผู้ใช้ต้องทำการล็อกอินเข้าสู่ระบบก่อน โดยกรอก Username และ Password จากนั้นกดปุ่ม Login แต่หากเป็นการใช้งานครั้งแรกให้ผู้ใช้กดที่ปุ่ม Sign up now เพื่อเป็นการลงทะเบียนเข้าสู่เว็บไซต์ แสดงดังรูปที่ ก.2



รูปที่ ก.2 หน้าจอเข้าสู่ระบบของเว็บไซต์

ก.3 หน้าจอการลงทะเบียนเพื่อเข้าใช้งานระบบ

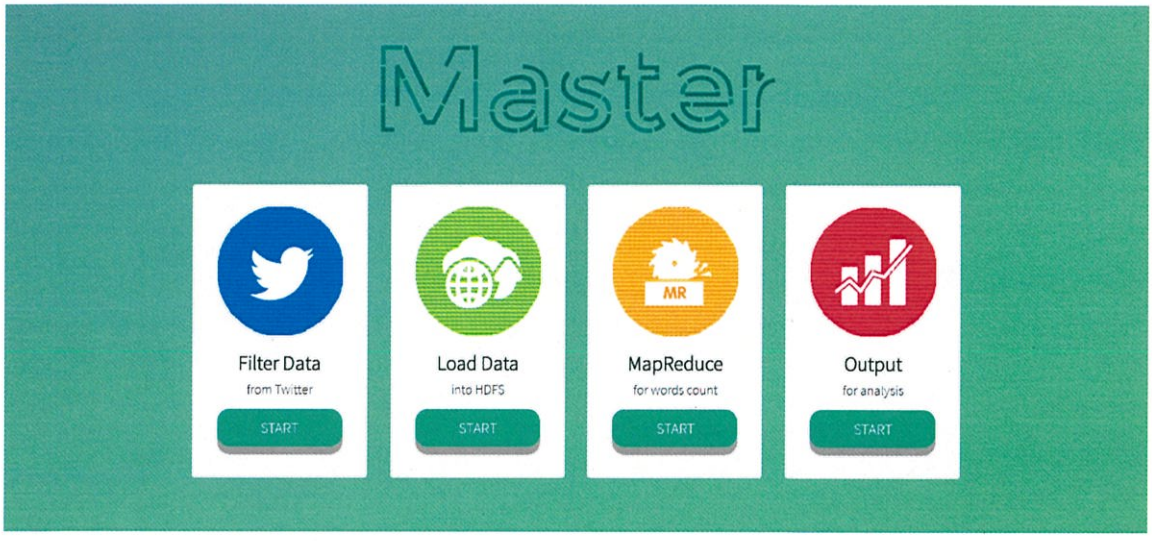
หน้าจอลงทะเบียนสำหรับผู้ใช้ที่เข้าใช้งานเว็บไซต์เป็นครั้งแรก โดยเมื่อผู้ใช้กรอกรายละเอียดเสร็จเรียบร้อยแล้วให้กดที่ปุ่ม Sign in เพื่อเป็นการลงทะเบียนเพื่อเข้าใช้งานเว็บไซต์ แสดงดังรูปที่ ก.3



รูปที่ ก.3 หน้าจอลงทะเบียนของเว็บไซต์

ก.4 หน้าจอเมนูหลักของเว็บไซต์

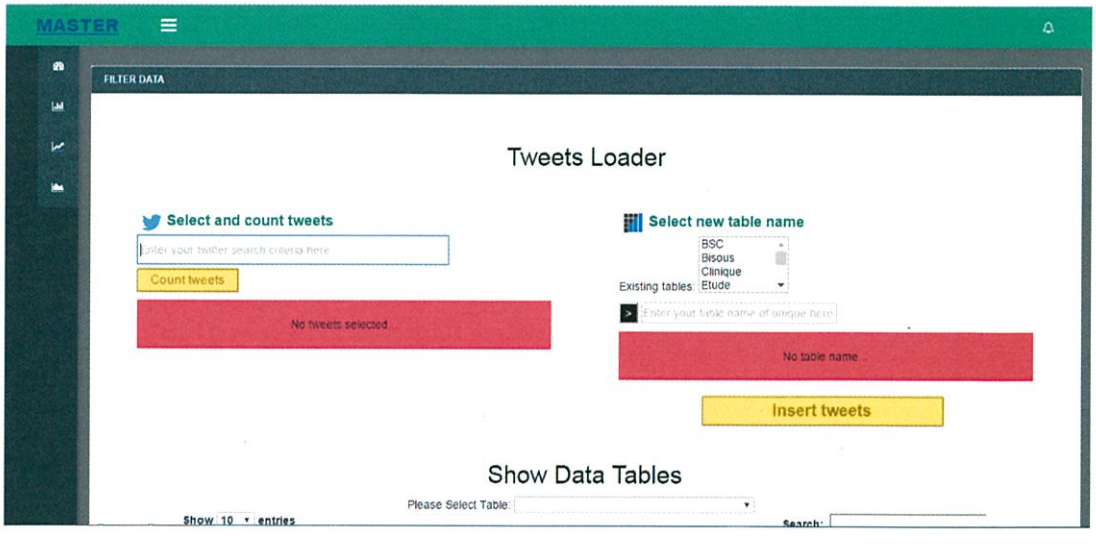
เมื่อเข้าสู่เว็บไซต์จะแสดงหน้าจอฟังก์ชันหลักประกอบไปด้วย 4 ฟังก์ชันคือ Filter Data, Load Data, MapReduce และ Output หากผู้ใช้ต้องการทำในทุกฟังก์ชันจะต้องเรียงตามลำดับข้างต้น จากนั้นกดปุ่ม START เพื่อเป็นการเริ่มกระบวนการ แสดงดังรูปที่ ก.4



รูปที่ ก.4 หน้าจอเมนูหลักของเว็บไซต์

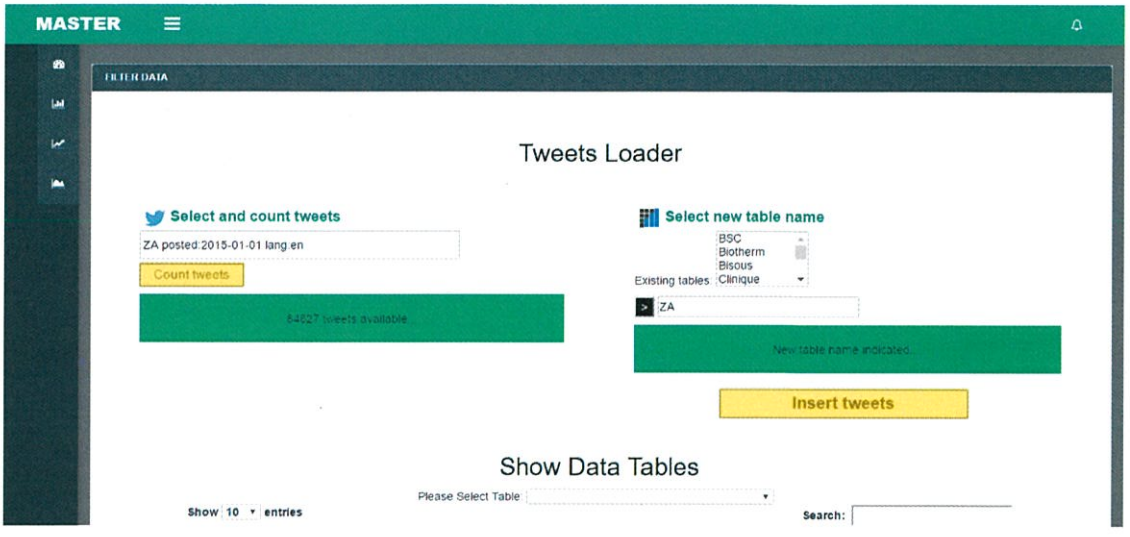
ก.5 หน้าจอฟังก์ชันกรองคำจากทวิตเตอร์

ฟังก์ชัน Filter Data หรือขั้นตอนการกรองคำ โดยจะใช้แหล่งข้อมูลจากทวิตเตอร์ ซึ่งฟังก์ชันนี้เป็นฟังก์ชันที่ให้ผู้ใช้สามารถระบุคำที่ต้องการกรองลงไป โดยเริ่มต้นที่ Select and count tweets แสดงดังรูปที่ ก.5



รูปที่ ก.5 หน้าจอฟังก์ชันการกรองคำ

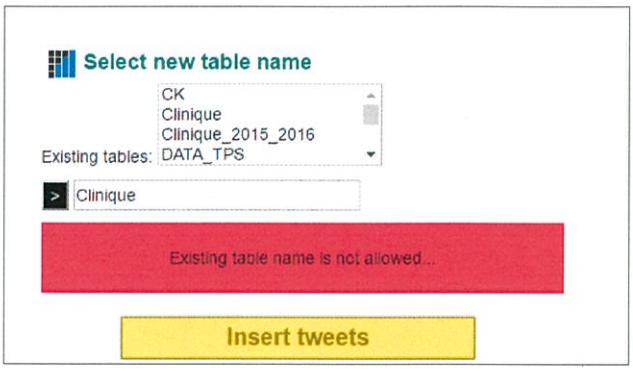
รูปที่ ก.6 จะใช้คำว่า “ZA” เป็นคำตัวอย่างในการกรองคำ โดยมีคำสั่งในการกรองคำคือ “ZA posted 2015-01-01 lang en” ซึ่งมีหมายความว่าจะเลือกเอาทวีตที่มีคำว่า ZA อยู่ตั้งแต่วันที่ 1 มกราคม 2015 จนถึงปัจจุบันและเลือกเฉพาะภาษาอังกฤษ (แต่ก็จะมีปะปนภาษาอื่นมาบ้างเพราะว่าผู้ใช้ทวิตเตอร์บางคนใส่ไม่ตรงตามความจริง) เมื่อพิมพ์คำสั่งข้างต้นเสร็จแล้วให้กดปุ่ม Count tweets จากนั้นโปรแกรมจะขึ้นปุ่มสีเขียวเพื่อแสดงจำนวนทวีตว่ามีคนพูดถึงจำนวนเท่าใด แสดงดังรูปที่ ก.6



รูปที่ ก.6 ขั้นตอนการกรองคำ

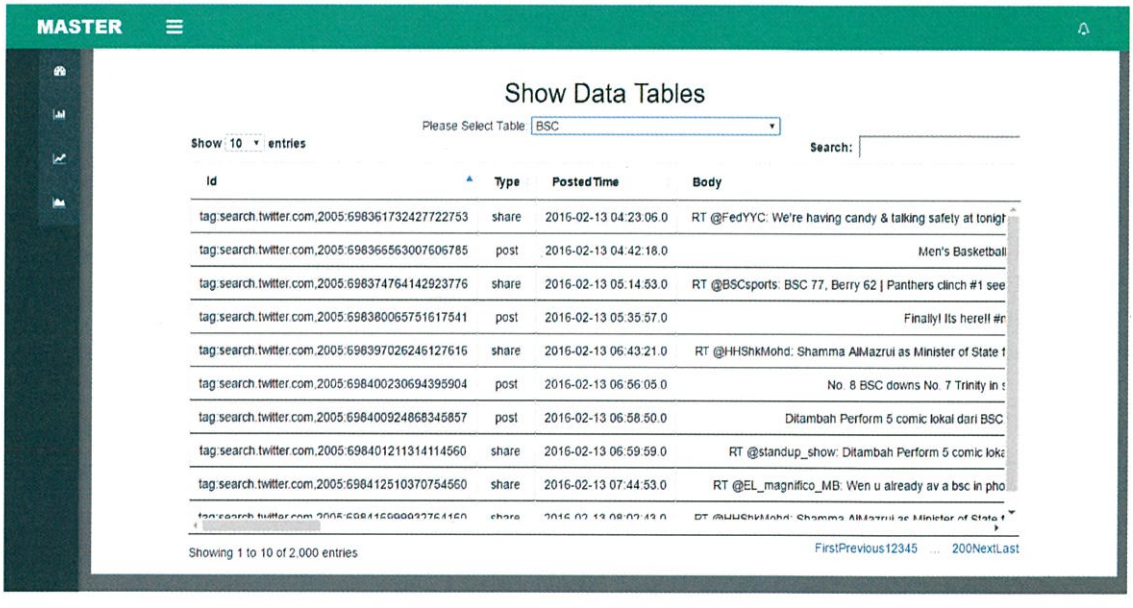
จากนั้นทำการตั้งชื่อตาราง และเพื่อให้เกิดความสะดวกในการใช้งานควรตั้งชื่อเป็นชื่อเดียวกับคำที่ใช้กรอง โดยในที่นี้ตั้งชื่อตารางว่า ZA หากชื่อตารางไม่ซ้ำ โปรแกรมก็จะขึ้นปุ่มสีเขียวเพื่อบอกว่าสามารถใช้ชื่อตารางนี้ได้ แสดงดังรูปที่ ก.7 แต่ถ้าชื่อตารางซ้ำกับตารางเดิมที่มีอยู่แล้ว โปรแกรมจะขึ้นแสดงให้เห็นเป็นปุ่มสีแดง แสดงดังรูปที่ ก.8 จากนั้นกดปุ่ม Insert tweets

รูปที่ ก.7 การสร้างตารางที่สามารถใช้งานได้



รูปที่ ก.8 การสร้างตารางที่ไม่สามารถใช้งานได้

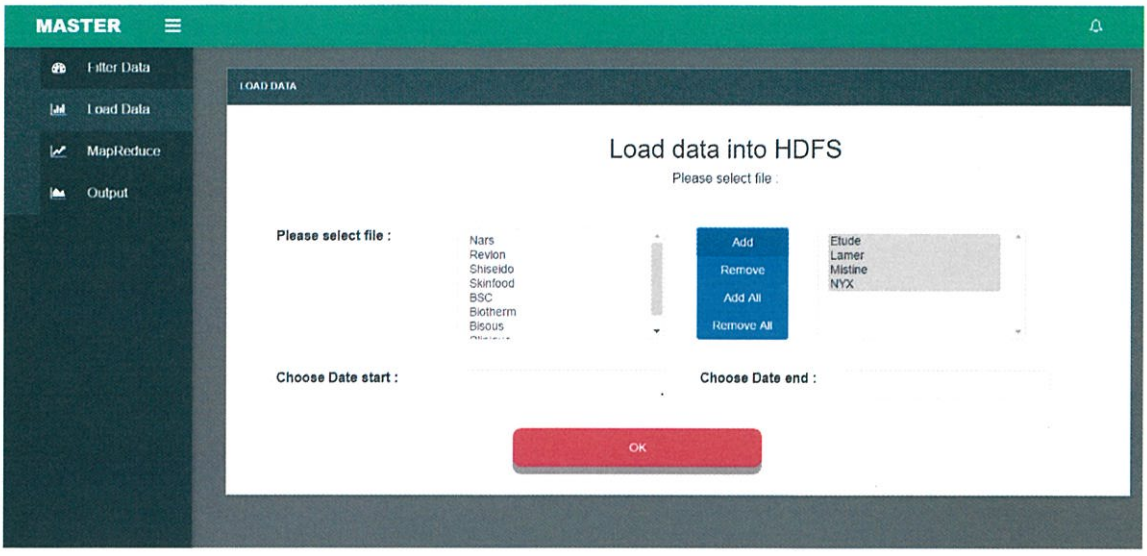
เมื่อทำการกรองคำเสร็จเรียบร้อยแล้ว สามารถคลิกเลือกชื่อตารางตามต้องการได้ เพื่อเลือกดูรายละเอียดของตารางนั้น โดยตารางที่ได้มาจะได้มาจากตอนที่ทำการกรองคำข้างต้น แสดงดังรูปที่ ก.9



รูปที่ ก.9 รายละเอียดของข้อมูลที่ได้กรองมาจากทวิตเตอร์

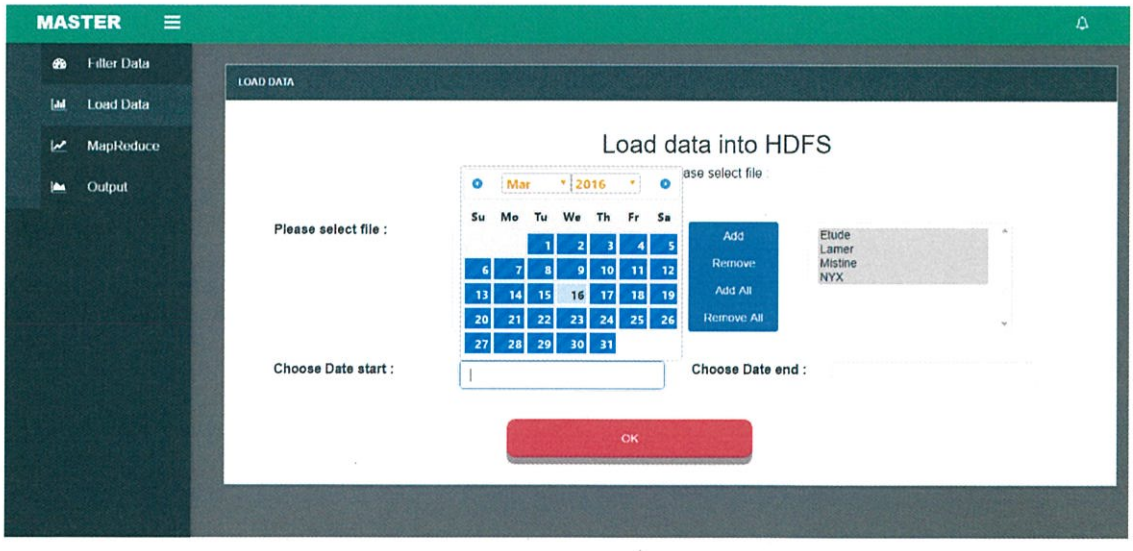
ก.6 หน้าจอฟังก์ชันโหลดข้อมูลไปยังฮาดูป

ฟังก์ชัน Load Data หรือขั้นตอนการโหลดข้อมูลนำไปเก็บไว้บนฮาดูป (HDFS) ผู้ใช้สามารถเลือกตารางที่ต้องการจะไปเก็บบนฮาดูป โดยคลิกที่ปุ่ม Add โดยสามารถเลือกได้หลายๆตารางในการโหลด 1 ครั้ง แสดงดังรูปที่ ก.10



รูปที่ ก.10 ขั้นตอนการโหลดข้อมูลไปเก็บไว้บนฮาดูป (1)

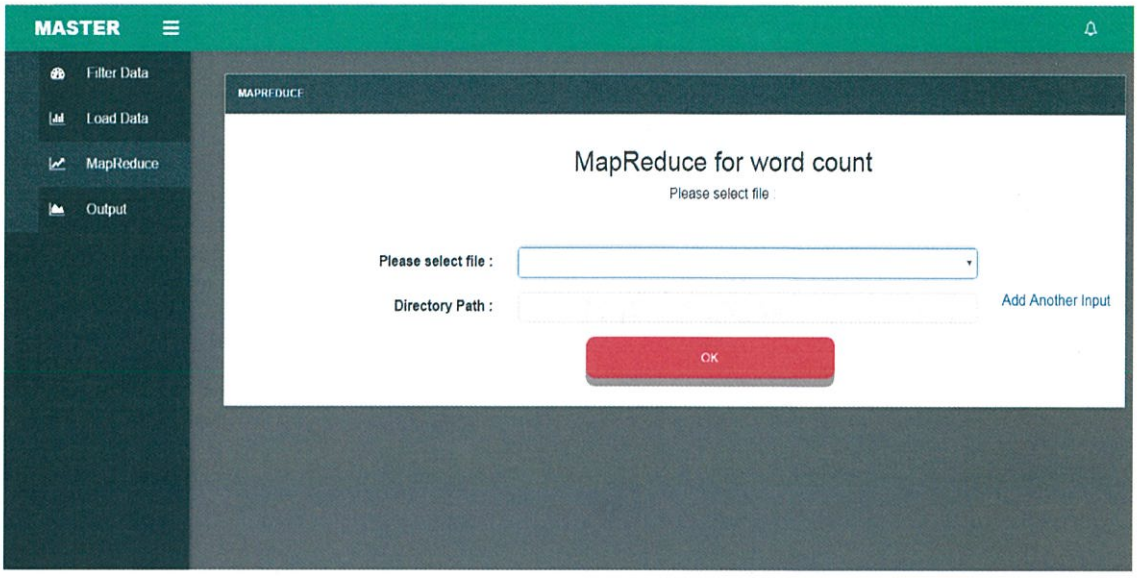
โดยจะต้องเลือกช่วงเวลาที่ต้องการ โดยระบุวันเริ่มต้นและวันสิ้นสุดให้กับกระบวนการนี้ จากนั้นกดปุ่ม OK เพื่อเป็นการเริ่มต้นการโหลดข้อมูล แสดงดังรูปที่ ก.11



รูปที่ ก.11 ขั้นตอนการโหลดข้อมูลไปเก็บไว้บนฮาดูป (2)

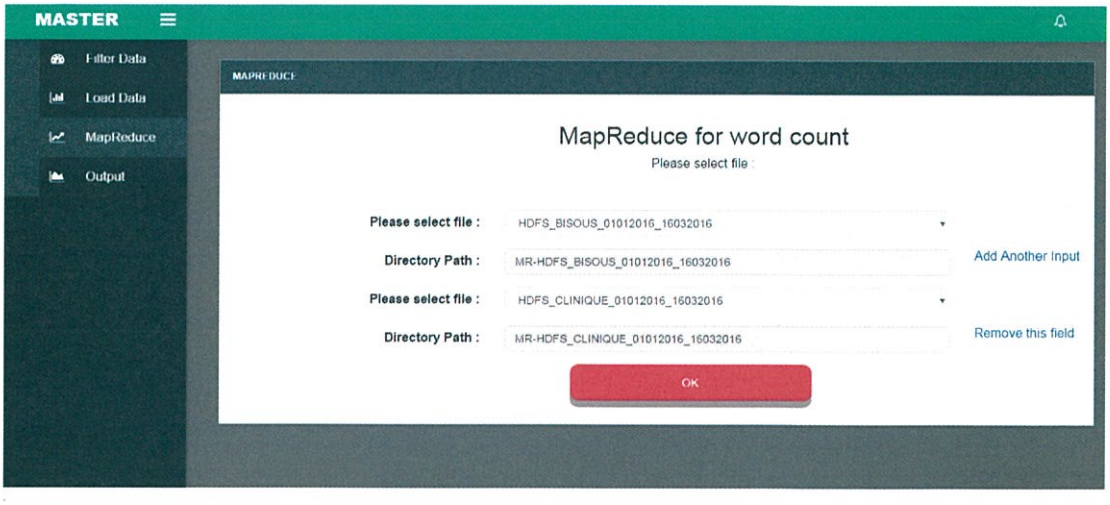
ก.7 หน้าจอฟังก์ชันแมปรีดิวซ์

ฟังก์ชัน MapReduce หรือขั้นตอนการตัดคำ (Word Count) โดยขั้นตอนนี้จะนำไฟล์ที่เก็บอยู่บนฮาดูปนำมาทำการแมปรีดิซ์ เพื่อเป็นการนับคำ แสดงดังรูปที่ ก.12



รูปที่ ข.12 หน้าจอฟังก์ชันแมปรีดิวซ์

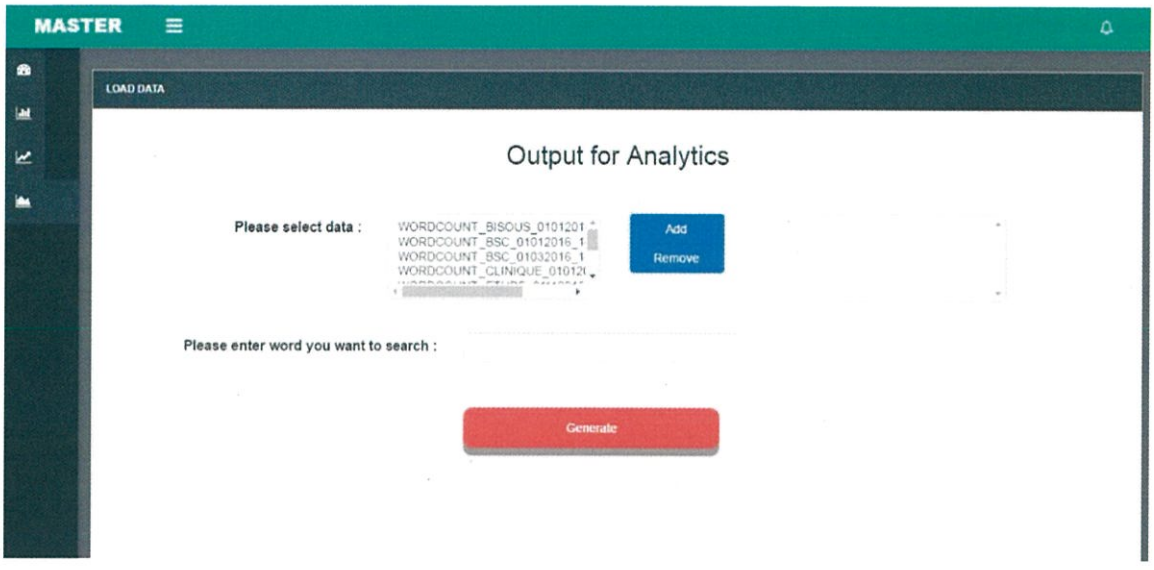
ผู้ใช้สามารถกดปุ่ม Add Another Input เพื่อทำการเพิ่มตารางที่ต้องการทำแมปรีดิวซ์ โดยสามารถเพิ่มได้หลายๆตาราง จากนั้นกดปุ่ม OK เพื่อเป็นการเริ่มต้นกระบวนการแมปรีดิวซ์ แสดงดังรูปที่ ข.13



รูปที่ ข.13 ขั้นตอนการทำแมปรีดิวซ์

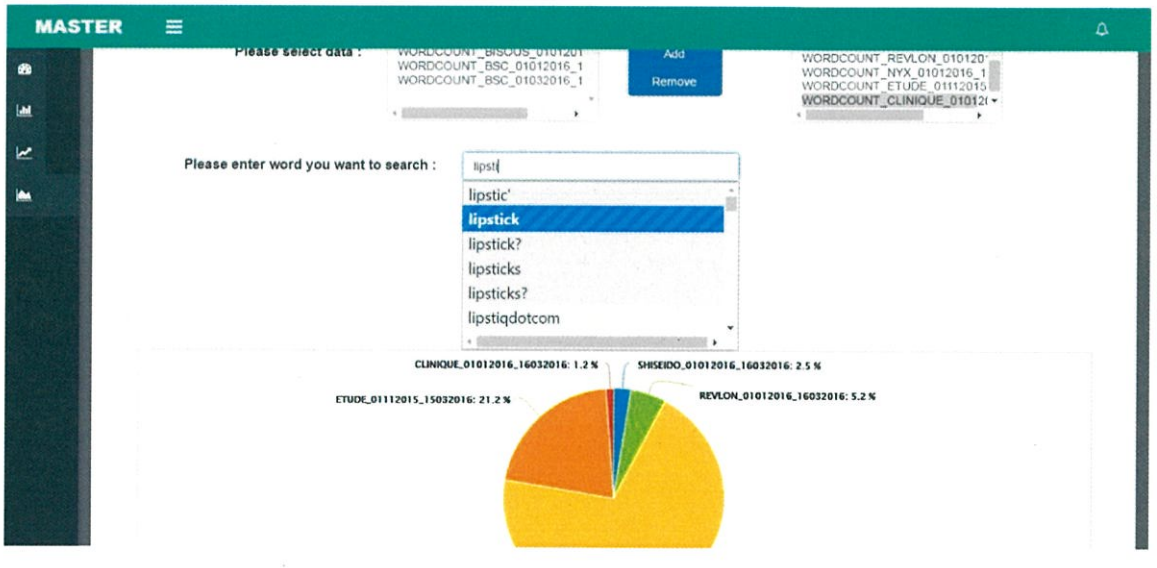
ก.8 หน้าจอฟังก์ชันการแสดงผล

ฟังก์ชัน Output หรือหน้าแสดงผลลัพธ์ โดยหน้าจอนี้จะแสดงกราฟเปรียบเทียบแบรนด์เครื่องสำอางและกราฟวิเคราะห์อารมณ์และความรู้สึก เพื่อเป็นส่วนหนึ่งที่ช่วยให้ผู้ใช้นำไปใช้ในการประกอบการตัดสินใจในด้านธุรกิจหรือด้านต่างๆได้ แสดงดังรูปที่ ก.14



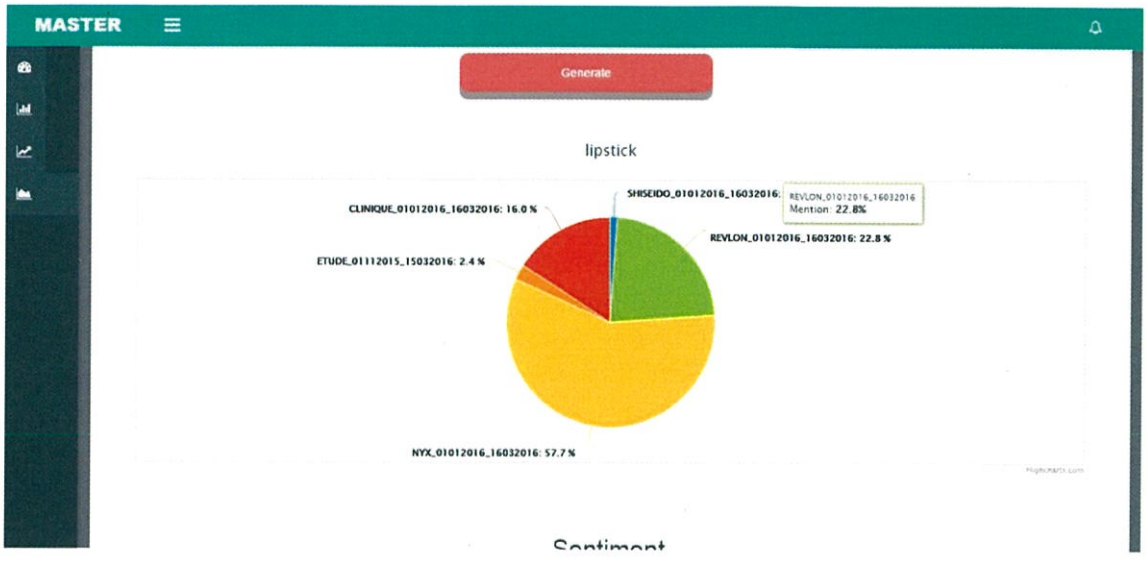
รูปที่ ก.14 หน้าจอแสดงกราฟเปรียบเทียบแบรนด์เครื่องสำอาง

ผู้ใช้สามารถเลือกตารางได้หลายๆตาราง โดยการกดปุ่ม Add แต่หากต้องการลบสามารถกดที่ปุ่ม Remove และพิมพ์คำที่ต้องการ โดยจะมีตัวอย่างคำค้นหาเพื่อช่วยให้ผู้ใช้สามารถพิมพ์คำได้สะดวกยิ่งขึ้น จากนั้นกด Generate แสดงดังรูปที่ ก.15



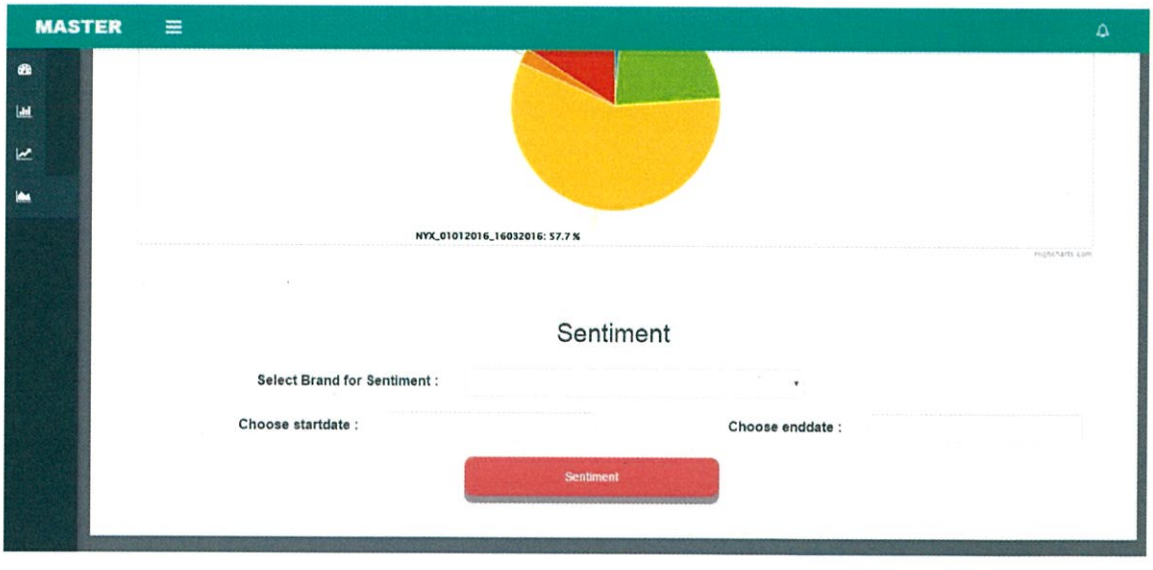
รูปที่ ก.15 ขั้นตอนการแสดงกราฟเปรียบเทียบ

เมื่อกดปุ่ม Generate ผลลัพธ์จะออกมาในรูปแบบของกราฟวงกลม โดยจะแสดงให้เห็นว่าแบรนด์ใดมีอันดับสูงสุด โดยแยกตามสีและคิดผลลัพธ์ออกมาเป็นเปอร์เซ็นต์ แสดงดังรูปที่ ก.16



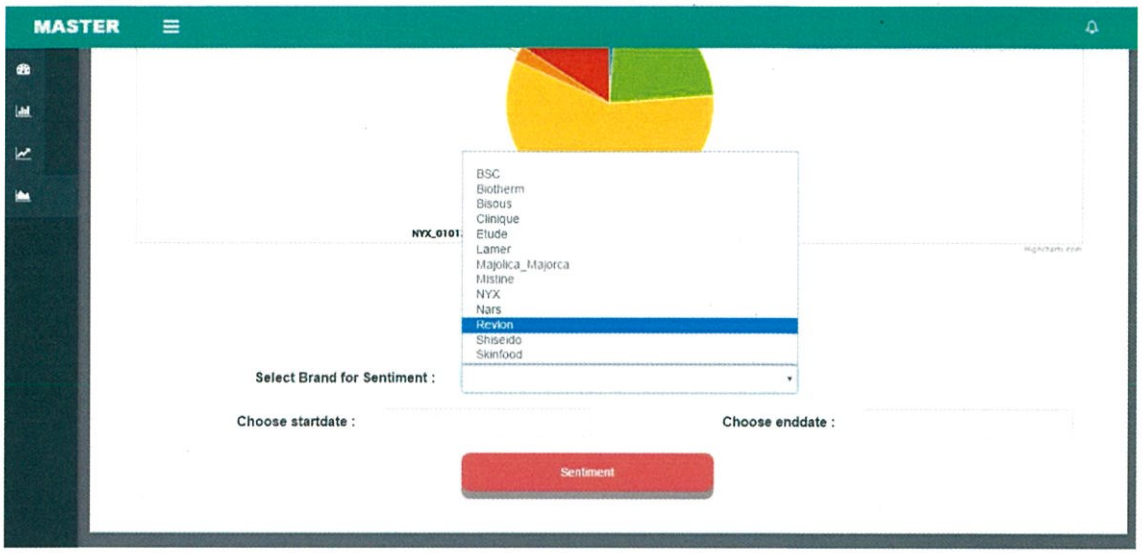
รูปที่ ก.16 หน้าจอแสดงกราฟเปรียบเทียบ

ส่วนต่อมาคือส่วนของการวิเคราะห์ความรู้สึกของข้อความ โดยจะวิเคราะห์ความรู้สึกของข้อความว่าเป็นเชิงบวก เชิงลบ เป็นกลาง และคลุมเครือ ซึ่งจะแสดงผลลัพธ์ในรูปแบบของกราฟวงกลม แสดงดังรูปที่ ก.17

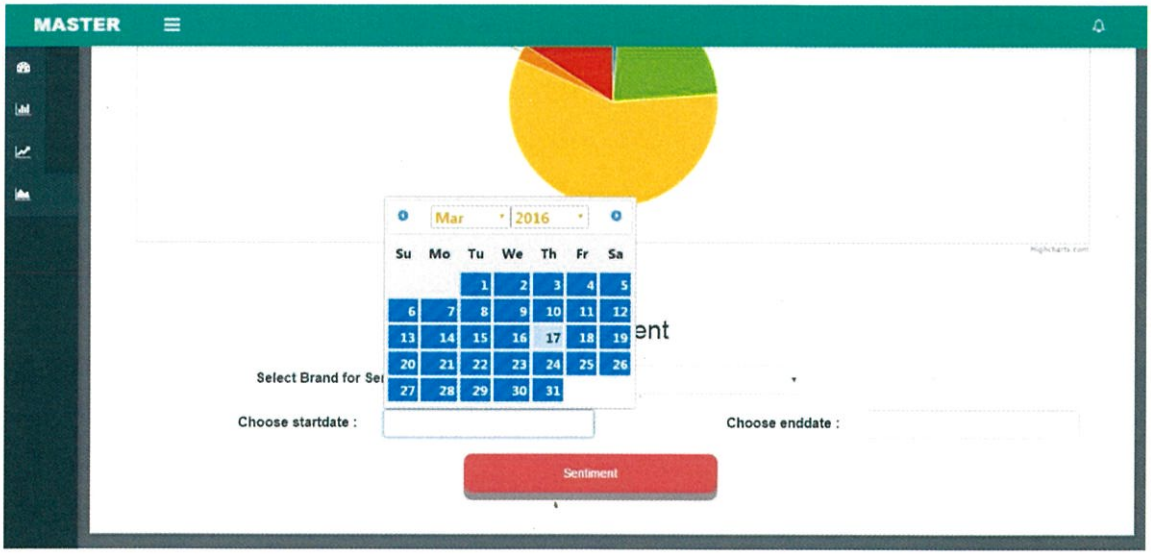


รูปที่ ก.17 หน้าจอแสดงผลลัพธ์อารมณ์และความรู้สึก

ผู้ใช้สามารถเลือกแบรนด์ที่ต้องการจะทราบการวิเคราะห์อารมณ์และความรู้สึก แสดงดังรูปที่ ก.18 โดยจะต้องเลือกช่วงเวลาที่ต้องการจะทำการวิเคราะห์ ด้วยการระบุวันเริ่มต้นและวันสิ้นสุด แสดงดังรูปที่ ก.19 จากนั้นกดปุ่ม Sentiment

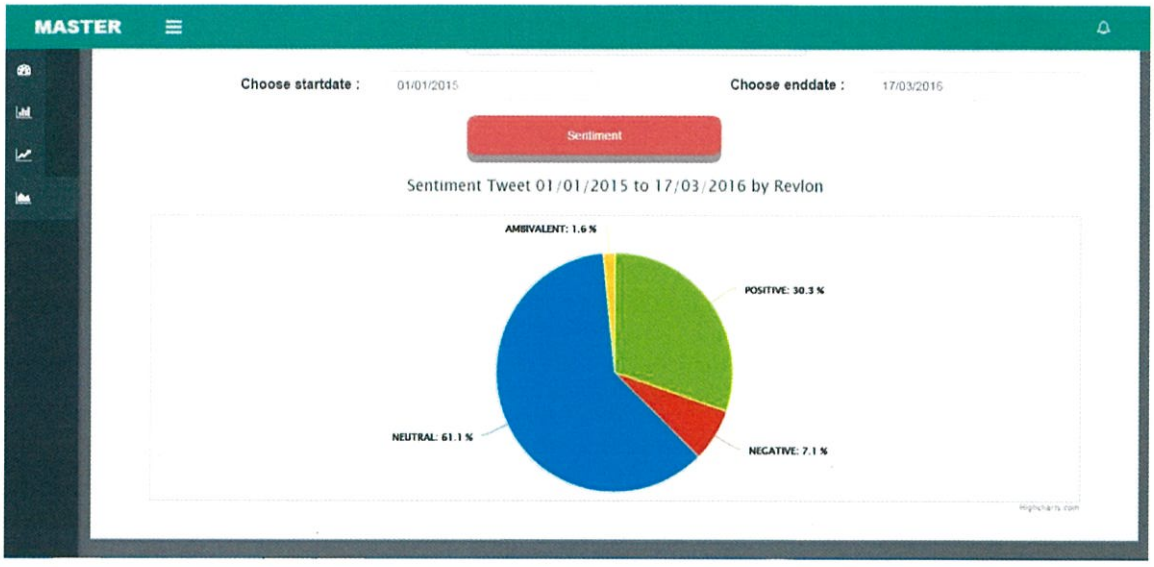


รูปที่ ก.18 ขั้นตอนการแสดงกราฟอารมณ์และความรู้สึก (1)



รูปที่ ก.19 ขั้นตอนการแสดงกราฟอารมณ์และความรู้สึก (2)

โดยผลลัพธ์จะแสดงจำนวนออกมาในรูปแบบของกราฟวงกลม โดยแต่ละสีจะมีความหมายดังนี้ สีเขียว : POSITIVE (บวก) สีแดง : NEGATIVE (ลบ) สีฟ้า : NEUTRAL (กลางๆ) และสีเหลือง : AMBIVALENT (คลุมเครือ) ซึ่งจะคิดออกมาเป็นเปอร์เซ็นต์ แสดงดังรูปที่ ก.20



รูปที่ ก.20 กราฟอารมณ์และความรู้สึก

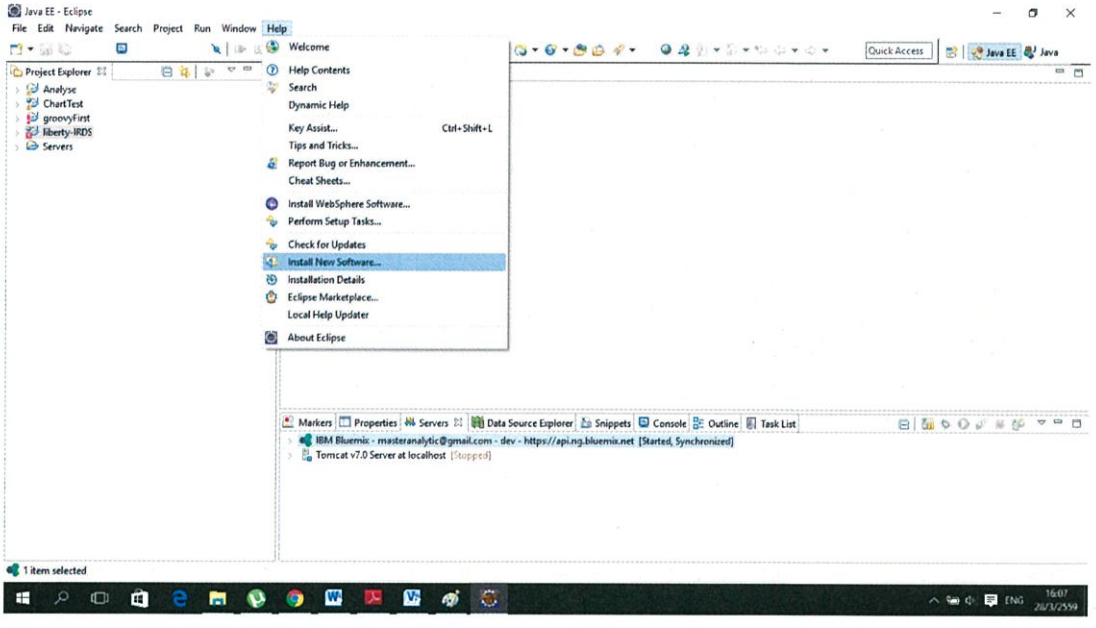
ภาคผนวก ข.
การติดตั้งซอฟต์แวร์

ข.1 การติดตั้งกรูวี (Groovy)

ภาษากรูวี [1] (Groovy) เป็นภาษาโปรแกรมเชิงวัตถุสำหรับแพลตฟอร์มจาวานอกเหนือจากภาษาจาวา โดยอาจจะมองกรูวีเป็นเหมือนภาษาสคริปต์สำหรับแพลตฟอร์มจาวาก็ได้ เนื่องจากมีคุณลักษณะหลายอย่างเหมือนกับภาษาสคริปต์ เช่น ไพทอน (Python), รูบี้ (Ruby), เพิร์ล (Perl), และ สมอลทอล์ค (Smalltalk) โปรแกรมที่เขียนด้วยภาษากรูวีจะถูกคอมไพล์เป็นจาวาไบต์โค้ด

ขั้นตอนการติดตั้งมีรายละเอียดดังนี้

1) เปิดโปรแกรม Eclipse เพื่อทำการติดตั้งกรูวี โดยการคลิกที่แถบเมนู Help > Install New Software แสดงดังรูปที่ ข.1

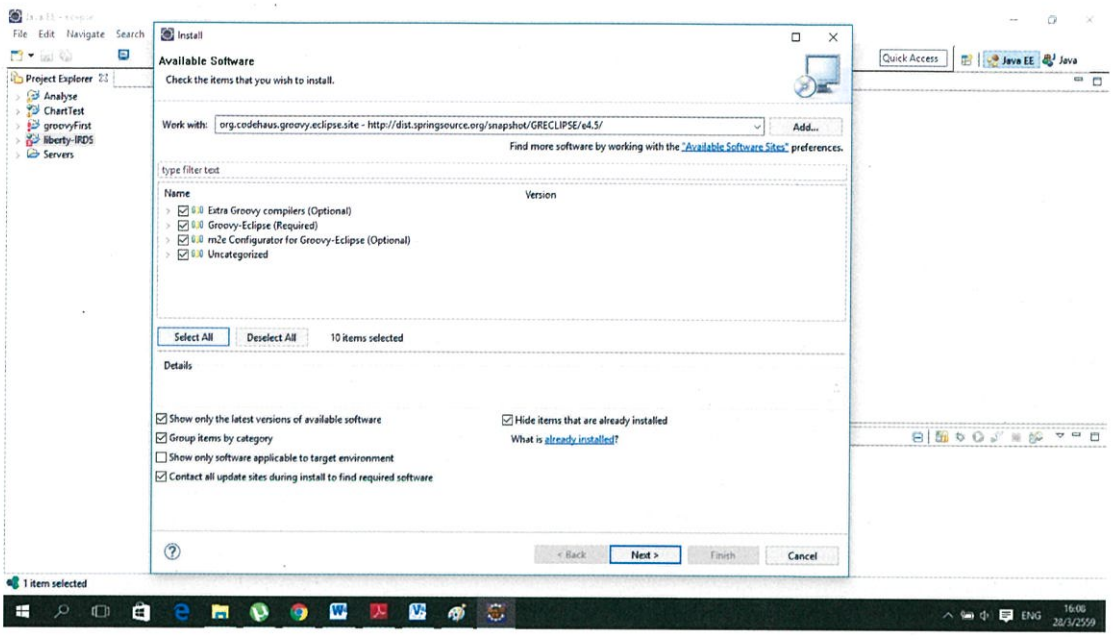


รูปที่ ข.1 ขั้นตอนการติดตั้งกรูวี (1)

---

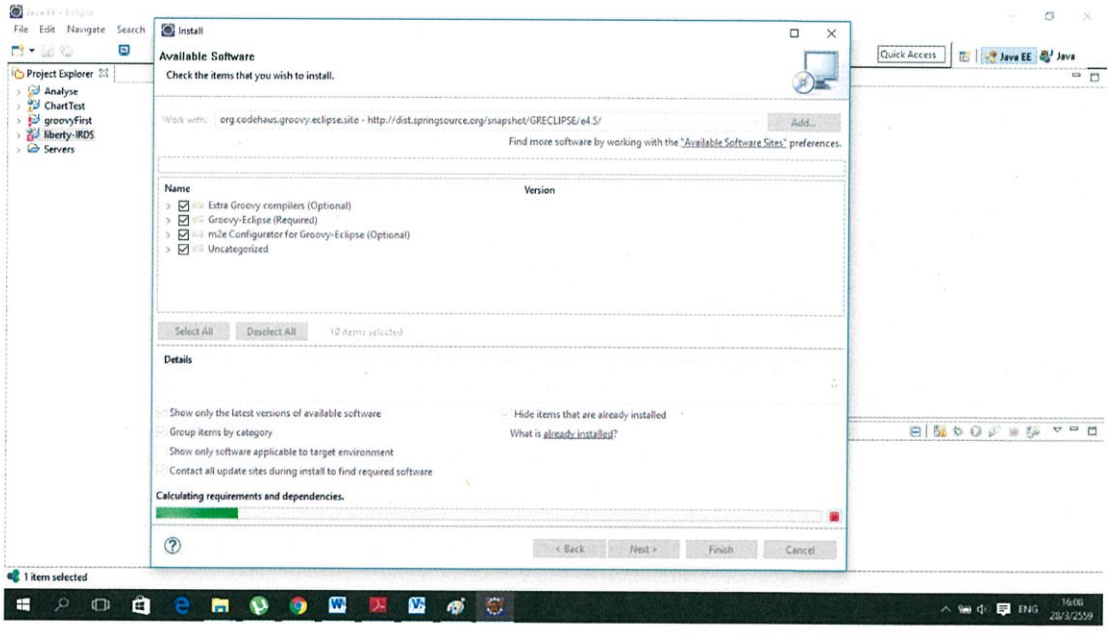
[1] วิกิพีเดีย สารานุกรมเสรี. **ภาษากรูวี.** [Online]. Available: https://th.wikipedia.org/wiki/%.

2) กำหนดเส้นทาง http://dist.springsource.org/snapshot/GRECLIPSE/e4.5/ แล้วเลือก Select All จากนั้นกด Next แสดงดังรูปที่ ข.2



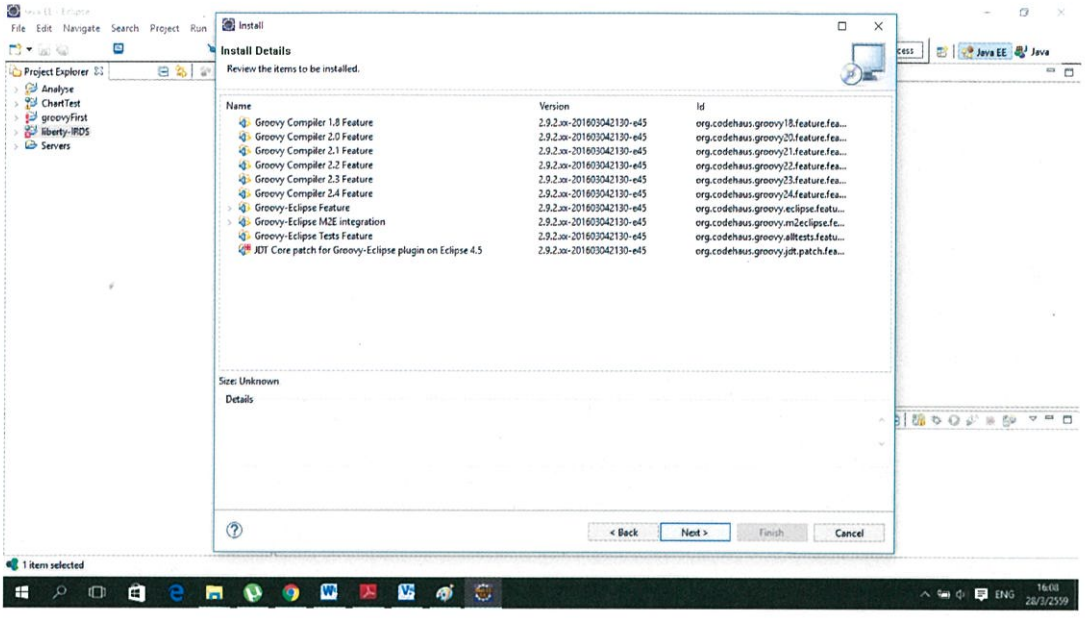
รูปที่ ข.2 ขั้นตอนการติดตั้งกรูวี (2)

3) จากนั้นรอจนกว่าการติดตั้งจะเสร็จสิ้น แสดงดังรูปที่ ข.3



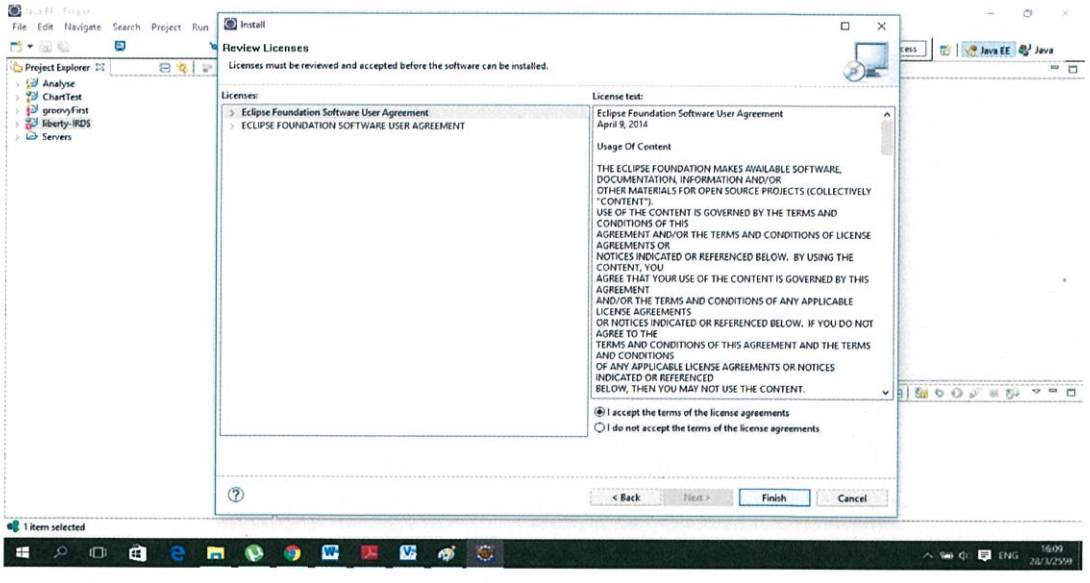
รูปที่ ข.3 ขั้นตอนการติดตั้งกรูวี (3)

4) เมื่อติดตั้งเสร็จแล้วจะแสดงหน้าจอบอกถึงรายละเอียดของกรูวี จากนั้นกดปุ่ม Next แสดงดังรูปที่ ข.4



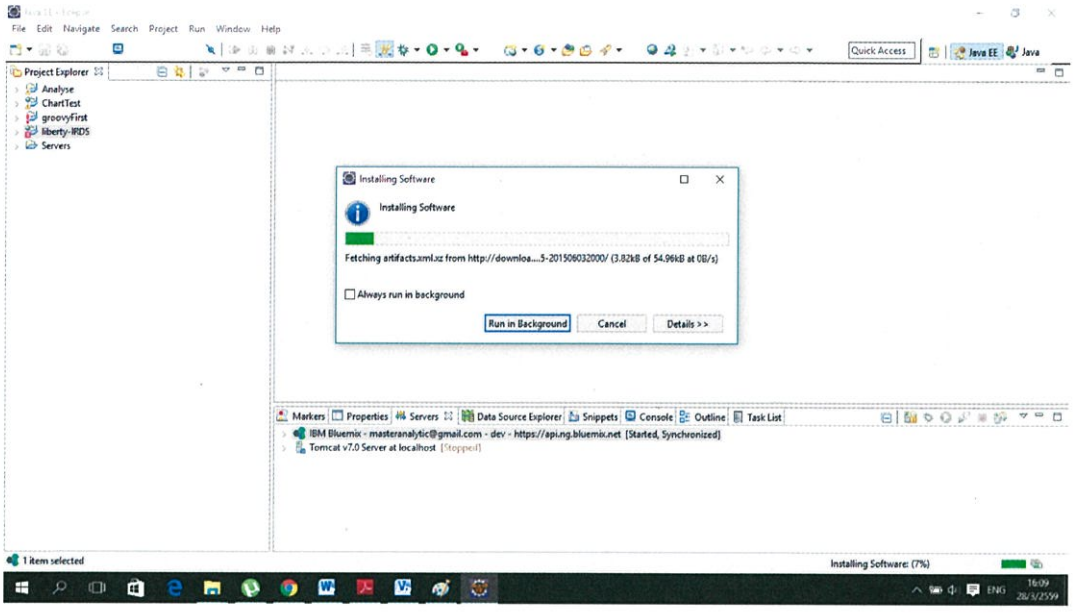
รูปที่ ข.4 ขั้นตอนการติดตั้งกรูวี (4)

5) จากนั้นจะแสดงหน้าจอข้อตกลงต่างๆ เมื่อทำการอ่านละเอียดเสร็จแล้ว ให้กดยอมรับ โดยเลือกที่ I accept the terms of the license agreements จากนั้นกดปุ่ม Finish แสดงดังรูปที่ ข.5



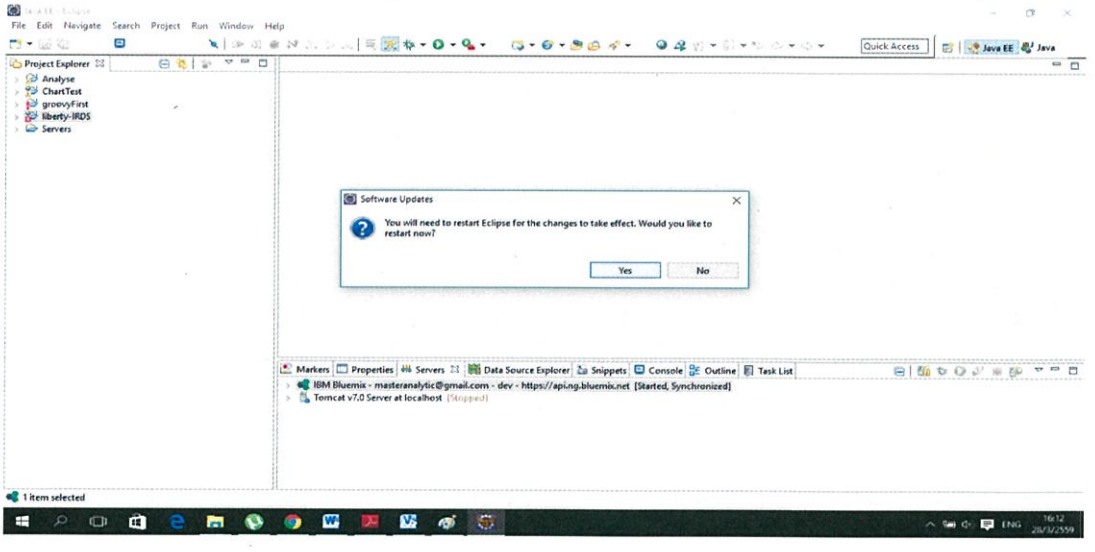
รูปที่ ข.5 ขั้นตอนการติดตั้งกรูวี (5)

6) เมื่อทำการยอมรับข้อตกลงแล้ว จะเป็นการเริ่มทำการติดตั้งกรูวี รอจนกว่าจะทำการติดตั้งเสร็จสิ้น แสดงดังรูปที่ ข.6



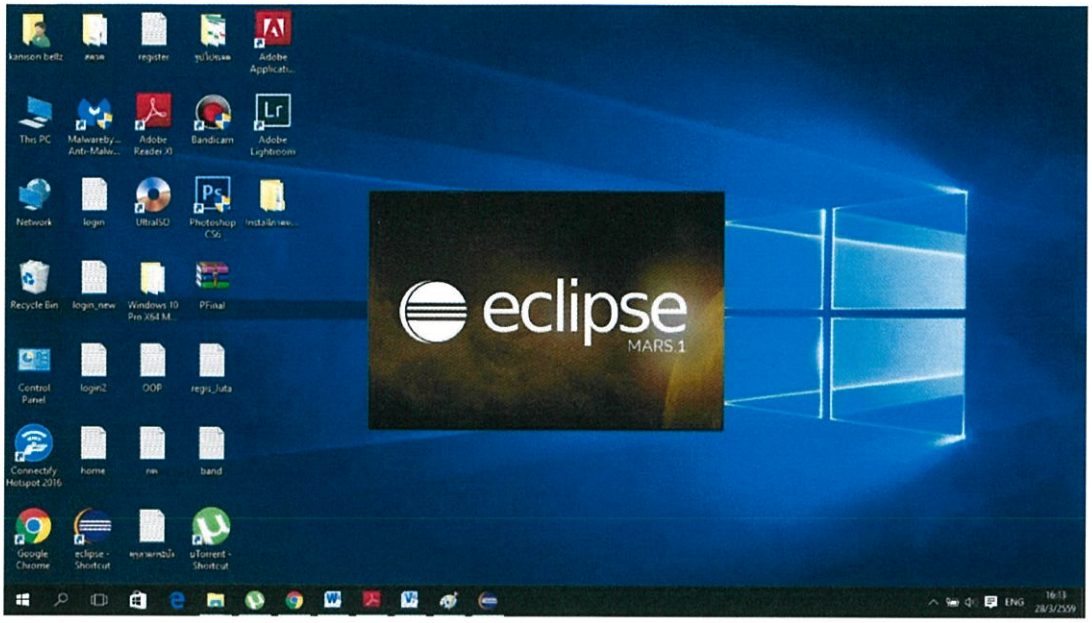
รูปที่ ข.6 ขั้นตอนการติดตั้งกรูวี (6)

7) เมื่อติดตั้งกรูวี เสร็จแล้ว โปรแกรม Eclipse จะแจ้งเตือนว่า ต้องการที่จะรีสตาร์ท Eclipse ในตอนนี้หรือไม่ เพื่อเป็นการอัพเดทกรูวีที่ได้ทำการติดตั้งไปเมื่อซักครู่ ให้คลิกที่ปุ่ม Yes แสดงดังรูปที่ ข.7



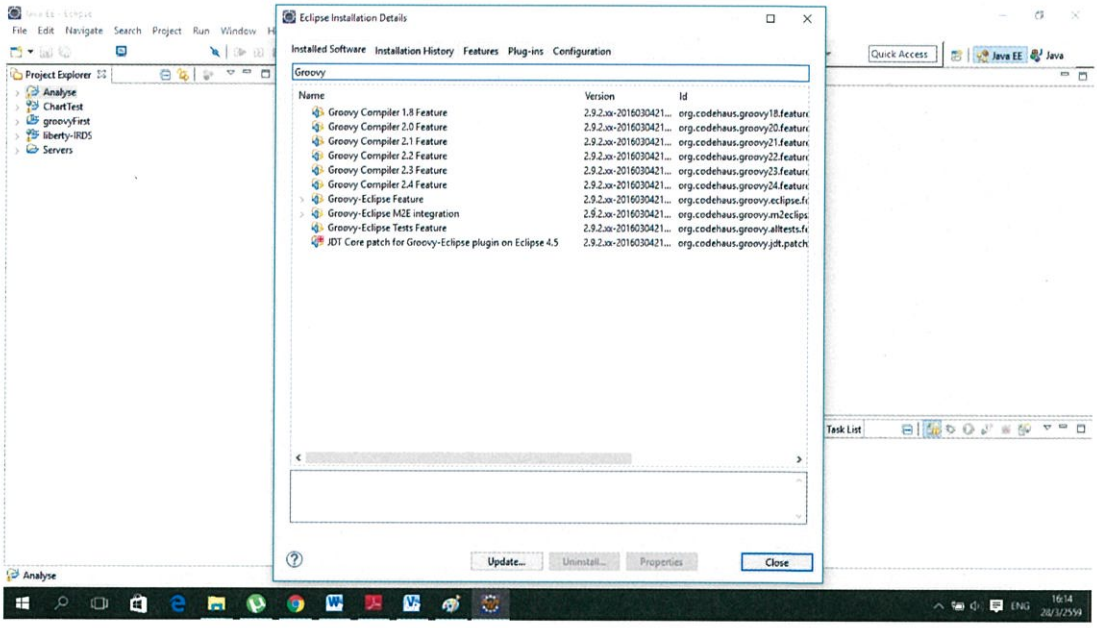
รูปที่ ข.7 ขั้นตอนการติดตั้งกรูวี (7)

8) โปรแกรม Eclipse จะทำการรีสตาร์ทขึ้นมาใหม่ แสดงดังรูปที่ ข.8



รูปที่ ข.8 ขั้นตอนการติดตั้งกรูวี (8)

9) เมื่อเข้ามาในโปรแกรม Eclipse จะเห็นว่าได้มีการติดตั้งกรูวีเรียบร้อยแล้ว แสดงดังรูปที่ ข.9



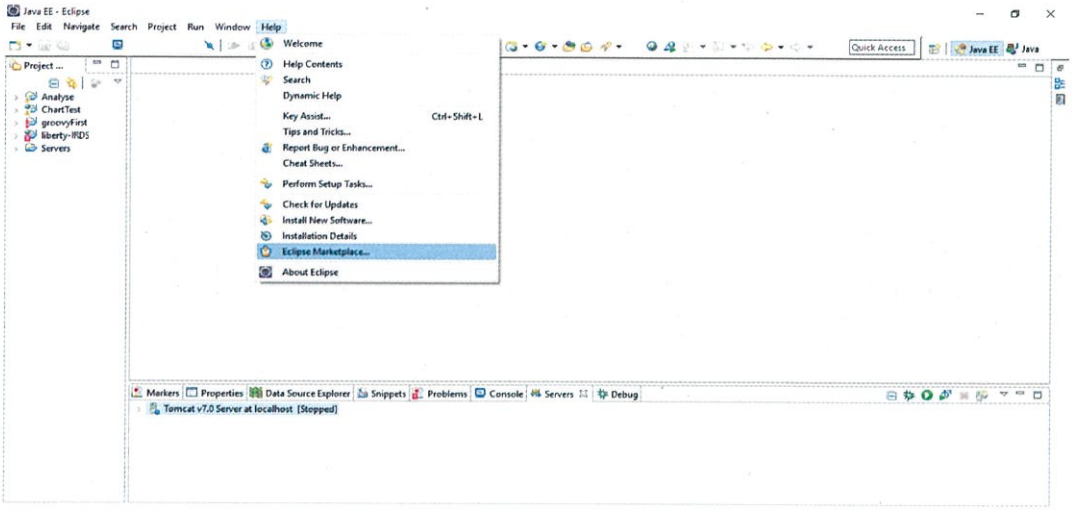
รูปที่ ข.9 ขั้นตอนการติดตั้งกรูวี (9)

ข.2 การติดตั้ง IBM Eclipse Tools for Bluemix

IBM Bluemix ได้เพิ่มการรองรับการแก้จุดบกพร่องของ JavaScript รองรับ Node.js รองรับ Eclipse Mars รุ่นล่าสุด และรองรับการเพิ่มแบบสาธารณะ (Incremental Publish)

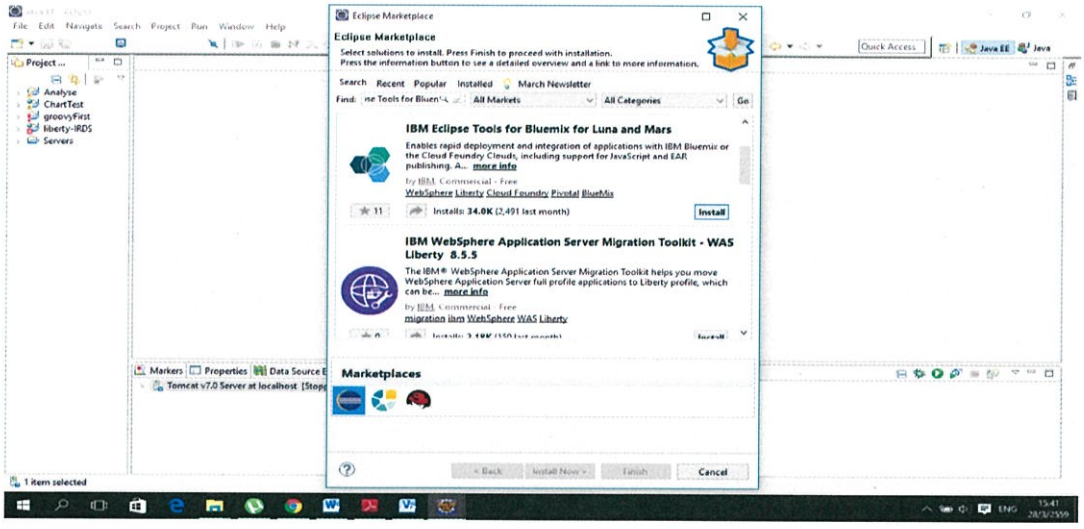
ขั้นตอนการติดตั้งมีรายละเอียดดังนี้

1) เข้าโปรแกรม Eclipse จากนั้นกดเลือกที่แถบเมนู Help > Eclipse Masketplace แสดงดังรูปที่ ข.10



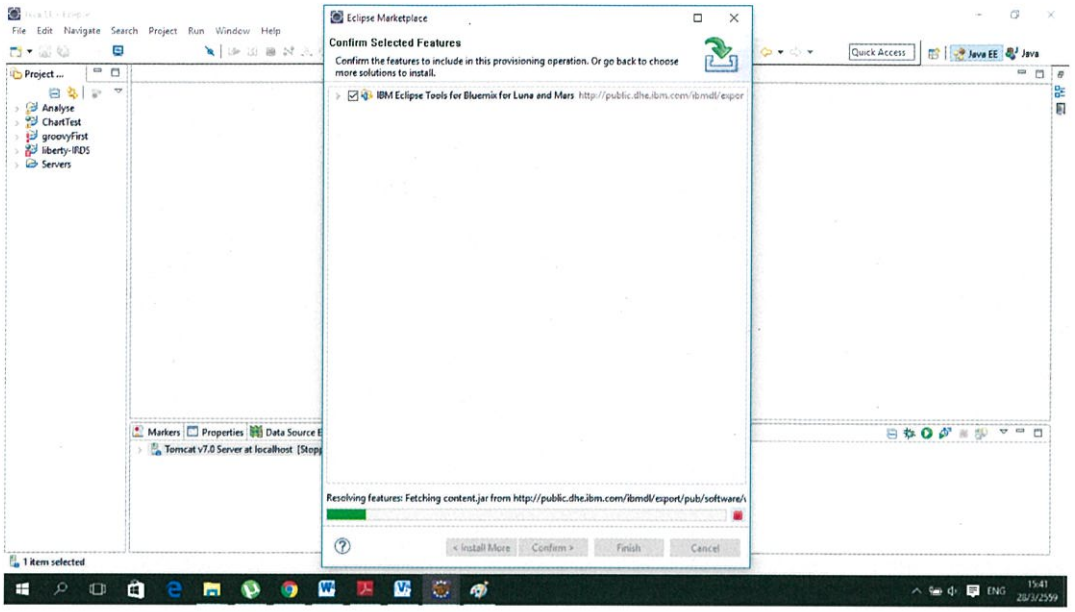
รูปที่ ข.10 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (1)

2) เมื่อเข้ามาแล้วให้ใส่คำค้นหา IBM Eclipse Tools for Bluemix for Luna and Mars และกดปุ่ม Install แสดงดังรูปที่ ข.11



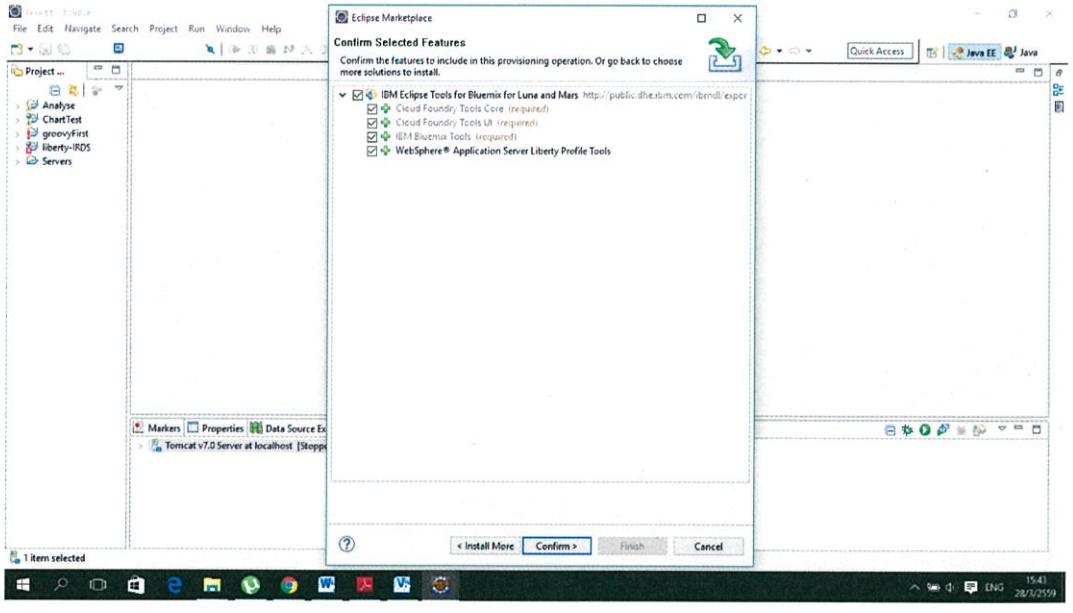
รูปที่ ข.11 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (2)

3) จากนั้นให้รอการติดตั้ง IBM Eclipse Tools for Bluemix for Luna and Mars แสดงดังรูปที่ ข.12



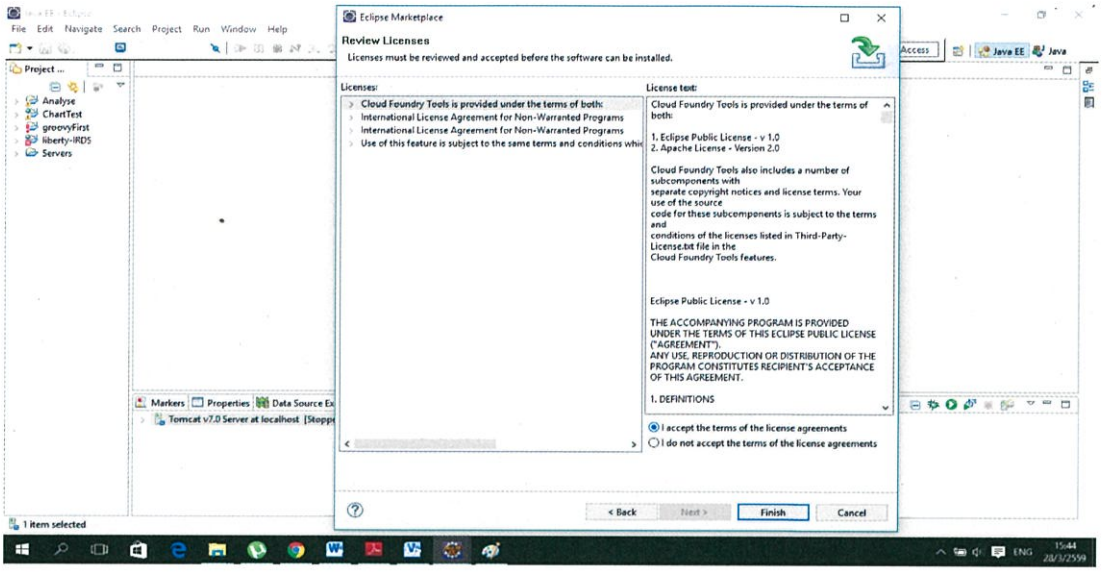
รูปที่ ข.12 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (3)

4) เมื่อทำการติดตั้งเสร็จแล้ว ขั้นตอนต่อไปคือการยืนยันสำหรับการติดตั้ง โดยหน้าจอนี้จะแสดงรายละเอียดของ IBM Eclipse Tools for Bluemix for Luna and Mars จากนั้นทำการยืนยันการติดตั้งโดยการกดปุ่ม Confirm > แสดงดังรูปที่ ข.13



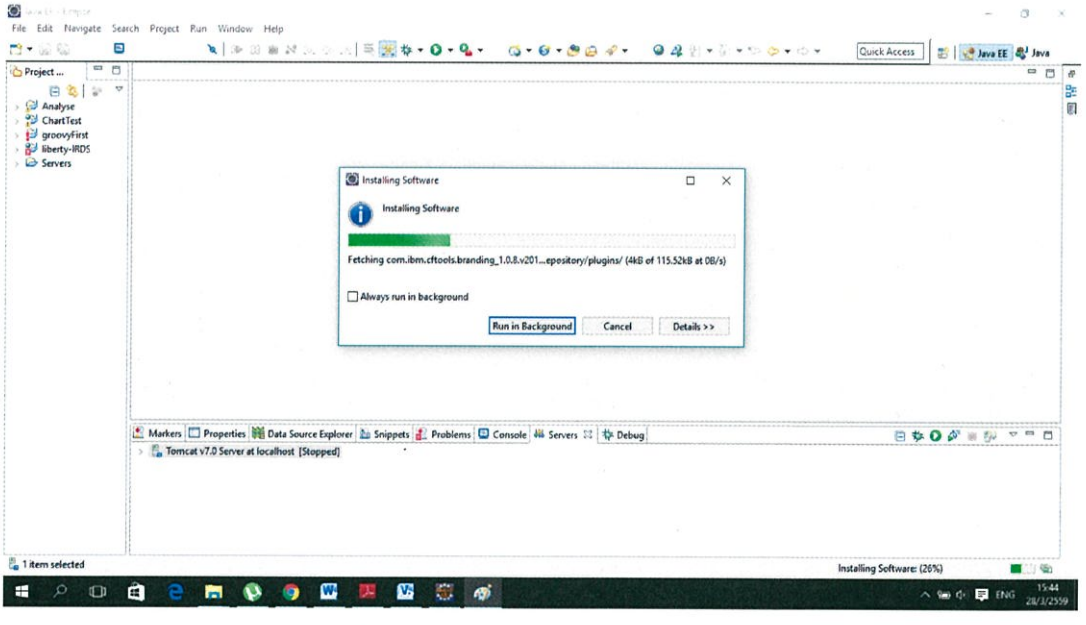
รูปที่ ข.13 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (4)

5) ขั้นตอนต่อไปเป็นการยอมรับข้อตกลงต่างๆ เมื่อทำการอ่านรายละเอียดเรียบร้อยแล้ว ให้เลือกที่ I accept the terms of the license agreements เพื่อเป็นการยอมรับข้อตกลง จากนั้นกดปุ่ม Finish แสดงดังรูปที่ ข.14



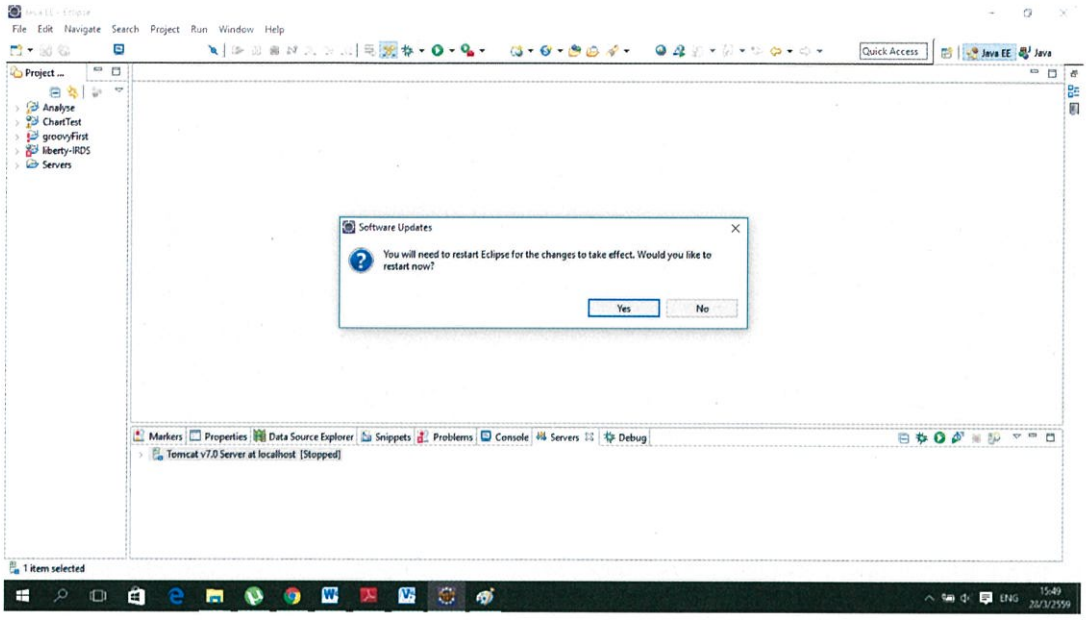
รูปที่ ข.14 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (5)

6) จากนั้นรอจนกว่าการติดตั้งจะเสร็จเรียบร้อย แสดงดังรูปที่ ข.15



รูปที่ ข.15 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (6)

7) เมื่อติดตั้ง IBM Eclipse Tools for Bluemix for Luna and Mars เสร็จแล้ว โปรแกรม Eclipse จะแจ้งเตือนว่า ต้องการที่จะรีสตาร์ท Eclipse ในตอนนี้หรือไม่ เพื่อเป็นการอัพเดทกรูวีที่ได้ทำการติดตั้งไป จากนั้นให้กดปุ่ม Yes แสดงดังรูปที่ ข.16

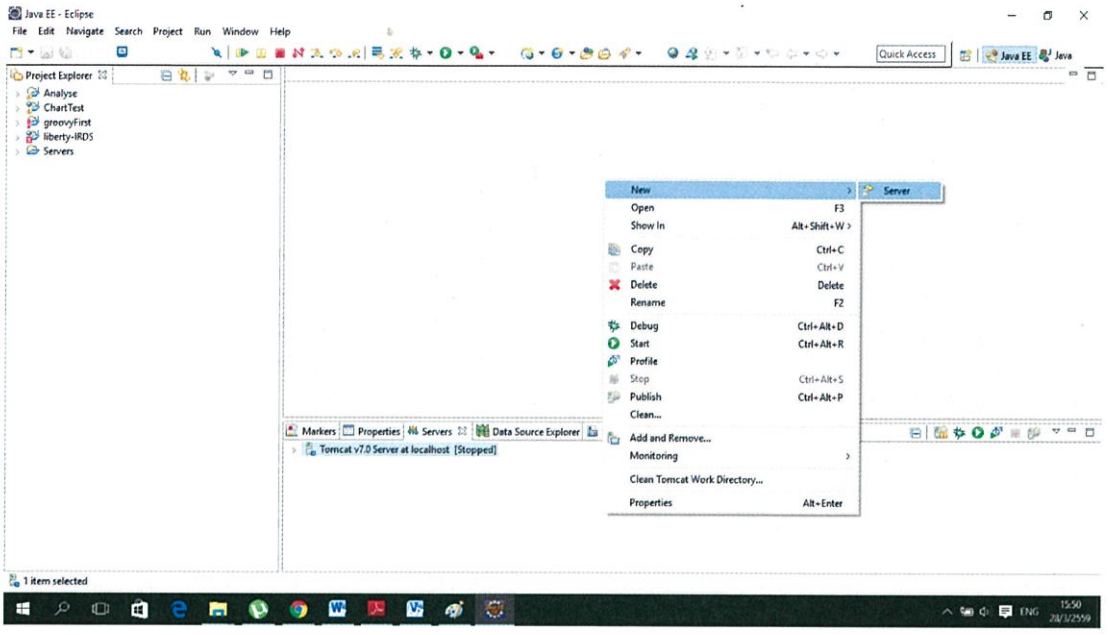


รูปที่ ข.16 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (7)

8) โปรแกรม Eclipse จะทำการรีสตาร์ทขึ้นมาใหม่ แสดงดังรูปที่ ข.17

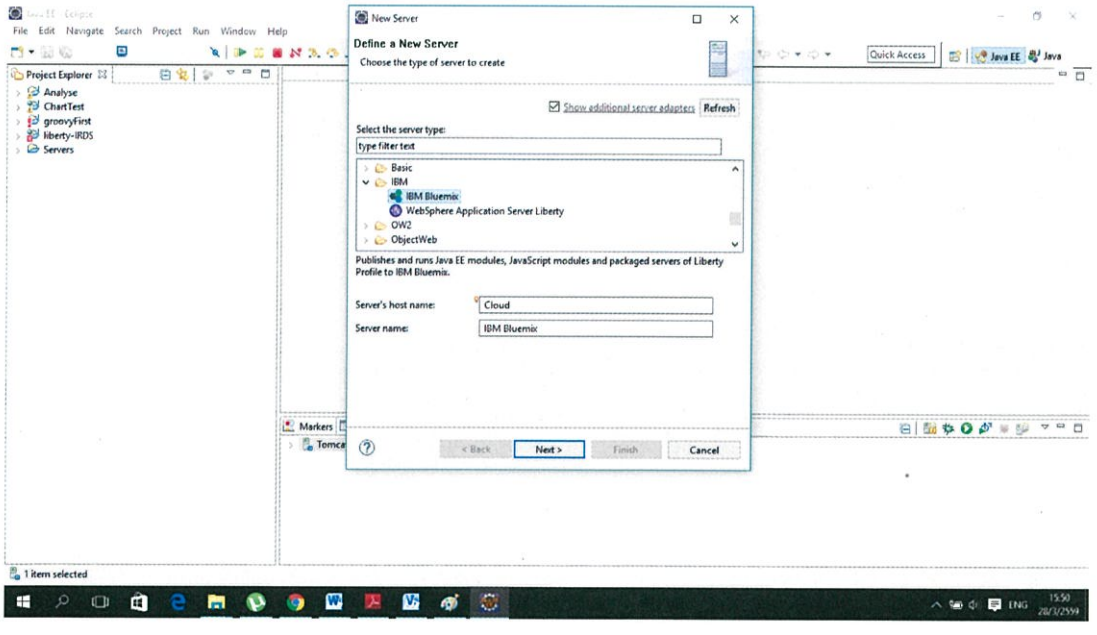
รูปที่ ข.17 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (8)

9) หลังจากทำการรีสตาร์ทโปรแกรม Eclipse ให้คลิกขวาที่ Servers จากนั้นเลือก New > Server แสดงดังรูปที่ ข.18



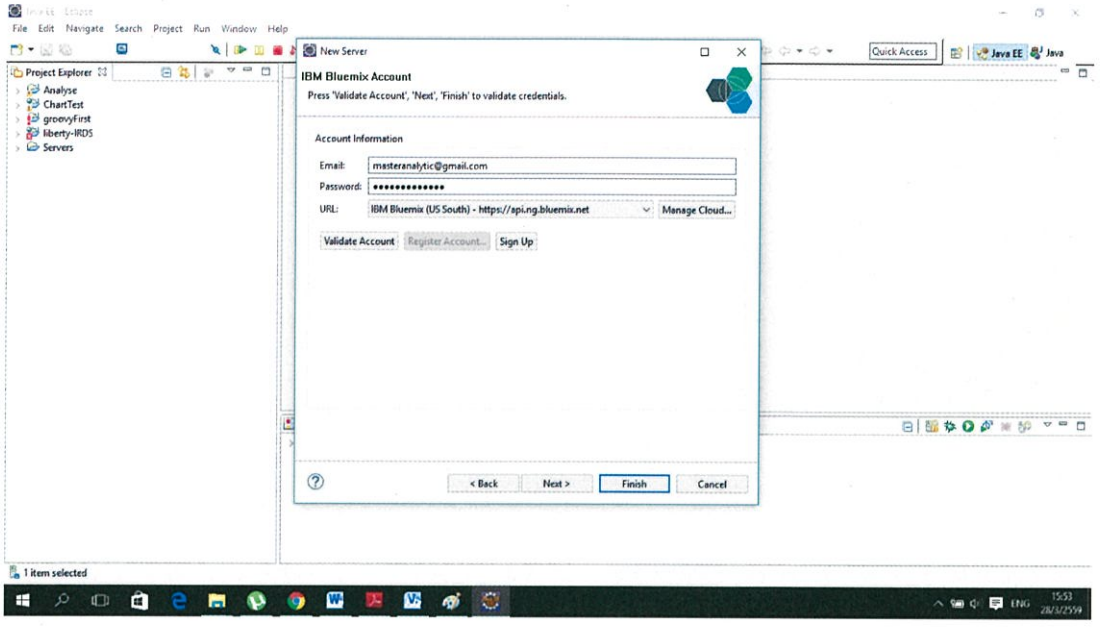
รูปที่ ข.18 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (9)

10) จากนั้นเลือก IBM Bluemix แล้วกดปุ่ม Next > แสดงดังรูปที่ ข.19



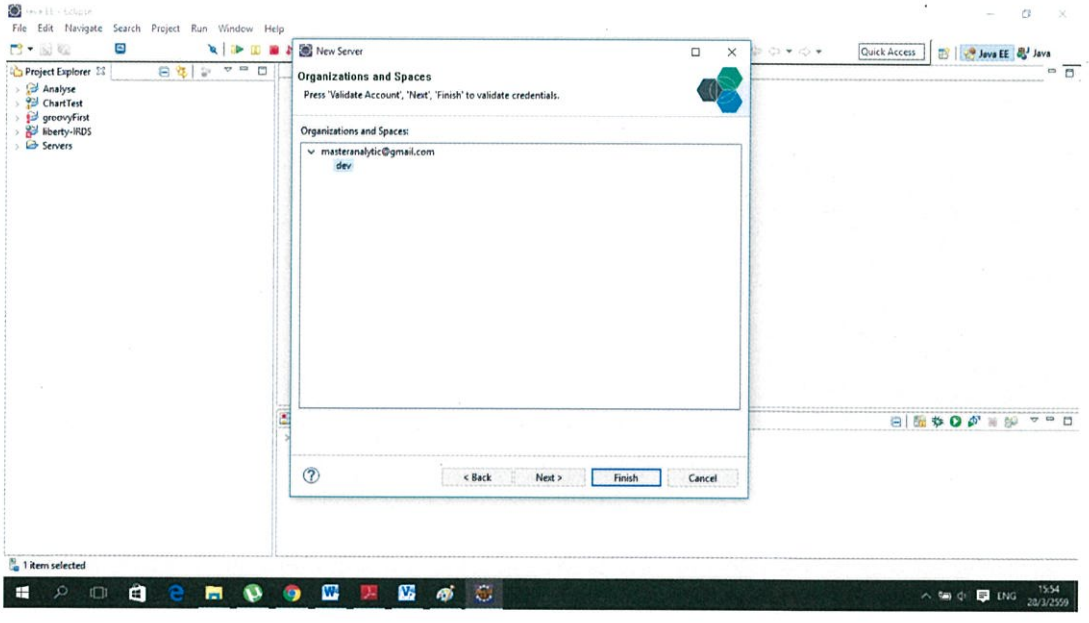
รูปที่ ข.19 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (10)

11) ใส่ Email และ Password ของ IBM Bluemix ที่ได้สมัครไว้ จากนั้นกดปุ่ม Next > แสดงดังรูปที่ ข.20



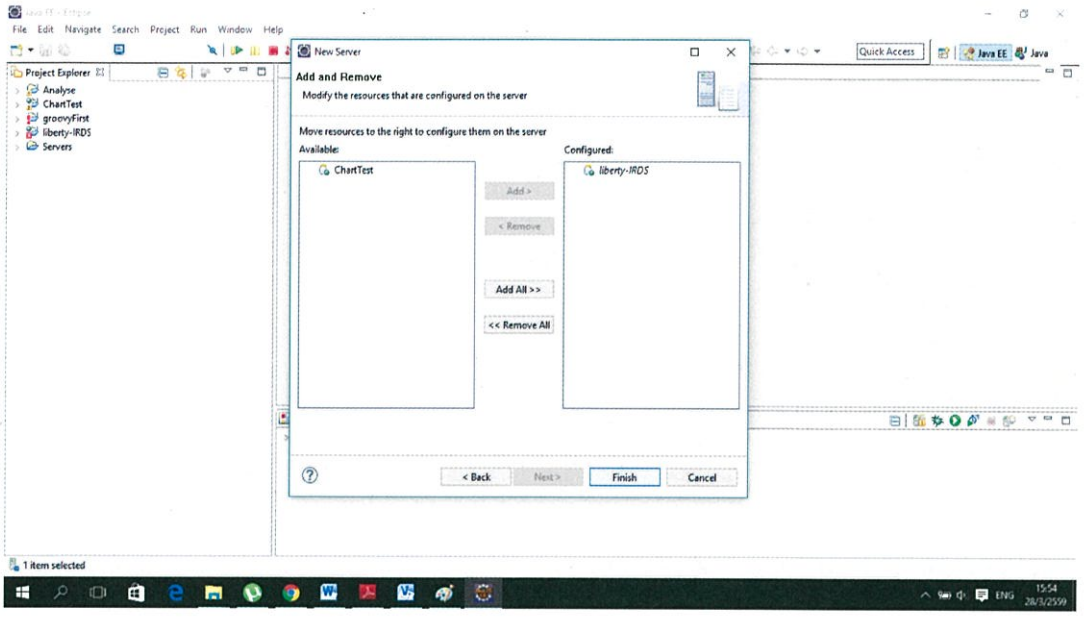
รูปที่ ข.20 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (11)

12) เลือกพื้นที่จากนั้นกดปุ่ม Next > แสดงดังรูปที่ ข.21



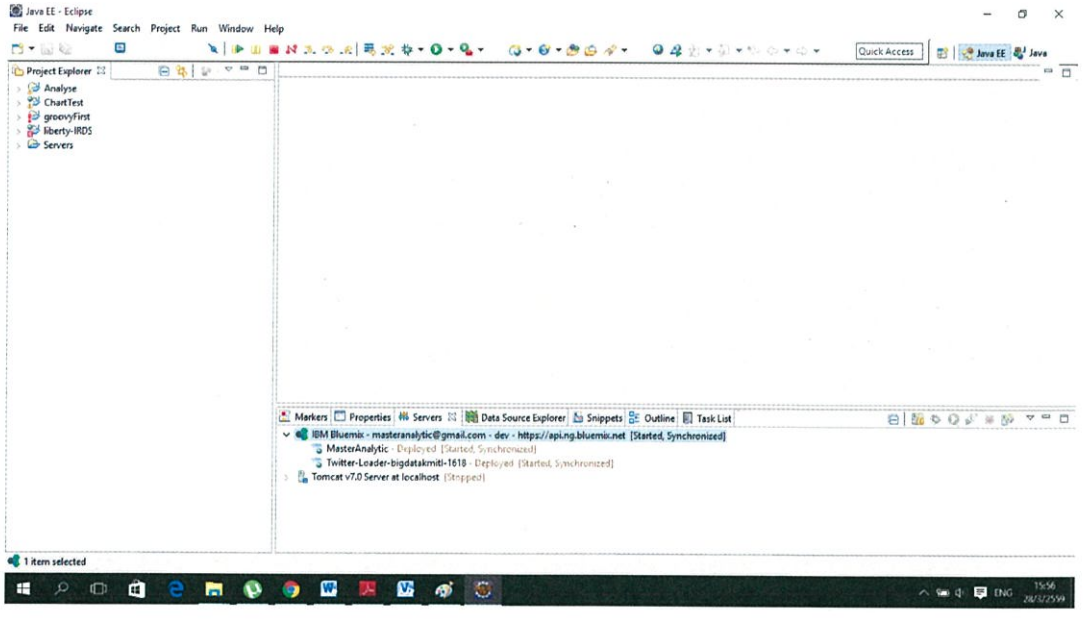
รูปที่ ข.21 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (12)

13) ให้กดปุ่ม Finish แสดงดังรูปที่ ข.22



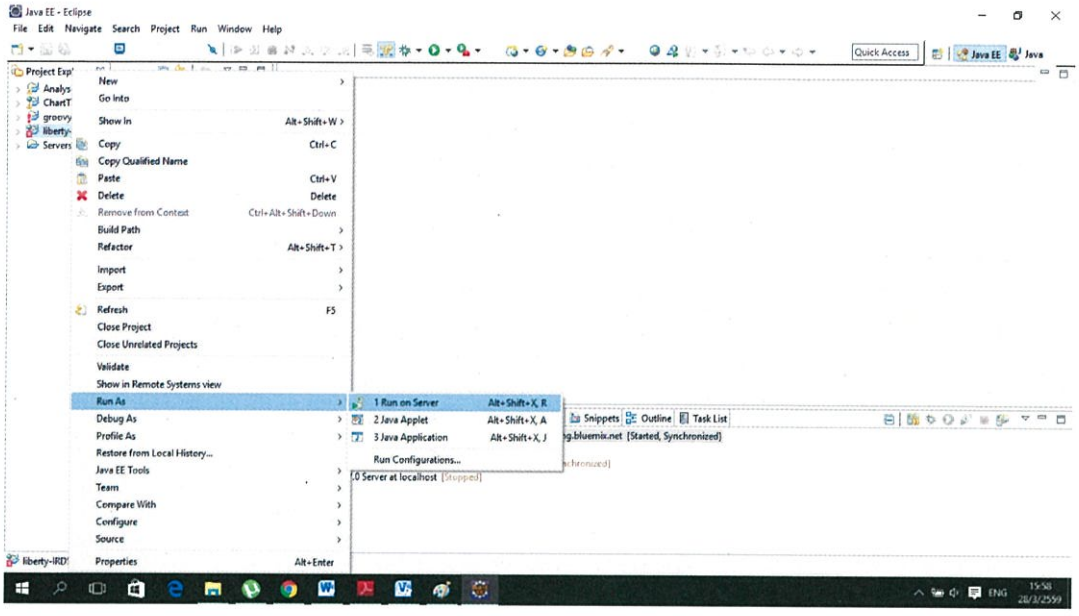
รูปที่ ข.22 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (13)

14) เมื่อทำการติดตั้ง IBM Bluemix เสร็จเรียบร้อยแล้ว จะเห็นว่า IBM Bluemix ได้ถูกเปิดใช้งานแล้ว แสดงดังรูปที่ ข.23



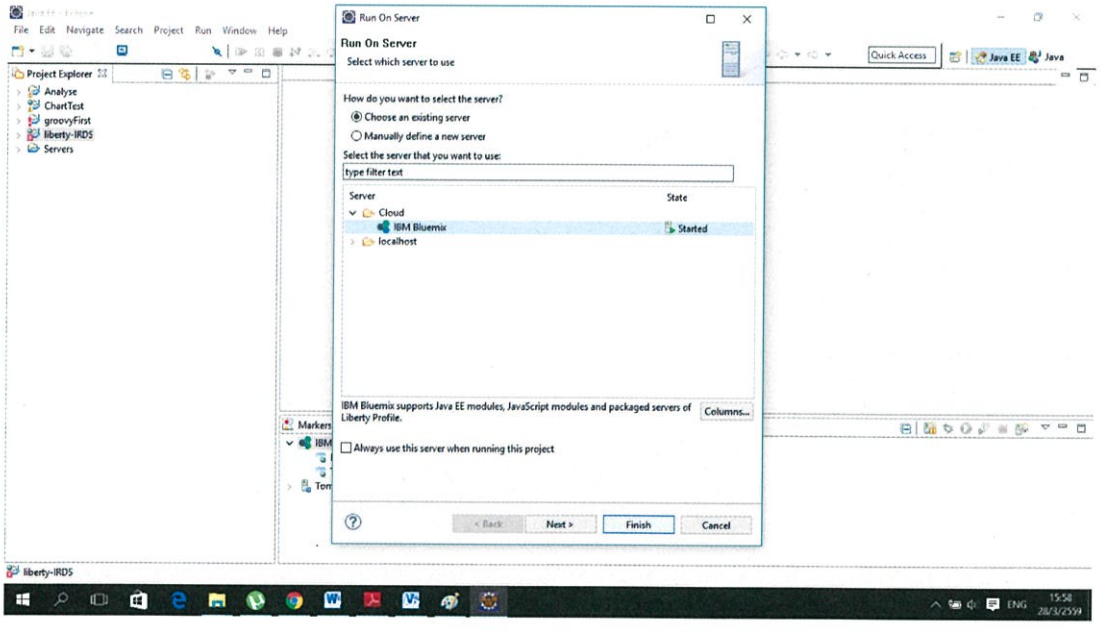
รูปที่ ข.23 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (14)

15) จากนั้นทำการคลิกขวาที่ไฟล์งาน เลือก Run As > Run on Server แสดงดังรูปที่ ข.24



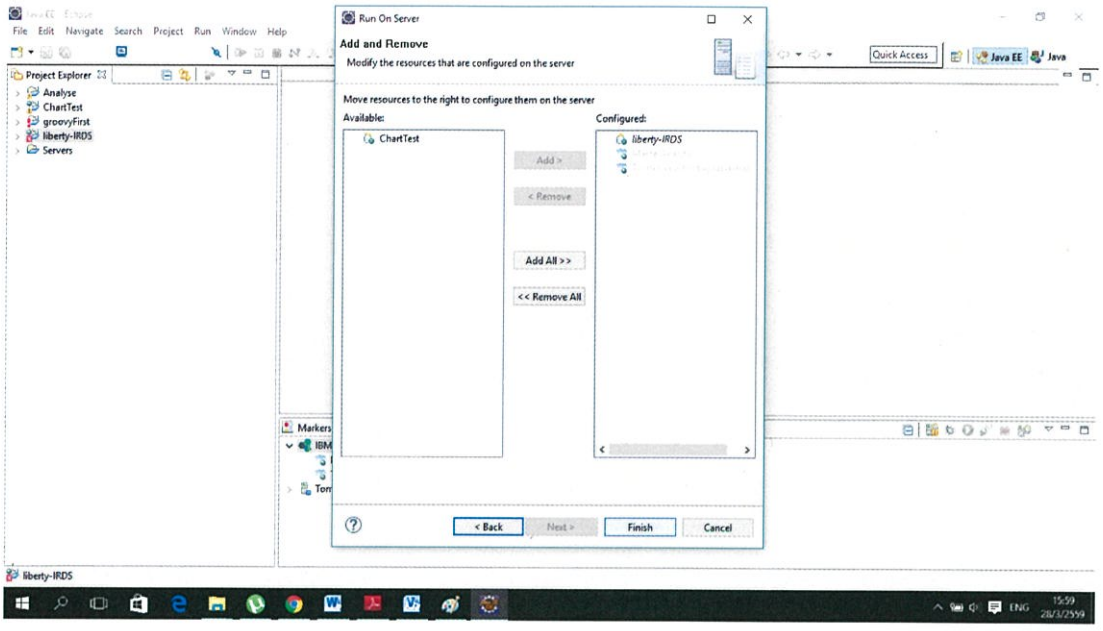
รูปที่ ข.24 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (15)

16) เลือก IBM Bluemix จากนั้นกดปุ่ม Next > แสดงดังรูปที่ ข.25



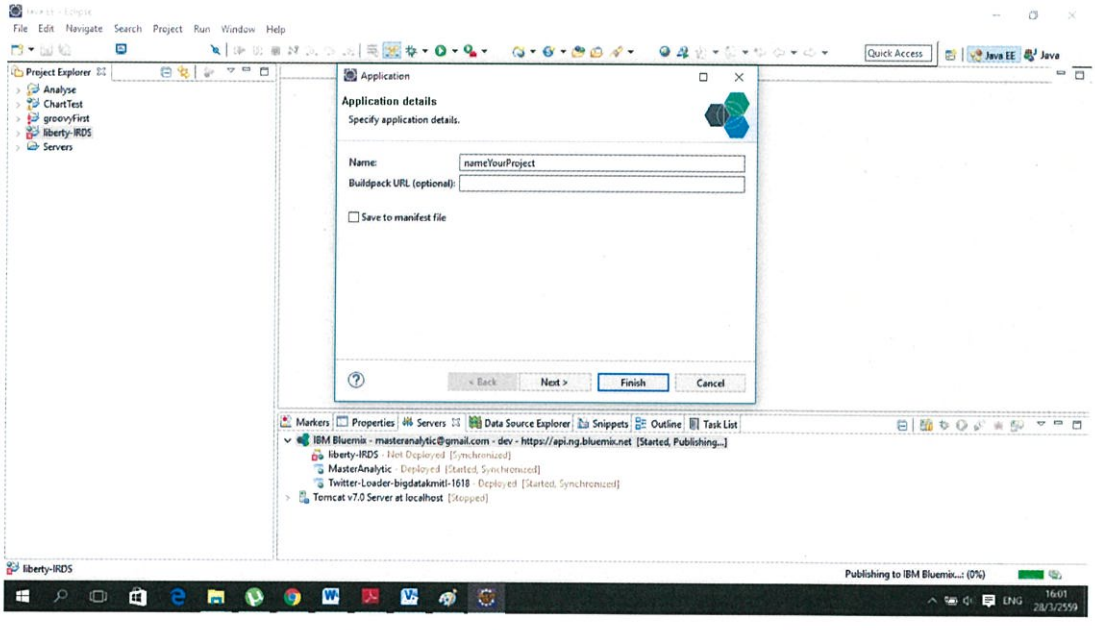
รูปที่ ข.25 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (16)

17) จากนั้นกดปุ่ม Finish แสดงดังรูปที่ ข.26



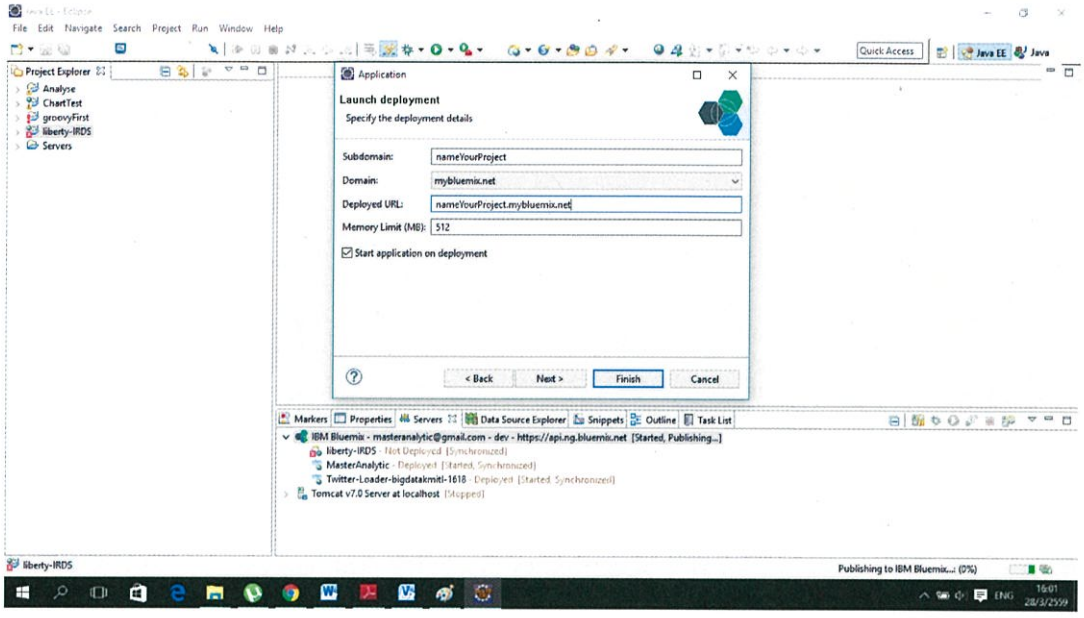
รูปที่ ข.26 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (17)

18) จากนั้นทำการตั้งชื่อโปรเจค ในตัวอย่างตั้งชื่อเป็น Name : nameYourProject จากนั้นกดปุ่ม Next > แสดงดังรูปที่ ข.27



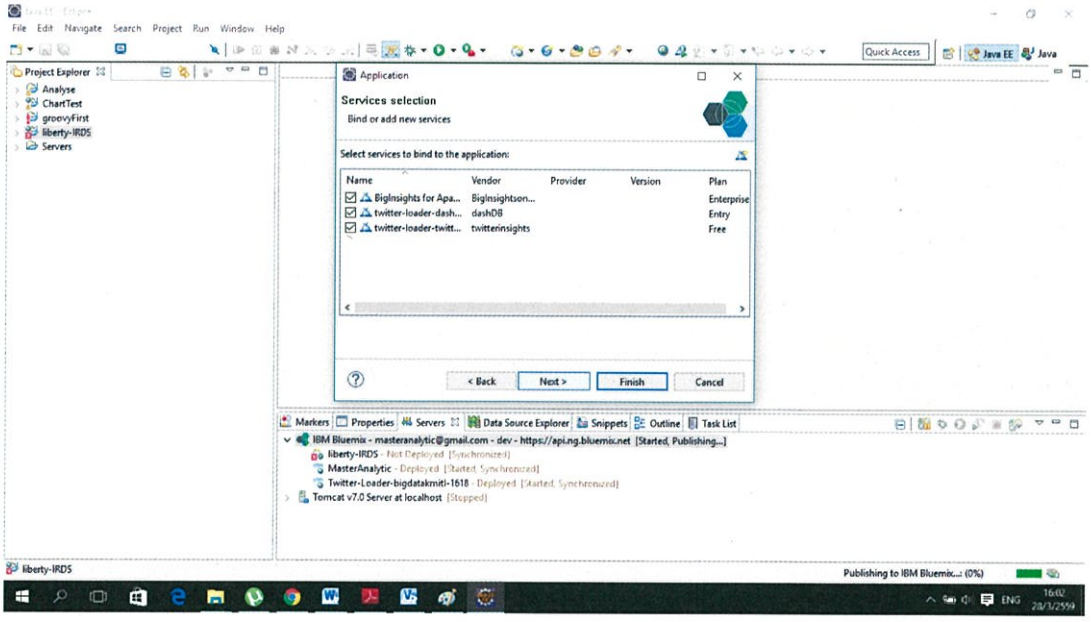
รูปที่ ข.27 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (18)

19) จะได้ชื่อเว็บที่จะนำขึ้นเว็บเป็น nameYourProject.mybluemix.net จากนั้นกดปุ่ม Finish แสดงดังรูปที่ ข.28



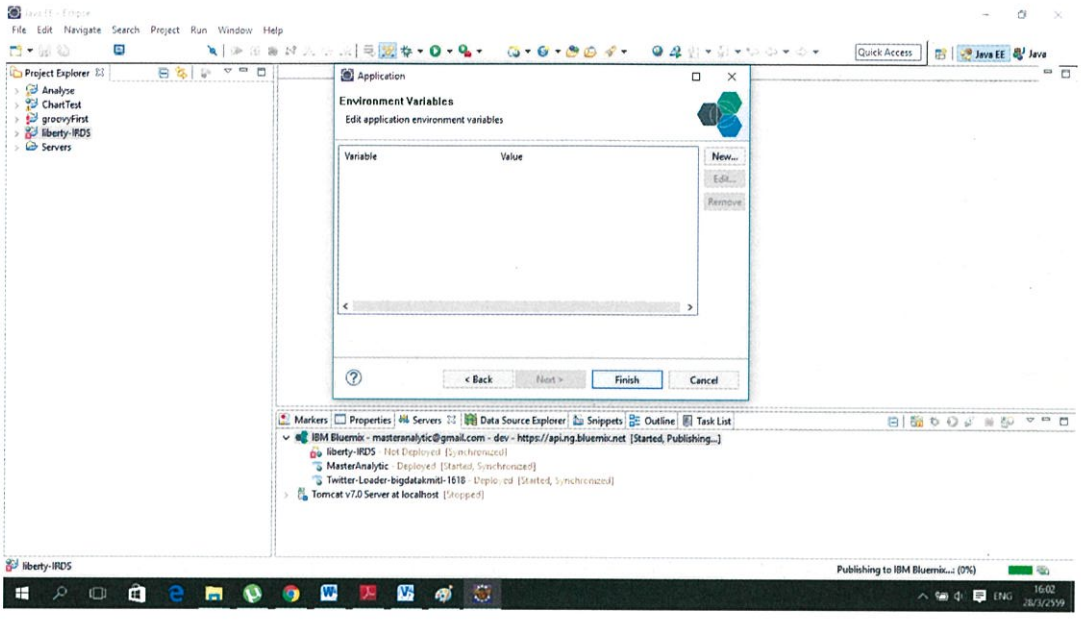
รูปที่ ข.28 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (19)

20) จากนั้นจะปรากฏหน้าจอ Services selection ทำการเลือก Service และกดปุ่ม Next > แสดงดังรูปที่ ข.29



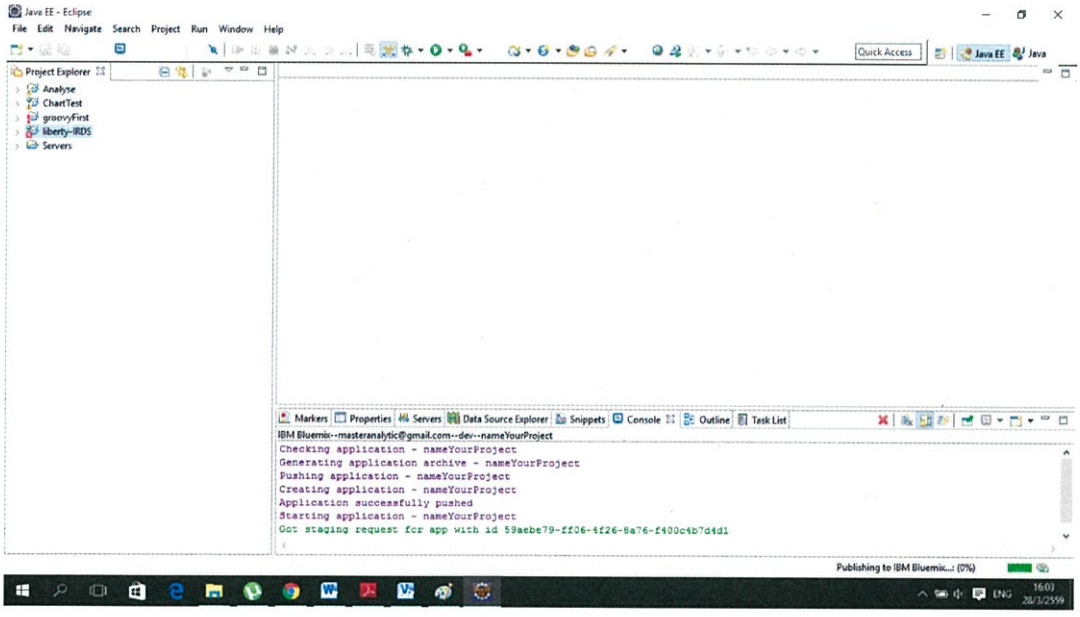
รูปที่ ข.29 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (20)

21) กดปุ่ม Finish เพื่อเป็นการเสร็จสิ้นกระบวนการติดตั้ง แสดงดังรูปที่ ข.30



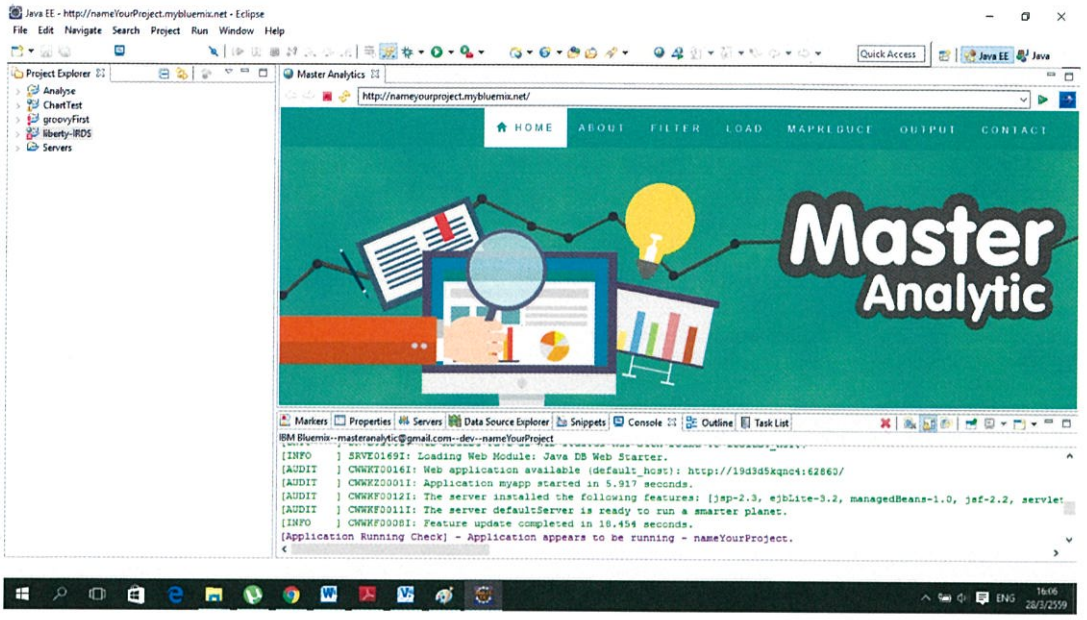
รูปที่ ข.30 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (21)

22) โปรแกรมจะทำการประมวลผล รอจนกว่าจะทำการติดตั้งเสร็จสิ้น แสดงดังรูปที่ ข.31



รูปที่ ข.31 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (22)

23) เมื่อทำการประมวลผลเสร็จแล้ว จะแสดงหน้าเว็บที่ได้ทำการติดตั้งไว้บน IBM Bluemix แสดงดังรูปที่ ข.32



รูปที่ ข.32 ขั้นตอนการติดตั้ง IBM Eclipse Tools for Bluemix (23)

ภาคผนวก ค.
ผลงานที่ได้รับรางวัล

การประชุมวิชาการ AUCC Conference ครั้งที่ 4
(the 4th ASEAN Undergraduate Conference in Computing : AUC^2 2016)

การประชุมวิชาการนี้จัดขึ้นระหว่างวันที่ 27 ถึง 29 เมษายน พ.ศ.2559 ที่คณะวิทยาศาสตร์และสังคมศาสตร์ มหาวิทยาลัยบูรพา วิทยาเขตสระแก้ว จังหวัดสระแก้ว มีมหาวิทยาลัยเข้าร่วมจำนวน 43 สถาบัน มีนักศึกษาจากมหาวิทยาลัยดังกล่าวส่งผลงานเข้าร่วมเป็นจำนวนทั้งหมด 403 โครงการ สำหรับปัญหาพิเศษนี้ได้รับรางวัล "Very Good Paper Award" (รางวัลที่ 3) ในประเภทการนำเสนอผลงานทางวิชาการแบบปากเปล่า (Oral Presentation)

1) **ภาพถ่ายขณะเข้าร่วมการประชุมวิชาการ**

รูปที่ ค.1 ภาพถ่ายคณะผู้จัดทำปัญหาพิเศษขณะเข้าร่วมการนำเสนอผลงาน

The 4th ASEAN Undergraduate Conference in Computing (AUC² 2016)

Certificate of Oral Presentation

presented to

คณิศร เสมพืช, จีราวรรณ เจียมใจ, ธนโชติ สมันตธนกุล และ วรางคณา กิ้มปาน

for a contribution in the field of computing entitled

เว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์

upon with the success of **Very Good Paper Award**

held on April 27th - 29th, 2016 at Faculty of Science and Social Sciences,
Burapha University Sakaeo Campus, Thailand

Seree Chinodom
Seree Chinodom
General Chair of AUC² 2016

รูปที่ ค.2 ประกาศนียบัตรรางวัล “Very Good Paper Award”

2) บทความที่นำเสนอ

เว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์
Web Application for Managing Big Data from Twitter Using IBM Bluemix

ธนโชติ สมันตธนกุล คณิศร เสมพืช จีราวรรณ เจียมใจ และวรางคณา กิ้มปาน
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
tanachot.samun@gmail.com, ksb_onepiece.pkw@hotmail.com, magentak.jrw@gmail.com

บทคัดย่อ

บทความนี้นำเสนอเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์ โดยผู้ใช้สามารถกรองข้อมูลมาจากทวิตเตอร์แล้วนำข้อมูลเหล่านี้ไปเก็บในฮาดูปซึ่งสามารถเก็บข้อมูลได้จำนวนมากและมีกระบวนการ MapReduce ฟังก์ชันสำหรับนับจำนวนคำที่พบในประโยค จากนั้นเว็บแอปพลิเคชันจะนำข้อมูลไปแสดงเป็นกราฟซึ่งช่วยให้ผู้ใช้เข้าใจง่ายและสามารถนำไปวิเคราะห์ข้อมูลตามที่ต้องการได้

คำสำคัญ: *เว็บแอปพลิเคชัน ทวิตเตอร์ ข้อมูลขนาดใหญ่ ฮาดูป แมปรีดิวซ์ กราฟ*

Abstract

This paper presents web application for managing big data from Twitter using IBM Bluemix. Users are able to filter Hashtag data from Twitter and then the data are stored into Hadoop. Hadoop then enables the users to store large volumes of data. Moreover, it provides the process map and reduces the functions for counting the number of interesting keywords. At last, the results will be shown in graphs which make the users easily understand and analyze the data.

Keywords: *Web application, Twitter, Big Data, Hadoop, MapReduce, Graph*

1. บทนำ

ในยุคปัจจุบันมีการใช้อุปกรณ์สื่อสารประเภทพกพา (Smart Phone) และแท็บเล็ตกันอย่างแพร่หลาย มีแอปพลิเคชันถูกพัฒนามาเพื่อสนับสนุนแพลตฟอร์มดังกล่าวมากมาย รวมถึงความนิยมในการใช้โซเชียลเน็ตเวิร์คไม่ว่าจะเป็น ทวิตเตอร์ (Twitter) เฟซบุ๊ก (Facebook) หรือไฟล์จำพวกมีเดีย (Media) ที่มีขนาดใหญ่ เป็นต้น และการทำธุรกิจหรือธุรกรรมออนไลน์ต่างๆ ทำให้มีข้อมูลเกิดขึ้นในระบบเป็นจำนวนมากทั้งข้อมูลที่มีโครงสร้าง ข้อมูลกึ่งโครงสร้างและข้อมูลที่ไม่มีโครงสร้าง

การจัดการกับข้อมูลจำนวนมากและเกิดขึ้นตลอดเวลาซึ่งไม่สามารถทำได้ด้วยวิธีการจัดเก็บไว้ในฐานข้อมูลรูปแบบเดิมๆ ดังนั้นจึงมีวิธีการจัดการข้อมูลขนาดใหญ่ (Big Data) ซึ่งวิธีการจัดการกับ Big Data นั้น ณ ปัจจุบันมีเครื่องมือที่ได้รับความนิยมอย่างแพร่หลายและมีชื่อเสียงตัวหนึ่งเข้ามาช่วยจัดการ ได้แก่ ฮาดูป (Hadoop) ที่พัฒนามาจากอาปาเช่โอเพนซอร์สเทคโนโลยี (Apache Open Source Technology) ทำหน้าที่เป็นแหล่งจัดเก็บข้อมูลแบบกระจาย (Distributed Storage) ที่สามารถเก็บข้อมูลขนาดใหญ่และนำมาประมวลผลได้ ข้อมูลต่างๆที่เกิดขึ้นเหล่านี้ก่อให้เกิดโอกาสมากมายโดยการนำข้อมูลมาประมวลผลและวิเคราะห์ให้เกิดมูลค่าได้

ดังนั้น บทความนี้นำเสนอการศึกษาและพัฒนาเว็บแอปพลิเคชันที่ช่วยจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ทำให้ผู้ใช้งานสามารถใช้ได้ง่ายขึ้น เช่น การกรองข้อมูลมาจากทวิตเตอร์ การนำข้อมูลไปเก็บไว้ในฮาดูป การ MapReduce และการแสดงผลลัพธ์ออกมาในรูปแบบต่างๆ โดยผลลัพธ์ที่นำมาแสดงจะช่วยให้ผู้ใช้งานสามารถนำข้อมูลเหล่านี้ไปวิเคราะห์ต่อได้ เช่น กราฟแสดงผลการเปรียบเทียบจาก

ฟังก์ชัน MapReduce กราฟแสดงอารมณ์และความรู้สึกจากข้อความ (Sentiment Analysis) การดูอันดับความนิยมของคำที่กรองมาจากทวิตเตอร์ เป็นต้น

2. ทฤษฎีที่เกี่ยวข้อง

2.1 ไอบีเอ็มบลูมิกซ์ (IBM Bluemix)

Bluemix [1] เป็นระบบที่ให้บริการประมวลผลบนกลุ่มเมฆของ IBM ซึ่งเน้นการให้บริการแพลตฟอร์มสำหรับการพัฒนาแอปพลิเคชัน (Platform as a Service : PaaS) โดยใช้แพลตฟอร์มสำหรับการประมวลผลแบบกลุ่มเมฆซึ่งให้บริการโดย VMware Bluemix สามารถใช้งานได้ทั้งในรูปแบบส่วนต่อประสานแบบชุดคำสั่ง (Command Line Interfaces : CLI) ผ่านส่วนต่อประสานแบบชุดคำสั่งสำหรับการประมวลผลแบบกลุ่มเมฆ (Cloud Foundry Cli) ปกติและการทำงานผ่านหน้าเว็บไซต์ Bluemix โดยตรง แพลตฟอร์มที่ให้บริการใน Bluemix มี 4 แพลตฟอร์ม หลัก คือ java, node.js, ruby on rails และ ruby sinatra นอกจากนี้ยังเป็นเซอร์วิสซึ่งสามารถเพิ่มเข้ามาได้ในภายหลังได้และยังมีเทมเพลตสำหรับการเริ่มต้นทำโปรเจคเว็บไซต์หรือเว็บแอปพลิเคชัน ที่มีการเตรียมไฟล์ต่างๆให้อย่างครบครันและจะทำให้มีการทำงานที่เร็วขึ้น แข็งแกร่ง ยืดหยุ่นและสมบูรณ์แบบมากขึ้น (Boilerplates) โดยรวมเอารันไทม์และเซอร์วิสเข้ามาให้บริการร่วมกันสร้างเป็นบริการเฉพาะทาง เช่น Internet of Things platform, Mobile Cloud, Node Cached Starter, Big Data เป็นต้น

2.2 ระบบฮาดูป (Hadoop System)

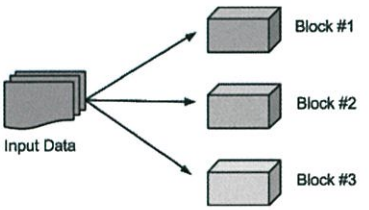
ระบบฮาดูปเป็นซอฟต์แวร์แบบโอเพ่นซอร์ส (Open Source) สำหรับการสร้างระบบการประมวลผลแบบกระจาย (Distributed Computing) โดยอาศัยแนวคิดของ Google File System ใช้สำหรับรันแอปพลิเคชันบนระบบคลัสเตอร์ขนาดใหญ่และสนับสนุนการทำงานแบบขนาน

ฮาดูปสามารถแบ่งออกเป็น 2 ส่วนคือ ส่วน HDFS (Hadoop Distributed File System) และส่วน MapReduce ซึ่ง HDFS มีความสามารถในการจัดเก็บข้อมูลขนาดใหญ่แบบกระจาย ส่วน MapReduce เป็นส่วนประมวลผลและวิเคราะห์ข้อมูล โดยทั้ง HDFS และ MapReduce ได้ถูกออกแบบมาเพื่อให้เฟรมเวิร์คสามารถจัดการกับโหนดที่ทำงานผิดพลาดให้สามารถทำงานต่อได้โดยอัตโนมัติ

2.3 ระบบจัดการไฟล์แบบกระจายของฮาดูป

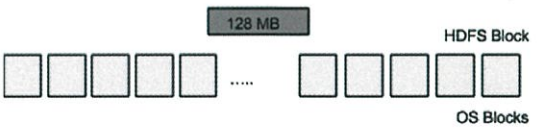
ระบบจัดการไฟล์แบบกระจายของฮาดูป [2] คือส่วนจัดเก็บข้อมูลหลักเป็นระบบแฟ้มข้อมูลแบบกระจายซึ่งระบบจัดการไฟล์แบบกระจายของฮาดูปสามารถสร้างแบบจำลองเป็นกลุ่มข้อมูลบนคลัสเตอร์กระจายข้อมูลที่ต้องการเก็บไว้ในไฟล์บนคอมพิวเตอร์หลายๆเครื่อง (คอมพิวเตอร์แต่ละเครื่องเรียกว่าโหนด) โดยกลุ่มไฟล์หนึ่งไฟล์อาจถูกแยกออกเป็นหลายส่วนแต่ละส่วนมีขนาดเท่าๆกัน แล้วเก็บไว้ต่างโหนดกัน ทำให้ได้กลุ่มโหนดที่ถูกมองเป็นฮาร์ดดิสก์หนึ่งส่วนที่มีขนาดใหญ่ โดยขนาดของฮาร์ดดิสก์สามารถเพิ่มได้ไม่จำกัด การเพิ่มขนาดฮาร์ดดิสก์ไม่กระทบข้อมูลเดิม และเมื่อโหนดใดโหนดหนึ่งเสียระบบไฟล์ทั้งหมดยังสามารถทำงานได้จึงทำให้มีความน่าเชื่อถือของระบบสูง

โดยระบบจัดการไฟล์แบบกระจายของฮาดูป [3] ทำให้การสำรองข้อมูลบนโหนดและเตรียมแบนด์วิดท์ไว้สำหรับการส่งข้อมูลข้ามคลัสเตอร์และเก็บไฟล์ต่างๆเป็นกลุ่มไฟล์ดังรูปที่ 1



รูปที่ 1. การแบ่งไฟล์เป็นกลุ่มของระบบจัดการไฟล์แบบกระจายของฮาดูป

ขนาดกลุ่มไฟล์ของระบบจัดการไฟล์แบบกระจายของฮาดูปโดยปกติมีขนาด 64MB แต่ในระบบไฟล์ของลีนุกซ์ (Linux) มีขนาดบล็อก 4KB แต่อาจจะใช้บล็อกไฟล์ของ HDFS ขนาด 128MB ดังรูปที่ 2



รูปที่ 2. กลุ่มไฟล์ของระบบจัดการไฟล์แบบกระจายของฮาดูป

2.4 แมปรีดิวซ์

MapReduce [4] เป็นเฟรมเวิร์ก (Framework) ในการเขียนโปรแกรมแบบหนึ่ง ที่ช่วยในงานประมวลผลที่มีชุดของข้อมูลจำนวนมากเป็นการทำงานแบบขนานและเหมาะกับการทำงานแบบกระจาย ซึ่งอาศัยเครื่องคอมพิวเตอร์หลายๆ เครื่องช่วยกันประมวลผลข้อมูล

โปรแกรม MapReduce ประกอบด้วยสองส่วนหลักๆ คือ Map เป็นขั้นตอนการประมวลผลข้อมูลที่รับเข้ามา (Input data) และ Reduce เป็นขั้นตอนการรวบรวมผลเพื่อนำไปเป็นผลลัพธ์สุดท้าย โดยทั่วไปสามารถเขียนโปรแกรมใน MapReduce เพื่อคำนวณผลลัพธ์ได้ซึ่งแสดงในรูปที่ 3

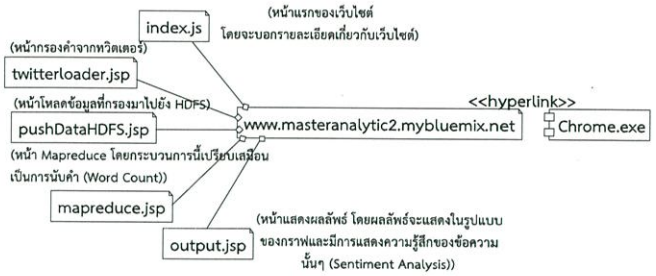
```
Map(String key, String value) :
    // key: document name
    // value: document contents
for each word w in value:
    EmitIntermediate(w, "1");

reduce(String key, Iterator values) :
    // key: a word
    // value: a list of counts
    int result = 0;
    for each v in values:
        result += ParseInt(v);
        Emit (AsString (result));
```

รูปที่ 3. ฟังก์ชัน Map และ Reduce

จากรูปที่ 3 อัลกอริทึมของฟังก์ชัน Map ซึ่งทำหน้าที่รับข้อมูลอินพุตที่เป็นชื่อเอกสาร (Key) และเนื้อหาเอกสาร (Value) เพื่อจัดกลุ่มคำ ส่วนฟังก์ชัน Reduce ทำหน้าที่รับข้อมูลเข้าที่เป็นค่า (Key, Word) และรายการของกลุ่มคำ (List of counts) เพื่อลดจำนวนผลลัพธ์ ลักษณะการทำงานของฟังก์ชัน Map/Reduce มีรูปแบบการเขียนโปรแกรมให้มีการประมวลผลแบบขนาน(Parallel Programming Model) คือ การนำการประมวลผลมาแบ่งย่อยออกเป็นส่วนๆ โดยแต่ละส่วนสามารถทำงานหรือประมวลผลได้ในเวลาเดียวกัน ดังนั้นหลักการทำงานของโปรแกรมแบบขนาน (Parallel Programming) สามารถทำงานได้เร็วกว่าซึ่งในการทำ MapReduce กับข้อมูลจะใช้รูปแบบการประมวลผลแบบขนาน จึงสามารถจัดการกับความซับซ้อนในการกระจายการประมวลผลให้เกิดความสมดุลการประมวลผลแบบขนานได้

3. การออกแบบเว็บไซต์

เว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์ประกอบไปด้วยฟังก์ชันหลักๆด้วยกัน 4 ฟังก์ชัน คือ ฟังก์ชันกรองคำจากทวิตเตอร์ ฟังก์ชันโหลดข้อมูลไปยังฮาดูป ฟังก์ชัน MapReduce และฟังก์ชันการแสดงผล สามารถสร้างไดอะแกรม ดังรูปที่ 4



รูปที่ 4. ไดอะแกรมแสดงฟังก์ชันหลัก

จากรูปที่ 4 สามารถอธิบายแต่ละฟังก์ชันได้ดังนี้

1) ฟังก์ชันกรองคำจากทวิตเตอร์

เป็นฟังก์ชันซึ่งผู้ใช้สามารถกรองคำที่ต้องการจากทวิตเตอร์ได้โดยการพิมพ์คำที่ต้องการลงไปในช่องที่กำหนด แล้วตั้งชื่อตารางที่จะนำข้อมูลลงไปเก็บ โดยระยะเวลาในการดึงคำนั้นจะขึ้นอยู่กับจำนวนคำที่พบในทวิตเตอร์ ถ้าข้อมูลน้อยก็จะใช้เวลาไม่นานในทางกลับกันถ้าข้อมูลมากก็จะใช้เวลามากขึ้นตามลำดับ

2) ฟังก์ชันโหลดข้อมูลไปยังฮาดูป

เป็นฟังก์ชันซึ่งผู้ใช้สามารถนำตารางที่เก็บข้อมูลที่ได้สร้างไว้ในฟังก์ชันกรองคำจากทวิตเตอร์ไปเก็บไว้บนฮาดูป

3) ฟังก์ชัน MapReduce

เป็นฟังก์ชันที่ผู้ใช้ต้องนำตารางที่เก็บไว้บนฮาดูปนำมาทำ MapReduce โดยกระบวนการ MapReduce มีลักษณะเหมือนกับการนับคำ (Word Count) ซึ่งผลลัพธ์ของการ MapReduce แสดงออกมาเป็นลักษณะกราฟ

4) ฟังก์ชันการแสดงผล

เป็นฟังก์ชันซึ่งผู้ใช้สามารถดูผลลัพธ์ที่ได้จากการทำกระบวนการต่างๆที่ผ่านมา โดยผลลัพธ์แสดงในรูปแบบของกราฟการเปรียบเทียบและกราฟที่ผู้ใช้กรองคำมาจากทวิตเตอร์สามารถแบ่งแยกเป็นความรู้สึกด้านบวก ด้านลบ เป็นกลาง หรือคลุมเครือ ซึ่งช่วยเป็นส่วนหนึ่งในการตัดสินใจของผู้ใช้ในเชิงธุรกิจได้

4. ผลการทดลอง

4.1 การใช้งานฟังก์ชันต่างๆ

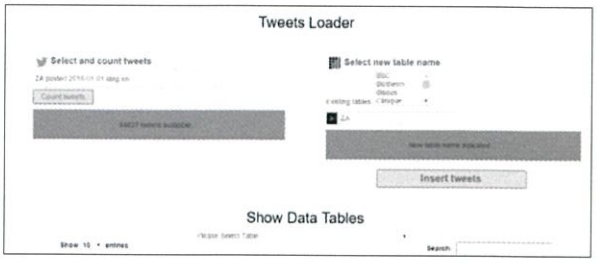
หน้าเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์แสดงดังรูปที่ 5 และรูปที่ 6 และผู้ใช้สามารถเลือกฟังก์ชันต่างๆได้แสดงดังรูปที่ 7 ถึงรูปที่ 11

รูปที่ 5. หน้าแรกเว็บแอปพลิเคชัน

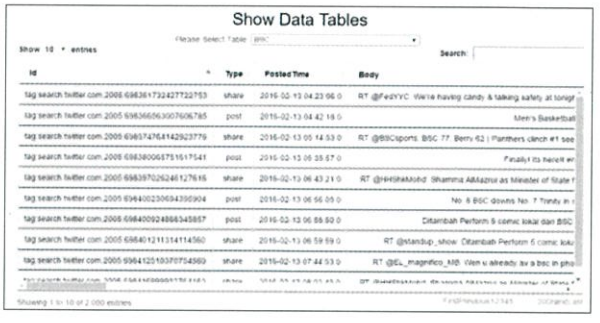
ฟังก์ชันหลักในการทำงานซึ่งประกอบด้วย ฟังก์ชันกรองคำจากทวิตเตอร์ ฟังก์ชันโหลดข้อมูลไปยังฮาดูป ฟังก์ชัน MapReduce และฟังก์ชันการแสดงผล แสดงดังรูปที่ 6

รูปที่ 6. หน้าจอเลือกฟังก์ชันในการทำงาน

การทำงานของฟังก์ชันกรองคำจากทวิตเตอร์ ซึ่งใช้คำว่า “ZA” เป็นคำตัวอย่างในการกรองคำ โดยมีคำสั่งในการกรองคำคือ “ZA posted 2015-01-01 lang en” มีหมายความว่าให้เลือกเอาทวีตเตอร์ที่มีคำว่า ZA อยู่ตั้งแต่วันที่ 1 มกราคม 2015 จนถึงปัจจุบันและเลือกเฉพาะภาษาอังกฤษ เมื่อพิมพ์คำที่ต้องการในช่องทางซ้ายแล้วกด Count Tweets สถานะจะเปลี่ยนเป็นสีเขียวพร้อมบอกจำนวนของทวิตเตอร์ที่ค้นหาเจอ หลังจากนั้นให้ตั้งชื่อตารางที่จะนำข้อมูลไปเก็บในช่องทางขวามือ เมื่อกรอกข้อมูลทั้งสองช่องครบแล้วสีของสถานะทั้งด้านซ้ายและด้านขวาจะเปลี่ยนเป็นสีเขียว ซึ่งแสดงให้เห็นว่าฟังก์ชันการกรองคำพร้อมทำงาน จากนั้นกด Insert Tweets เพื่อเก็บข้อมูลลงในตารางซึ่งสามารถดูข้อมูลในตารางได้ในส่วนของ Show Data Tables แสดงดังรูปที่ 7 และรูปที่ 8

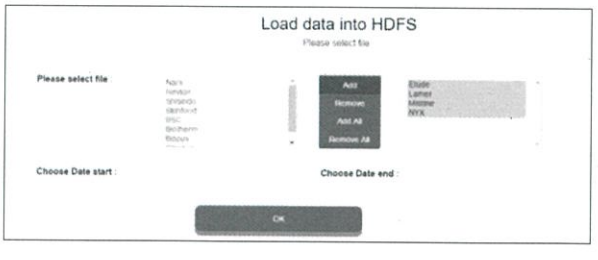


รูปที่ 7. ฟังก์ชันกรองคำจากทวิตเตอร์



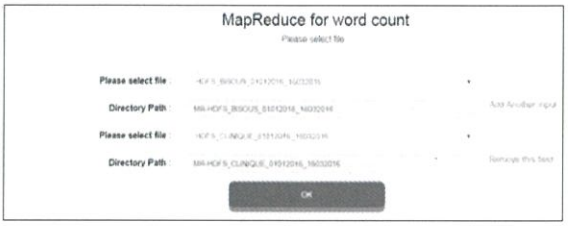
รูปที่ 8. ตารางข้อมูลจากฟังก์ชันกรองคำมาจากทวิตเตอร์

การนำตารางที่เก็บข้อมูลจากฟังก์ชันกรองคำจากทวิตเตอร์ไปเก็บไว้ใน HDFS ที่อยู่ในฮาดูป โดยผู้ใช้สามารถเลือกตารางที่ต้องการได้จากทางซ้ายมือแล้วกด ADD หลังจากนั้นชื่อตารางจะถูกเพิ่มไว้ทางขวามือและผู้ใช้สามารถเลือกช่วงเวลาที่ต้องการเก็บข้อมูลได้โดยระบุวันที่เริ่มต้นและวันที่สิ้นสุด หลังจากนั้นเมื่อกดปุ่ม OK ข้อมูลทั้งหมดจะถูกนำไปเก็บไว้ใน HDFS แสดงดังรูปที่ 9



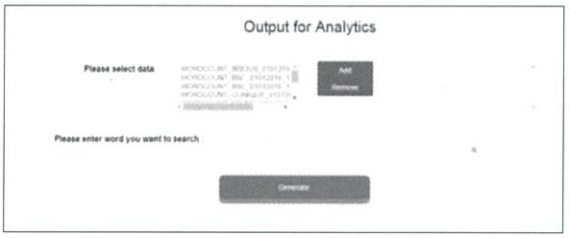
รูปที่ 9. ฟังก์ชันโหลดข้อมูลไปยังฮาดูป

ขั้นตอนการนำไฟล์ที่เก็บไว้อยู่บนฮาดูปนำมาทำ MapReduce เพื่อเป็นการนับคำ โดยผู้ใช้สามารถกดปุ่ม Add Another Input เพื่อทำการเพิ่มตารางที่ต้องการทำ MapReduce ซึ่งสามารถเพิ่มได้หลายๆตารางในเวลาเดียวกัน กระบวนการ MapReduce จะนับคำเพื่อแสดงจำนวนความถี่ของคำที่พบทั้งหมดแสดงดังรูปที่ 10



รูปที่ 10. ฟังก์ชัน MapReduce

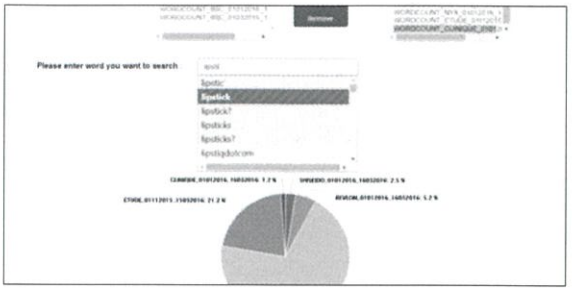
หน้าจอของฟังก์ชันการแสดงผลซึ่งแสดงผลลัพธ์ในรูปแบบของกราฟเปรียบเทียบและกราฟวิเคราะห์อารมณ์และความรู้สึก เพื่อเป็นส่วนหนึ่งที่ช่วยให้ผู้ใช้นำไปใช้ในการประกอบการตัดสินใจในด้านธุรกิจหรือด้านต่างๆได้แสดงดังรูปที่ 11



รูปที่ 11. ฟังก์ชันการแสดงผล

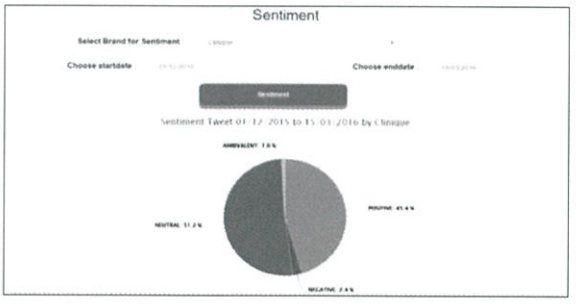
4.2 การนำผลการวิเคราะห์มาใช้งาน

การเปรียบเทียบแบรนด์ของเครื่องสำอางในแต่ละผลิตภัณฑ์ โดยผู้ใช้สามารถเลือกตารางได้หลายๆตาราง แล้วกดปุ่ม Add จากนั้นพิมพ์คำที่ต้องการนำมาวิเคราะห์ เช่น Lipstick, Eyeshadow เป็นต้น แล้วกด Generate โดยผลลัพธ์จะออกมาในรูปแบบของกราฟเปรียบเทียบผลิตภัณฑ์แต่ละแบรนด์ด้วยกราฟวงกลมและคิดผลลัพธ์ออกมาในรูปแบบเปอร์เซ็นต์เพื่อให้ผู้ใช้งานเข้าใจได้ง่าย แสดงดังรูปที่ 12



รูปที่ 12. กราฟเปรียบเทียบผลิตภัณฑ์แต่ละแบรนด์

กราฟการวิเคราะห์อารมณ์และความรู้สึกของข้อความโดยผู้ใช้สามารถเลือกคำที่กรองมาจากทวิตเตอร์เพื่อนำมาวิเคราะห์และสามารถเลือกช่วงเวลาที่ต้องการทำการวิเคราะห์ด้วยการระบุวันที่เริ่มต้นและวันที่สิ้นสุด แล้วกด Sentiment ผลลัพธ์แสดงออกมาในรูปแบบของกราฟวงกลมโดยแต่ละสีของกราฟจะมีความหมายดังนี้ สีเขียว หมายถึง POSITIVE (ด้านบวก) สีแดง หมายถึง NEGATIVE (ด้านลบ) สีฟ้า หมายถึง NEUTRAL (ด้านกลางๆ) และสีเหลือง หมายถึง AMBRVALENT (ด้านคลุมเครือ) แสดงดังรูปที่ 13



รูปที่ 13. กราฟแสดงอารมณ์และความรู้สึก

5. บทสรุป

จากผลการทดลองเว็บแอปพลิเคชันจัดการข้อมูลขนาดใหญ่จากทวิตเตอร์ด้วยไอบีเอ็มบลูมิกซ์ จะเห็นได้ว่าเว็บแอปพลิชันช่วยให้ผู้ใช้สามารถใช้งานและวิเคราะห์ข้อมูลต่างๆได้ง่ายขึ้นด้วยฟังก์ชันที่ใช้งานง่ายและเพิ่มความสะดวกรวดเร็วในการทำงาน อีกทั้งยังมีฟังก์ชันแสดงผลลัพธ์ที่ช่วยเพิ่มความเข้าใจให้กับผู้ใช้งานในรูปแบบของกราฟข้อมูลเปรียบเทียบและกราฟแสดงอารมรณ์และความรู้สึก ซึ่งข้อมูลเหล่านี้สามารถนำไปวิเคราะห์เพื่อให้เกิดประโยชน์

ต่างๆได้อีกมากมายทั้งในเชิงธุรกิจหรือด้านงานวิจัยทางด้าน Data Science

6. เอกสารอ้างอิง

[1] Thaiopensource.org, มาเล่น Bluemix บริการ PaaS จาก IBM กัน, [Online], Available: http://thaiopensource.org/, เข้าถึงเมื่อวันที่ 15 กุมภาพันธ์ 2559.

[2] Ximplesoft.com, Is Hadoop Suitable for BigData, [Online], Available: http://www.simplesoft.com/blog/219, เข้าถึงเมื่อวันที่ 8 ตุลาคม 2558.

[3] รวีรัตน์ จตุราพิศพรชัย และลลนาวัลย์ มนตรีธนสาร, การพัฒนาฐานข้อมูลขนาดใหญ่ด้วยฮาดูป, ปริญญานิพนธ์วิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2555.

[4] สุวรรณี ธูปจีน, เทคนิคการเพิ่มประสิทธิภาพบนกรอบการทำงานของ Map/Reduce, [Online], Available: http://it.kmutnb.ac.th/teacher/DrSucha2155811392.pdf, เข้าถึงเมื่อวันที่ 10 ตุลาคม 2558.