

การประมาณค่าและการจำลองโมเดลทางคณิตศาสตร์สำหรับงานพัสดุคงคลัง  
ในห่วงโซ่อุปทาน

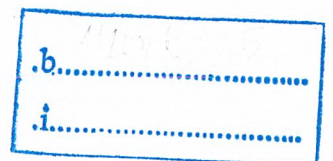
USING MATHEMATICAL MODEL AND APPROXIMATION  
ALGORITHM TO DETERMINE INVENTORY IN SUPPLY CHAIN



นางสาวภาวดี	อิสรงค์
นางสาวสุคนธ์	อนุสุนัย
นางสาวอริตภรณ์	บัวแก้ว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมการวัดคุม  
ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2546

เลขหมู่.....  
เลขทะเบียน..... 550๖๖  
วัน,เดือน,ปี - 7 เม.ย. 2548



**USING MATHEMATICAL MODEL AND APPROXIMATION  
ALGORITHM TO DETERMINE INVENTORY IN SUPPLY CHAIN**

**PAWADEE            ITSARONG  
SUKON              ANUSUNAI  
ATHITAPORN        BUAKAEW**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING  
DEPARTMENT OF INSTRUMENTATION ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2003**

ภาควิชาวิศวกรรมการวัดคุม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ การประมาณค่าและการจำลองโมเดลทางคณิตศาสตร์สำหรับงานพัสดุดัง  
คลังในห่วงโซ่อุปทาน

USING MATHEMATICAL MODEL AND APPROXIMATION  
ALGORITHM TO DETERMINE INVENTORY IN SUPPLY CHAIN

นักศึกษาผู้จัดทำ นางสาวภาวดี อิศรงค์ รหัสประจำตัว 44015527

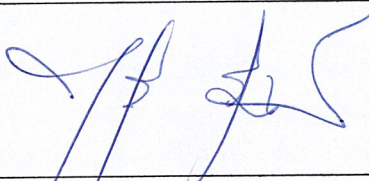
นางสาวสุคนธ์ อนุสุนัย รหัสประจำตัว 44015539

นางสาวอริศภรณ์ บัวแก้ว รหัสประจำตัว 44015544

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2546

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
ดร. ฤดี มาสุขจันทร์	

วัน/เดือน/ปี ที่สอบ 24 มีนาคม 2547

สถานที่สอบ ณ ห้องสอบปริญญาานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว



(รศ. ประสิทธิ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

หัวข้อปริญญานิพนธ์	การประมาณค่าและการจำลองโมเดลทางคณิตศาสตร์สำหรับงานพัสดุ คลังในห่วงโซ่อุปทาน		
	USING MATHEMATICAL MODEL AND APPROXIMATION ALGORITHM TO DETERMINE INVENTORY IN SUPPLY CHAIN		
นักศึกษาผู้จัดทำ	นางสาวภาวดี	อิสรงค์	รหัสประจำตัว 44015527
	นางสาวสุกนธ์	อนุสุนัย	รหัสประจำตัว 44015539
	นางสาวธิดาภรณ์	บัวแก้ว	รหัสประจำตัว 44015544
อาจารย์ที่ปรึกษา	ดร. ฤดี มาสุจินท์		
ปีการศึกษา	2546		

### บทคัดย่อ

การประมาณค่าและการจำลองโมเดลทางคณิตศาสตร์ในงานพัสดุคลัง แบบห่วงโซ่อุปทานนี้ จะอาศัยการวิเคราะห์ข้อมูลจากการวิจัยดำเนินงาน และสร้างระบบจำลองทางคณิตศาสตร์ที่เหมาะสมกับปัญหานั้น เป็นเครื่องมือที่ช่วยในการตัดสินใจ เพื่อนำไปปฏิบัติงานให้ได้ผลดีที่สุด เช่น หาดัชนีต้นทุนต่ำสุด หรือกำไรสูงสุด โดยในโครงการนี้จะวางแผนการในงานอุตสาหกรรมการผลิต เช่น การวางแผนการผลิต การวางแผนการลงทุน เป็นต้น เพื่อการจัดเก็บข้อมูลต่างๆ อย่างเป็นระเบียบเราจึงสร้างฐานข้อมูลโดยใช้โปรแกรม Microsoft Access มาทำการจัดเก็บข้อมูลต่างๆ เช่น ข้อมูลของโรงงานผลิต ข้อมูลของลูกค้า รายละเอียดต่างๆ เกี่ยวกับตัวสินค้า จำนวนความต้องการสินค้า โดยได้ทำการสร้างรูปแบบที่ใช้งานได้ง่ายขึ้นด้วย Visual Basic 6.0 ซึ่งได้รับความนิยมเป็นอย่างมากในการนำมาใช้พัฒนาโปรแกรมบน Windows นำมาใช้ในการติดต่อกับฐานข้อมูล อีกทั้งยังสามารถจัดเก็บข้อมูล แสดงกราฟจำนวนของสินค้า หรือจำนวนสินค้าคลังได้อีกด้วย นอกจากนี้ เราจะใช้ Genetic Local Search Algorithm ช่วยในการหาคำตอบที่เหมาะสมของระบบในการหาจำนวนการผลิตที่ดี เพื่อให้แผนการผลิตมีประสิทธิภาพมากขึ้น

<b>Thesis Title</b>	Using Mathematical Model and Approximation Algorithm to Determine Inventory in Supply Chain	
<b>Authors</b>	Miss. Pawadee	Itsarong
	Miss. Sukon	Anusunai
	Miss. Athitaporn	Buakaew
<b>Thesis Advisor</b>	Dr. Ruedee	Masuchun
<b>Year</b>	2003	

## ABSTRACT

This thesis presents the technique to determine inventory level in supply chain using the mathematical model and the approximation algorithm in operations research. First, a small supply chain is modeled whose data are recorded in the database developed in Microsoft Access. A database application was also created to easily and conveniently modify the database. The mathematical model with the objective function of minimizing the inventory cost and the shortage cost of the product plan are then constructed. Since the approximation algorithms are widely accepted to obtain the good solution for the large-scaled problems in real time, one of the approximation algorithms called a genetic local search algorithm written in Visual Basic 6.0 is chosen to solve for a good production plan.

## กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเพราะได้รับความเมตตาจาก ดร. ฤดี มาสุจันทร์ ที่ได้ให้คำแนะนำและให้คำปรึกษาแก่ผู้จัดทำปริญญาานิพนธ์ตลอดมา ผู้จัดทำรู้สึกซาบซึ้ง และขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์ภาควิชาวิศวกรรมการวัดคุม ที่ให้คำแนะนำอันเป็นประโยชน์ต่อการทำปริญญาานิพนธ์ฉบับนี้

ที่สำคัญ ขอกราบขอบพระคุณ พ่อ แม่ อันเป็นที่รักยิ่ง ที่สนับสนุนและเป็นกำลังใจเสมอมา ในการทำปริญญาานิพนธ์ฉบับนี้

ขอขอบคุณเพื่อนทุกคน ที่เอื้อเฟื้อหนังสือ และข้อมูลต่างๆ ที่เป็นประโยชน์ในการทำปริญญาานิพนธ์

ท้ายที่สุดนี้คุณค่าและประโยชน์อันพึงมีจากปริญญาานิพนธ์ฉบับนี้ ผู้จัดทำขอบแต่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่ออังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VI
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและเหตุจูงใจของการวิจัย	1
1.2 วัตถุประสงค์ของปริญญาานิพนธ์	1
1.3 ขอบเขตของปริญญาานิพนธ์	2
1.4 รายละเอียดของปริญญาานิพนธ์	2
บทที่ 2 ความรู้เบื้องต้นเกี่ยวกับระบบคลัง	3
2.1 ระบบคลังห่วงโซ่อุปทาน	3
2.2 สินค้าคลัง (Inventory)	3
2.3 การควบคุมสินค้าคลัง	4
2.4 ค่าใช้จ่ายในการเก็บสินค้าคลัง	4
บทที่ 3 หลักการและทฤษฎีที่ใช้ในการออกแบบ	5
3.1 หลักการ Optimization	5
3.2 ฐานข้อมูล (Database)	8
3.2.1 Microsoft Access	9
3.2.2 การสร้างฐานข้อมูลด้วย Microsoft Access	10
3.2.3 วิธีการออกแบบตาราง	11
3.2.4 การออกแบบตารางที่ใช้ในการศึกษาโปรเจกต์	14
3.2.5 การกำหนดความสัมพันธ์ของตาราง	24
3.3 การเขียนโปรแกรมด้วย Visual Basic	25
3.3.1 การเขียนโปรแกรมด้วย Visual Basic ขั้นพื้นฐาน	25

## สารบัญ(ต่อ)

	หน้า
3.3.2 ส่วนประกอบของหน้าต่าง Visual Basic 6.1	27
3.3.3 การจัดการกับ โปรเจ็คต์และฟอร์ม	30
3.3.4 ขั้นตอนในการพัฒนาโปรแกรมของ Visual Basic	34
3.3.5 วิธีการติดต่อกับฐานข้อมูลของ MS-Visual Basic 6.0	45
3.3.6 Data Form Wizard	50
3.3.7 การออกแบบการสร้างเมนูและทูลบาร์ของระบบสินค้าคงคลัง	53
3.3.8 ขั้นตอนการสร้างทูลบาร์	55
<b>บทที่ 4 การทดลองโครงงาน</b>	<b>59</b>
4.1 การทดสอบโปรแกรม	59
<b>บทที่ 5 สรุปการทำงานของโครงการและแนวทางการพัฒนา</b>	<b>68</b>
5.1 สรุปการทำงานของโครงงาน	68
5.2 แนวทางการพัฒนา	68
<b>บรรณานุกรม</b>	<b>69</b>
<b>ภาคผนวก</b>	<b>70</b>

# สารบัญภาพ

ภาพที่	หน้า
3.1 แสดงแผนผัง การทำงานบนหลักการของ Genetic local search algorithm	7
3.2 แสดงหน้าต่างในการเข้าสู่โปรแกรม Microsoft Access	10
3.3 แสดงมุมมองการออกแบบตาราง	11
3.4 แสดงการออกแบบตารางในมุมมองออกแบบ	11
3.5 แสดงชื่อของตารางต่างๆ	13
3.6 แสดงข้อมูลในตาราง Supplier ในมุมมองออกแบบ	14
3.7 แสดงข้อมูลในตาราง Supplier ในมุมมองแผ่นข้อมูล	14
3.8 แสดงข้อมูลในตาราง Manu ในมุมมองออกแบบ	15
3.9 แสดงข้อมูลของตาราง Manu ในมุมมองแผ่นข้อมูล	15
3.10 แสดงข้อมูลของตาราง Retailer ในมุมมองออกแบบ	16
3.11 แสดงข้อมูลของตาราง Retailer ในมุมมองแผ่นข้อมูล	16
3.12 แสดงข้อมูลของตาราง SupplierProduct ในมุมมองออกแบบ	17
3.13 แสดงข้อมูลของตาราง SupplierProduct ในมุมมองแผ่นข้อมูล	17
3.14 แสดงข้อมูลของตาราง ManuProduct ในมุมมองออกแบบ	18
3.15 แสดงข้อมูลของตาราง ManuProduct ในมุมมองแผ่นข้อมูล	18
3.16 แสดงข้อมูลของตาราง ProductDetail ในมุมมองออกแบบ	19
3.17 แสดงข้อมูลของตาราง ProductDetail ในมุมมองแผ่นข้อมูล	19
3.18 แสดงข้อมูลของตาราง ManuOrder ในมุมมองออกแบบ	20
3.19 แสดงข้อมูลของตาราง ManuOrder ในมุมมองแผ่นข้อมูล	20
3.20 แสดงข้อมูลของตาราง RetailerOrder ในมุมมองออกแบบ	21
3.21 แสดงข้อมูลของตาราง RetailerOrder ในมุมมองแผ่นข้อมูล	21
3.22 แสดงข้อมูลของตาราง RetailerSale ในมุมมองออกแบบ	22
3.23 แสดงข้อมูลของตาราง RetailerSale ในมุมมองแผ่นข้อมูล	22
3.24 แสดงข้อมูลของตาราง Genetic ในมุมมองออกแบบ	23
3.25 แสดงข้อมูลของตาราง Genetic ในมุมมองแผ่นข้อมูล	23
3.26 แสดงความสัมพันธ์ของตารางแต่ละตาราง	24
3.27 แสดงการเรียก Project ใหม่ขึ้นมาใช้งาน	25
3.28 แสดง การเรียกใช้ Project เดิมมาเก็บไว้ใน Directory	26

## สารบัญภาพ(ต่อ)

ภาพที่	หน้า
3.29 รูปแสดง Project ที่เคยถูกเรียกมาใช้	27
3.30 แสดงส่วนประกอบของหน้าต่างของ Visual Basic	27
3.31 แสดงรูปฟอร์มเปล่า	28
3.32 แสดง Tool Box	28
3.33 แสดง Tool Bar	29
3.34 แสดง Properties Window	29
3.35 แสดง Form Layout Window	30
3.36 แสดงการ save ฟอร์ม	31
3.37 แสดงการเลือกเมนู File -> Make ExProject.exe	32
3.38 แสดงการตั้งชื่อไฟล์จะได้นามสกุลเป็นจุด EXE	32
3.39 แสดงการเรียกเมนู Start -> Run -> Browse	33
3.40 แสดงรูปแบบ Form เมื่อเปลี่ยนเป็น .EXE	33
3.41 แสดงแถบ และ Properties	35
3.42 แสดง TextBox และ Properties	35
3.43 แสดงปุ่ม command และ Properties	36
3.44 แสดงเฟรม และ Properties	36
3.45 แสดงแถบเลื่อนในแนวนอนและแนวตั้ง	37
3.46 แสดง Timer และ Properties	39
3.47 แสดงการเพิ่มฟอร์มชนิด MDI เข้ามาในใช้ใน VBIDE	43
3.48 แสดงฟอร์มชนิด MDI ที่ถูกเพิ่มเข้ามาใน VBIDE	43
3.49 แสดงลักษณะไอคอนที่ปรากฏอยู่ใน Project Explorer ที่กำกับฟอร์มชนิด MDI	44
3.50 ไอคอนของฟอร์มชนิด MDI และฟอร์มชนิด Child	44
3.51 แสดงการสร้างฟอร์มเพื่อเชื่อมฐานข้อมูลจาก Data Control	45
3.52 แสดงการ Run Program หลังการเชื่อมฐานข้อมูลจาก Data Control แล้ว	46
3.53 แสดงการเลือกเมนูคำสั่ง Add-Ins>Add-In Manager	50
3.54 แสดงการ Run Program หลังการเชื่อมฐานข้อมูลจาก Data Control แล้ว	53
3.55 แสดงการเลือกคำสั่ง Add Components	53
3.56 แสดงไดอะล็อกบ็อกซ์ Components	54

## สารบัญภาพ(ต่อ)

ภาพที่	หน้า
3.57 แสดง icon ของคอนโทรล Microsoft Windows Common Control 6.0 ในแถบเครื่องมือ	54
3.58 แสดงการเพิ่มฟอร์มชนิด MDI เข้ามาใน VBIDE	55
3.59 แสดงฟอร์มชนิด MDI ที่ถูกเพิ่มเข้ามาใน VBIDE	55
3.60 แสดงหลังจากใส่รูปภาพให้กับคอนโทรล ImageList	56
3.61 แสดงการกำหนดคอนโทรล ImageList1 ให้กับคอนโทรล ToolBar1	56
3.62 แสดงการกำหนดจำนวนปุ่มที่จะปรากฏในคอนโทรล ToolBar1	57
3.63 แสดงการสร้างเมนูบนฟอร์มด้วย Menu Editor	57
3.64 แสดงการสร้างฟอร์มลูกโดยการเลือกที่ Properties	58
4.1 แสดงหน้าจอหลัก	59
4.2 แสดงรายละเอียดของแหล่งวัตถุดิบ	60
4.3 แสดงรายละเอียดของการสั่งซื้อวัตถุดิบของ Manufacturer	60
4.4 แสดงรายละเอียดของวัตถุดิบของ Supplier	61
4.5 แสดงรายละเอียดของผู้ค้ารายหลัก	61
4.6 แสดงรายละเอียดของสินค้าที่ผลิตโดย Manufacturer	62
4.7 แสดงข้อมูลของแผนการผลิตสินค้า	62
4.8 แสดงข้อมูลของผู้ค้ารายย่อย	63
4.9 แสดงรายละเอียดของการสั่งซื้อสินค้าแต่ละชนิด	63
4.10 แสดงการขายสินค้าของผู้ค้ารายย่อยแต่ละคน	64
4.11 แสดงการรันโปรแกรมการคำนวณบนหลักการของ Genetic Local Search Algorithm	64
4.12 กราฟแสดงการสั่งซื้อสินค้าในแต่ละเดือน	65
4.13 กราฟแสดงผลรวมการสั่งซื้อสินค้าในแต่ละเดือน	65
4.14 กราฟแสดงแผนการผลิตสินค้า	66
4.15 กราฟแสดงการเปรียบเทียบระหว่างปริมาณการสั่งซื้อสินค้าและการได้รับสินค้า	66

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและเหตุจูงใจของการวิจัย

การประเมินค่าและการจำลองโมเดลทางคณิตศาสตร์ (Mathematical model) ในงานพัสดุคงคลัง (Inventory) แบบห่วงโซ่อุปทาน (Supply chain) โดยอาศัยการวิเคราะห์ข้อมูลจากการวิจัยดำเนินงาน (Operation Research) และระบบจำลองทางคณิตศาสตร์ที่เหมาะสมกับปัญหานั้น เพื่อการจัดเก็บข้อมูลอย่างมีระเบียบ ซึ่งจะทำให้ง่ายต่อการใช้งานและค้นหาข้อมูล เราจึงสร้างฐานข้อมูล (Database) โดยใช้โปรแกรม Microsoft Access อีกทั้งในการทำงานกับฐานข้อมูลที่มีอยู่นั้น ยังสามารถหาข้อสรุปหรือแนวโน้มบางอย่างจากข้อมูลบางอย่างได้อย่างสะดวกโดยใช้วิธีการทางสถิติโดยการสร้างกราฟจากฐานข้อมูล ซึ่งจะทำให้ข้อมูลของเรานั้นน่าสนใจ ผู้ใช้งานสามารถเข้าใจได้ง่ายขึ้น ส่วนการใช้โปรแกรม Visual Basic 6 เนื่องจากเป็นภาษาคอมพิวเตอร์ที่ได้รับความนิยมนำมาใช้ในการพัฒนาโปรแกรมบน Windows โดยสามารถสร้างโปรแกรมประยุกต์ (Application) ที่ใช้ในการติดต่อกับฐานข้อมูล เพื่อให้ผู้ใช้งานสามารถติดต่อกับฐานข้อมูลและเรียกใช้งานฐานข้อมูลได้ง่าย ไม่ว่าจะเป็นการจัดเก็บข้อมูล การแสดงผล การค้นหา ฯ ในส่วนของการคำนวณหาคำตอบของแบบจำลองทางคณิตศาสตร์นี้ เราจะใช้ Genetic local search algorithm มาช่วยในการหาคำตอบที่เหมาะสมของระบบ โดยการเขียนโปรแกรมด้วยภาษา Visual Basic

### 1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาหลักการและปัญหาของระบบคงคลังในห่วงโซ่อุปทานที่สามารถพบได้ ในสถานการณ์ปัจจุบัน
2. ศึกษาและพัฒนา software เพื่อนำมาใช้ทดแทน optimization software ในปัจจุบัน ที่มีข้อเสียคือ ราคาสูง และไม่สามารถนำมาแก้ไขปัญหาระบบห่วงโซ่อุปทาน ที่มีขนาดใหญ่แบบ real-time ได้
3. ประยุกต์ใช้โปรแกรม Visual Basic ในการสร้าง software ได้
4. ศึกษาและเรียนรู้ระบบฐานข้อมูลเพื่อนำไปพัฒนาระบบคงคลังในห่วงโซ่อุปทาน ให้เป็นระบบ paperless ได้

### 1.3 ขอบเขตของปริญญานิพนธ์

1. ศึกษาหลักการของ optimization software ที่ใช้ทั่วไปในปัจจุบันและ software ที่จะนำมาใช้ใน project
2. สร้างแบบจำลองทางคณิตศาสตร์ในงานพัสดุคงคลัง ที่มีระบบขนาดเล็กได้
3. ใช้ Genetic local search algorithm หาคำตอบของแบบจำลองเพื่อที่จะคำนวณหาความสมดุลของวัสดุคงคลังที่ผู้ผลิต

### 1.4 รายละเอียดของปริญญานิพนธ์

ในปริญญานิพนธ์ที่ได้ทำ

1. เขียนโปรแกรมที่ใช้ในการคำนวณ คือ Genetic Local Search โดยใช้ Visual Basic 6.0
2. เขียนโปรแกรมอำนวยความสะดวกในการใช้งานติดต่อระหว่าง Visual Basic 6.0 กับ Microsoft Access และการแสดงผล
3. เขียนโปรแกรมการจัดการฐานข้อมูล

## บทที่ 2

# ความรู้เบื้องต้นเกี่ยวกับระบบคลัง

### 2.1 ระบบคลังห่วงโซ่อุปทาน

ห่วงโซ่อุปทานเป็นการศึกษาโดยพิจารณาทุกๆ หน่วยงานในระบบที่มีส่วนเกี่ยวข้องกับ การผลิตสินค้า หรือผลิตภัณฑ์ เช่น Supplies, Manufacturers, Warehouses, Store มีวัตถุประสงค์ เพื่อ

- การผลิตและส่งออกของสินค้าให้เป็นที่ไปตามปริมาณที่เหมาะสม
- ให้สินค้าได้ถูกส่งต่อไปยังตำแหน่งต่างๆ ในห่วงโซ่อุปทานที่เหมาะสม
- การผลิตและส่งออกของสินค้าได้ดำเนินการในช่วงที่เหมาะสม
- ลดค่าใช้จ่ายของทั้งระบบห่วงโซ่อุปทานให้อยู่ในระดับที่ต่ำที่สุด แต่ยังคงไว้ซึ่งระดับ

ความพึงพอใจที่ทุกหน่วยงานยอมรับได้

จะเห็นว่าการศึกษาล่วงโซ่อุปทานเป็นการศึกษาทั้งระบบร่วมกัน มิได้เป็นการพิจารณา เพียงจุดใดจุดหนึ่ง เนื่องจากผลเสียที่เกิดจากการพิจารณาที่จุดใดจุดหนึ่งมีความรุนแรงกว่าผลเสียที่เกิดจากการพิจารณาทุกจุดร่วมกัน เนื่องจากห่วงโซ่อุปทานเป็นระบบที่มีขนาดใหญ่ประกอบไปด้วยระบบย่อยต่างๆ มากมายรวมถึงเวลาที่มีจำกัด โดยโครงการนี้เป็นการศึกษาเพียงระบบเดียวของห่วงโซ่อุปทาน นั่นก็คือ ระบบคลัง

### 2.2 สินค้าคงคลัง (Inventory)

สินค้าคงคลัง (Inventory) หมายถึง สินค้าหรือวัสดุที่เก็บไว้เพื่อการใช้งานหรือไว้จำหน่าย ในอนาคต ซึ่งขึ้นอยู่กับหน่วยงานหรือองค์กรที่จะมีการจัดเก็บสินค้าคงคลังในปริมาณมากน้อยแตกต่างกัน ถ้าหน่วยงานหรือองค์กรมีการจัดเก็บปริมาณสินค้าคงคลังในจำนวนที่มากเกินไป ก็จะทำให้หน่วยงานหรือองค์กรจะต้องมีค่าใช้จ่ายในการเก็บบำรุง รักษาที่เพิ่มขึ้น แต่ถ้าหน่วยงานหรือองค์กรใดมีการจัดเก็บปริมาณสินค้าในจำนวนที่น้อยเกินไป ก็จะทำให้หน่วยงานหรือองค์กรมีสินค้าไม่พอที่จะใช้งานหรือจำหน่าย

สินค้าคงคลังที่จัดเก็บไว้ในหน่วยงานหรือองค์กรสามารถจำแนกได้ 4 ประเภท ดังนี้

1. วัตถุดิบและชิ้นส่วนเพื่อการผลิต
2. สินค้าคงคลังในระหว่างกระบวนการผลิต
3. ผลิตภัณฑ์สำเร็จรูป
4. ชิ้นส่วนของเครื่องจักรเครื่องมือต่างๆ

## 2.3 การควบคุมสินค้าคงคลัง

- การเก็บสินค้าคงคลังของหน่วยงานหรือองค์กรต่างๆ มีสาเหตุและเหตุผลของการจัดเก็บที่แตกต่างกันออกไป เหตุผลทั่วไปที่ต้องทำการจัดเก็บสินค้าคงคลังประกอบด้วย
  - เพื่อให้มีจำหน่ายเมื่อลูกค้าต้องการ ลูกค้าที่ต้องการสินค้าหรือผลิตภัณฑ์อาจจะสั่งซื้อสินค้าหรือผลิตภัณฑ์เมื่อใดก็ได้ การเก็บสินค้าคงเหลือจะช่วยประกันว่าจะไม่เสียโอกาสในการทำการใดๆ จากการจำหน่ายสินค้า เมื่อลูกค้าต้องการ
  - เพื่อปรับเรียบการผลิต (production smoothing) สำหรับกรณีที่มีความต้องการมีลักษณะเป็นไปตามฤดูกาล และกรณีที่วัตถุดิบที่ใช้ในการผลิตมีปริมาณมาก น้อยตามแต่ฤดูกาล
  - เพื่อประโยชน์ในการสั่งซื้อวัตถุดิบหรือชิ้นส่วนที่ละจำนวนมาก การสั่งซื้อสินค้าทีละจำนวนมากจะทำให้ได้ราคาต่อหน่วยถูกลงกว่าการสั่งซื้อทีละน้อยๆ
  - เพื่อแก้ปัญหากรณีที่วัตถุดิบอาจเน่าเสีย ในอุตสาหกรรมการผลิตที่ใช้วัตถุดิบจากพืชผลทางการเกษตร วัตถุดิบอาจจะเกิดการเน่าเสียได้ถ้าทิ้งไว้นานเกินไป ดังนั้นการผลิตจะต้องแปรสภาพสินค้าทางการเกษตรให้เป็นผลิตภัณฑ์สำเร็จรูปให้หมดในเวลาอันสั้น และเก็บผลิตภัณฑ์สำเร็จรูปในลักษณะสินค้าคงคลังเพื่อใช้ในการจำหน่ายต่อไป
  - เพื่อป้องกันการขาดแคลน เพราะถ้าหากเกิดการขาดแคลนจะทำให้เกิดผลเสียหายต่อกระบวนการผลิตถึงกับจะต้องหยุดการผลิต หรืออาจจะทำให้หมดความน่าเชื่อถือจากลูกค้าได้
  - เพื่อแยกกระบวนการผลิตให้เป็นอิสระต่อกัน การเก็บสินค้าคงเหลือในรูปแบบของผลิตภัณฑ์ที่สำเร็จรูประหว่างการผลิตจะสามารถช่วยให้การดำเนินการของแต่ละส่วนในกระบวนการผลิตแยกจากกันอิสระ

ปัญหาการตัดสินใจในเรื่องการควบคุมสินค้าคงคลังมี 2 ประการ ดังนี้

1. จำนวนที่จะสั่งซื้อสินค้าหรือผลิตภัณฑ์ในแต่ละครั้งควรมีปริมาณเท่าไรจึงจะเหมาะสม
2. เวลาที่เหมาะสมในการสั่งซื้อ

## 2.4 ค่าใช้จ่ายในการเก็บสินค้าคงคลัง

2.4.1 ค่าใช้จ่ายในการเก็บรักษาสินค้าคงคลัง คือ ค่าใช้จ่ายที่เกิดจากการเก็บรักษาสินค้าคงคลังให้อยู่ในสภาพที่สามารถใช้งานได้ตลอดเวลา เช่น ค่าเช่าพื้นที่เก็บของ ค่าน้ำ ค่าไฟฟ้า ค่าจ้างคนดูแลรักษา ฯลฯ

2.4.2 ค่าใช้จ่ายในยามขาดแคลน (Shortage Cost) คือ ค่าใช้จ่ายที่เกิดขึ้นเมื่อเกิดการขาดแคลนสินค้าคงคลัง เช่น การขาดโอกาสทำการใดๆ เพราะไม่มีสินค้าจำหน่ายให้แก่ลูกค้า การเสียค่าปรับเมื่อไม่มีสินค้าหรือผลิตภัณฑ์ส่งให้แก่ลูกค้าตามที่ได้ตกลงไว้

## บทที่ 3

# หลักการและทฤษฎีที่ใช้ในการออกแบบ

### 3.1 หลักการ Optimization

จากการวิจัยศึกษาทางด้านอัลกอริทึมเพื่อใช้แก้ปัญหา Optimization ที่มีขนาดใหญ่และไม่สามารถหาคำตอบได้ในเวลาอันรวดเร็วด้วย Optimization ในปัจจุบัน พบว่าวิธี Approximation เป็นวิธีที่สามารถให้คำตอบที่ยอมรับได้ ในเวลาอันรวดเร็ว ถึงแม้ไม่ใช่คำตอบที่ดีที่สุดแต่เมื่อพิจารณาถึงเวลาที่ใช้ในการได้มาซึ่งคำตอบที่ดีที่สุด วิธี Approximation เป็นวิธีที่นักวิจัยทั่วโลกยอมรับ

ตัวอย่างอัลกอริทึมต่างๆ ตามวิธี Approximation คือ genetic algorithm, tabu search algorithm และ simulated annealing algorithm ทั้ง 3 อัลกอริทึมนี้รวมถึงอัลกอริทึมอื่นๆ ตามวิธี Approximation จะใช้หลักการของการสุ่มเป็นส่วนสำคัญในการหาคำตอบ และการได้มาซึ่งคำตอบที่ยอมรับได้ก็เกิดจากการทำซ้ำกันหลายๆ ครั้ง รายละเอียดโดยสรุปของทั้ง 3 อัลกอริทึมสามารถเขียนได้ดังนี้

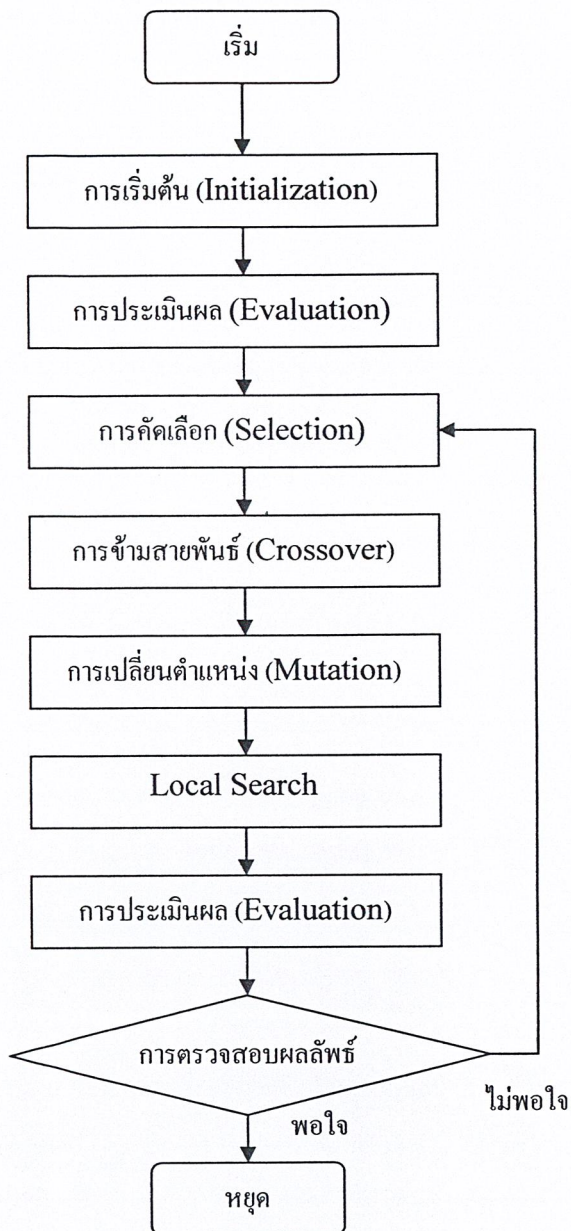
Genetic algorithm: กำเนิดขึ้นจากหลักการพันธุศาสตร์ที่ว่าสิ่งมีชีวิตจะมีการพัฒนาตามลำดับ รุ่นลูกหลานย่อมมีสายพันธุ์ที่มีการพัฒนาว่ารุ่นปู่ย่าตายาย ลำดับขั้นตอนของ genetic algorithm มีดังนี้

1. การเริ่มต้น คือการกำเนิดของประชากรหรือการสุ่มกลุ่มของผลลัพธ์
2. การคัดเลือก บนพื้นฐานของหลักการที่ว่า ประชากรที่มีความแข็งแรงกว่าย่อมมีโอกาสในการอยู่รอดสูงกว่าประชากรที่อ่อนแอกว่า เมื่อนำมาประยุกต์ใช้กับปัญหา optimization นั่นก็คือคำตอบที่ให้ผลลัพธ์ที่ดีกว่าก็จะมีโอกาสสูงกว่าที่จะถูกเลือกไปใช้ในรอบต่อไป
3. การแลกเปลี่ยนทางพันธุกรรม บนพื้นฐานของหลักการที่ว่า โอกาสของการพัฒนาสายพันธุ์จะไม่สามารถเกิดได้ถ้าไม่มีการแลกเปลี่ยนทางพันธุกรรมเกิดขึ้น สายพันธุ์ใหม่ที่แข็งแรงหรือดีกว่าก็就会被คัดเลือกเพื่อการพัฒนาของประชากรใน genetic ถัดไป เมื่อนำมาประยุกต์ใช้กับปัญหา optimization นั่นก็คือคำตอบใหม่ที่ให้ผลลัพธ์ที่ดีกว่าก็จะมีโอกาส สูงกว่าที่จะถูกเลือกไปใช้ใน รอบถัดไป
4. การตรวจสอบผลลัพธ์หยุดเมื่อได้คำตอบที่พอใจแล้ว ถ้ายังไม่พอใจกลับไปขั้นตอนที่ 2 อีกครั้ง

Tabu search algorithm: ถูกคิดค้นเพื่อแก้ไขข้อเสียข้อหนึ่งของวิธี Approximation นั่นก็คือ การวนเวียนอยู่ในห้วงของผลลัพธ์ที่ไม่ใช่ผลลัพธ์ที่ดีที่สุด โดยใช้หลักการของการสร้างรายการต้องห้าม ที่ประกอบด้วยรายการของคำตอบที่ได้ในรอบที่ผ่านมา หลักการของ คือทิศทางของคำตอบที่ได้ มาจากการสุ่มในแต่ละครั้งจะถูกลบทิ้งไว้ในรายการต้องห้าม คำตอบที่จะถูกนำไปใช้ในในรอบถัดไปจะต้องไม่มีทิศทางเหมือนกับทิศทางที่ถูกลบทิ้งเอาไว้ในรายการต้องห้าม เพื่อป้องกันไม่ให้เกิดการวนเวียนกลับไป ณ จุดเดิมอีก

Simulated annealing algorithm: มีหลักการที่คล้ายกับ genetic algorithm ที่ว่าทั้งผลลัพธ์ที่ดีกว่าและแย่กว่ามีโอกาสที่จะไปสู่รอบถัดไป แต่ผลลัพธ์ที่ดีกว่าจะมีโอกาสสูงกว่าผลลัพธ์ที่แย่กว่า นอกจากอัลกอริทึมทั้ง 3 นี้แล้ว ยังมีอัลกอริทึมอื่นอีกรวมทั้งอัลกอริทึมที่เกิดจากการผสมผสานระหว่างอัลกอริทึมทั้ง 3 นี้ เช่น Genetic local search algorithm และ Genetic simulated annealing algorithm จากการศึกษาพบว่าอัลกอริทึมที่เกิดจากการผสมผสานกันนี้ให้ผลลัพธ์ที่ดีกว่า โดยเฉพาะอย่างยิ่ง Genetic local search algorithm ดังนั้น Genetic local search algorithm จะถูกนำมาใช้เพื่อแก้ปัญหาระบบคลังในห่วงโซ่อุปทานของโครงการนี้

หลักการของ Genetic local search algorithm เป็นการผสมผสานระหว่างหลักการของ Genetic algorithm กับ Random search (Local search) นั่นก็คือ การใช้ Genetic algorithm เพื่อหาคำตอบโดยรวมจากประชากรที่มีอยู่หลังจากนั้นใช้ Local search โดยการสุ่มตัวอย่าง บนขอบเขตของคำตอบที่ได้จาก Genetic algorithm เพื่อเพิ่มความถูกต้องภาพที่ 3.1 แสดงแผนผัง การทำงานบนหลักการของ Genetic local search algorithm



ภาพที่ 3.1 แสดงแผนผัง การทำงานบนหลักการของ Genetic local search algorithm

คำอธิบายแผนผังการทำงานบนหลักการของ Genetic local search algorithm

1. การเริ่มต้น (Initialization) เป็นการสร้างกลุ่มของประชากรอย่างอิสระ
2. การประเมินผล (Evaluation) เป็นการคำนวณผลลัพธ์ของแต่ละประชากรที่ได้จากข้อ 1
3. การคัดเลือก (Selection) เป็นการเลือกประชากรที่จะนำไปใช้ในรอบต่อไป โดยประชากรที่ให้ผลลัพธ์ตามข้อ 2 ที่ดีที่สุดจะมีโอกาสมากที่สุดที่จะถูกเลือก นั่นก็คือมีโอกาสมากที่สุดที่จะมีจำนวนมากที่สุดในกลุ่มของประชากรที่ถูกเลือก
4. การข้ามสายพันธุ์ (Crossover) เป็นหนึ่งในขั้นตอนของการแลกเปลี่ยนทางพันธุกรรม ในขั้นตอนนี้จะเป็นการสุ่มเลือกประชากรจากข้อ 3 มาทีละคู่ เพื่อทำการแลกเปลี่ยนสายพันธุ์ระหว่างประชากร
5. การเปลี่ยนตำแหน่ง (Mutation) เป็นหนึ่งในขั้นตอนของการแลกเปลี่ยนสายพันธุ์ทางพันธุกรรม ในขั้นตอนนี้จะเป็นการสับเปลี่ยนสายพันธุ์ภายในแต่ละประชากรที่ได้มาจากข้อ 4
6. Local Search เป็นการสุ่มตัวอย่างบนขอบเขตของคำตอบที่ได้จากข้อ 5
7. การประเมินผล (Evaluation) เป็นการคำนวณผลลัพธ์ของแต่ละประชากรที่คำนวณได้จากข้อ 6
8. การตรวจสอบผลลัพธ์ เพื่อตรวจสอบความเหมาะสมและถูกต้องของคำตอบ เมื่อใดที่ได้คำตอบเป็นที่น่าพอใจก็สามารถที่จะหยุดและใช้คำตอบที่ดีที่สุดเป็นคำตอบของปัญหา แต่ถ้าคำตอบที่ได้ไม่เป็นที่น่าพอใจ ให้กลับไปที่ขั้นตอนของการคัดเลือกใหม่ กฎทั่วไปที่ใช้เป็นตัววัดความพอใจมี 2 วิธีหลักๆ คือ

8.1 การตรวจสอบที่จำนวนรอบ โดยทั่วไปจะเป็นการกำหนดจำนวนรอบที่ใช้ในการหาคำตอบ เช่น 10,000 รอบ

8.2 การตรวจสอบผลลัพธ์ที่ดีที่สุดในรอบปัจจุบัน เทียบกับผลลัพธ์ที่ดีที่สุดในรอบก่อนหน้า ถ้าผลลัพธ์ที่ดีที่สุดไม่มีการเปลี่ยนแปลงในทางที่ดีขึ้นเป็นนัยสำคัญก็สามารถหยุดได้ และใช้คำตอบที่ดีที่สุดจากรอบปัจจุบันเป็นคำตอบของปัญหา

### 3.2 ฐานข้อมูล (Database)

ฐานข้อมูล (Database) คือ วิธีการจัดเก็บข้อมูลที่มีความสัมพันธ์กันอย่างมีระเบียบ ซึ่งให้ง่ายต่อการค้นหา และการใช้งาน ซึ่งรูปแบบของฐานข้อมูลที่นิยมใช้โดยทั่วไปคือ ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เป็นรูปแบบการจัดเก็บข้อมูลในฐานข้อมูลที่มีสัมพันธ์กัน จะเก็บข้อมูลในรูปของตารางต่าง ๆ ที่มีความสัมพันธ์กัน

### 3.2.1 Microsoft Access

Microsoft Access เป็นโปรแกรมจัดการฐานข้อมูลที่มีความสามารถสูงในการใช้งาน และง่ายต่อการออกแบบที่จะนำไปใช้งาน ในการประยุกต์กับ โปรแกรม Visual Basic

#### 3.2.1.1 องค์ประกอบของฐานข้อมูล

##### Database Application

เป็น Application ที่สร้างไว้ให้ผู้ใช้ได้ใช้งานได้สะดวกยิ่งขึ้น ซึ่งมีรูปแบบการติดต่อกับฐานข้อมูลแบบเมนู หรือกราฟิก

##### Database Management System

ระบบจัดการฐานข้อมูล เป็นซอฟต์แวร์ที่ทำหน้าที่จัดการข้อมูลในฐานข้อมูล การจัดเก็บ การแสดงผล การค้นหา การสำรวจข้อมูล ฯลฯ โดยจะเป็นเครื่องมือในการทำงาน และเป็นตัวกลางที่เชื่อมผ่านระหว่าง Application ฐานข้อมูลที่สร้างขึ้นกับตัวข้อมูลในฐานข้อมูล

##### Database Server

เป็นคอมพิวเตอร์ที่ให้บริการในการจัดการกับฐานข้อมูล

##### Data

เป็นตัวเนื้อหาของข้อมูลที่ใช้งาน ซึ่งจะถูกรวบรวมในหน่วยความจำของ Database Server โดยจะถูกเรียกมาใช้งานจากระบบจัดการฐานข้อมูล

##### Database Administrator

เป็นผู้ดูแลข้อมูลในฐานข้อมูลผ่านระบบการจัดการฐานข้อมูล

#### 3.2.1.2 นิยามต่างๆ ของ Microsoft Access

Database คือ ไฟล์ซึ่งประกอบด้วยข้อมูลและเครื่องมือต่างๆ ทั้งหมด ที่ใช้สำหรับการดำเนินการกับข้อมูล

Table คือ ตารางที่ใช้ในการจัดเก็บข้อมูล ซึ่งจะประกอบด้วย Field และ Record

Queries คือ แบบสอบถามข้อมูลจากฐานข้อมูล ใช้ในการสอบถามข้อมูล เพื่อเรียกดูข้อมูลใน Table และจะช่วยในการทำงานให้สะดวกมากยิ่งขึ้น

Form เป็นส่วนที่ใช้ในการออกแบบหน้าจอเป็นแบบฟอร์มต่างๆ เพื่อรับข้อมูลจากผู้ใช้งานและนำไปจัดเก็บใน Table

Report เป็นส่วนที่ใช้ในการสร้างรายงานต่างๆ

Macro ใช้ในการสร้างชุดคำสั่งโดยรวมขั้นตอนการทำงานต่างๆ ไว้ในคำสั่งเดียว

Modules ใช้ในการเขียนโปรแกรม เพื่อให้ Microsoft Access มีการทำงานที่ดียิ่งขึ้นชุดคำสั่งที่เขียนใน Module สามารถเรียกใช้ได้ด้วยทุกๆ Form

### 3.2.1.3 Relation Database

คือ กลุ่มของข้อมูลที่มีความสัมพันธ์ ซึ่งมีการจัดเก็บไว้ในตาราง (Table) ในตารางแต่ละตารางจะประกอบไปด้วย Record และใน Record จะประกอบไปด้วยไปด้วยกลุ่มของ Field เพราะ ฉะนั้นในแต่ละ Table จะประกอบไปด้วย Record ซึ่งจะอยู่ในรูปของ Row และมี Field จะอยู่ในรูปของ Column และในแต่ละ Table จะต้องมีการกำหนด Field ที่ใช้กำหนดเป็น Key ซึ่งจะมี Primary Key ที่ใช้เป็น Key หลักของ Table และ Foreign Key ที่เป็น Field ที่ใช้ในการอ้างอิงถึงข้อมูลใน Table ที่สัมพันธ์กัน

#### 3.2.1.4 ประเภทของความสัมพันธ์ระหว่าง Table แบ่งได้ 3 ประเภท

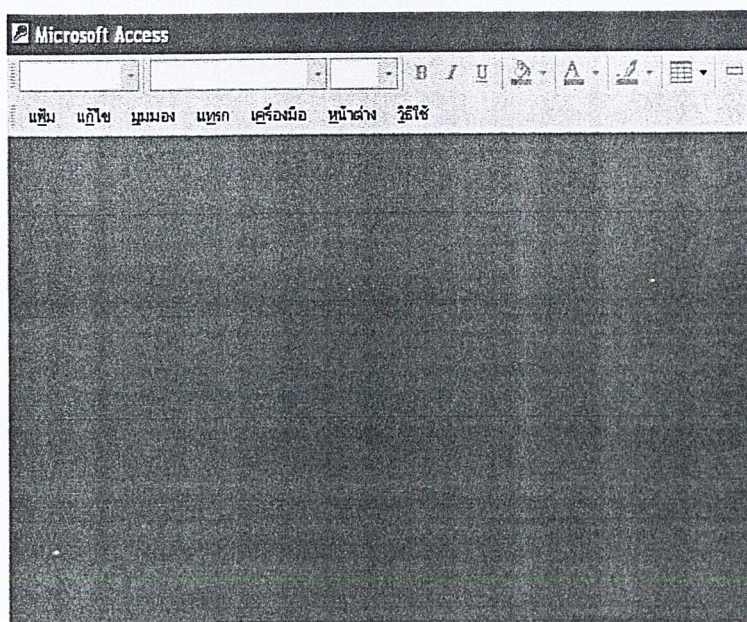
ประเภทที่ 1 แบบ One-to-Many คือ ข้อมูล 1 Record ใน Table หนึ่งจะมีความสัมพันธ์กับข้อมูลในอีก Table หนึ่งมากกว่า 1 Record

ประเภทที่ 2 แบบ One-to-One คือ ข้อมูลแต่ละ Record ใน Table หนึ่งจะมีความสัมพันธ์กับข้อมูลในอีก Table หนึ่งเพียง Record

ประเภทที่ 3 แบบ Many-to-Many คือ หลาย Record ใน Table หนึ่งจะมีความสัมพันธ์กับอีกหลาย Record ในอีก Table

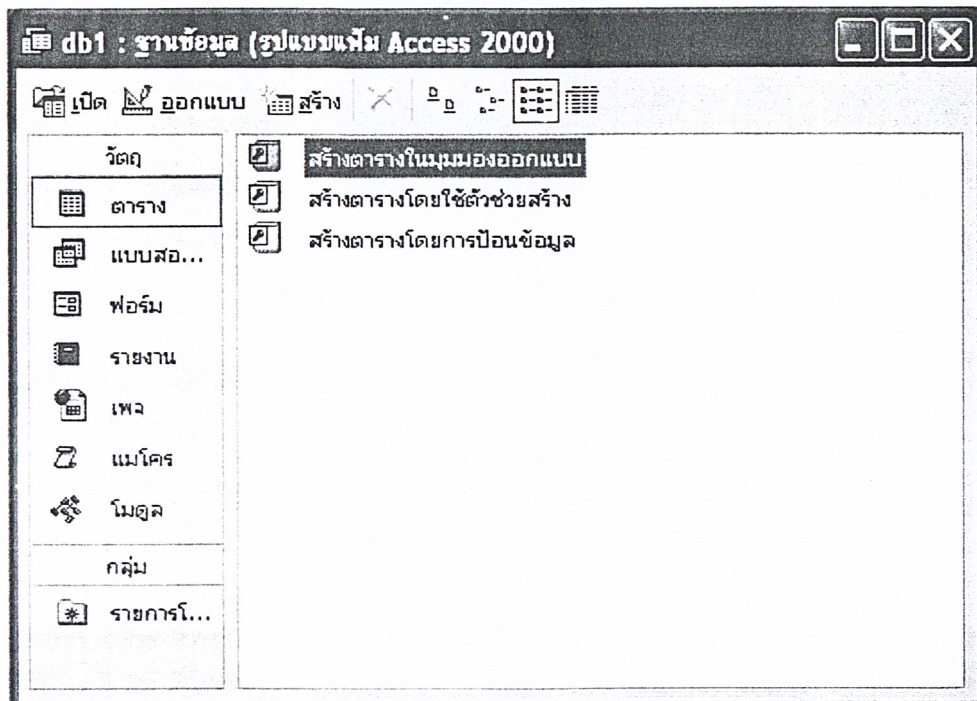
## 3.2.2 การสร้างฐานข้อมูลด้วย Microsoft Access

3.2.2.1 เปิดโปรแกรม Microsoft Access จะปรากฏรูปแบบดังภาพที่ 3.2 ถ้าเป็นการสร้างข้อมูลครั้งแรก ในช่อง Field Name ให้ตั้งชื่อฐานข้อมูลที่สร้างขึ้นจากนั้นให้คลิกที่ปุ่ม Create



ภาพที่ 3.2 แสดงหน้าต่างในการเข้าสู่โปรแกรม Microsoft Access

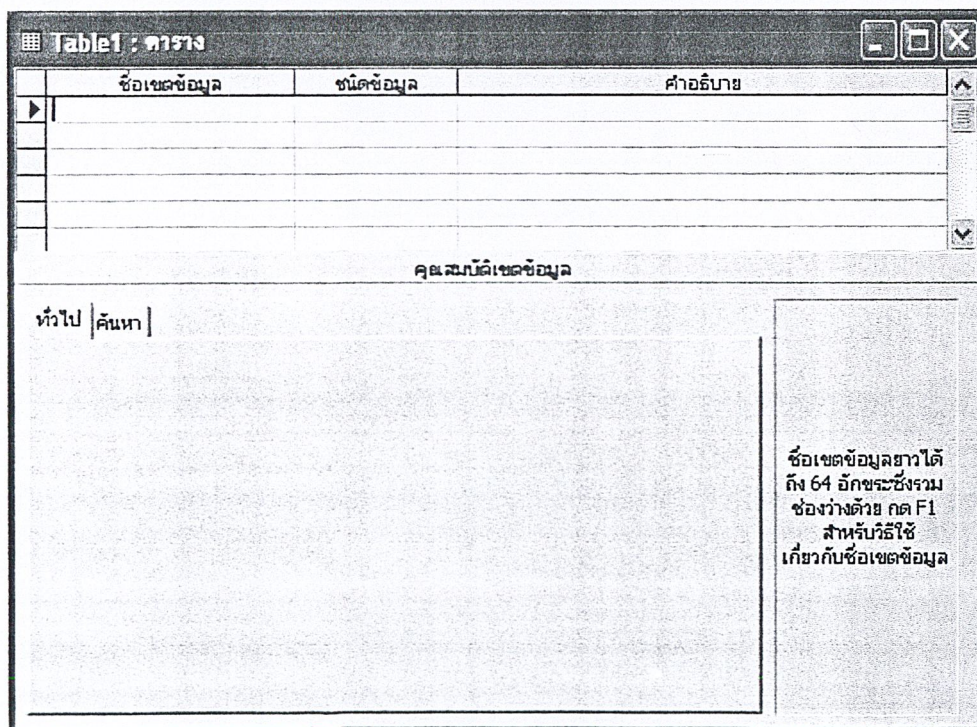
### 3.2.2.2 จากนั้นจะเข้าสู่ฐานข้อมูล ที่ได้สร้างไว้ดังแสดงในภาพที่ 3.3



ภาพที่ 3.3 แสดงมุมมองการออกแบบตาราง

### 3.2.3 วิธีการออกแบบตาราง

จากภาพที่ 3.3 ให้เลือก สร้างตารางในมุมมองออกแบบ จะปรากฏดังภาพที่ 3.4



ภาพที่ 3.4 แสดงการออกแบบตารางในมุมมองออกแบบ

จากในภาพที่ 3.4 จะมีส่วนประกอบของการออกแบบตารางต่างๆ ดังนี้

Field Name	Field ต่างๆ ที่ใช้ในการออกแบบ
Data Type	ชนิดข้อมูลที่จัดเก็บให้กับ Field Name นั้น
Description	เป็นคำอธิบายของลักษณะ Filed Name นั้น
Field Properties	เป็นส่วนที่ใช้ในการกำหนดคุณสมบัติต่างๆ ของ Field

### 3.2.3.1 การกำหนด Primary key

1. เลือก Field ที่ต้องการให้เป็น Primary Key ซึ่งก็คือ Field ที่ต้องการไม่ให้ข้อมูลมีค่าที่ ซ้ำกัน ในแต่ละ Record และไม่สามารถที่จะเป็นค่าว่างได้
2. ไปที่ Toolbar จากนั้นคลิกที่รูปกุญแจ
3. จะสามารถสังเกตได้ว่าที่ Field ที่เราต้องการให้เป็น Primary Key จะปรากฏรูปกุญแจอยู่ที่ข้างหน้า




### 3.2.3.2 การกำหนดความสัมพันธ์ระหว่าง Table

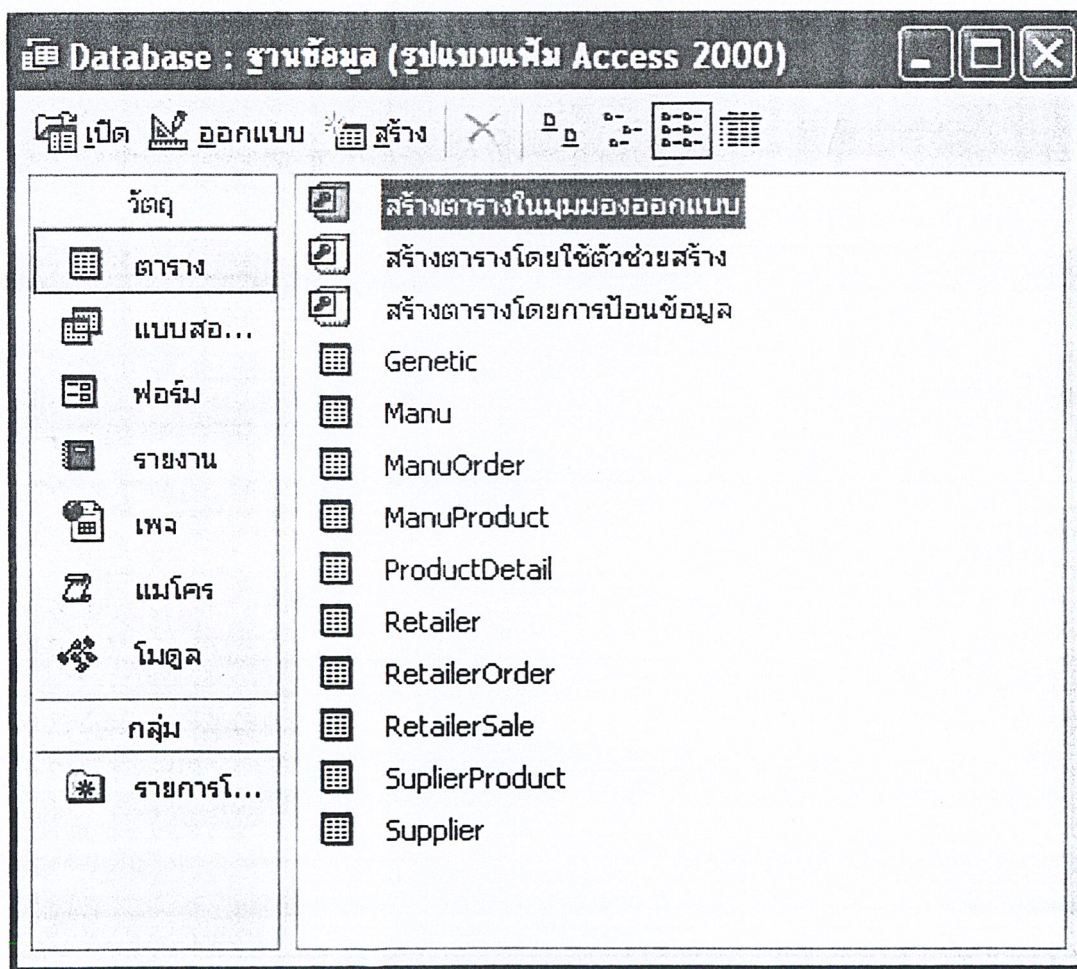
การสร้างความสัมพันธ์ระหว่าง Table ที่เรียกว่า Relationship มีประโยชน์สำหรับผู้ออกแบบฐานข้อมูล ซึ่งสามารถจะช่วยดูแลความถูกต้องของข้อมูล (Data Integrity) และความสอดคล้องของข้อมูล (Consistency) เพื่อใช้ในการเพิ่ม แก้ไข ลบ ข้อมูล ทำให้ช่วยลดภาระในการเขียนโปรแกรมลงได้

ประโยชน์ของการสร้างความสัมพันธ์ระหว่างตาราง

1. ช่วยควบคุมความถูกต้องของข้อมูล (Data Integrity) ในตารางแต่ละตารางได้มีการกำหนด Primary Key และบางตารางได้มีการกำหนด Foreign key ซึ่ง key ทั้งสองได้มีการควบคุมความถูกต้องของข้อมูลอยู่แล้ว แต่ถ้าไม่มีการกำหนดความสัมพันธ์ระหว่างตาราง ก็จะไม่สามารถควบคุมความถูกต้องของ key ทั้งสองได้ เพราะระบบฐานข้อมูลจะมองตารางในฐานข้อมูลเป็นตารางที่ไม่ขึ้นอยู่ต่อกัน
2. มีความสอดคล้องในความสัมพันธ์ของข้อมูล (Consistency) การกำหนดความสัมพันธ์ระหว่างตารางยังช่วยลดภาระในการเขียนโปรแกรม เนื่องจากความสัมพันธ์ที่เกี่ยวเนื่องกันระหว่างตาราง

สัญลักษณ์และความหมายที่ใช้ในการกำหนดความสัมพันธ์ระหว่างตาราง มีดังนี้

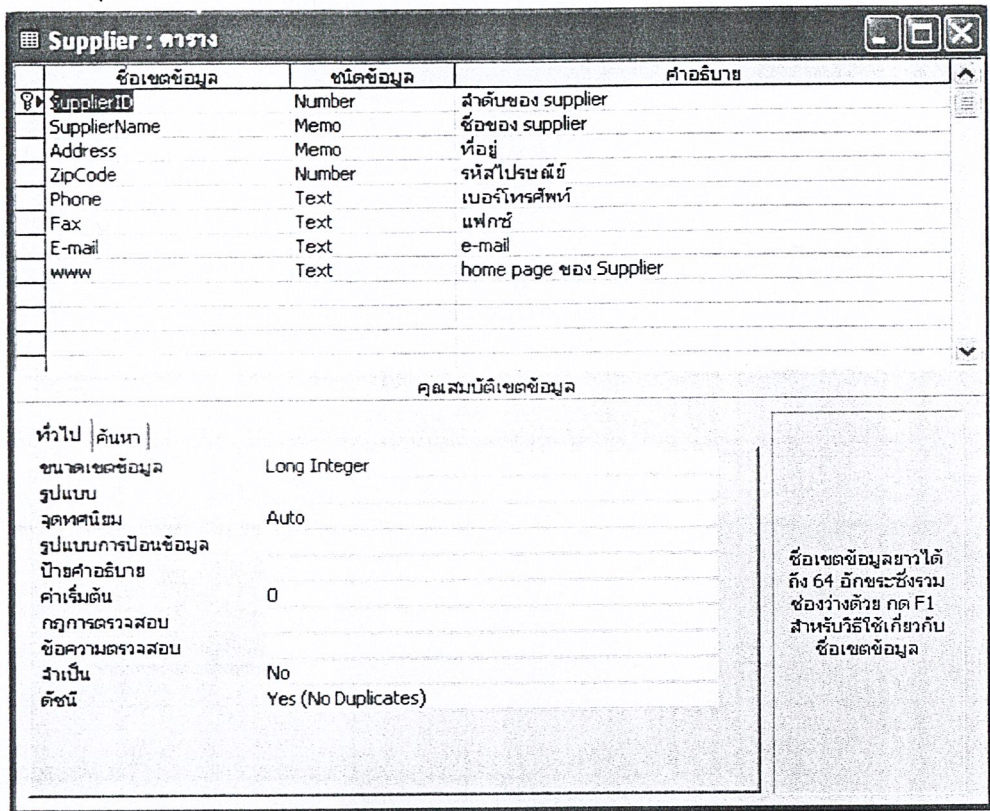
- |    |   |         |  |
|----|---|---------|--|
| 1. |  | หมายถึง | ความสัมพันธ์แบบหนึ่ง เช่น หนึ่งคน หนึ่งบริษัท            |
| 2. |  | หมายถึง | ความสัมพันธ์แบบหลาย เช่น หลายคน หลายบริษัท               |
| 3. |  | หมายถึง | เส้นที่ใช้ในการเชื่อมโยงความสัมพันธ์ระหว่างตารางสองตาราง |



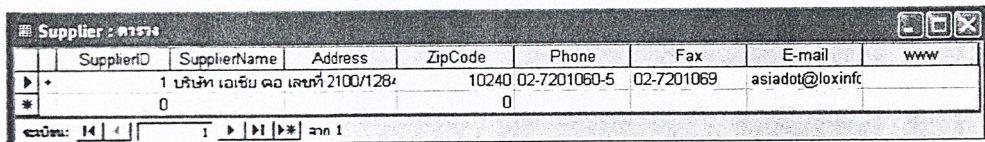
ภาพที่ 3.5 แสดงชื่อของตารางต่างๆ

### 3.2.4 การออกแบบตารางที่ใช้ในการศึกษาโปรเจกต์

3.2.4.1 ตาราง Supplier ซึ่งเป็นตารางที่มีข้อมูลและรายละเอียดเกี่ยวกับ Supplier หรือแหล่งของวัตถุดิบ มี Field ต่างๆ ดังนี้



ภาพที่ 3.6 แสดงข้อมูลในตาราง Supplier ในมุมมองออกแบบ



ภาพที่ 3.7 แสดงข้อมูลในตาราง Supplier ในมุมมองแผ่นข้อมูล

- SupplierID เป็น Field ที่ใช้สำหรับลำดับของ SupplierID
- SupplierName เป็น Field สำหรับชื่อของ Supplier
- Address เป็น Field สำหรับที่อยู่ของ Supplier
- ZipCode เป็น Field สำหรับใส่รหัสไปรษณีย์
- Phone เป็น Field สำหรับเบอร์โทรศัพท์
- Fax เป็น Field สำหรับเบอร์ Fax
- e-mail เป็น Field สำหรับ e-mail address
- www เป็น Field สำหรับ Home Page

3.2.4.2 ตาราง Manu ซึ่งจะเป็นตารางที่มีข้อมูลและรายละเอียดเกี่ยวกับโรงงานผู้ผลิต มี Field ต่างๆดังนี้

ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
ManuID	Number	ลำดับของโรงงานผู้ผลิต
ManuName	Memo	ชื่อของโรงงานผู้ผลิต
Address	Memo	ที่อยู่ของโรงงานผู้ผลิต
ZipCode	Number	รหัสไปรษณีย์
Phone	Text	เบอร์โทรศัพท์
Fax	Text	แฟกซ์
E-mail	Text	e-mail
www	Text	home page ของ Manufacture

คุณสมบัติเขตข้อมูล

ทั่วไป   ค้นหา	
ขนาดเขตข้อมูล	Long Integer
รูปแบบ	
จุดทศนิยม	Auto
รูปแบบการป้อนข้อมูล	
ป้ายคำอธิบาย	
ค่าเริ่มต้น	0
กฎการตรวจสอบ	
ข้อความตรวจสอบ	
จำเป็น	No
ดัชนี	Yes (No Duplicates)

ชื่อเขตข้อมูลยาวได้ถึง 64 อักขระซึ่งรวมช่องว่างด้วย กด F1 สำหรับวิธีใช้เกี่ยวกับชื่อเขตข้อมูล

ภาพที่ 3.8 แสดงข้อมูลในตาราง Manu ในมุมมองออกแบบ

ManuID	ManuName	Address	ZipCode	Phone	Fax	E-mail	www
+	บริษัท จีเนิส โอ เลขที่ 7 อาคารศูนย์		10320	02-6420610	02-6420620	graphic@inksc-	
*	0		0				

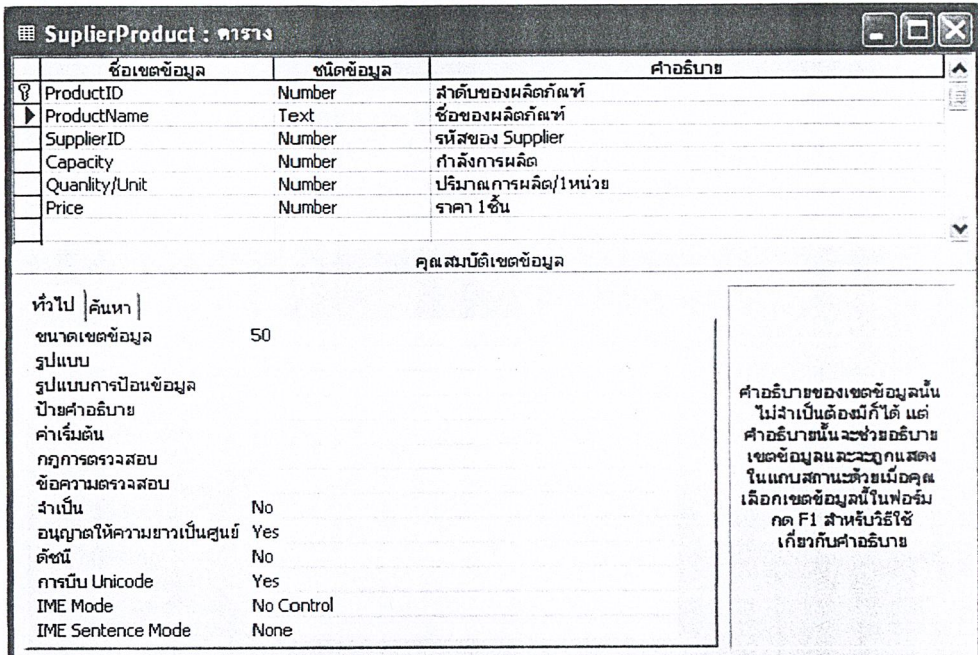
ระเบียน: 14 | < | 1 | > | \* | จาก 1

ภาพที่ 3.9 แสดงข้อมูลของตาราง Manu ในมุมมองแผ่นข้อมูล

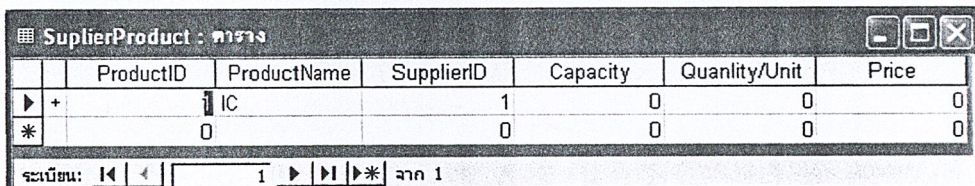
- ManuID เป็น Field ที่ใช้สำหรับลำดับของโรงงานผู้ผลิต
- ManuName เป็น Field สำหรับชื่อของโรงงานผู้ผลิต
- Address เป็น Field สำหรับที่อยู่ของโรงงานผู้ผลิต
- ZipCode เป็น Field สำหรับไปรษณีย์
- Phone เป็น Field สำหรับเบอร์โทรศัพท์
- Fax เป็น Field สำหรับเบอร์ Fax
- e-mail เป็น Field สำหรับ e-mail address
- www เป็น Field สำหรับ Home Page ของโรงงานผู้ผลิต



3.2.4.4 ตาราง SupplierProduct ข้อมูลและรายละเอียดเกี่ยวกับสินค้าหรือผลิตภัณฑ์ที่มีการผลิต โดย Supplier ซึ่งมี Filed ต่างๆ ดังนี้



ภาพที่ 3.12 แสดงข้อมูลของตาราง SupplierProduct ในมุมมองออกแบบ



ภาพที่ 3.13 แสดงข้อมูลของตาราง SupplierProduct ในมุมมองแผ่นข้อมูล

- ProductID เป็น Field สำหรับลำดับของสินค้าหรือผลิตภัณฑ์ที่ Supplier เป็นผู้ผลิต
- ProductName เป็น Field สำหรับชื่อของสินค้าหรือผลิตภัณฑ์
- SupplierID เป็น Field สำหรับรหัสของ Supplier ที่เป็นผู้ผลิตสินค้า หรือผลิตภัณฑ์นี้
- Capacity เป็น Field สำหรับกำลังการผลิตของสินค้าหรือผลิตภัณฑ์นี้ ที่ Supplier สามารถจะผลิตได้
- Price เป็น Field สำหรับราคาของสินค้าหรือผลิตภัณฑ์นี้
- Quantity/Unit เป็น Field ที่แสดงจำนวนของผลิตภัณฑ์ใน 1 หน่วย

3.2.4.5 ตาราง ManuProduct แสดงข้อมูลและรายละเอียดที่เกี่ยวข้องกับสินค้าหรือผลิตภัณฑ์ที่มีการผลิตโดย Manufacturer ซึ่งมี Field ต่างๆ ดังนี้

ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
ProductID	Number	รหัสของผลิตภัณฑ์
ProductName	Text	ชื่อของผลิตภัณฑ์
ManuID	Number	รหัสของโรงงานผู้ผลิต
Capacity	Number	กำลังการผลิต
Quantity/Unit	Number	ปริมาณของสินค้าใน 1 หน่วย
InventoryCost	Number	ค่าใช้จ่ายในการเก็บสินค้าต่อหน่วย
Price	Number	ราคา ชิ้น

คุณสมบัติเขตข้อมูล	
ทั่วไป  ค้นหา	
ขนาดเขตข้อมูล	Long Integer
รูปแบบ	
จุดทศนิยม	Auto
รูปแบบการป้อนข้อมูล	
ป้ายคำอธิบาย	
ค่าเริ่มต้น	0
กฎการตรวจสอบ	
ข้อความตรวจสอบ	
จำเป็น	No
ดัชนี	Yes (No Duplicates)

ชื่อเขตข้อมูลยาวได้ถึง 64 อักขระซึ่งรวมช่องว่างด้วย กัด F1 สำหรับวิธีใช้เกี่ยวกับชื่อเขตข้อมูล

ภาพที่ 3.14 แสดงข้อมูลของตาราง ManuProduct ในมุมมองออกแบบ

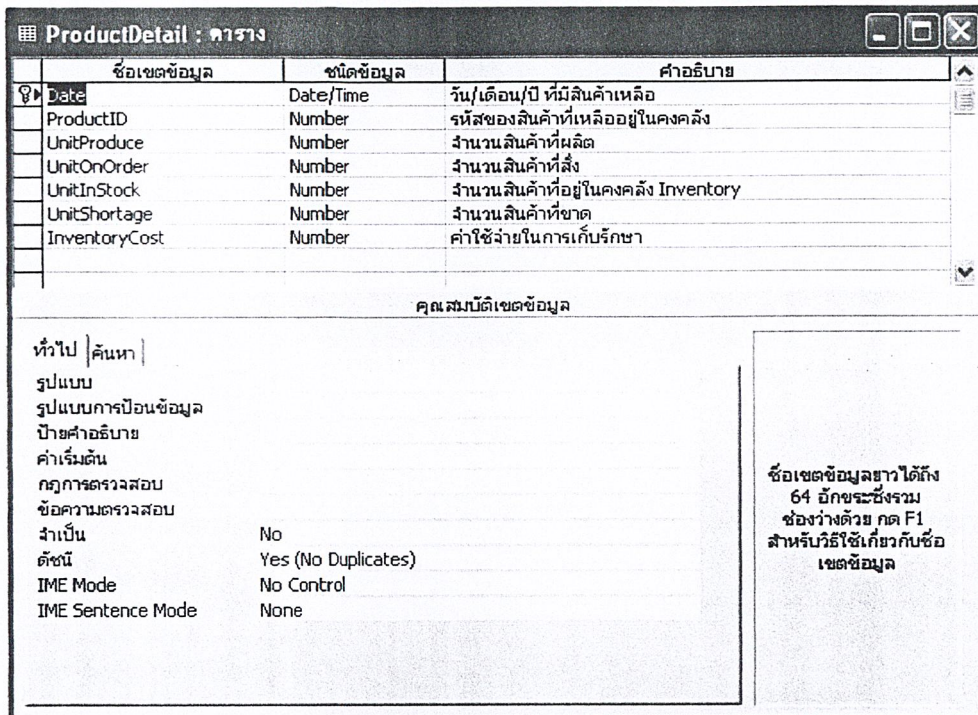
ProductID	ProductName	ManuID	Capacity	Quantity/Unit	InventoryCost	Price
1	Mainbord	1	7000	1	12	0
0		0	0	0	0	0

ฉบับที่: 1 จาก 1

ภาพที่ 3.15 แสดงข้อมูลของตาราง ManuProduct ในมุมมองแผ่นข้อมูล

- ProductID เป็น Field สำหรับลำดับของสินค้าหรือผลิตภัณฑ์ที่ Manufacturer เป็นผู้ผลิต
- ProductName เป็น Field สำหรับชื่อของสินค้าหรือผลิตภัณฑ์
- ManuID เป็น Field สำหรับรหัสของ ที่เป็น Manufacturer ผู้ผลิตสินค้าหรือ ผลิตภัณฑ์นี้
- Capacity เป็น Field สำหรับกำลังการผลิตของสินค้าหรือผลิตภัณฑ์นี้ ที่ Manufacturer สามารถจะผลิตได้
- Quantity/Unit เป็น Field ที่แสดงจำนวนของผลิตภัณฑ์ใน 1 หน่วย
- InventoryCost เป็น Field สำหรับค่าใช้จ่ายในการเก็บบำรุงรักษาสินค้า
- Price เป็น Field สำหรับราคาของสินค้าหรือผลิตภัณฑ์นี้

3.2.4.6 ตาราง ProductDetail เป็นตารางเกี่ยวกับรายละเอียดของสินค้าแต่ละชนิด ซึ่งมี Field ต่างๆ ดังนี้



ภาพที่ 3.16 แสดงข้อมูลของตาราง ProductDetail ในมุมมองออกแบบ

Date	ProductID	UnitProduce	UnitOnOrder	UnitInStock	UnitShortage	InventoryCost
1/1/2546	1		0	0		
1/2/2546	1		0	0		
1/3/2546	1		0	0		
1/4/2546	1		0	0		
1/5/2546	1		0	0		
1/6/2546	1		0	0		

ภาพที่ 3.17 แสดงข้อมูลของตาราง ProductDetail ในมุมมองแผ่นข้อมูล

- Date เป็น Field สำหรับ วัน/เดือน/ปี ที่ผลิตสินค้า
- ProductID เป็น Field สำหรับรหัสของสินค้า
- UnitProduce เป็น Field สำหรับปริมาณที่ควรผลิตสินค้าใน 1 เดือน
- UnitOnOrder เป็น Field ที่แสดงจำนวนของสินค้าหรือผลิตภัณฑ์ที่ได้รับการตั้งโดย Retailer
- UnitInStock เป็น Field ที่แสดงจำนวนของสินค้าหรือผลิตภัณฑ์ที่ยังเหลืออยู่ในคลัง
- InventoryCost เป็น Field สำหรับค่าใช้จ่ายในการเก็บบำรุงรักษาสินค้า



ManuID	เป็น Field ที่แสดงรหัสของ Manufacturer ที่สั่งซื้อสินค้า
UnitOnOrder	เป็น Field ที่แสดงปริมาณการสั่งซื้อสินค้าหรือผลิตภัณฑ์
UnitRecived	เป็น Field ที่แสดงปริมาณสินค้าที่ได้รับ
UnitInStock	เป็น Field ที่แสดงปริมาณสินค้าที่ยังเหลือคงคลัง
TotalPrice	เป็น Field ที่แสดงราคาของสินค้าหรือผลิตภัณฑ์ทั้งหมด

3.2.4.8 ตาราง RetailerOrder ซึ่งเป็นตารางที่แสดงข้อมูลและรายละเอียดเกี่ยวกับการสั่งซื้อสินค้าหรือผลิตภัณฑ์ ที่ Retailer เป็นผู้สั่งซื้อจาก Manufacturer ซึ่งมี Field ต่างๆ ดังนี้

ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
OrderID	Number	ลำดับของการสั่งซื้อสินค้า
DateOrder	Date/Time	วัน/เดือน/ปี ที่สั่งซื้อสินค้า
DateReceive	Date/Time	วัน/เดือน/ปี ที่ได้รับสินค้า
RetailerID	Number	รหัสของผู้ค้ารายย่อย
ProductID	Number	รหัสของผลิตภัณฑ์
UnitOnOrder	Number	จำนวนที่สั่งซื้อสินค้า
UnitRecived	Number	จำนวนสินค้าที่ได้รับ
UnitShortage	Number	จำนวนสินค้าที่ขาด
UnitInStock	Number	จำนวนสินค้าที่ยังเหลืออยู่ในคงคลัง (Inventory)
ShortageCost	Number	ราคาค่าปรับที่ได้เมื่อ Manufacturer ส่งของไม่ทันตามกำหนด
TotalPrice	Number	ราคารวมของสินค้าทั้งหมด

คุณสมบัติเขตข้อมูล

ทั่วไป | ค้นหา |

ขนาดเขตข้อมูล: Long Integer

รูปแบบ: Auto

จุดทศนิยม: 0

รูปแบบการป้อนข้อมูล: No

ป้ายคำอธิบาย: Yes (No Duplicates)

ค่าเริ่มต้น: 0

กฎการตรวจสอบ: No

ข้อความตรวจสอบ: Yes (No Duplicates)

ค่าเป็น: No

ดัชนี: Yes (No Duplicates)

ชื่อเขตข้อมูลยาวได้ถึง 64 อักขระซึ่งรวมช่องว่างด้วย กด F1 สำหรับวิธีใช้เกี่ยวกับชื่อเขตข้อมูล

ภาพที่ 3.20 แสดงข้อมูลของตาราง RetailerOrder ในมุมมองออกแบบ

OrderID	DateOrder	DateReceive	RetailerID	ProductID	UnitOnOrder	UnitRecived	UnitShortage	UnitInStock	ShortageCost	TotalPrice
1	1/1/2546	15/1/2546	1	1	756	0	0	0	0	0
2	1/2/2546	15/2/2546	1	1	1236	0	0	0	0	0
3	1/3/2546	15/3/2546	1	1	365	0	0	0	0	0
4	1/4/2546	15/4/2546	1	1	891	0	0	0	0	0
5	1/5/2546	15/5/2546	1	1	1925	0	0	0	0	0
6	1/6/2546	15/6/2546	1	1	2368	0	0	0	0	0
7	1/1/2546	15/1/2546	2	1	1756	0	0	0	0	0
8	1/2/2546	15/2/2546	2	1	936	0	0	0	0	0
9	1/3/2546	15/3/2546	2	1	1395	0	0	0	0	0
10	1/4/2546	15/4/2546	2	1	1990	0	0	0	0	0
11	1/5/2546	15/5/2546	2	1	1825	0	0	0	0	0
12	1/6/2546	15/6/2546	2	1	968	0	0	0	0	0

ภาพที่ 3.21 แสดงข้อมูลของตาราง RetailerOrder ในมุมมองแผ่นข้อมูล

OrderID	เป็น Field ที่แสดงลำดับการสั่งซื้อสินค้าหรือผลิตภัณฑ์
DateOrder	เป็น Field ที่แสดง วัน/เดือน/ปี ที่สั่งซื้อสินค้าหรือผลิตภัณฑ์
DateReceive	เป็น Field ที่แสดง วัน/เดือน/ปี ที่ได้รับสินค้าหรือผลิตภัณฑ์
ProductID	เป็น Field ที่แสดงรหัสของสินค้าหรือผลิตภัณฑ์ที่สั่งซื้อ
RetailerID	เป็น Field ที่แสดงรหัสของ Retailer ที่สั่งซื้อสินค้า
UnitOnOrder	เป็น Field ที่แสดงปริมาณการสั่งซื้อสินค้าหรือผลิตภัณฑ์
UnitRecived	เป็น Field ที่แสดงปริมาณสินค้าที่ได้รับ
UnitInStock	เป็น Field ที่แสดงปริมาณของสินค้าหรือผลิตภัณฑ์ที่อยู่ใน คงคลัง
ShortageCost	เป็น Field ที่แสดงปริมาณค่าปรับทั้งหมดที่ได้รับ
TotalPrice	เป็น Field ที่แสดงราคาของสินค้าหรือผลิตภัณฑ์ทั้งหมด

3.2.4.9 ตาราง RetailerSale ซึ่งเป็นตารางที่แสดงข้อมูลและรายละเอียดเกี่ยวกับการขายสินค้าหรือผลิตภัณฑ์ ที่ Retailer ซึ่ง Field ต่าง ๆ ดังนี้

ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
SaleID	Number	ลำดับที่ของการขายสินค้า
Date	Date/Time	วัน/เดือน/ปี ที่ขายสินค้า
ProductID	Number	รหัสของผลิตภัณฑ์
RetailerID	Number	รหัสของผู้ค้ารายย่อย
UnitSale	Number	จำนวนสินค้าที่ขาย
UnitInStock	Number	จำนวนสินค้าที่เหลือในคงคลัง inv
Price	Number	ราคาขายสินค้าไป
TotalPrice	Number	ราคารวมของสินค้าทั้งหมด

คุณสมบัติเขตข้อมูล

หัวไป | ค้นหา |

รูปแบบ

รูปแบบการป้อนข้อมูล

ป้ายคำอธิบาย

ค่าเริ่มต้น

กฎการตรวจสอบ

ข้อความตรวจสอบ

จำเป็น No

ดัชนี No

IME Mode No Control

IME Sentence Mode None

ชื่อเขตข้อมูลยาวได้ถึง 64 อักขระซึ่งรวมช่องว่างด้วย กด F1 สำหรับวิธีใช้เกี่ยวกับชื่อเขตข้อมูล

ภาพที่ 3.22 แสดงข้อมูลของตาราง RetailerSale ในมุมมองออกแบบ

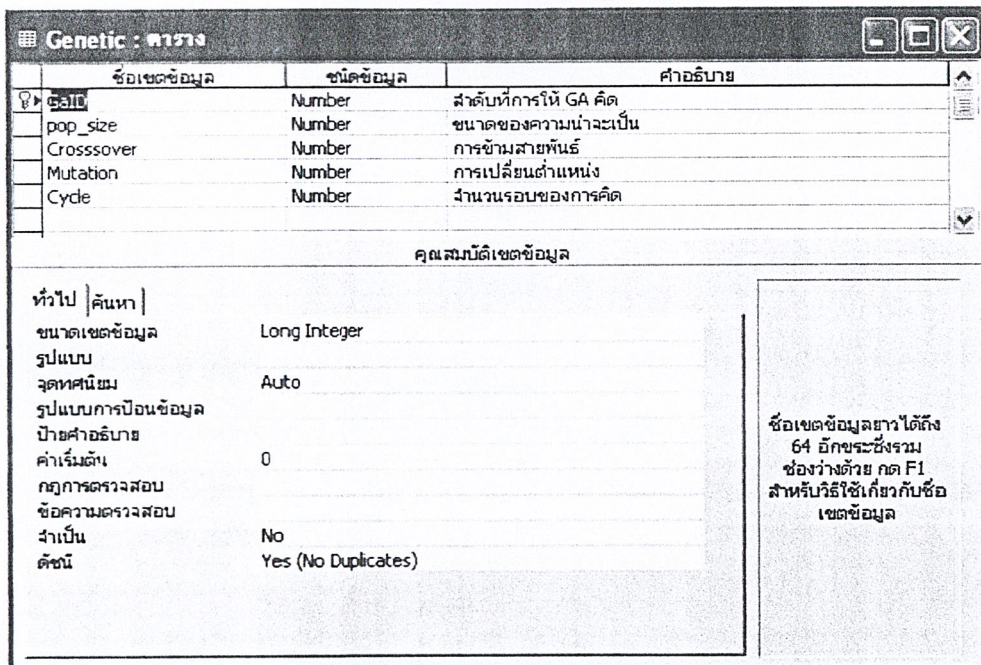
SaleID	Date	ProductID	RetailerID	UnitSale	UnitInStock	Price	TotalPrice
1	15/1/2546	1	1	0	0	0	0

รวมบันทึก: 1 จาก 1

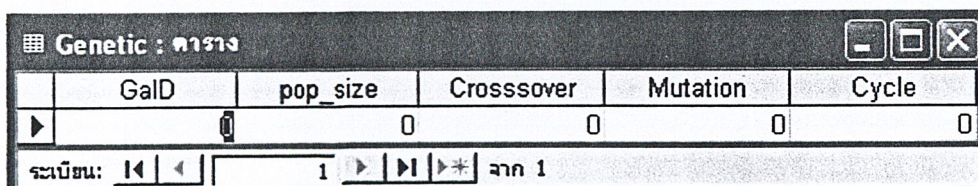
ภาพที่ 3.23 แสดงข้อมูลของตาราง RetailerSale ในมุมมองแผ่นข้อมูล

SaleID	เป็น Field ที่แสดงลำดับการขายสินค้าหรือผลิตภัณฑ์
Date	เป็น Field ที่แสดง วัน/เดือน/ปี ที่ขายสินค้าหรือผลิตภัณฑ์
ProductID	เป็น Field ที่แสดงรหัสของสินค้าหรือผลิตภัณฑ์ที่ได้ขายไป
RetailerID	เป็น Field ที่แสดงรหัสของ Retailer ที่ขายสินค้าหรือผลิตภัณฑ์
UnitSale	เป็น Field ที่แสดงปริมาณของสินค้าหรือผลิตภัณฑ์ที่ขาย
UnitInStock	เป็น Field ที่แสดงปริมาณของสินค้าที่ยังเหลืออยู่ในคลัง
Price	เป็น Field ที่แสดงราคาของสินค้าที่ขายไป
TotalPrice	เป็น Field ที่แสดงราคาของสินค้าที่ขายไปทั้งหมด

#### 3.2.4.10 ตาราง Genetic เป็นตารางที่มีข้อมูลเกี่ยวกับ Genetic



ภาพที่ 3.24 แสดงข้อมูลของตาราง Genetic ในมุมมองออกแบบ



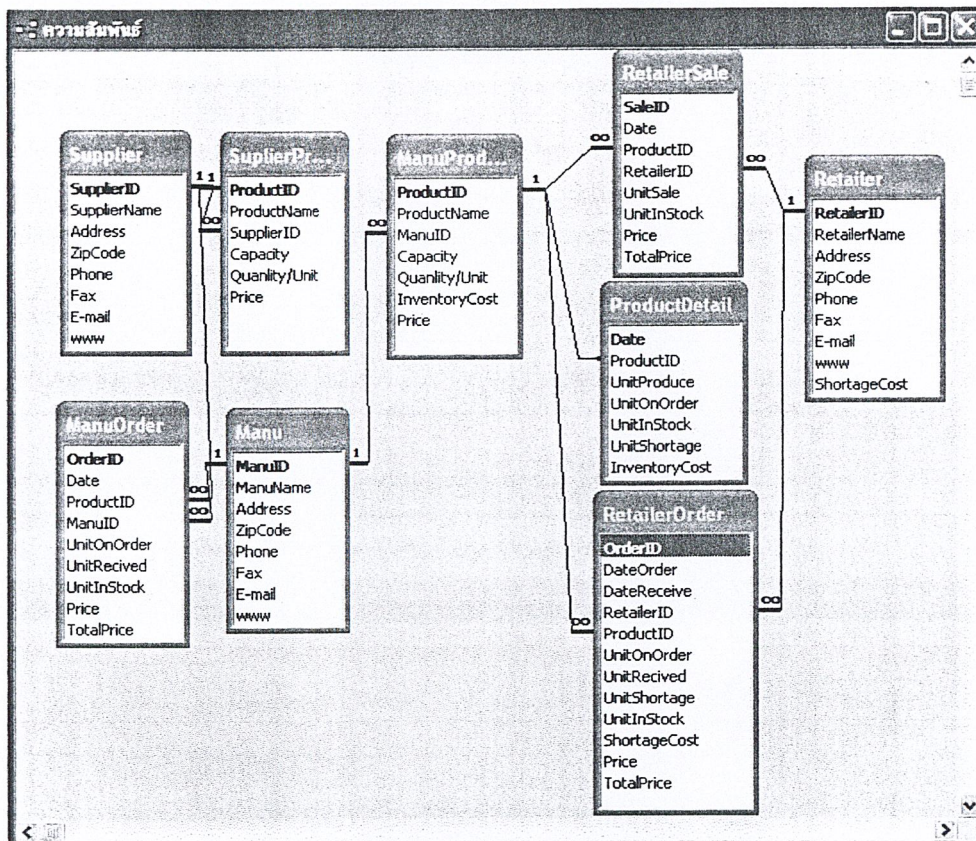
ภาพที่ 3.25 แสดงข้อมูลของตาราง Genetic ในมุมมองแผ่นข้อมูล

GaID	เป็น Field สำหรับลำดับที่ของการคำนวณ Genetic
pop size	เป็น Field สำหรับขนาดความน่าจะเป็น
Crossover	เป็น Field สำหรับการข้ามสายพันธ์

Mutation เป็น Field สำหรับการเปลี่ยนตำแหน่ง  
 Cycle เป็น Field สำหรับจำนวนรอบของการคำนวณ

### 3.2.5 การกำหนดความสัมพันธ์ของตาราง

กำหนดความสัมพันธ์ของตาราง ซึ่งใน Project จะประกอบไปด้วยตารางดังต่อไปนี้ ตาราง Supplier, Manu, ManuOrder, ManuProduct, ProductDetail, SupplierProduct, Retailer, RetailerOrder, RetailerSale และ Genetic ซึ่งแต่ละตารางจะมีความสัมพันธ์กันดังรูป



ภาพที่ 3.26 แสดงความสัมพันธ์ของตารางแต่ละตาราง

ซึ่งความสัมพันธ์ของตารางสามารถที่จะอธิบายได้นี้ จากข้อมูลในตาราง Supplier ซึ่งมี Field SupplierID เป็น Primary Key ซึ่งรหัสของ Supplier แต่ละรายไม่สามารถที่จะมีชื่อที่ซ้ำกันได้ ตารางของ Supplier จะเชื่อมความสัมพันธ์กับตาราง SupplierProduct ซึ่งเป็นตารางที่มีข้อมูลเกี่ยวกับวัตถุดิบที่ Supplier เป็นผู้ผลิต ทั้งสองตารางนี้มี Field ที่เชื่อมความสัมพันธ์ คือ SupplierID ซึ่งเป็นความสัมพันธ์แบบ One-to-Many คือ Supplier หนึ่งรายสามารถที่จะมีวัตถุดิบได้หลายชนิด

จากตาราง Manu ซึ่งเป็นตารางที่มีข้อมูลและรายละเอียดเกี่ยวกับ Manufacturer ซึ่งมี Field ManuID เป็น Primary Key และเชื่อมความสัมพันธ์กับตาราง ManuProduct ด้วย ManuID ซึ่ง

Manufacturer เป็นผู้ผลิตสินค้าหรือผลิตภัณฑ์นี้ และมีความสัมพันธ์กับตาราง ManuOrder ซึ่งเชื่อมความสัมพันธ์ด้วย ManuID ซึ่งตารางนี้เป็นตารางเกี่ยวกับการสั่งซื้อวัตถุดิบที่ Manufacturer เป็นผู้สั่งซื้อจาก Supplier และตาราง ManuOrder มีความสัมพันธ์กับตาราง SupplierProduct ซึ่งเป็นตารางที่เกี่ยวข้องวัตถุดิบที่ Supplier เป็นผู้ผลิต เชื่อมความสัมพันธ์กันด้วย ProductID

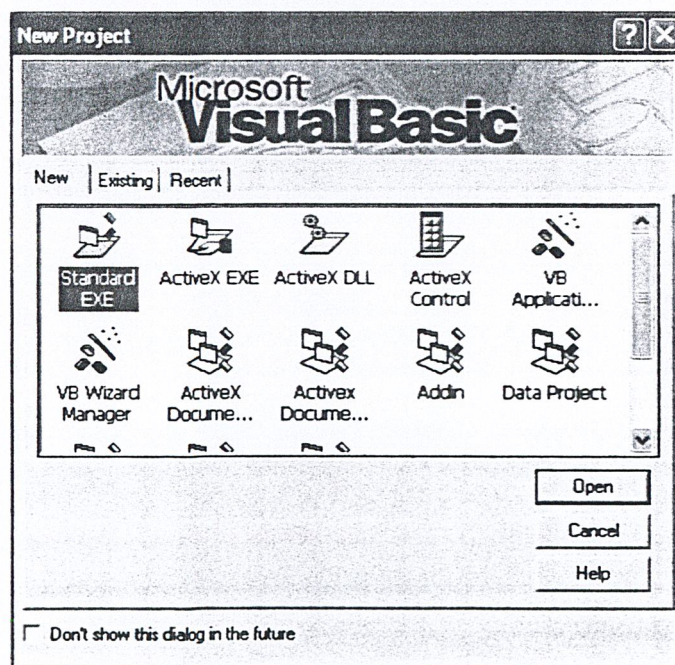
จากตาราง Retailer ซึ่งเป็นตารางที่มีข้อมูลเกี่ยวกับรายละเอียดของผู้ค้ารายย่อยแต่ละราย ซึ่งมีความสัมพันธ์กับตาราง RetailerOrder และ RetailerSale และใช้ Field RetailerID เชื่อมความสัมพันธ์ระหว่างตารางทั้ง 3 และในตาราง Retailer RetailerID จะเป็น Primary key คีย์คือ รหัสของ Retailer แต่ละรายไม่สามารถที่จะซ้ำกันได้ ส่วนตาราง RetailerOrder และ RetailerSale มีความสัมพันธ์กับตาราง ManuProduct ด้วย ProductID ที่ผู้ค้ารายย่อยแต่ละรายจะสั่งซื้อหรือขายสินค้าอะไรไปบ้าง และจากตาราง ManuProduct จะมีความสัมพันธ์กับตาราง ProductDetail แบบ หนึ่ง-ต่อ-หนึ่ง ด้วย ProductID

### 3.3 การเขียนโปรแกรมด้วย Visual Basic

#### 3.3.1 การเขียนโปรแกรมด้วย Visual Basic ขั้นพื้นฐาน

สิ่งแรกที่จะพบเมื่อเข้าสู่โปรแกรม Microsoft Visual Basic 6.0 ได้แก่ จอภาพที่ใช้สำหรับ Project จะเป็นชื่อที่ใช้เรียกแทนระบบงานที่พัฒนาขึ้นด้วยการเขียนโปรแกรม Visual Basic ประกอบด้วย 3 Tab ดังนี้

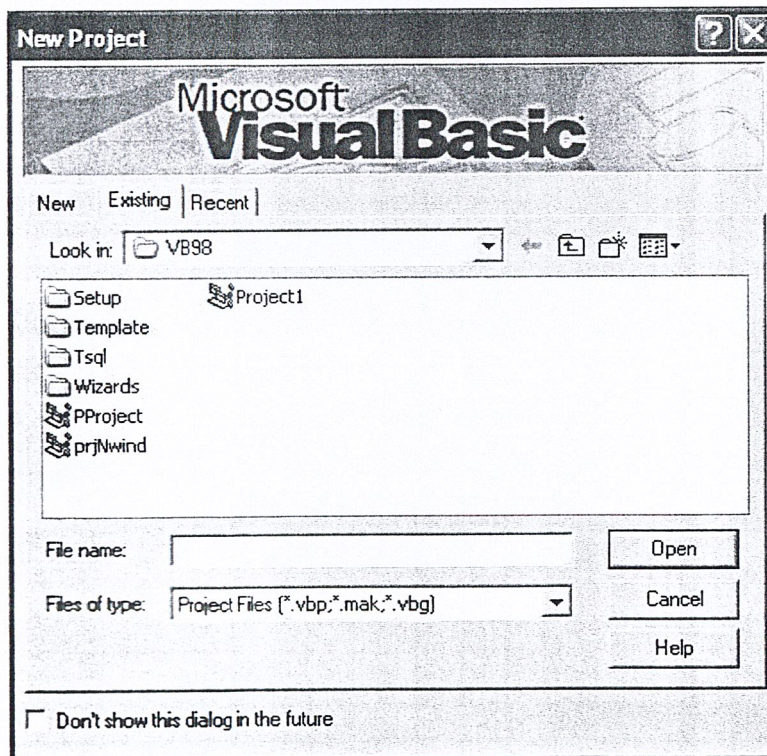
1. Tab "New" เป็นจอภาพที่ประกอบด้วย Icon ต่างๆ ที่ใช้สำหรับเรียกใช้ Project ใหม่



ภาพที่ 3.27 แสดงการเรียก Project ใหม่ขึ้นมาใช้งาน

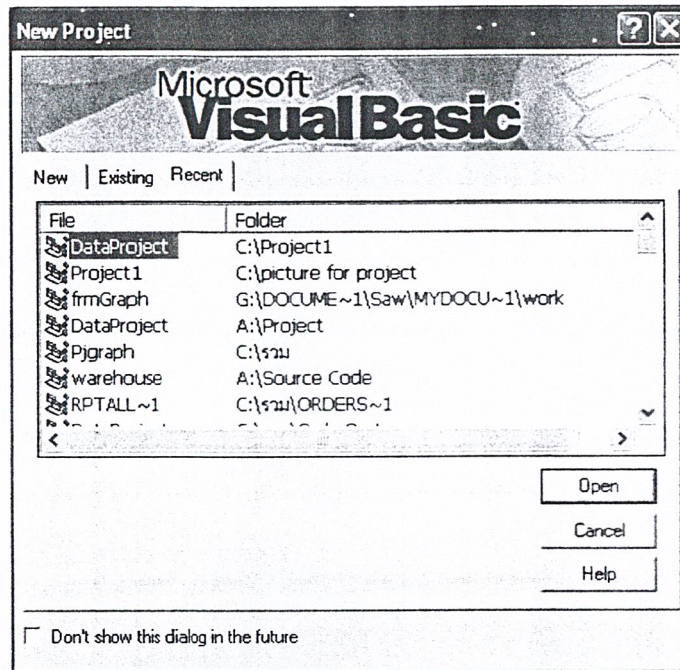
Standard.EXE	ใช้สร้างโปรแกรมทั่วไปในรูปแบบ GUI
ActiveX.EXE	สร้างโปรแกรมติดต่อกับโปรแกรมอื่นในรูปแบบ OLE
ActiveX.DLL	เหมือนกับ ActiveX.EXE แต่จะเป็นนามสกุล DLL (ไม่สามารถรันด้วยตัวเองได้)
ActiveXControl	ใช้สร้างคอนโทรลขึ้นเองในโปรแกรม
VB Application Wizard	สร้างโปรแกรมตามโครงสร้างของ Database
Addin	ใช้เพิ่ม Utility ใน Visual Basic
ActiveX Document DLL	ใช้รันบน Internet Explorer แต่มีนามสกุลเป็น DLL
ActiveX Document EXE	ใช้รันบน Internet Explorer แต่มีนามสกุลเป็น EXE

2. Tab “Existing” เป็นจอภาพสำหรับเรียกใช้ Project เดิมที่พัฒนาขึ้นแล้วมาเก็บเอาไว้ใน Directory ต่างๆ ขึ้นมาใช้งาน



ภาพที่ 3.28 แสดงการเรียกใช้ Project เดิมมาเก็บไว้ใน Directory

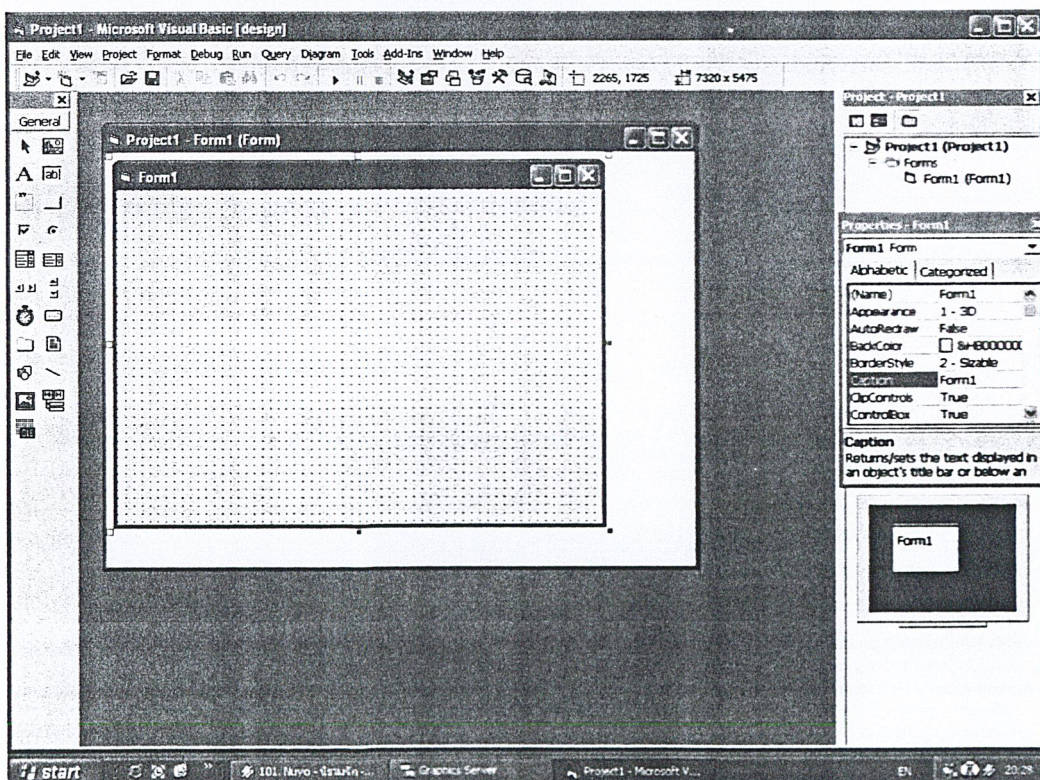
3. Tab “Recent” เป็นจอภาพที่แสดง Project ล่าสุดที่เคยถูกเรียกขึ้นมาพัฒนา



ภาพที่ 3.29 แสดง Project ที่เคยถูกเรียกมาใช้

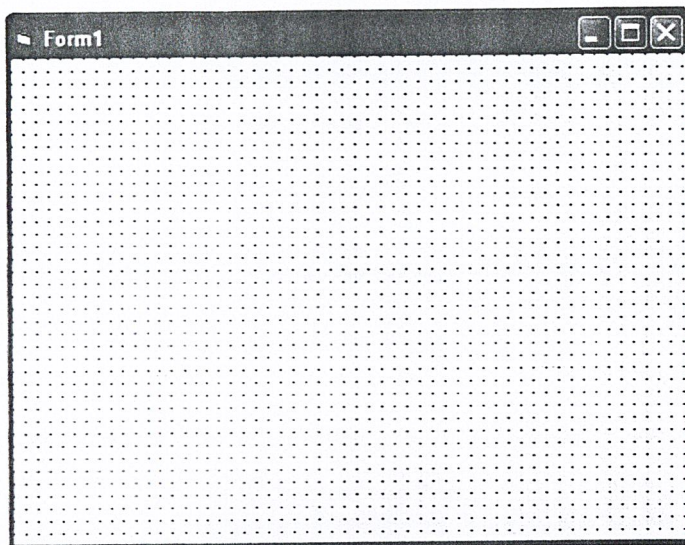
### 3.3.2 ส่วนประกอบของหน้าต่าง Visual Basic 6.1

ถ้าเริ่มต้นงานใหม่ให้เลือก New -> Standard.EXE-> Open ดังแสดงตามภาพที่ 3.30



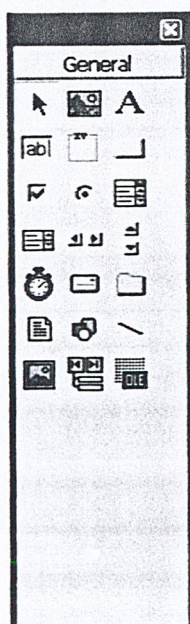
ภาพที่ 3.30 แสดงส่วนประกอบของหน้าต่างของ Visual Basic

Form เป็นพื้นที่สำหรับออกแบบ โดยนำคอนโทรลต่างๆ มาวางตามเหมาะสม ค่าต่างให้กำหนดในส่วนช่องProperties ทุกครั้งของการเปิด Project ใหม่จะได้ฟอร์มเปล่าดังภาพ



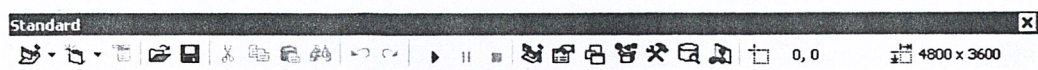
ภาพที่ 3.31 แสดงรูปฟอร์มเปล่า

Toolbox เป็นแถบเครื่องมือที่ประกอบไปด้วย Icon ต่างๆ ที่เรียกว่า “Control” ซึ่งจะใช้ร่วมกับ Form เพื่อสร้างจอภาพ Project แต่ละ Control จะใช้เป็นเครื่องมือที่ใช้สำหรับสร้างส่วนที่ใช้ติดต่อกับผู้ใช้ หรือที่เรียกว่า “User Interface” เช่น ข้อความต่างๆ ช่องว่างสำหรับรับข้อมูลจากคีย์บอร์ดปุ่มต่างๆ ฯลฯ เป็นต้น จะถูกนำไปใช้งานด้วย โดยการนำ Control ที่ต้องการไปวางลงบน Form Control แต่ละตัวจะมีชื่อที่แตกต่างกันไป เมื่อต้องการดูชื่อของ Control ใดก็เพียงแต่เลื่อน Mouse ไปชี้ที่ Control นั้นชื่อของ Control จะปรากฏขึ้นให้เห็น



ภาพที่ 3.32 แสดง ToolBox

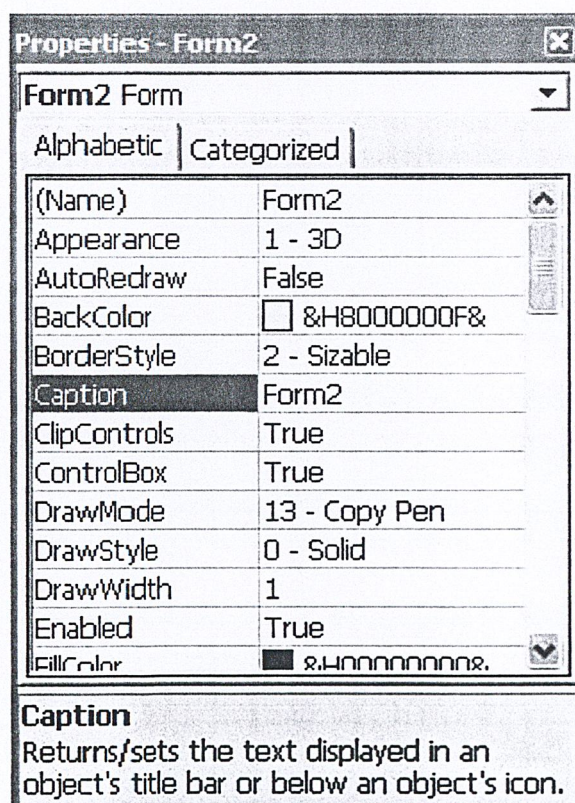
- Toolbar เป็นแถบเครื่องมือที่ประกอบด้วย Icon ต่าง ๆ ดังรูป ซึ่ง Toolbar ทำหน้าที่เป็นผู้ช่วยในการพัฒนาโปรแกรม ซึ่งเมื่อเลื่อน Mouse ไปชี้ยัง Icon ใด ก็จะมีปรากฏชื่ออยู่ใต้ Icon นั้นแต่ละ Icon จะมีหน้าที่ต่างกันไป



ภาพที่ 3.33 แสดง Tool Bar

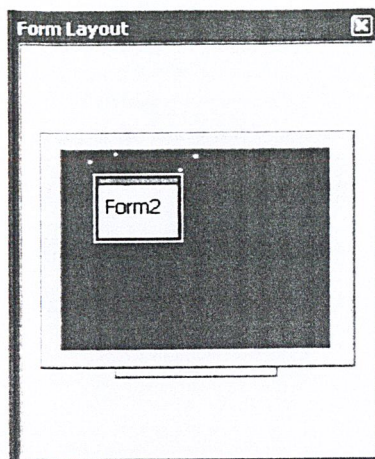
- Project Explorer Window เป็นส่วนสำหรับเรียก Form ต่างๆ ขึ้นมาแก้ไขในกรณีที่มี Project ประกอบด้วย Form มากกว่า 1 Form

- Properties Window เป็นจอภาพที่ใช้สำหรับกำหนดคุณสมบัติ (Property) ให้กับ Form และ Object ต่างๆ ที่ปรากฏอยู่บน Form ซึ่งจะอำนวยความสะดวกในการออกแบบฟอร์มโดยไม่ต้องเขียนโปรแกรมให้ยุ่งยาก



ภาพที่ 3.34 แสดง Properties Window

- Form Layout Window เป็นหน้าต่างที่ใช้กำหนดตำแหน่งของฟอร์มบนหน้าจอในขณะรันโปรแกรม โดยกรอบที่อยู่ภายใน ณ ตำแหน่งใด ฟอร์มที่แสดงในขณะรันโปรแกรมก็จะแสดงตำแหน่งตามนั้น ทำให้จัดตำแหน่งของ Form ได้สะดวกขึ้น



ภาพที่ 3.35 แสดง Form Layout Window

- Project โดยทั่วไปในงานหนึ่งๆ มักจะประกอบไปด้วยหลายๆ จอภาพ ดังนั้นการพัฒนาโปรแกรม จึงนิยมแยกแต่ละจอภาพออกเป็น โปรแกรมเพื่อสะดวกต่อการแก้ไขตามหลักของการเขียนโปรแกรม แล้วจึงนำแต่ละโปรแกรมย่อยมาประกอบกันขึ้นเป็นระบบโดยการ Compile ไฟล์เหล่านั้นรวมกันเป็น Executed Program (ไฟล์นามสกุล EXE) เพื่อนำไปใช้งาน

พรอพเพอร์ตี้ (Properties), วัตถุ (Object), และอีเวนต์ (Event)

ในภาษา Visual Basic นั้น วัตถุ (Object) จะหมายถึงเครื่องมือคอนโทรลต่างๆ และในแต่ละวัตถุจะมีลักษณะเฉพาะเช่น ความยาว ความกว้าง เรียกว่า พรอพเพอร์ตี้ (Properties) ในแต่ละวัตถุยังมีพฤติกรรมประจำตัวเรียกว่า เมธอด (Method) เช่น การถูกเมาส์ชี้ เป็นต้น ในการเขียนโปรแกรมนั้นจะมี อีเวนต์ (Event) ซึ่งในความหมายจะคล้ายกับเมธอด ดังนั้นอาจสรุปได้ว่า

: Properties คือ การกำหนดคุณลักษณะของคอนโทรลและฟอร์ม

: Method คือ สิ่งที่คอนโทรลและฟอร์มสามารถทำได้ เช่น สามารถวาดรูปได้

: Event หมายถึง ความสามารถตอบสนองของฟอร์มหรือคอนโทรลตัวนั้น เช่น กด Enter หรือ ESC บนแป้นพิมพ์

### 3.3.3 การจัดการกับโปรเจกต์และฟอร์ม

#### 3.3.3.1 การ Run และการเลิกงาน Project

ในการ Run Project ที่พัฒนาขึ้นด้วย Visual Basic สามารถ Run ได้ทั้งโดยการใช้ Interpreter และการใช้ Compiler กล่าวคือเราสามารถทดลอง Run สิ่งต่างๆ ที่เราจัดทำขึ้นไปพร้อมๆ กับการแก้ไขโปรแกรมจนกระทั่งเสร็จสมบูรณ์ แล้วจึง Compile โปรแกรมให้อยู่ในรูปของ Executed Program เพื่อนำไปใช้งานได้เช่นเดียวกัน แต่ในเบื้องต้นนี้ จะขอกล่าวถึงส่วนที่เป็น Interpreter เป็นลำดับแรก

ในการ Run Project ที่เราจัดทำขึ้นนั้นทำได้ 3 วิธี

- วิธีที่ 1 Run โดยกด F5
- วิธีที่ 2 คลิกที่ Icon “Run” ใน ToolBar
- วิธีที่ 3 เลือกจากเมนู Run และ Start ตามลำดับ

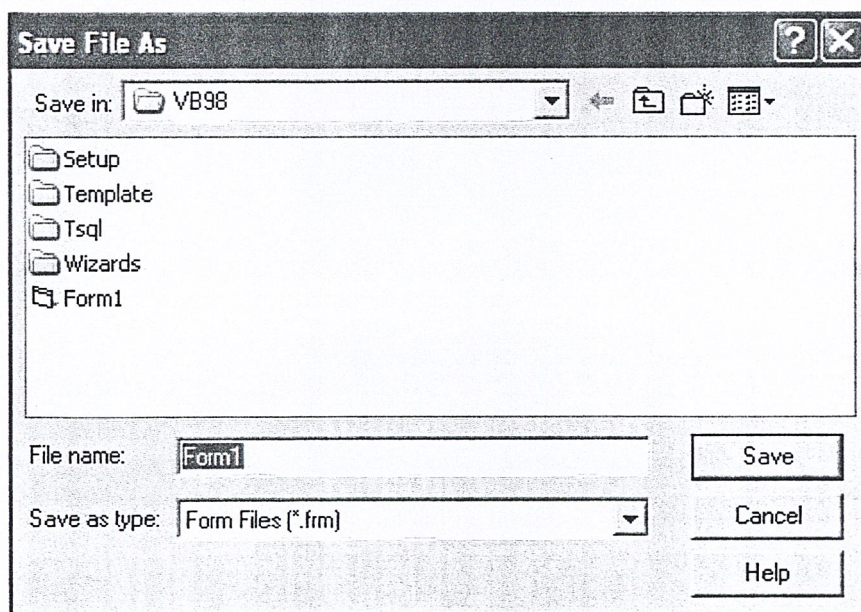
ในการเลิกงาน Project ทำได้ 2 วิธี

- วิธีที่ 1 คลิกที่ Icon Run ใน ToolBar
- วิธีที่ 2 เลือกจากเมนู Run และ End ตามลำดับ

### 3.3.3.2 การบันทึก Form และ Project

ในการบันทึก Project จะต้องบันทึกทั้งส่วนของ Form และ Project โดย Form จะถูกบันทึกลงในไฟล์นามสกุล FRM ในขณะที่ Project จะถูกบันทึกลงในไฟล์นามสกุล VBP ในการบันทึก Form ทำได้ดังนี้

1. คลิกที่ Icon “Save” ใน ToolBar จะปรากฏจอภาพดังภาพที่ 3.36



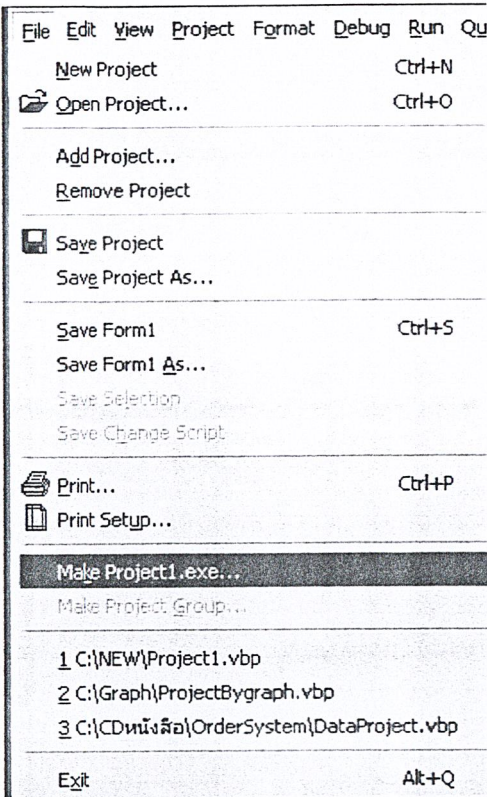
ภาพที่ 3.36 แสดงการ save ฟอর্ম

2. ใส่ชื่อไฟล์ที่จะใช้เก็บฟอर्म ซึ่ง Visual Basic จะใช้ชื่อเริ่มต้นเป็นชื่อเดียวกับฟอर्मเสมอ ดังนั้นจึงปรากฏชื่อ Form1 มาให้ผู้ใช้สามารถเปลี่ยนแปลงชื่อได้ตามต้องการ จากนั้นให้คลิกปุ่ม Save

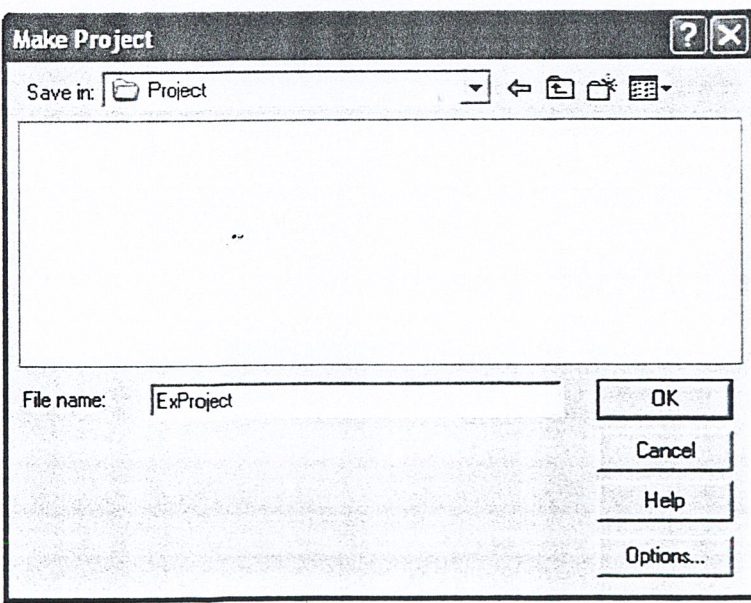
3. จะปรากฏจอภาพสำหรับบันทึก Project ซึ่งก็เช่นเดียวกับ Form เป็นชื่อ Project1 ให้เปลี่ยนชื่อตามต้องการแล้วคลิกปุ่ม Save

### 3.3.3.3 การแปลงไฟล์เป็น Executed file (.EXE)

หลังจากการรันโปรแกรมเสร็จสมบูรณ์แล้ว โปรแกรมดังกล่าวยังเป็นแค่โปรแกรมต้นฉบับ หรือเรียกว่า Source Code เมื่อต้องการไปใช้งานจึงต้องแปลงเป็น Executed file (.EXE) ซึ่งเป็นไฟล์ที่มีนามสกุลเป็น EXE โดยเลือกที่เมนู File -> Make ExProject.exe เสียก่อน



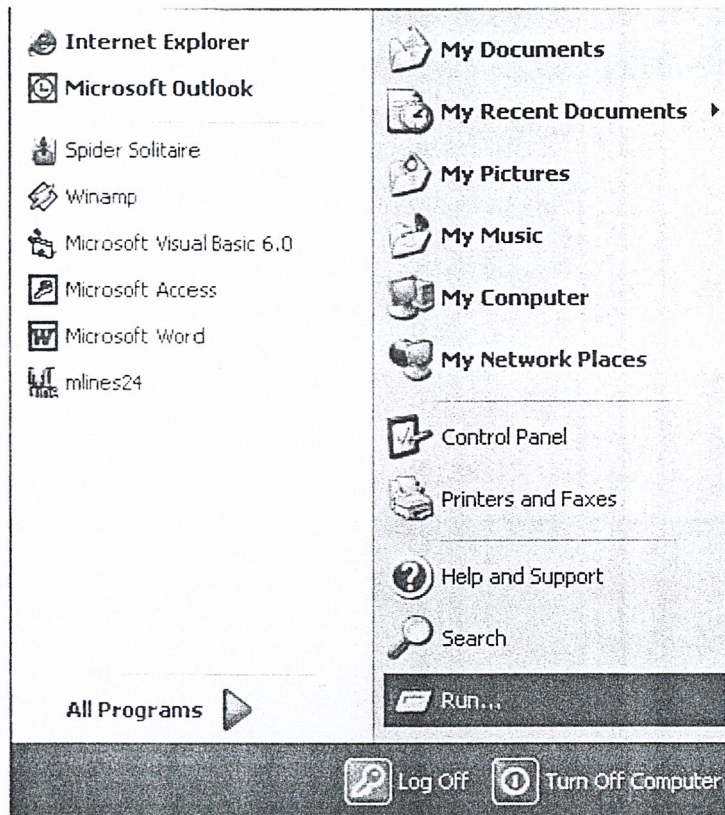
ภาพที่ 3.37 แสดงการเลือกเมนู File -> Make ExProject.exe



ภาพที่ 3.38 แสดงการตั้งชื่อไฟล์จะได้นามสกุลเป็นจุด EXE

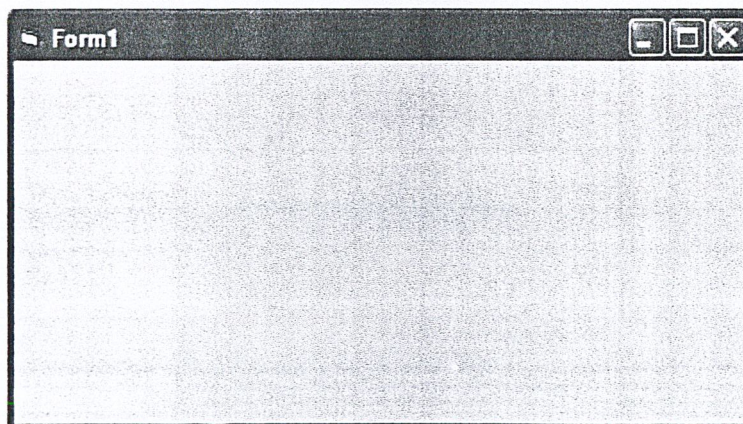
### 3.3.3.4 วิธีการเรียกโปรแกรมที่แปลงเป็น Executed file (.EXE)

1. เรียกเมนู Start -> Run -> Browse



ภาพที่ 3.39 แสดงการเรียกเมนู Start -> Run -> Browse

2. เลือกโปรแกรมที่ได้แปลงเป็น EXE จากตัวอย่างที่ผ่านมาแล้วคลิกเลือก ExProject.EXE เมื่อเรียกโปรแกรมจะปรากฏหน้าต่างที่ฟอร์ม DeskTop ซึ่งเป็นฟอร์มที่ถูกแปลงเป็น EXE แล้ว



ภาพที่ 3.40 แสดงรูปแบบ Form เมื่อเปลี่ยนเป็น .EXE

### 3.3.4 ขั้นตอนในการพัฒนาโปรแกรมของ Visual Basic

ประกอบด้วยขั้นตอนหลัก 2 ขั้นตอน

ขั้นตอนที่ 1 การสร้างจอภาพของโปรแกรม ในขั้นตอนนี้จะนำ Form มาออกแบบเพื่อใช้ในการติดต่อกับผู้ใช้ หรือที่เรียกว่าการออกแบบ “User Interface” ในการพัฒนาโปรแกรมแบบเดิม แต่ในส่วนนี้ Visual Basic สามารถทำได้อย่างง่ายคยจากการพัฒนาโปรแกรมแล้ว เพียงแต่นำเอาคอนโทรลต่างๆ ใน ToolBox ที่ต้องการใช้งานมาสร้างไว้บนฟอร์ม

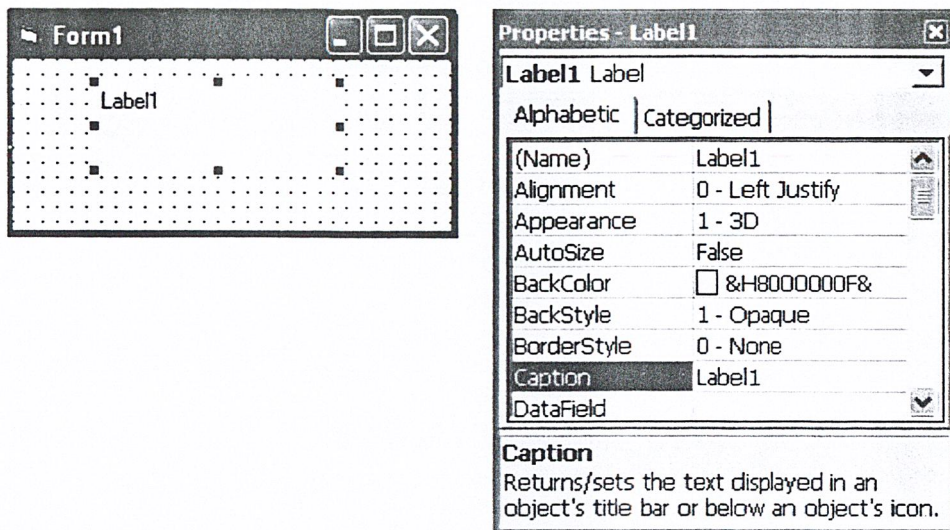
ขั้นตอนที่ 2 การเขียนโปรแกรม เมื่อวางคอนโทรลต่างๆ ลงบน Form เป็นที่เรียบร้อยแล้ว (คอนโทรลต่างๆ เมื่อถูกนำมาวางไว้บนฟอร์มจะเรียกว่า “Object” ขั้นตอนที่ตามมาได้แก่ การเขียนโปรแกรมเพื่อกำหนดการทำงานให้แก่ Object ภายในเหตุการณ์ต่างๆ(Event) ที่จะเกิดขึ้นกับจอภาพนั้นๆ

#### 3.3.4.1 Event-Driven Program กับ Visual Basic

แนวความคิดในการเขียนโปรแกรมแบบ Event-Driven จะเปลี่ยนมาสนใจกับเหตุการณ์ (Event) ที่จะเกิดขึ้นในโปรแกรมมากกว่าการกำหนดขั้นตอนการทำงานของโปรแกรมในแบบเดิม เช่น ถ้ามีการ “เลื่อนเมาส์” เกิดขึ้น จะให้โปรแกรมทำอะไร หรือมีการกดปุ่มที่ 1 ขึ้น จะให้โปรแกรมทำอะไร เป็นต้น อย่างไรก็ตาม ก็ยังต้องอาศัยแนวความคิดในการเขียนโปรแกรมแบบเดิมอยู่บ้าง เนื่องจากการกำหนดการทำงานให้แก่แต่ละ Event ยังต้องกำหนดอย่างเป็นขั้นตอนอยู่ดี การเขียนโปรแกรมแบบ Event-Driven ใน Visual Basic จะเป็นการเขียนโปรแกรมให้กับ Object ต่างๆ ที่ปรากฏอยู่บนฟอร์ม โดยพิจารณาว่าแต่ละ Object จะมี Event อะไรเกิดขึ้นบ้าง แล้วจึงเลือกเขียนโปรแกรมเฉพาะ Event นั้น

#### 3.3.4.2 การใช้งานปุ่มคอนโทรลต่างๆ

เลเบล (Label) เป็นคอนโทรลที่แสดงข้อความบนฟอร์ม โดยคลิกที่ปุ่มเลเบลแล้วนำเมาส์ไปสร้างขึ้นบนฟอร์ม กำหนด Properties ได้ดังนี้



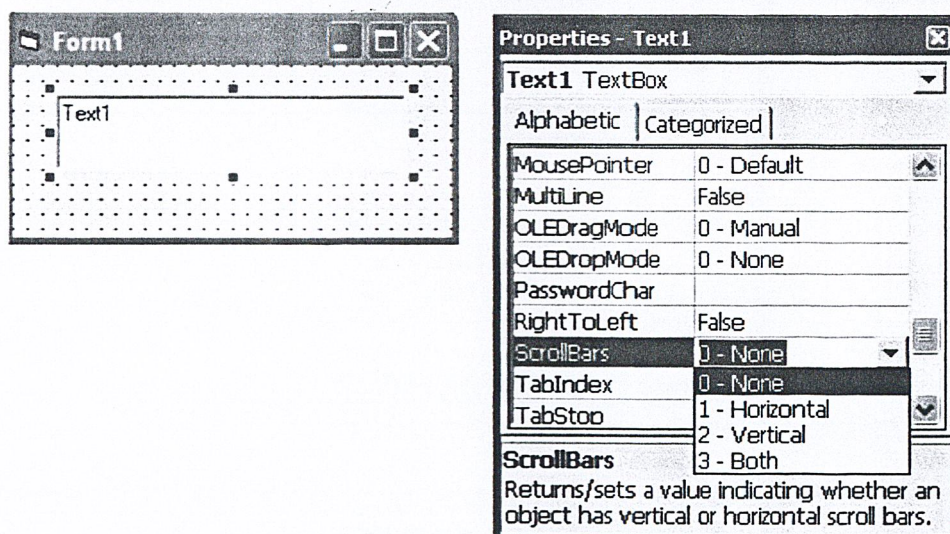
ภาพที่ 3.41 แสดงเลเบล และ Properties

Name : กำหนดชื่อของเลเบล ซึ่งใช้สำหรับเขียนโปรแกรม (Coding)

Caption: เปลี่ยนข้อความของเลเบล

Autosize: กำหนดให้ขอบเขตของเลเบลพอดีกับข้อความในเลเบล

แท็บซ์บ็อกซ์ (TextBox) เป็นช่องรับข้อมูลทางแป้นพิมพ์หรือรับข้อมูลจากคอนโทรลตัวอื่น โดยปกติช่อง TextBox จะรับข้อมูลและแสดงข้อมูลเพียงแถวเดียว แต่ถ้าข้อมูลมากกว่า 1 ช่องจะสามารถรับได้ กำหนด Properties ได้ดังนี้

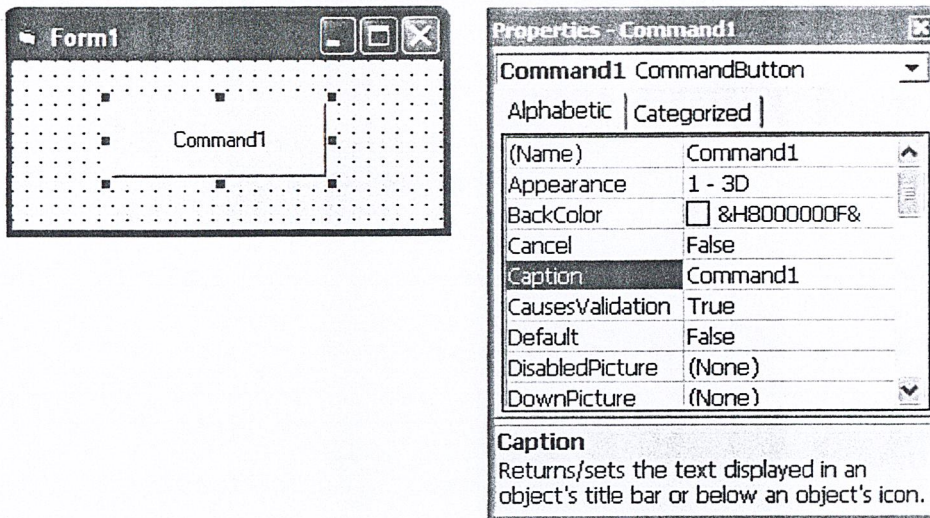


ภาพที่ 3.42 แสดง TextBox และ Properties

MultiLine : กำหนดให้ TextBox ใช้งานได้หลายบรรทัด

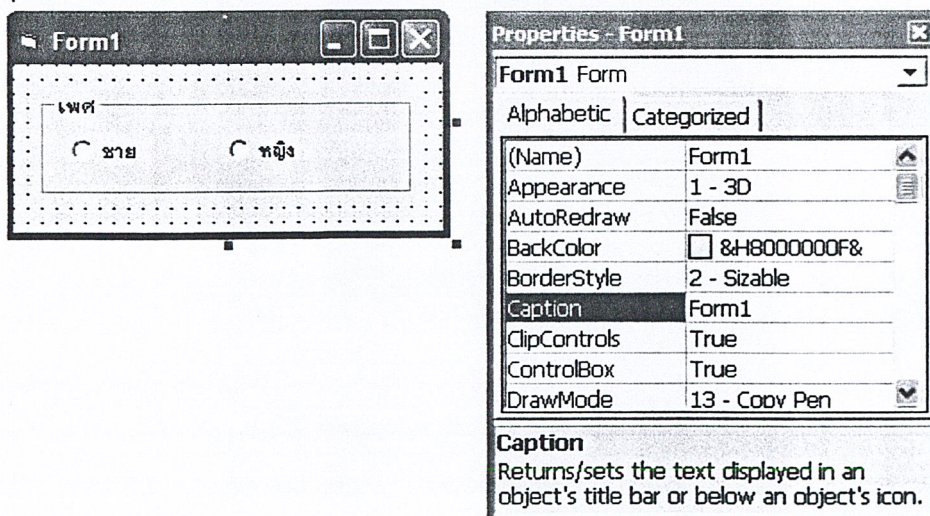
ScrollBars : สามารถใช้แถบเลื่อนช่วยในการดูข้อมูลในกรณีที่ไม่สามารถดูข้อมูลได้หมด โดยต้องกำหนดเป็น MultiLine ไว้แล้ว มีค่าให้เลือกคือ 0-ไม่มี, 1-แถบเลื่อนในแนวนอน, 2-แถบเลื่อนในแนวตั้ง 3-แถบเลื่อนทั้งแนวนอนและแนวตั้ง

ปุ่มคอมมานด์ (Command Button) เป็นปุ่มคอนโทรลที่มีไว้ตอบสนองการทำงานด้วยการคลิกเมาส์ที่ปุ่ม หรือกด Enter แทนการคลิก แต่จะกด Enter ต้องกำหนด Focus เสียก่อน ซึ่งจะกำหนดได้ในการเขียนโปรแกรม



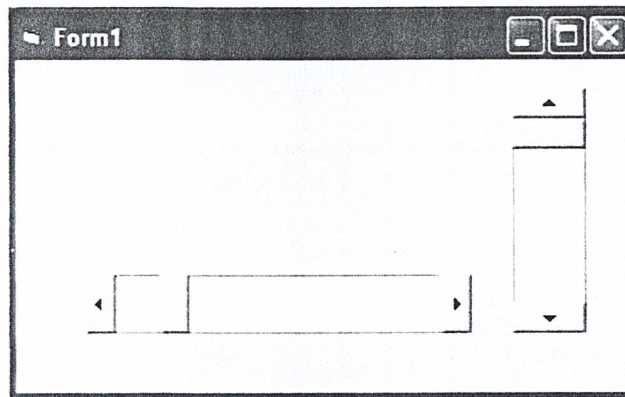
ภาพที่ 3.43 แสดงปุ่ม command และ Properties

เฟรม (Frame) เป็นกรอบของตัวเลือก มักใช้สำหรับการแบ่งกลุ่ม เช่น กลุ่มเพศ กลุ่มอายุ เป็นต้น



ภาพที่ 3.44 แสดงเฟรม และ Properties

แถบเลื่อนแนวนอน (HScrollBar), แถบเลื่อนแนวตั้ง (VScrollBar) แสดงแถบเลื่อนในแนวนอนและแนวตั้ง โดยทั่วไปมักใช้เพิ่มหรือลดค่าเท่านั้น เช่น การปรับสีเพื่อลดความสว่างของแสง การปรับความดังของเสียง เป็นต้น แถบเลื่อนในแนวตั้งและแนวนอนจะเป็นอิสระไม่ขึ้นกับคอนโทรลตัวอื่น



ภาพที่ 3. 45 แสดงแถบเลื่อนในแนวนอนและแนวตั้ง

อิมเมจ (Image), พิกเจอร์บ็อกซ์ (PictureBox) ใช้ในการแสดงรูปภาพบนฟอร์ม โดย Image จะสามารถปรับขนาดได้ตามเหมาะสม แต่จะไม่สามารถใช้ร่วมกับคอนโทรลวาดรูปได้ ส่วน PictureBox จะสามารถแสดงรูปภาพเคลื่อนไหว เช่น ภาพ \*.AVI โดยกำหนด Path และชื่อไฟล์รูปภาพ และยังสามารถแสดงรูปภาพในแบบ Bmp, Dib, Jpeg, Gif, Ico, Cur, Wmf, ได้ อีกทั้งยังสามารถแปลงไปในส่วนของ PictureBox บนฟอร์มได้

คอนโทรล ImageList คอนโทรล ImageList มีหน้าที่สำหรับเก็บรูปภาพ แล้วเรียงเรียงรูปภาพดังกล่าวแบบอาร์เรย์ เมื่อคุณต้องการใช้รูปภาพใด ก็จะใช้การอ้างอิงลำดับของรูปภาพไว้นั่นเอง โดยที่รูปภาพแรกมีลำดับเท่ากับ 1 ให้คลิกขวาที่คอนโทรล ImageList แล้วเลือกคำสั่ง Properties

#### แท็บ General

ตัวเลือก 16x16 หมายถึง ต้องการเก็บรูปภาพที่มีขนาด 16x16 พิกเซล

ตัวเลือก 32x32 หมายถึง ต้องการเก็บรูปภาพที่มีขนาด 32x32 พิกเซล

ตัวเลือก 48x48 หมายถึง ต้องการเก็บรูปภาพที่มีขนาด 48x48 พิกเซล

ตัวเลือก Custom หมายถึง ต้องการเก็บรูปภาพที่มีขนาดตามที่กำหนดไว้ในช่อง Height: และ Width:

#### แท็บ Images

ให้คลิกปุ่ม Insert Picture เพื่อเลือกรูปภาพที่คุณต้องการเก็บไว้ในคอนโทรล ImageList

ช่อง Index หมายถึง ลำดับของรูปภาพ

- ช่อง Key หมายถึง การตั้งชื่อให้รูปภาพสำหรับอ้างอิงในการเรียกใช้งาน คุณจะกำหนดหรือไม่ก็ได้ เพราะว่าคุณสามารถอ้างอิงถึงรูปภาพนั้น โดยใช้ลำดับของรูปภาพก็ได้ โดยดูลำดับของรูปภาพจากช่อง Index

- ช่อง Image Count หมายถึง จำนวนรูปภาพที่เก็บอยู่ในคอนโทรล ImageList

คอนโทรล ToolBar หลังจากเก็บรูปภาพที่ต้องการไว้ในคอนโทรล ImageList แล้วต่อไปจะเป็นการนำรูปภาพที่เก็บไว้มาแสดงในคอนโทรล ToolBar เพื่อใช้เป็นเมนูรูปภาพแทนเมนูคำสั่งตามที่ต้องการ ให้คลิกขวาที่คอนโทรล ToolBar แล้วเลือกคำสั่ง Properties ดังรูป

แท็บ General ที่ช่อง ImageList: ให้เลือกชื่อคอนโทรล ImageList ที่ต้องการใช้งานร่วมกับคอนโทรล ToolBar กรณีนี้คือ ImageList1

แท็บ Buttons เพื่อเพิ่มปุ่มเข้าไปในคอนโทรล ToolBar โดยการคลิกปุ่ม Insert Button จำนวนปุ่มที่คุณเพิ่มต้องเท่ากับรูปภาพที่เก็บอยู่ในคอนโทรล ImageList ช่องที่น่าสนใจมี 3 ตัว คือ

- ช่อง Caption: ใช้สำหรับกำหนดข้อความที่จะปรากฏอยู่ในแต่ละปุ่ม

- ช่อง Key: ใช้สำหรับตั้งชื่อให้ปุ่ม คุณจะต้องกำหนดชื่อให้แต่ละปุ่มด้วยเพื่อใช้สำหรับตรวจสอบว่าผู้ใช้คลิกปุ่มใดในทูลบาร์

- ช่อง Image: ใช้สำหรับระบุลำดับที่ของปุ่มนี้จะใช้รูปที่อยู่ในคอนโทรล ImageList

#### คอนโทรล StatusBar

คอนโทรล StatusBar ใช้สำหรับแสดงสถานะ, แสดงข้อความ หรือแสดงรูปภาพทางด้านล่างของแบบฟอร์ม

คอนโทรล StatusBar เป็นคอนโทรลอีกตัวหนึ่งที่จะช่วยเสริมให้โปรเจกต์ของคุณมีลักษณะมาตรฐานมากขึ้น เราสามารถใช้คอนโทรล StatusBar เพื่อแสดงสถานะของปุ่มบางปุ่มที่อยู่บนคีย์บอร์ด, แสดงวันที่, แสดงเวลาและแสดงข้อความใดๆ เป็นต้น

ช่องที่ปรากฏอยู่ในคอนโทรล StatusBar จะเรียกว่า Panel ซึ่งสามารถเพิ่ม Panel ได้โดยการคลิกขวาที่คอนโทรล StatusBar แล้วเลือกคำสั่ง Properties จากนั้นคลิกที่แท็บ Panel

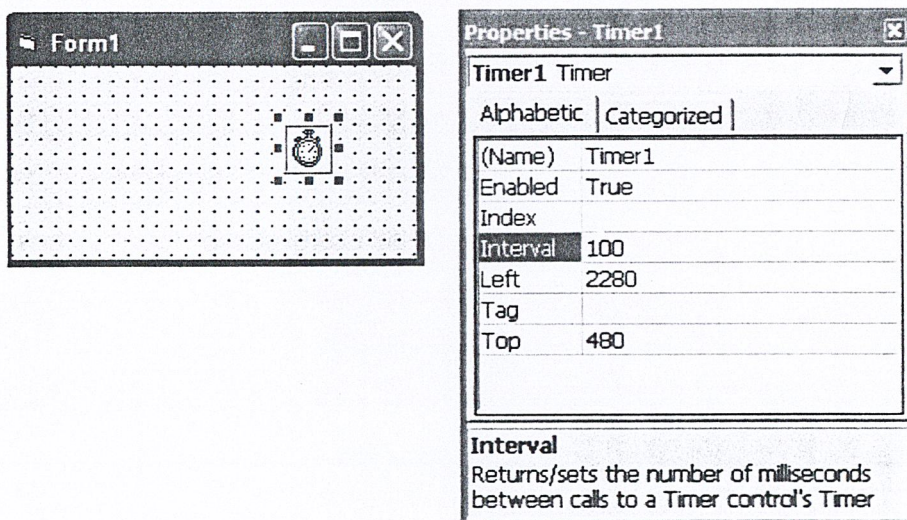
คลิกปุ่ม Insert Panel เพื่อเพิ่มจำนวน Panel คลิกปุ่ม Remove Panel เพื่อลดจำนวน Panel สำหรับความหมายของช่องอื่นๆ ที่น่าสนใจมีดังนี้

- ช่อง Text: หมายถึง ข้อความที่คุณต้องการให้ปรากฏใน Panel โดยที่ในช่อง Style: จะต้องมียค่าเท่ากับ 0-sbrText เท่านั้น

- ช่อง Style: หมายถึง การกำหนด ลักษณะของ Panel ว่า มีลักษณะอย่างไรคุณสามารถกำหนดได้ทั้งสิ้น 8 ประการดังนี้

- 0-sbrText กำหนดให้ Panel แสดงข้อความที่อยู่ในช่อง Text
- 1-sbrCaps กำหนดให้ Panel แสดงสถานะของปุ่ม Caps Lock
- 2-sbrNum กำหนดให้ Panel แสดงสถานะของปุ่ม Num Lock
- 3-sbrIns กำหนดให้ Panel แสดงสถานะของปุ่ม Insert
- 4-sbrScrl กำหนดให้ Panel แสดงสถานะของปุ่ม Scroll Lock
- 5-sbrTime กำหนดให้ Panel แสดงเวลา
- 6-sbrDate กำหนดให้ Panel แสดงวันที่
- 7-sbrKana กำหนดให้ Panel แสดงสถานะของการเลือกอักษรคาดคะเนในภาษาญี่ปุ่น

คอนโทรล Timer เป็นคอนโทรลที่ไม่ปรากฏให้เห็นในขณะรัน แต่จะทำงานเป็นช่วงเวลาที่กำหนดไว้เท่านั้น การกำหนดช่วงเวลาของไทมเมอร์จะกำหนดใน Properties Interval ของคอนโทรล ไทมเมอร์ การกำหนดเวลาจะอยู่ในช่วง 1-1000 มิลลิวินาที เช่น ถ้ากำหนดค่าใน Properties Interval =100 หมายถึง 1/10 วินาทีนั่นเอง



ภาพที่ 3. 46 แสดง Timer และ Properties

ค่าคอนโทรล (Data Control) ซึ่งคอนโทรลที่ใช้ในการเชื่อมข้อมูล จากฐานข้อมูลมายังคอนโทรลต่างๆ ก็คือ ค่าคอนโทรล (Data Control)

Properties สำคัญของ Data Control ได้แก่

- DataBaseName กำหนดชื่อไฟล์ของฐานข้อมูลที่จะติดต่อ
- RecordSource กำหนดตารางข้อมูลภายในฐานข้อมูลนั้นๆ เพราะบางฐานข้อมูลอาจมีหลายตาราง
- Connect กำหนดชนิดฐานข้อมูลที่เราติดต่อกับ เช่น Access, Dbase, FoxPro

- RecordsetType กำหนดได้ 3 ค่าคือ
  1. Tableกระทำได้กับเพียง 1 ตาราง
  2. Dynaset กระทำได้กับข้อมูลที่มีมากกว่า 1 ตาราง
  3. Snapshot กระทำได้มากกว่า 1 ตารางแต่ไม่สามารถแก้ไขข้อมูลได้

บาวนด์คอนโทรล (Bound Control) เมื่อดึงฐานข้อมูลมาเก็บไว้ในดาต้าคอนโทรลแล้ว ถ้านำไปใช้ในคอนโทรลต่างๆ คอนโทรลเหล่านั้นจะเรียกว่า บาวนด์คอนโทรล (Bound Control) เช่น เท็กซ์บ็อกซ์, ลิสต์บ็อกซ์

คอนโทรล DataGrid เป็นคอนโทรลที่ถูกจัดอยู่ในกลุ่ม Data-Aware Control มีรูปแบบการแสดงผลแบบตาราง

คอนโทรล MSFlexGrid เป็นคอนโทรลด้านฐานข้อมูล ที่ถูกออกแบบมาให้ใช้ในการแสดงผลข้อมูลที่มาจากตารางมากกว่า 1 ตารางในเวลาเดียวกัน

คอนโทรล ADO Data เป็นคอนโทรลที่ถูกออกแบบมาแทน Data Control โดยที่คอนโทรล ADO Data จะใช้เทคโนโลยี OLEDB เป็นกลไกในการเข้าถึงข้อมูลในฐานข้อมูล รูปแบบการติดต่อของคอนโทรล ADO Data กับฐานข้อมูล จะเป็นลักษณะการเชื่อมต่อ (Connection) เข้ากับฐานข้อมูลแทนการกำหนดชื่อฐานข้อมูล

อาร์เรย์ของคอนโทรล การออกแบบฟอร์มต่างๆ นั้นไม่จำกัดจำนวนคอนโทรลที่ใช้ ซึ่งแล้วแต่ว่างานนั้นจะต้องการคอนโทรลมากน้อยเพียงใด เมื่อคอนโทรลบางตัวมีการใช้ซ้ำๆ กันหลายตัว ก็สามารถใช้วิธี Copy คอนโทรลนั้นได้โดยคำสั่ง Edit -> Copy และ Edit -> Paste จากนั้นจึงใช้เมาส์ลากคอนโทรลนั้นๆ ไปวางยังตำแหน่งที่ต้องการ แต่การ Copy คอนโทรลถือว่าเป็นการสร้างคอนโทรลในรูปแบบอาร์เรย์ (Array)

โฟกัส (Focus) การกำหนดโฟกัส (Focus) เป็นการกำหนดขั้นตอนของการทำงาน โดยให้การทำงานเริ่มต้นไปอยู่ที่คอนโทรลตัวใดตัวหนึ่ง การที่โปรแกรมรอการทำงานที่ตัวใดตัวหนึ่งจะเรียกว่าการกำหนดโฟกัส ซึ่งเมื่อเรากด Enter ก็เหมือนกับการคลิกที่คอนโทรลตัวนั้น อาจกล่าวได้ว่า เป็นเหมือนการตั้งให้ Cursor ไปรอรับคำสั่งการทำงานในตำแหน่งต่างๆ ของโปรแกรม ภาษา เช่น คำสั่ง Go (X, Y) ใน Pascal เป็นต้น ซึ่งการกำหนดโฟกัสจะกำหนดในขั้นตอนของการเขียนโปรแกรม

### พรอพเพอร์ตี้อื่นๆ

Active Control	แสดงออปเจ็กปัจจุบันที่กำลังทำงานอยู่ ใช้ได้เฉพาะโปรแกรมที่รัน ไม่มีในพรอพเพอร์ตี้
Active Form	กำหนด Form Active กำหนดได้คอนเขียน Code เช่นเดียวกับ Active Control
Alignment	กำหนดตำแหน่งของข้อความบนออปเจ็ก โดย 0->ชิดซ้าย, 1->ชิดขวา, 2->กึ่งกลาง
Autoredraw	ใช้กำหนดให้ออปเจ็กหรือฟอร์มที่แสดงรูปภาพ โดย จัดเก็บ =True, ไม่จัดเก็บ=False ภาพนั้นจะถูกเก็บไว้ในหน่วยความจำ นอกเหนือจากการแสดงภาพด้วย
BackColor	ระบุสีพื้นหลัง
BorderStyle	กำหนดขอบ โดย 1= ปรากฏขอบ, 2= ไม่ปรากฏขอบ
Clipcontrol	กำหนดสีของฟอร์ม, ออปเจ็กที่ไม่ใช่รูปภาพ โดยกำหนดสีใหม่=True
Current x,y	ระบุตำแหน่งที่วาดภาพหรือข้อความในแกน x,y ส่วนใหญ่ใช้กับรูปภาพ
Caption	กำหนดข้อความสำหรับแสดงภายในคอนโทรล
Dragicon	ใช้กำหนดรูปภาพ Icon ให้ปรากฏขณะลากหรือวางเมาส์
Dragmode	0-> กำหนดแบบ Manual, 1->กำหนดแบบ Olecontrol, 2-> Auto ใช้ Event ไม่ได้คอนเขียน Code
Drawstyle	0-> เส้นปกติ, 1-> เส้นประ, 2-> เส้นจุด, 3->เส้นประสลับจุด, 4->เส้นประจุดจุด
Drawmode	ระบุ Mode ในการวาดรูป
Drawwidth	ระบุความกว้างของเส้น เช่น 1 หมายถึง 1 Pixel
Enabled	กำหนดให้คอนโทรลสามารถมีหรือไม่มีการตอบสนองต่อเหตุการณ์ที่ถูกกำหนดโดยผู้ใช้ เช่น Click โดยจะมีค่าเป็น True หรือ Fales
FillColor	ระบุสีในการวาดภาพ Circle, Line
Fillsytle	ระบุแบบของ Circle, Line
Fontitalic	กำหนดอักษรแบบเอียง
Fontname	ระบุแบบอักษรที่ใช้กับออปเจ็ก
FontSize	ระบุขนาดของตัวอักษร

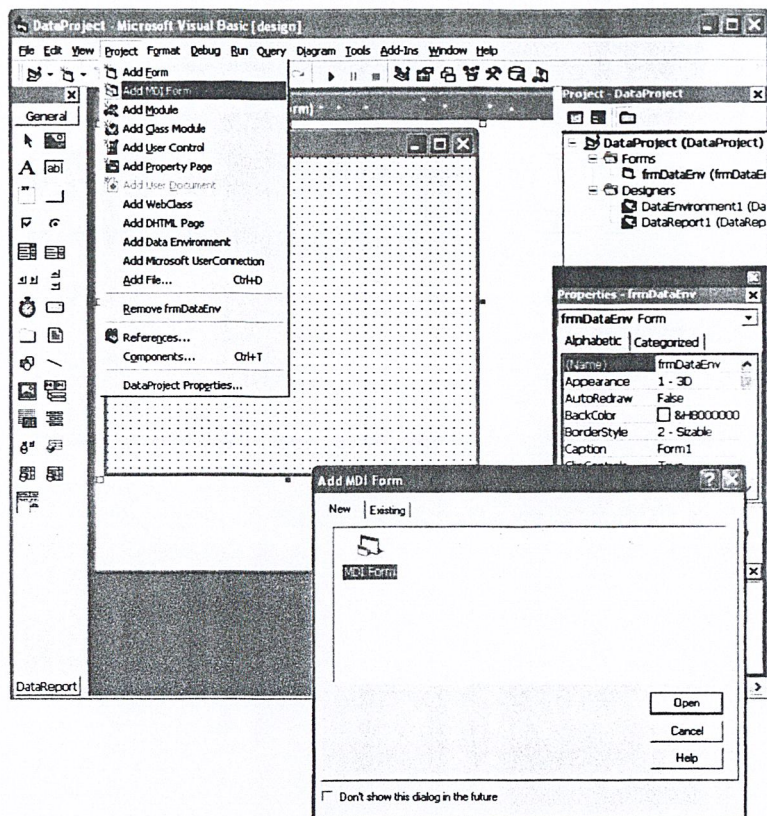
Fontstrikeout	กำหนดเส้นคาคกลางตัวอักษร
Forecolor	สีข้อความ
Height	ความสูงของออปเจ็ก
hWnd	แสดง Handle ซึ่งใช้กับ Form, Control
Icon	ระบุชื่อ Icon
Image	แสดง Handle ซึ่งใช้กับรูปภาพ
Index	ระบุลำดับ Control Array
Left	ระบุตำแหน่งเริ่มต้นในแนวแกน x มักใช้ในการกำหนด Code
Linkitem	กำหนดข้อมูลที่ใช้ส่งระหว่าง Client กับ Application โดยใช้กับ DDE
Name	กำหนดชื่อคอนโทรลสำหรับการอ้างอิงโดย Code ในการเขียนโปรแกรม
Page	กำหนดหน้าในการแสดงข้อมูล
Scale Height	ระบุขนาดสูงสุดในแกน x ของออปเจ็ก
Scale Left	ระบุตำแหน่งเริ่มต้นของพรอพเพอร์ตี้ Left
Scale Top	ระบุตำแหน่งเริ่มต้นของพรอพเพอร์ตี้ Top
Scale Change	แสดงจำนวนในการเลื่อน Scroll
Sorted	เรียงข้อมูล
Text	กำหนดข้อความบน Text Box, List Box, Combo Box
Top	ใช้ระบุตำแหน่งเริ่มต้นของออปเจ็กในแนวแกน y
Width	ความกว้างของออปเจ็ก
Window State	แสดงขนาดขอบหน้าต่างขณะรัน โดย 0-> ปกติ, 1->Minimize, 2->Maximize

### 3.3.4.3 การใช้งาน MDI Form

คำว่า MDI ย่อมาจากคำว่า Multiple Document Interface คือ ฟอรัมพิเศษที่ทำหน้าที่เป็นฟอรัมแม่ (Parent Form) เพื่อใช้เก็บฟอรัมลูก (Child Form) ไว้ ส่งผลให้สามารถแสดงฟอรัมลูกได้หลายฟอรัมในเวลาเดียวกัน ดังนั้นใน 1 โปรเจกต์สามารถมี MID Form ได้เพียง 1 ฟอรัมเท่านั้น แต่สามารถมีฟอรัมลูกได้หลายฟอรัม

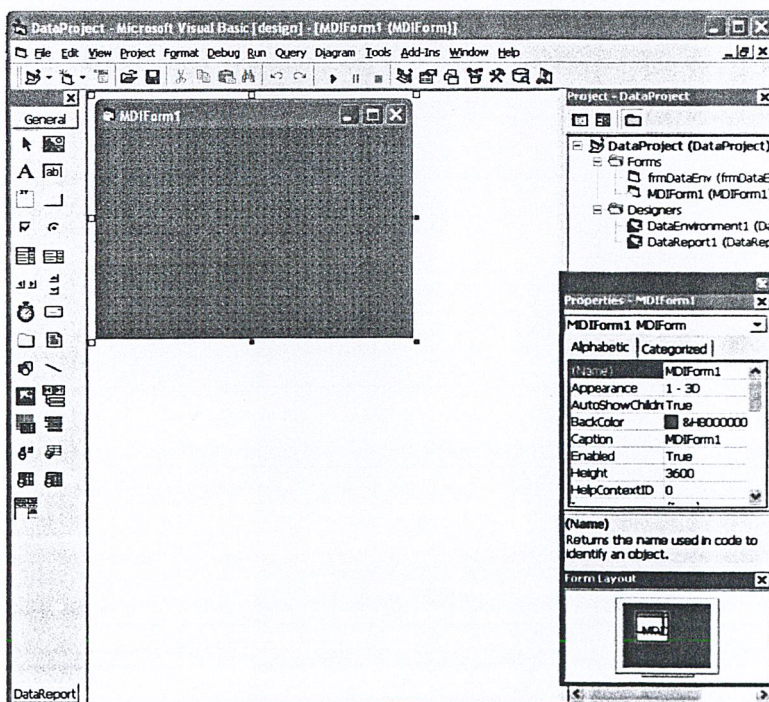
การเพิ่มฟอรัมชนิด MDI เข้ามาใน VBIDE

1. ใน VBIDE ชนิด Data Project หรือ Standard.EXE คลิกเมนู Project > Add MDI Form แล้วคลิกปุ่ม Open

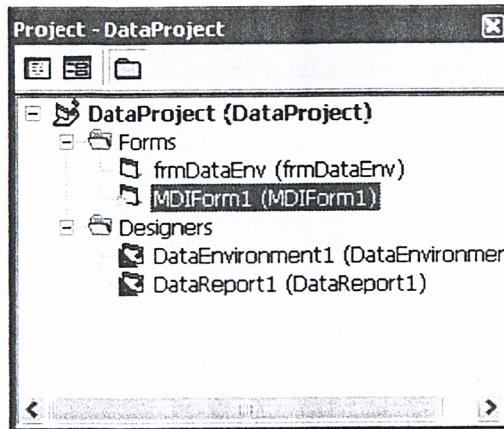


ภาพที่ 3.47 แสดงการเพิ่มฟอร์มชนิด MDI เข้ามาในใช้ใน VBIDE

2. จะเห็นได้ว่า ฟอร์มชนิด MDI จะถูกเพิ่มเข้ามาใน VBIDE ให้สังเกตที่หน้าต่าง Project Explorer ประกอบ

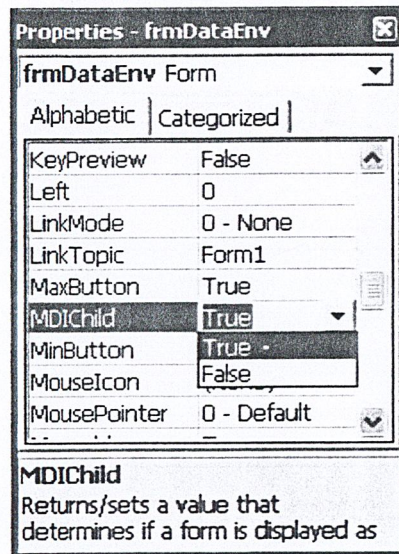


ภาพที่ 3.48 แสดงฟอร์มชนิด MDI ที่ถูกเพิ่มเข้ามาใน VBIDE



ภาพที่ 3.49 แสดงลักษณะ ไอคอนที่ปรากฏอยู่ใน Project Explorer ที่กำกับฟอร์มชนิด MDI

3. เมื่อสร้างฟอร์มปกติขึ้นมา ให้กำหนดคุณสมบัติ MDIChild ของฟอร์มปกติมีค่าเท่ากับ True เพื่อกำหนดให้ฟอร์มปกติเปลี่ยนเป็นฟอร์มลูก (Child Form)



ภาพที่ 3.50 ไอคอนของฟอร์มชนิด MDI และฟอร์มชนิด Child

การสร้างเมนูบนฟอร์มด้วย Menu Editor

Menu Editor เป็นโปรแกรมสร้างเมนูบนฟอร์มซึ่งสามารถสร้างได้โดยคลิกปุ่ม หรือใช้คำสั่ง Tool > Menu Editor หรือกด CTRL+E

รายการต่างๆ ที่ปรากฏอยู่ในกรอบ Menu Editor จะมีความหมายดังนี้

Caption : กำหนดข้อความที่ต้องการ ปรากฏบนฟอร์ม

Name : กำหนดชื่อของเมนู ซึ่งจะมีผลในคอนเขียนโปรแกรม และที่สำคัญทุกเมนูต้องตั้งชื่อ

Index : ใช้เมื่อต้องการกำหนดชื่อ Object บนเมนูเป็น Array

Checked : กำหนดคุณลักษณะทางกราฟิก เช่น มีเครื่องหมายถูกที่หน้าเมนู

Enabled : กำหนดคุณลักษณะให้เมนูใช้งานได้หรือไม่

Visible : กำหนดแสดงหรือซ่อนเมนู

### 3.3.5 วิธีการติดต่อกับฐานข้อมูลของ MS-Visual Basic 6.0

โปรแกรม MS-Visual Basic 6.0 มีความสามารถในการจัดการฐานข้อมูลได้ ดังนี้

#### 3.3.5.1 การใช้ค้ำคอนโทรล (Data Control)

การใช้ค้ำคอนโทรล (Data Control) เป็นวิธีการที่ง่ายและสะดวกที่สุดในการติดต่อกับฐานข้อมูล เนื่องจากค้ำคอนโทรลจะติดต่อกับฐานข้อมูลและจัดการกับฐานข้อมูลในตารางอัตโนมัติ เช่น การเปิดฐานข้อมูล การแสดงและแก้ไขข้อมูลในตาราง อย่างไรก็ตามการใช้ค้ำคอนโทรลยังมีข้อจำกัดอยู่พอสมควร เช่น ในการลบข้อมูล

ตัวอย่าง การติดต่อกับฐานข้อมูลด้วย Data Control

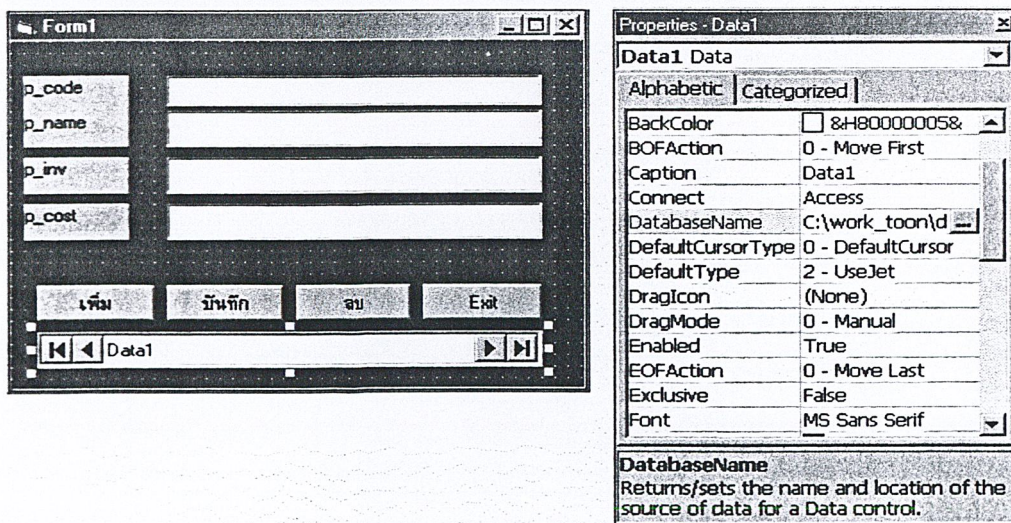
1. การสร้างฟอร์มและสร้าง Data Control แล้วจะต้องกำหนดคุณสมบัติ (Property) ไปยังคอนโทรลอื่นดังนี้

Dataname กำหนดชื่อไฟล์ของฐานข้อมูล

RecordSource กำหนดตารางข้อมูลภายในฐานข้อมูลนั้นๆ

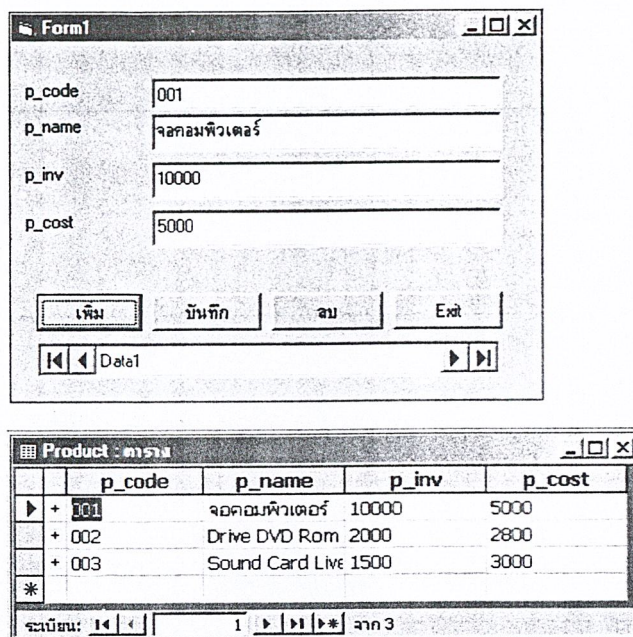
Connect กำหนดชนิดฐานข้อมูล เช่น Access

2. เมื่อกำหนด Data Control แล้วก็จะต้องสร้างคอนโทรลอื่นๆ เช่น Textbox ของ p\_code, p\_name, p\_inv และ p\_cost เพื่อที่จะเชื่อมฐานข้อมูลจากค้ำคอนโทรลและกำหนดฟิลด์ที่ต้องการแสดงออกทางคอนโทรล



ภาพที่ 3.51 แสดงการสร้างฟอร์มเพื่อเชื่อมฐานข้อมูลจาก Data Control

3. ในการจัดการฐานข้อมูลจะต้องสร้างปุ่มคอมมานด์ เช่น เพิ่ม,บันทึก,ลบ, Exit และเขียนโปรแกรมในส่วนของปุ่มต่างๆ เพิ่มเติมเพื่อการจัดการฐานข้อมูล



ภาพที่ 3.52 แสดงการ Run Program หลังการเชื่อมฐานข้อมูลจาก Data Control แล้ว

### 3.3.5.2 การใช้ค่าตัวออบเจกต์ (Data Object)

วิธีนี้จะต้องเขียนโปรแกรมเพื่อติดต่อกับฐานข้อมูล โดยการใช้ออบเจกต์ต่างๆ ที่ VB6 มีมาให้โดยการเพิ่ม ลบ หรือแก้ไขข้อมูล จะต้องเขียนโปรแกรมจัดการเอง แต่ข้อดีของวิธีนี้ก็คือสามารถติดต่อกับข้อมูลหลายๆ ตารางพร้อมกัน สามารถแสดงคิวรีตอนรันโปรแกรมได้ และสามารถควบคุมความผิดพลาดต่างๆ ได้ดีกว่าใช้ค่าคอนโทรล รวมทั้งสามารถใช้ภาษา SQL เพื่อจัดการกับฐานข้อมูลได้อีกด้วย

### 3.3.5.3 การจัดการฐานข้อมูลโดยใช้ Structure Query Language (SQL)

Structure Query Language (SQL) เป็นภาษามาตรฐานที่ใช้ในการจัดการฐานข้อมูลโดยเฉพาะที่เป็นแบบ RDBMS (Relation Database Management System) SQL เป็นภาษาที่มีประสิทธิภาพสูงสามารถเรียนรู้และใช้งานได้อย่างง่ายดาย จึงทำให้เป็นที่นิยมได้อย่างรวดเร็ว นอกจากนี้ยังสามารถใช้ในการจัดการกับฐานข้อมูลโดยผ่าน RDBMS เอง หรือจะตั้งผ่านโปรแกรมก็ได้ เช่น หากเราใช้ MS-Access เราก็สามารถพิมพ์คำสั่ง SQL ได้โดยตรงในโปรแกรมโปรแกรม Access หรือจะใช้คำสั่ง SQL ผ่าน VB6 หรือ Compiler อื่นๆ ก็ได้ เราจะใช้ SQL เพื่อทำการจัดการข้อมูลในฐานข้อมูลได้หลายอย่าง เช่น การแสดงข้อมูลจากฐานข้อมูลแบบมีเงื่อนไข, การเพิ่ม, การลบ และการนำข้อมูลจากหลายๆ ตารางมาแสดงร่วมกันได้ เป็นต้น

### 3.3.5.4 โครงสร้างของภาษา SQL

ภาษา SQL ประกอบไปด้วย 3 ส่วนใหญ่ๆ คือ

1. Data Definition Language (DDL) เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับจัดการโครงสร้างของฐานข้อมูล เช่น การสร้างฐานข้อมูล, ปรับปรุงโครงสร้างของฐานข้อมูล เป็นต้น ตัวอย่างการใช้งานกลุ่มคำสั่ง DDL นี้ก็คือ การสร้างฐานข้อมูลด้วย MS SQL Server ก็จะมีการใช้งานคำสั่งในกลุ่ม DDL เป็นหลัก

2. Data Manipulation Language (DML) เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับจัดการข้อมูลในฐานข้อมูล เช่น การแสดงข้อมูลแบบมีเงื่อนไข, การลบข้อมูล, การเพิ่มข้อมูล และการแสดงที่มาจากตารางหลายตาราง เป็น

กลุ่มฟังก์ชัน Aggregate Function เป็นฟังก์ชันพิเศษของภาษา SQL ที่ทำหน้าที่เฉพาะอย่าง เช่น การหาผลรวมใน Record , ค่าสูงสุด , ค่าต่ำสุด เป็นต้น เป็นกลุ่มฟังก์ชันที่มีประโยชน์มาก เพราะจะช่วยในการลดภาระในการเขียนโค้ดจัดการเอง

สำหรับการใช้งานภาษา SQL ร่วมกับ Visual Basic เพื่อจัดการฐานข้อมูล จะใช้งานกับกลุ่มคำสั่ง DML เป็นหลัก โดยจะอธิบายการใช้งานกลุ่มคำสั่ง DML ร่วมกับฟังก์ชัน Aggregate และการกำหนดเงื่อนไขโดยการใช้ตัวดำเนินการด้านต่างๆ

คำสั่งในกลุ่มของ DML มีคำสั่งพื้นฐานอยู่ 4 คำสั่ง

1. DELETE เป็นคำสั่งที่ใช้สำหรับลบข้อมูลหรือ Record ใดๆ ในตาราง
2. INSERT เป็นคำสั่งที่ใช้สำหรับการเพิ่มข้อมูลหรือเพิ่ม Record ใดๆ เข้าไปในตาราง
3. SELECT เป็นคำสั่งที่ใช้สำหรับเลือกข้อมูลหรือแสดง Record ใดๆ ที่ต้องการจากตารางอาจจะมาจากตารางเดียว หรือหลายตารางก็ได้
4. UPDATE ใช้สำหรับการแก้ไขข้อมูลหรือแก้ไข Record ใดๆ ในตาราง

ตัวดำเนินการ (Operator)

ตัวดำเนินการที่น่าสนใจ ได้แก่

- |    |     |   |
|----|-----|---|
| =  | คือ | เท่ากับ (Equal)                               |
| <> | คือ | ไม่เท่ากับ (Not Equal)                        |
| <  | คือ | น้อยกว่า (Less Than)                          |
| >  | คือ | มากกว่า (Greater Than)                        |
| <= | คือ | น้อยกว่าหรือเท่ากับ (Less Than or Equal To)   |
| >= | คือ | มากกว่าหรือเท่ากับ (Greater Than or Equal To) |

Like คือ เป็นการเปรียบเทียบโดยใช้ตัวอักษรพิเศษ (Wild Card Character) เข้าร่วม

ตัวดำเนินการด้านตรรกะ (Logical Operator)

ตัวดำเนินการด้านตรรกะที่นิยมใช้จะมีอยู่ 3 ชนิดคือ And , OR และ Not

### 3.3.5.5 กลุ่มฟังก์ชัน Aggregate

กลุ่มฟังก์ชัน Aggregate เป็นฟังก์ชันที่ใช้ในการคำนวณทางคณิตศาสตร์ โดยช่วยในการนำเสนอผลการค้นหาให้เป็นไปอย่างมีประสิทธิภาพ ซึ่งมีฟังก์ชันพื้นฐานดังนี้

AVG()	หน้าที่	หาค่าเฉลี่ยของ Field จาก Record ทั้งหมด
COUNT()	หน้าที่	นับจำนวนเรคคอร์ด
FIRST()	หน้าที่	หาค่าแรกในฟิลด์
LAST()	หน้าที่	หาค่าสุดท้ายในฟิลด์
MAX()	หน้าที่	หาค่ามากที่สุด หรือค่าสูงสุด
MIN()	หน้าที่	หาค่าน้อยที่สุด หรือค่าต่ำสุด
SUM()	หน้าที่	หาผลรวมทั้งหมดของฟิลด์

### 3.3.5.6 ลักษณะการใช้งานของกลุ่มคำสั่ง DML

คำสั่ง DELETE

เป็นคำสั่งที่ใช้สำหรับลบข้อมูล หรือลบเรคคอร์ดใดๆ ออกจากตาราง มีรูปแบบการใช้งาน 2 ลักษณะใหญ่ๆ คือ

รูปแบบที่ 1	DELETE FROM ชื่อตาราง WHERE เงื่อนไข
รูปแบบที่ 2	DELETE *FORM ชื่อตาราง

ชื่อตารางในที่นี้หมายถึง ชื่อตารางที่ต้องการลบ ส่วนเงื่อนไข จะหมายถึง เงื่อนไขในการลบข้อมูล หรือลบเรคคอร์ดใดๆ

สำหรับเครื่องหมาย \* หมายถึง ข้อมูลใดๆ หรือข้อมูลทุกเรคคอร์ด

คำสั่ง INSERT

เป็นคำสั่งที่ใช้สำหรับเพิ่มข้อมูล หรือเพิ่มเรคคอร์ดเข้าไปในตาราง ในกรณีที่ฟิลด์เป็นข้อมูลชนิด Text ต้องใช้เครื่องหมาย ' กำกับด้วยฟิลด์นั้นด้วย มีรูปแบบการใช้งาน 2 ลักษณะ

```

รูปแบบที่ 1  INSERT INTO tablename(field1, field2, ...) VALUES(value1,'value2', ...)
รูปแบบที่ 2  INSERT INTO tablename2
SELECT * FROM tablename1 WHERE criteria

```

ตัวแปร tablename หมายถึง ชื่อตารางที่ต้องการเพิ่มเรคคอร์ดเข้าไป

ตัวแปร tablename1 หมายถึง เลือกข้อมูลจากตารางที่ชื่อว่า tablename1 ตามเงื่อนไขที่ระบุไว้ในตัวแปร criteria แล้วนำมาเพิ่มที่ตาราง tablename2

ตัวแปร field-fieldn หมายถึง ชื่อของฟิลด์ต่างๆ ที่อยู่ในตารางที่ต้องการเพิ่มข้อมูล

ตัวแปร value-valuen หมายถึง ค่าของฟิลด์ที่จะเพิ่มเข้าไป โดยที่ค้องระบุค่าให้ตรงกับฟิลด์ด้วย

ตัวแปร criteria หมายถึง เงื่อนไขในการดึงข้อมูลจากตาราง tablename1

### คำสั่ง SELECT

ใช้สำหรับเลือกหรือดึงข้อมูล (Retrieve Data) ที่เราต้องการจากตารางที่ระบุไว้ เป็นคำสั่งที่มีความยืดหยุ่นสูงมาก เพราะฉะนั้นในการนำข้อมูลออกมาจากตารางมีมากมายหลายลักษณะ แต่มีรูปแบบการใช้งานพื้นฐาน 2 ลักษณะ คือ

```
SELECT *FROM ชื่อตาราง
```

หรือ

```
SELECT ฟิลด์ที่1, ฟิลด์ที่2,...,ฟิลด์ที่n FROM ชื่อตาราง WHERE เงื่อนไข
```

ชื่อตาราง หมายถึง ชื่อตารางที่ต้องการดึงข้อมูล

ตัวแปร ฟิลด์ที่1, ฟิลด์ที่2,...,ฟิลด์ที่n หมายถึง ชื่อฟิลด์ที่ต้องการดึงข้อมูล ถ้ามีมากกว่า 1 ฟิลด์จะใช้เครื่องหมาย , คั่นระหว่างฟิลด์

เงื่อนไข หมายถึง เงื่อนไขในการดึงข้อมูล อาจเป็นเงื่อนไขทางคณิตศาสตร์ หรือเป็นคำสั่ง SELECT ซ้อนอยู่ข้างในก็ได้

### คำสั่ง UPDATE

เป็นคำสั่งที่ใช้สำหรับแก้ไข หรือเปลี่ยนแปลงข้อมูลในเรคคอร์ดที่มีอยู่ในตาราง มีรูปแบบการใช้งานดังนี้

```
UPDATE ชื่อตาราง SET ชื่อฟิลด์ = ค่าที่กำหนด WHERE เงื่อนไข
```

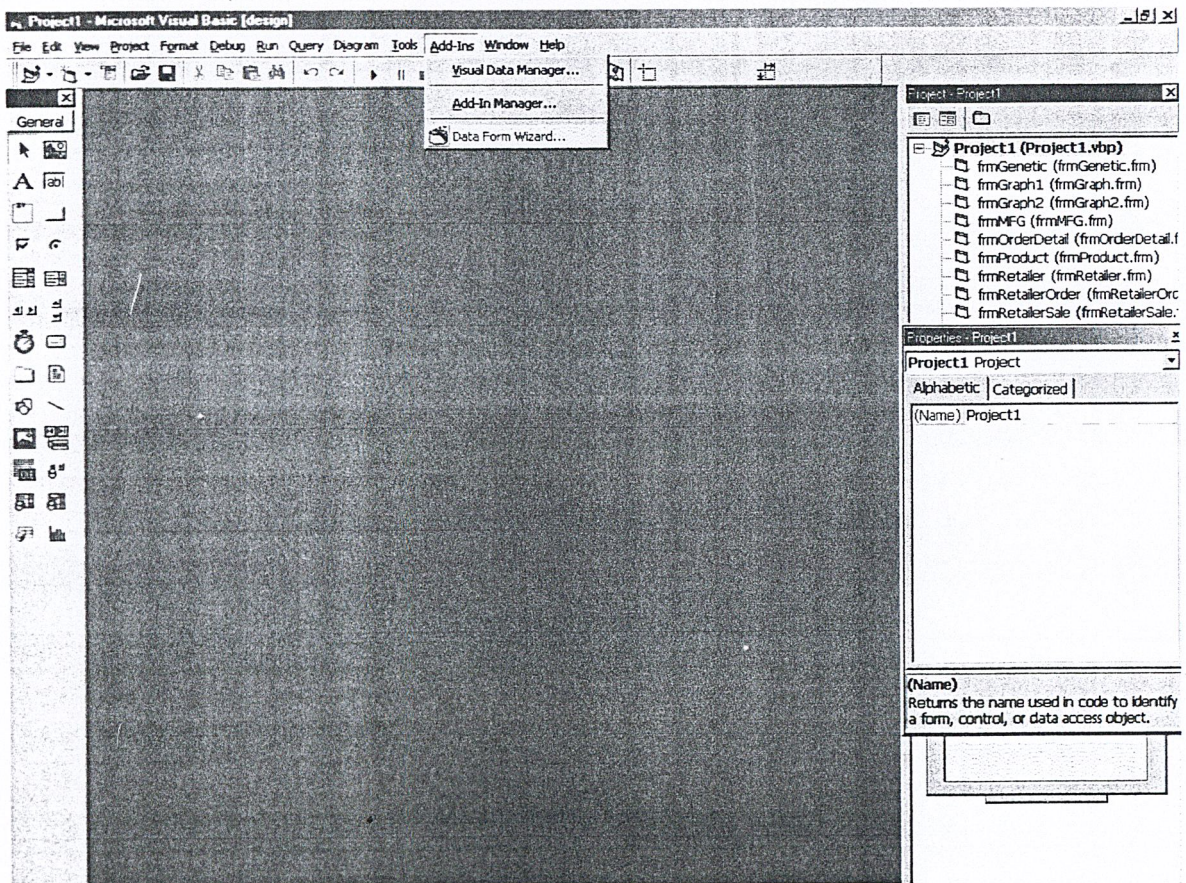
### 3.3.6 Data Form Wizard

เป็นเครื่องมือที่ให้แก่กับ VB6 ช่วยให้เราสามารถสร้างโปรแกรมเพื่อจัดการกับข้อมูลในตารางได้โดยอัตโนมัติ ส่วนการใช้ Data Control เพื่อติดต่อกับฐานข้อมูลจะมีข้อจำกัดอยู่พอสมควร ทำให้เราต้องเขียนโปรแกรมเพิ่มเติมเพื่อให้สามารถใช้งานได้จริง แต่เราสามารถสร้างโปรแกรมโดยไม่ต้องเขียนโปรแกรมใดๆ เพิ่มเติมและมีความสามารถครบถ้วนได้ด้วย Data Form Wizard

การจัดการฐานข้อมูลโดยใช้ Data Form Wizard

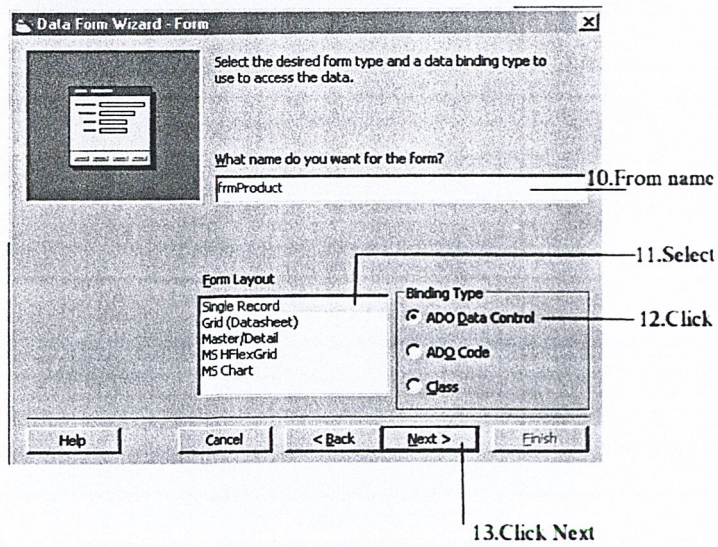
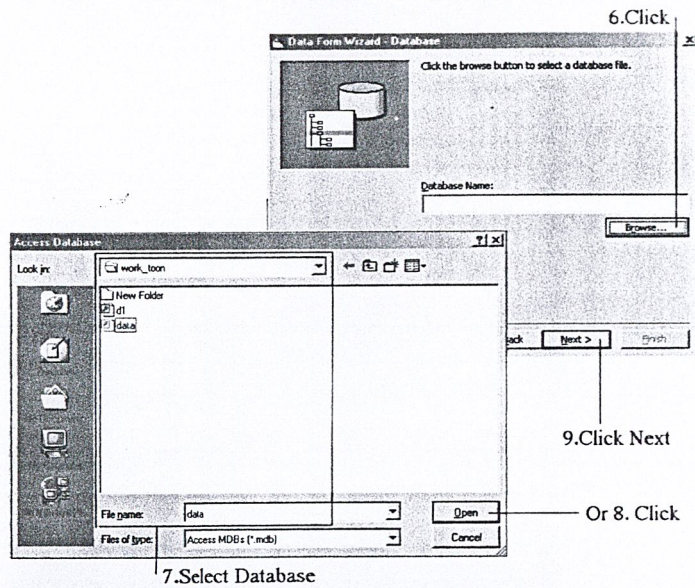
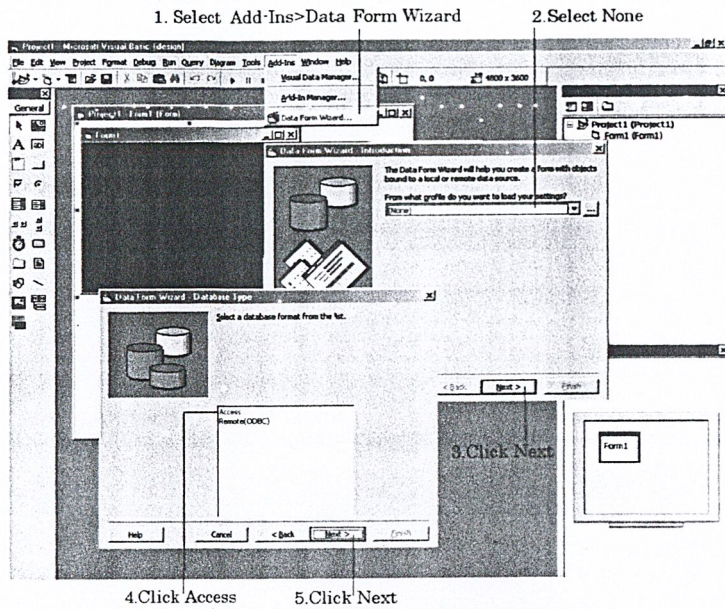
ก่อนที่จะสร้างโปรแกรมด้วย Data Form Wizard จะต้องทำตามขั้นตอนต่อไปนี้

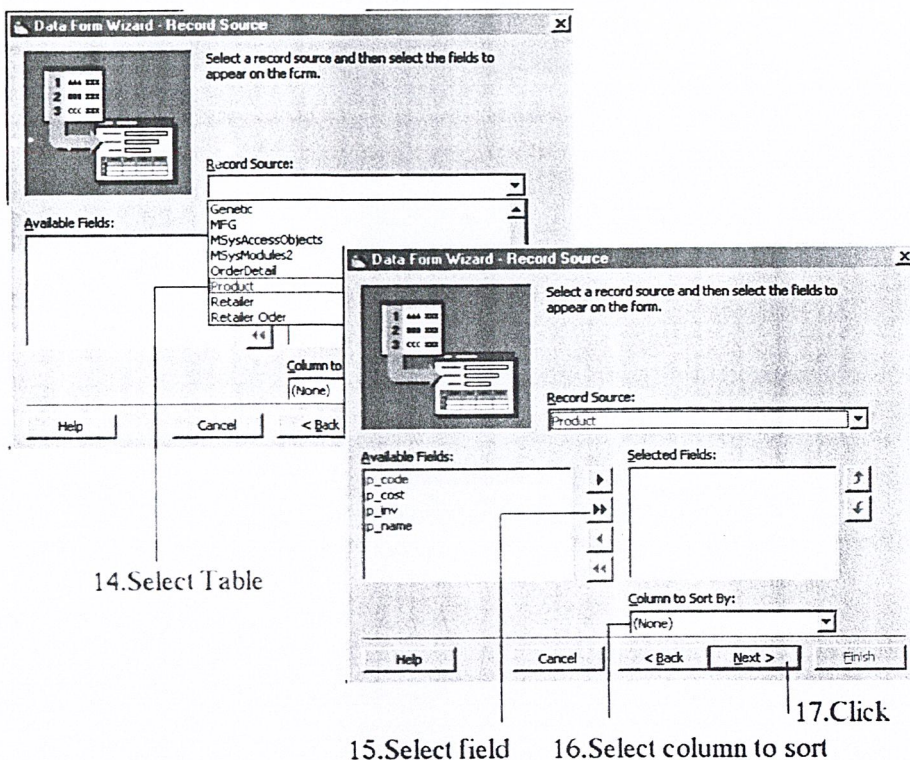
1. เลือกเมนูคำสั่ง Add-Ins>Add-In Manager
2. Click ที่ VB6 Data Form Wizard
3. เช็คข้อป้ช้้น Loaded/Unloaded
4. Click ปุ่ม OK



ภาพที่ 3.53 แสดงการเลือกเมนูคำสั่ง Add-Ins>Add-In Manager

หลังจากที่เราติดตั้ง Data form Wizard เรียบร้อยแล้ว เราจะยกตัวอย่างการสร้างโปรแกรมด้วย Data form Wizard เพื่อจัดการกับตารางสินค้า (Product)

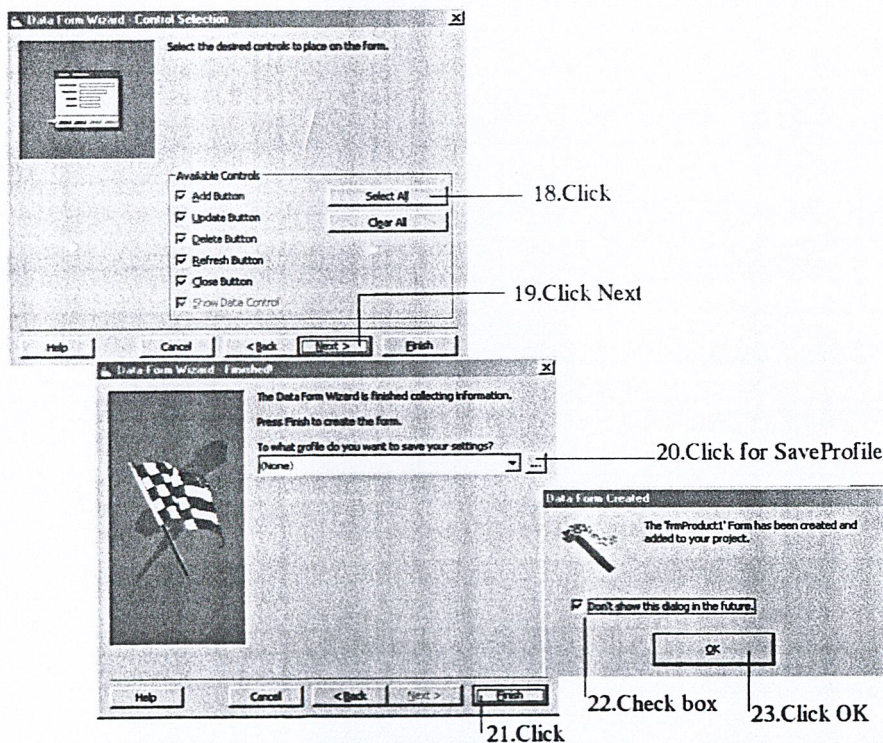




14. Select Table

15. Select field    16. Select column to sort

17. Click



18. Click

19. Click Next

20. Click for Save Profile

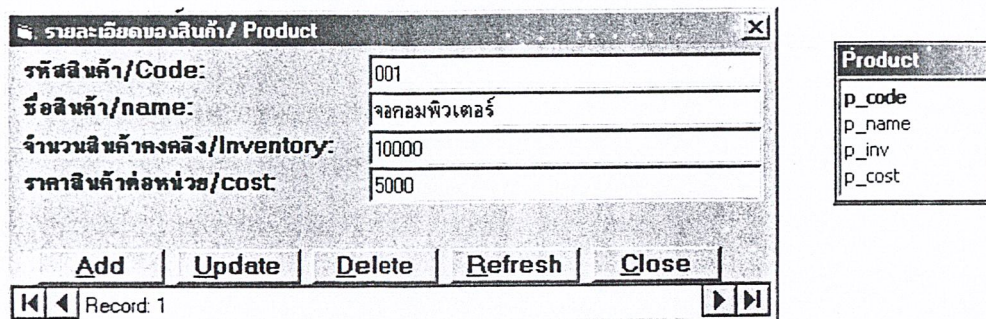
21. Click

22. Check box

23. Click OK

จะเห็นได้ว่าเราสามารถสร้างโปรแกรมเพื่อจัดการกับฐานข้อมูลได้อย่างง่ายดายโดยการ  
 ใช้ Data form Wizard โดยเราไม่ต้องเขียนโปรแกรมใดๆ เพิ่มเติมแต่ก็มีฟังก์ชันการใช้งานครบถ้วน  
 เพื่อความสะดวกและรวดเร็วเรามักจะใช้ Data form Wizard เพื่อสร้างโปรแกรมให้เราก่อน  
 หลังจากนั้นจึงนำฟอร์มที่ได้มาปรับเปลี่ยนตามความต้องการ

เมื่อทำการสร้างเสร็จเรียบร้อยแล้วจะมีลักษณะหน้าจอและโครงสร้างดังรูป



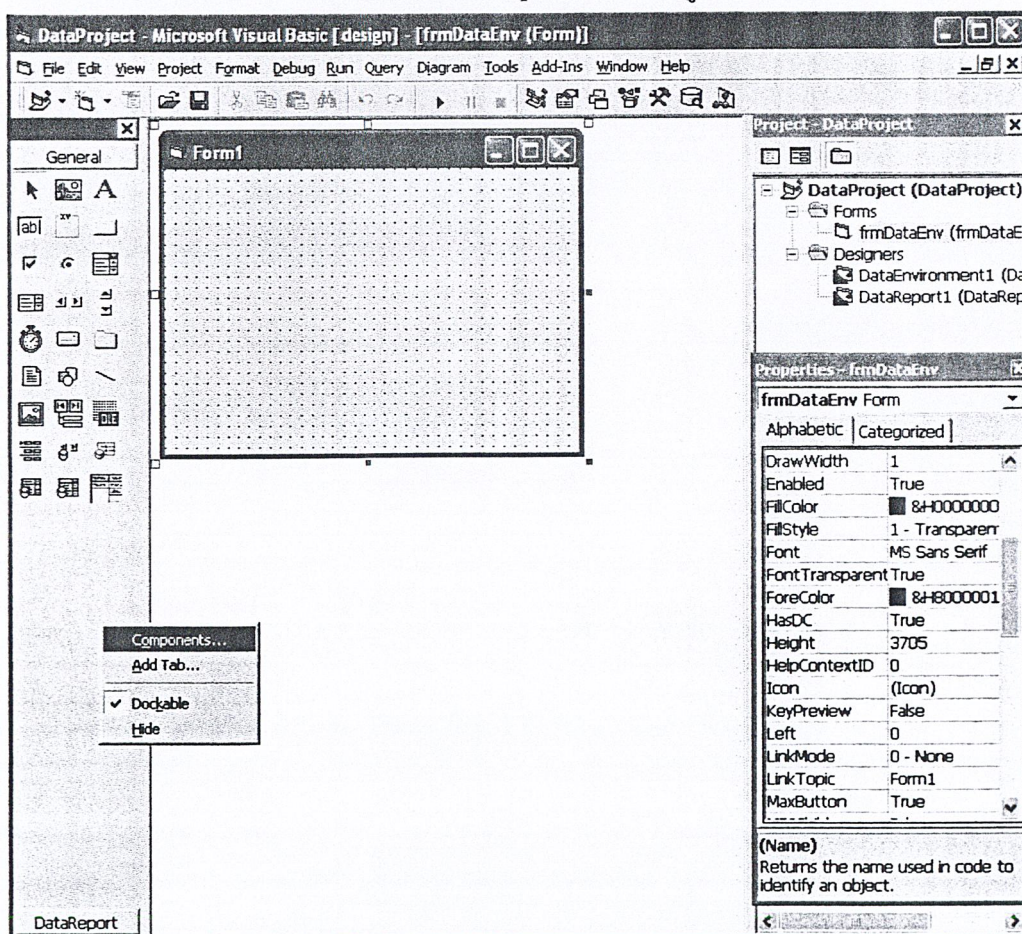
ภาพที่ 3.54 แสดงการ Run Program หลังการเชื่อมฐานข้อมูลจาก Data Control แล้ว

### 3.3.7 การออกแบบการสร้างเมนูและทูลบาร์ของระบบสินค้าคงคลัง

มีขั้นตอนต่างๆ ดังนี้

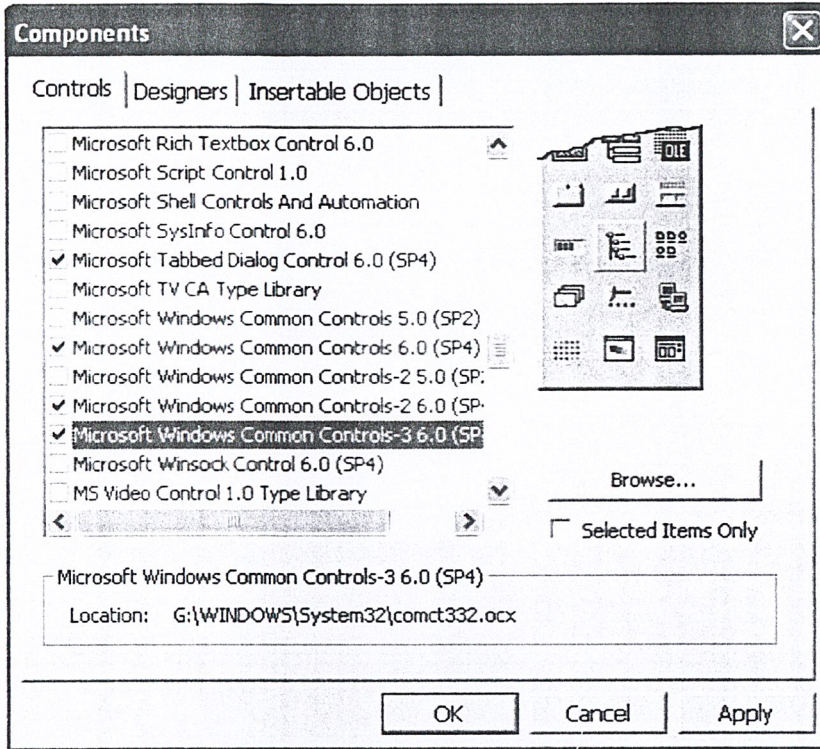
การเพิ่มคอนโทรล Microsoft Windows Common Controls 6.0 เข้ามาใน VBIDE

1. ใน VBIDE ชนิด Data Project ให้คลิกเมนู Project> Add Components... หรือคลิกขวาบริเวณแถบเครื่องมือของ VB แล้วเลือกคำสั่ง components... ดังรูป



ภาพที่ 3.55 แสดงการเลือกคำสั่ง Add Components

2. ในไดอะล็อกบ็อกซ์ Components ให้เลือกกลุ่มคอนโทรลเฉพาะ Microsoft Windows Common Controls 6.0 ที่มีคำว่า SP4 หรือ SP5 ต่อท้ายเท่านั้น และให้คุณเลือกคอนโทรล Microsoft Tabbed Dialog Control 6.0 (SP4 หรือ SP5) เข้ามาด้วย



ภาพที่ 3.56 แสดงไดอะล็อกบ็อกซ์ Components

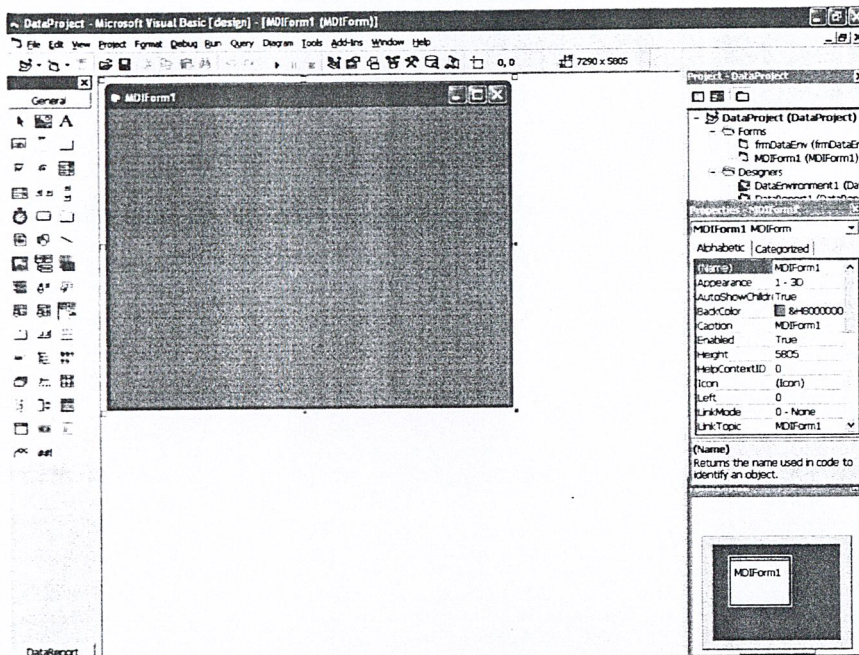
3. จะเห็นได้ว่า จะมีกลุ่มคอนโทรล Microsoft Windows Common Control 6.0 ถูกเพิ่มเข้ามาที่แถบเครื่องมือของ VBIDE ดังรูป



ภาพที่ 3.57 แสดง icon ของคอนโทรล Microsoft Windows Common Control 6.0 ในแถบเครื่องมือ

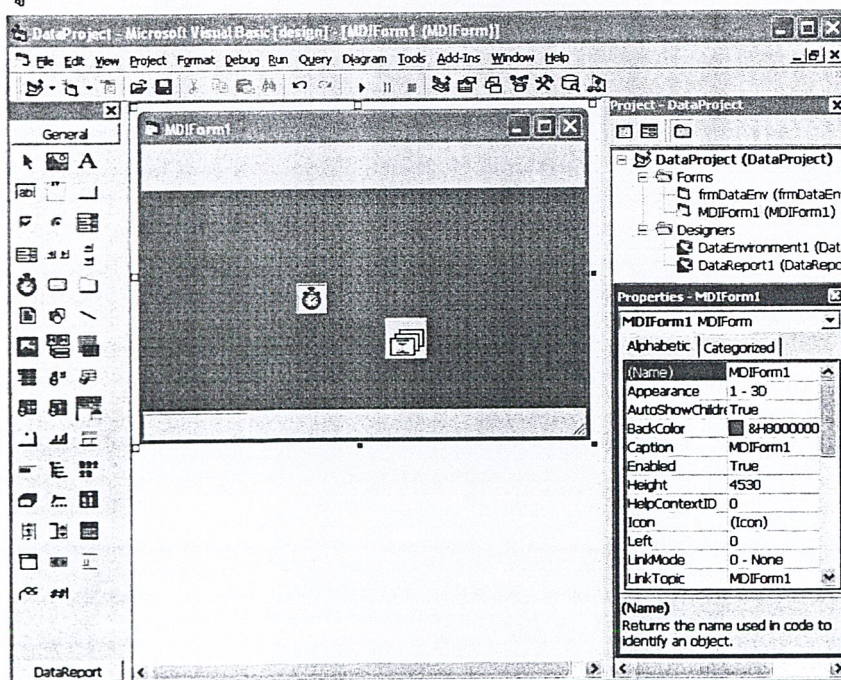
### 3.3.8 ขั้นตอนการสร้างทูลบาร์

1. ใน VBIDE ชนิด Data Project ให้เพิ่มฟอร์มชนิด MDI เข้ามา โดยการคลิกเมนู Project > Add MDI Form ดังรูป โดยที่คั้งชื่อฟอร์ม MDI ว่า frmMain ซึ่งจะทำหน้าที่เป็นฟอร์มหลักของระบบ



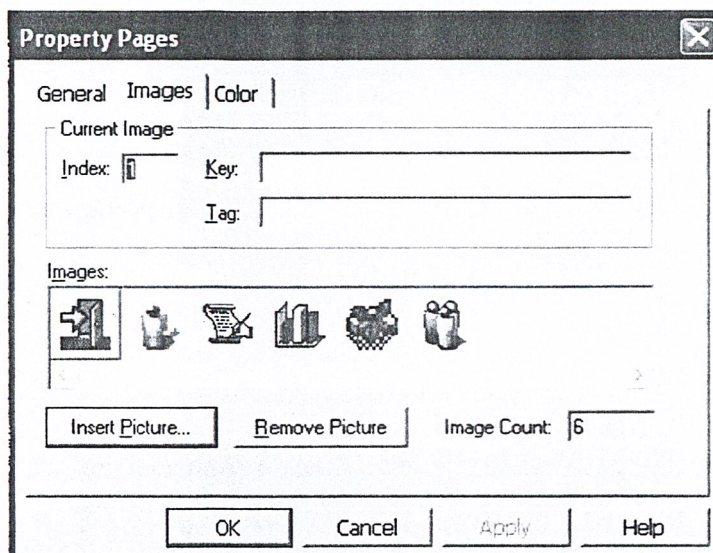
ภาพที่ 3.58 แสดงการเพิ่มฟอร์มชนิด MDI เข้ามาใน VBIDE

2. ให้เพิ่มคอนโทรล ToolBar1, ImageList1, StatusBar1 และคอนโทรล Timer1 ไปที่ frmMain ดังรูป



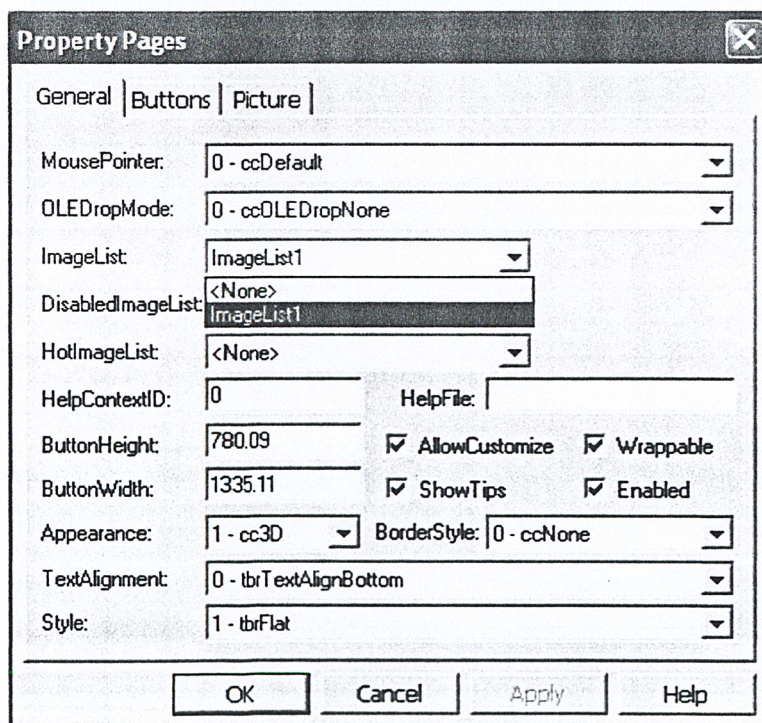
ภาพที่ 3.59 แสดงฟอร์มชนิด MDI ที่ถูกเพิ่มเข้ามาใน VBIDE

3. ให้คลิกขวาที่คอนโทรล ImageList1 แล้วเลือกคำสั่ง Properties จากนั้นคลิกที่แถบ Images ให้คลิกปุ่ม Insert Picture เพื่อเก็บรูปภาพไว้ในคอนโทรล ImageList1 ก่อน ในกรณีที่ ต้องการสร้างแถบทูลบาร์ เพื่อแทนเมนู 6 เมนู จึงเลือกใช้รูปภาพ 6 รูป



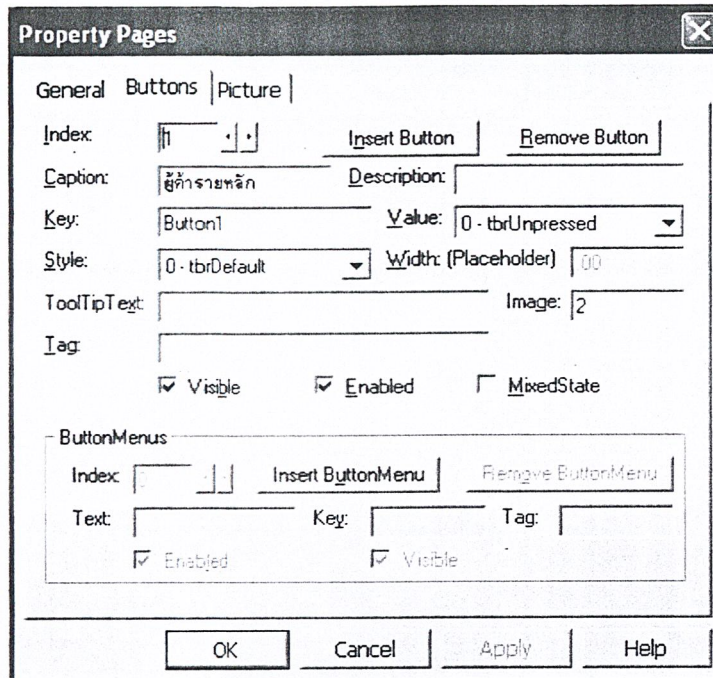
ภาพที่ 3.60 แสดงหลังจากใส่รูปภาพให้กับคอนโทรล ImageList

4. ให้คลิกขวาที่คอนโทรล ToolBar1 แล้วเลือกคำสั่ง Properties ที่ช่อง ImageList: ให้เลือกใช้รูปจากคอนโทรล ImageList1



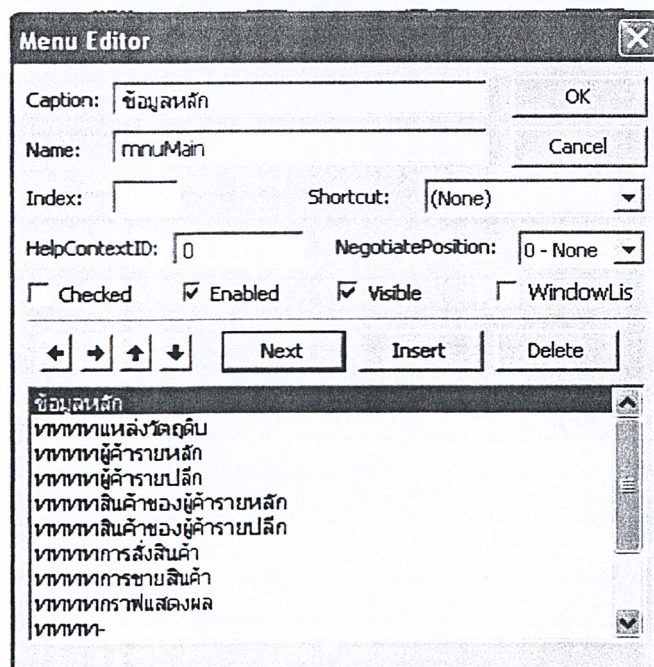
ภาพที่ 3.61 แสดงการกำหนดคอนโทรล ImageList1 ให้กับคอนโทรล ToolBar1

5. ที่แท็บ Buttons ให้เพิ่มปุ่มเข้าไป 6 ปุ่ม โดยการคลิกปุ่ม Insert Button ในแต่ละปุ่มให้เขียนข้อความกำกับไว้ในช่อง Caption: พร้อมกับเลือกรูปภาพที่ต้องการใช้ที่ช่อง Image: และที่ช่อง Key: ให้ตั้งชื่อให้กับปุ่มด้วย



ภาพที่ 3.62 แสดงการกำหนดจำนวนปุ่มที่จะปรากฏในคอนโทรล ToolBar1

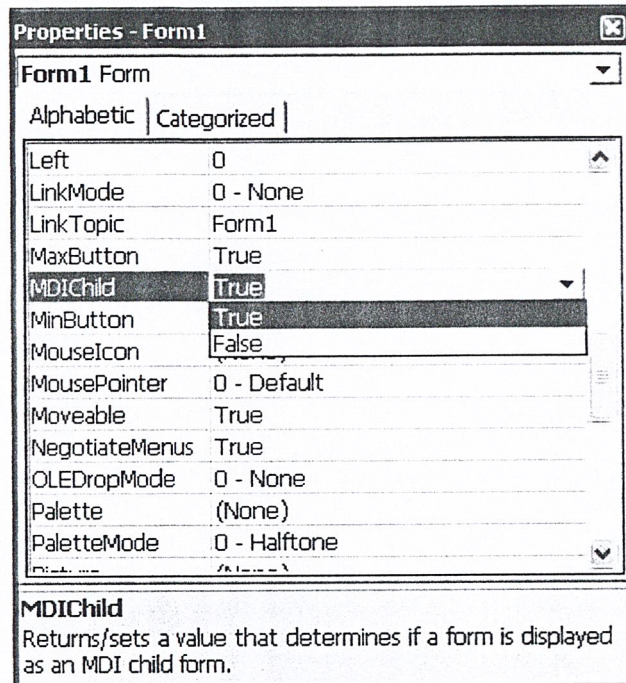
6. เพิ่มรายละเอียดให้ฟอร์ม โดยการสร้างเมนูบนฟอร์มด้วย Menu Editor สร้างเมนูดังภาพที่ 3.63 เพื่อใช้เรียกฟอร์มแต่ละฟอร์มทำงาน



ภาพที่ 3.63 แสดงการสร้างเมนูบนฟอร์มด้วย Menu Editor

หลังจากที่สร้างเมนู และทูลบาร์เรียบร้อยแล้วในส่วนต่อไปก็จะเริ่มเข้าสู่การเขียนโค้ด ให้โปรแกรมที่เราสร้างทำงานตามที่ต้องการ

7. สร้างฟอร์มปกติหรือฟอร์มลูกขึ้นมา โดยการเลือกที่ Properties ให้ MDIChild เป็น True เพื่อให้เป็นฟอร์มลูกของฟอร์มหลักที่สร้างขึ้น



ภาพที่ 3.64 แสดงการสร้างฟอร์มลูกโดยการเลือกที่ Properties

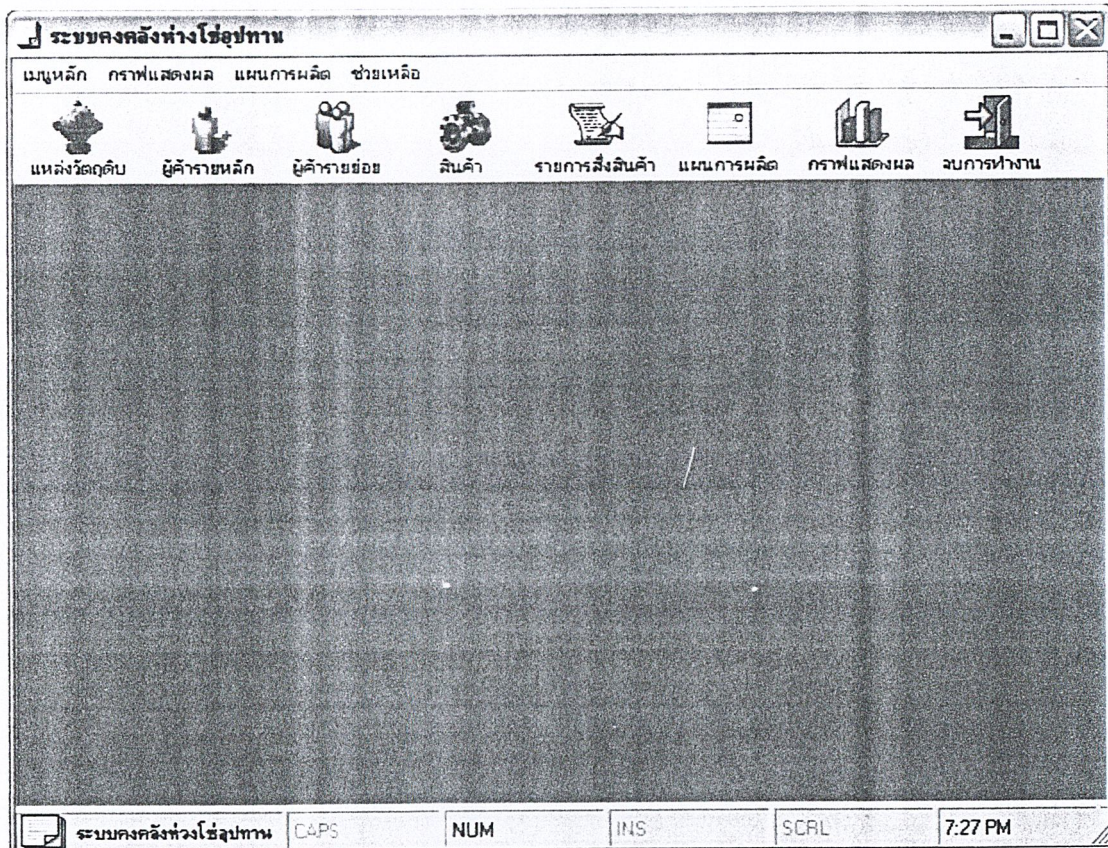
8. จากนั้นจึงเขียน Code การทำงานตามที่ต้องการภายในฟอร์มโดยแล้วแต่การทำงาน ถ้ามีฟอร์มหลักแล้ว เราก็สร้างฟอร์มลูกเพื่อเป็นฟอร์มย่อยให้กับฟอร์มหลักที่จะใช้ทำงานร่วมด้วย

# บทที่ 4

## การทดลองโครงการ

### 4.1 การทดสอบโปรแกรม

ในการทำงานของโปรแกรมมีการทำงานดังนี้  
เมื่อทำการรัน โปรแกรมที่เขียนไว้แล้วจะเป็นดังภาพที่ 4.1



ภาพที่ 4.1 แสดงหน้าจอหลัก

จากหน้าจอหลักจะมีหน้าต่างย่อย คือ แหล่งวัตถุดิบ ผู้ค้ารายหลัก ผู้ค้ารายย่อย สินค้า รายการสิ่งสินค้า แผนการผลิต กราฟแสดงผลและออกจากโปรแกรมโดยสามารถทำการเปลี่ยนแปลงข้อมูลได้ตลอดเวลาและนำไปเก็บไว้ในฐานข้อมูล

ระบบหลังห้องไอศูรียน (แหล่งวัตถุดิบ)

เมนูหลัก กราฟแสดงผล แผนกการผลิต ช่วยเหลือ

แหล่งวัตถุดิบ ผู้ขายหลัก ผู้ขายสำรอง สินค้า รายการสั่งซื้อสินค้า แผนกการผลิต กราฟแสดงผล งานประจำวัน

รายละเอียด

รหัสทั่วไป

รหัส Supplier: 1

ชื่อของSupplier: บริษัท เจริญ ตรา จำกัด

การติดต่อ

ที่อยู่: เลขที่ 2100/1294-95 โครงการพนาอินเพชร ถนนรามคำแหง ซอยเอก บางกะปิ กรุงเทพฯ

รหัสไปรษณีย์: 10240

เบอร์โทรศัพท์: 02-7201060-5

FAX: 02-7201069

โทรสาร: -

E-Mail: asiadot@bwinfo.co.th

ปุ่ม: << < > >> + เพิ่ม - ลบ บันทึก แก้ไข Exit

ระบบหลังห้องไอศูรียน CAPS NUM INS SCRL 23.09 28/3/2547

ภาพที่ 4.2 แสดงรายละเอียดของแหล่งวัตถุดิบ

ระบบหลังห้องไอศูรียน (ผู้ผลิตวัตถุดิบ)

เมนูหลัก กราฟแสดงผล แผนกการผลิต ช่วยเหลือ

แหล่งวัตถุดิบ ผู้ขายหลัก ผู้ขายสำรอง สินค้า รายการสั่งซื้อสินค้า แผนกการผลิต กราฟแสดงผล งานประจำวัน

รายละเอียดการสั่งซื้อ

ลำดับที่สั่งวัตถุดิบ: 1

วันที่สั่งวัตถุดิบ: 1/1/2546

รหัสวัตถุดิบ: 1

รหัสของผู้ผลิตที่ส่ง: 1

จำนวนการสั่งซื้อ: 0

จำนวนที่ได้รับ: 100

รายการวัตถุดิบ/หน่วย: -

รายการรวม: -

จำนวนวัตถุดิบที่คงเหลือ: 0

ปุ่ม: << < > >> + เพิ่ม - ลบ บันทึก แก้ไข Exit

ระบบหลังห้องไอศูรียน CAPS NUM INS SCRL 0.09 29/3/2547

ภาพที่ 4.3 แสดงรายละเอียดของการสั่งซื้อวัตถุดิบของ Manufacturer

ระบบคลังพัสดุโรงพยาบาล - [วัตถุดิบจาก Supplier]

เมนูหลัก กราฟแสดงผล แผนการผลิต ข้อมูล

แหล่งวัตถุดิบ ผู้ค้ารายหลัก ผู้ค้ารายอื่น สินค้า รายการสั่งซื้อสินค้า แผนการผลิต กราฟแสดงผล ระบบท่าทาง

ข้อมูลทั่วไป

รหัสวัตถุดิบ:

ชื่อวัตถุดิบ:

รหัสเจ้าของวัตถุดิบ:

ปริมาณวัตถุดิบ

กำลังการผลิต:

จากวัตถุดิบ/หน่วย:

จำนวนของวัตถุดิบใน 1 หน่วยการผลิต:

ข้อมูล: 01/03/2547

เพิ่ม ลบ บันทึก แก้ไข Exit

ระบบคลังพัสดุโรงพยาบาล CAPS NUM INS SCRL 003 29/3/2547

ภาพที่ 4.4 แสดงรายละเอียดของวัตถุดิบของ Supplier

ระบบคลังพัสดุโรงพยาบาล - [ผู้ค้ารายหลัก]

เมนูหลัก กราฟแสดงผล แผนการผลิต ข้อมูล

แหล่งวัตถุดิบ ผู้ค้ารายหลัก ผู้ค้ารายอื่น สินค้า รายการสั่งซื้อสินค้า แผนการผลิต กราฟแสดงผล ระบบท่าทาง

ข้อมูลทั่วไป

รหัสผู้ค้ารายหลัก:

ชื่อโรงงาน:

การติดต่อ

ที่อยู่:

รหัสไปรษณีย์:

เบอร์โทรศัพท์:

FAX:

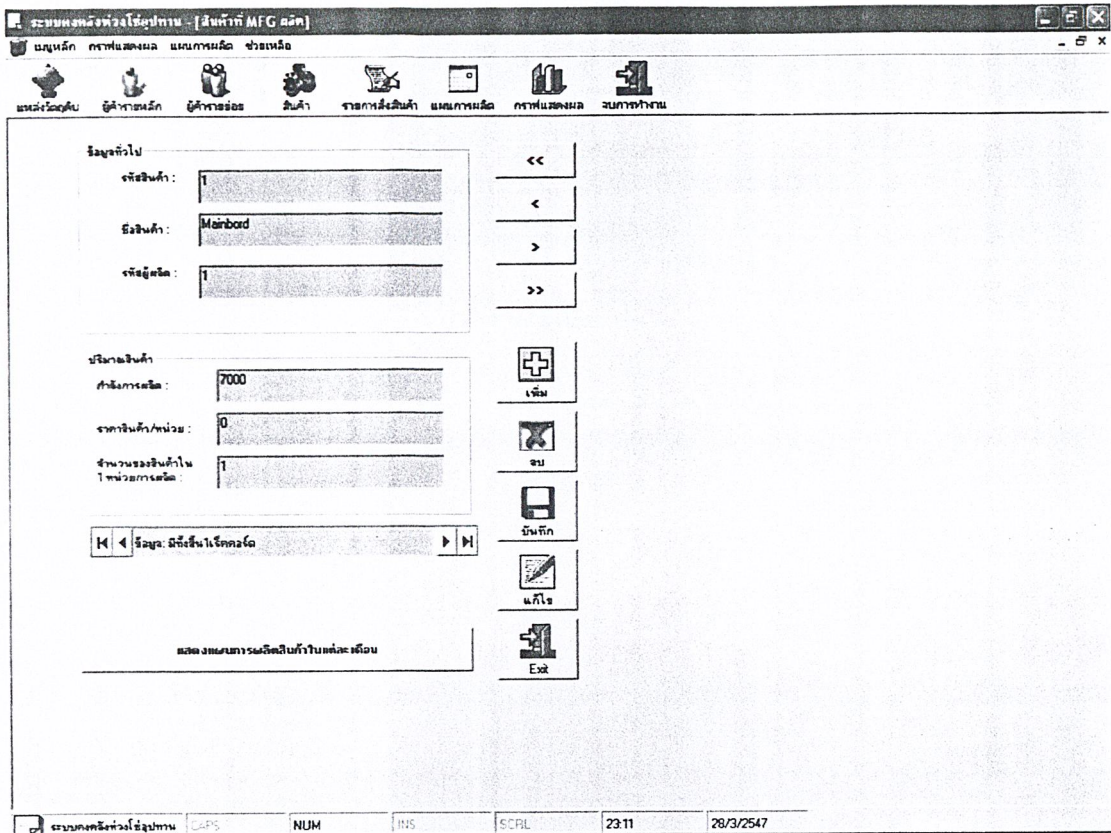
อีเมล:

ข้อมูล: 01/03/2547

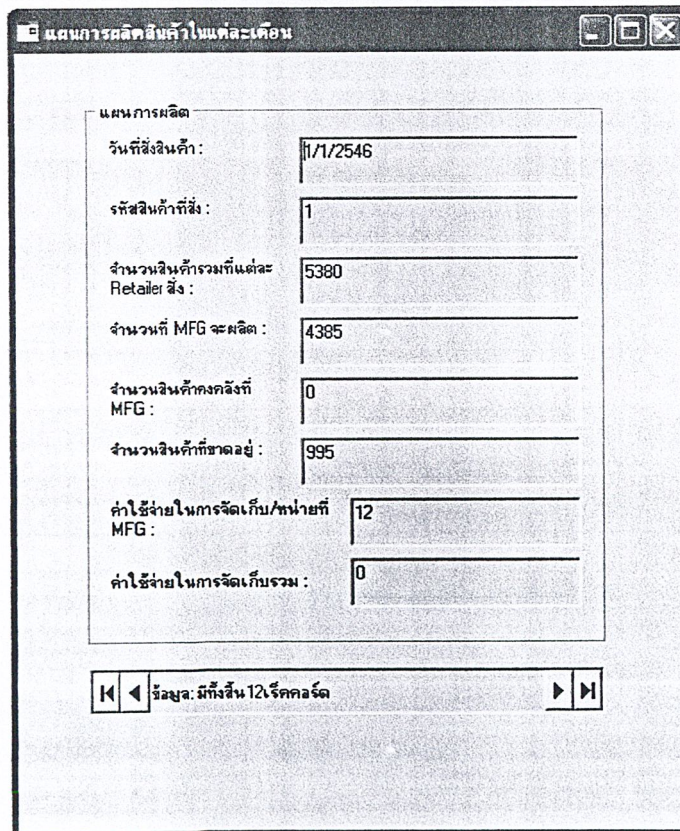
เพิ่ม ลบ บันทึก แก้ไข Exit

ระบบคลังพัสดุโรงพยาบาล CAPS NUM INS SCRL 2310 28/3/2547

ภาพที่ 4.5 แสดงรายละเอียดของผู้ค้ารายหลัก



ภาพที่ 4.6 แสดงรายละเอียดของสินค้าที่ผลิตโดย Manufacturer



ภาพที่ 4.7 แสดงข้อมูลของแผนการผลิตสินค้า

ระบบคงคลังห่วงโซ่อุปทาน - [ผู้ค้ารายย่อย]

เมนูหลัก กราฟแสดงผล แผนการמיד ผู้ช่วย

แหล่งข้อมูล ผู้ค้ารายเล็ก ผู้ค้ารายย่อย สินค้า รายการสั่งซื้อสินค้า แผนการמיד กราฟแสดงผล ระบบทำงาน

ข้อมูลทั่วไป

รหัสผู้ค้ารายย่อย: [ ]

ชื่อบริษัท: บริษัท เอสเอส เทคโนโลยี จำกัด

การติดต่อ

ที่อยู่: เขต 1 อาคารอินวิสิคธ ซอยประชา  
ราษฎร์บำรุงใหญ่ 5 ซอยราง กรุงเทพฯ

รหัสไปรษณีย์: 10320

เบอร์โทรศัพท์: 02-6902902

FAX: 02-6902719

อีเมล: ess-sytech@thai.com

ปุ่ม: เพิ่ม, ลบ, บันทึก, แก้ไข, Exit

สถานะ: << < > >>

สถานะ: << < > >> | ข้อมูล: มีทั้งสิ้น 5 รายการ

ระบบคงคลังห่วงโซ่อุปทาน CAPS NUM INS SCRL 23:10 28/3/2547

ภาพที่ 4.8 แสดงข้อมูลของผู้ค้ารายย่อย

ระบบคงคลังห่วงโซ่อุปทาน - [RetailerOrder]

เมนูหลัก กราฟแสดงผล แผนการמיד ผู้ช่วย

แหล่งข้อมูล ผู้ค้ารายเล็ก ผู้ค้ารายย่อย สินค้า รายการสั่งซื้อสินค้า แผนการמיד กราฟแสดงผล ระบบทำงาน

ข้อมูลการสั่งซื้อ

ลำดับการสั่งซื้อ: 1 <<

วันที่สั่งซื้อ: 1/1/2546 <

รหัสสินค้าที่สั่งซื้อ: 1 >

รหัสลูกค้าที่สั่งซื้อ: 1 >>

จำนวนการสั่งซื้อ: 756

วันที่รับสินค้า: 15/1/2546

จำนวนที่ไ้รับ: 756

จำนวนสินค้าที่ขาด: [ ]

ราคาสินค้า/หน่วย: [ ]

ราคารวม: [ ]

ค่าปรับ/หน่วยเมื่อสินค้าขาดมี: 30

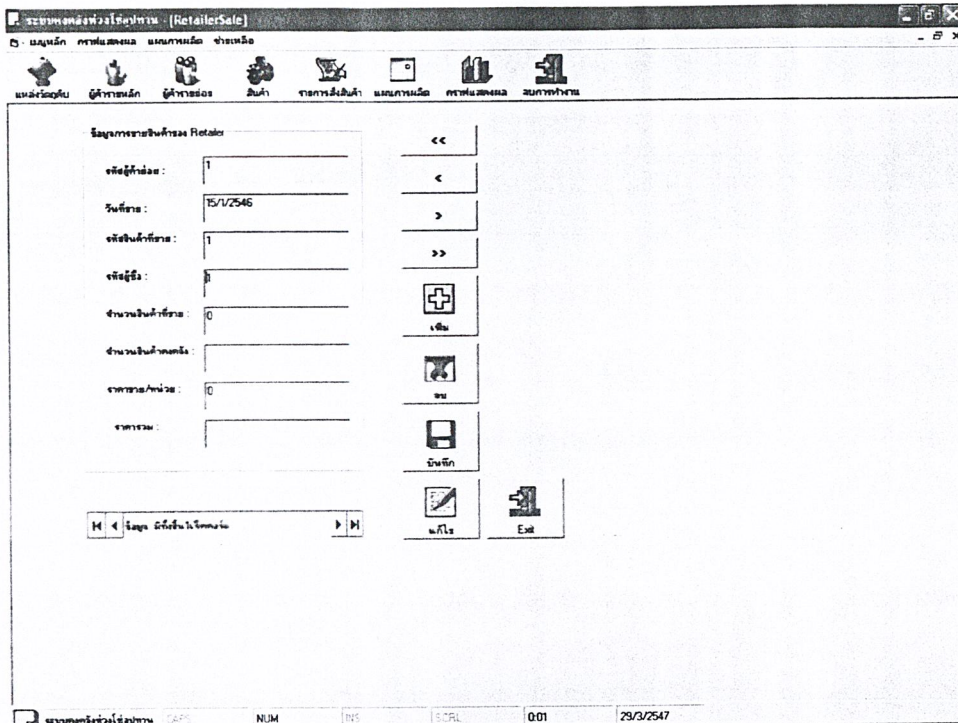
ค่าไ้จ่ายเมื่อสินค้าขาดมี: 0

ปุ่ม: เพิ่ม, ลบ, บันทึก, แก้ไข, Exit

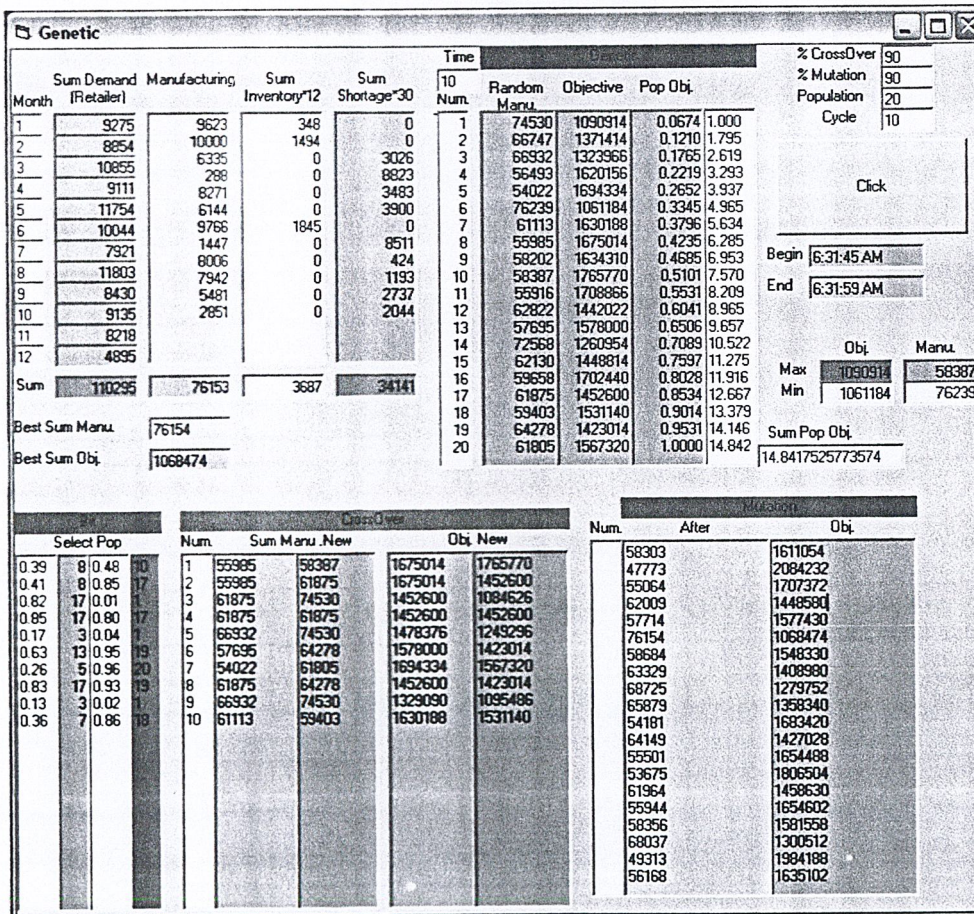
สถานะ: << < > >> | ข้อมูล: มีทั้งสิ้น 60 รายการ

ระบบคงคลังห่วงโซ่อุปทาน CAPS NUM INS SCRL 0:16 29/3/2547

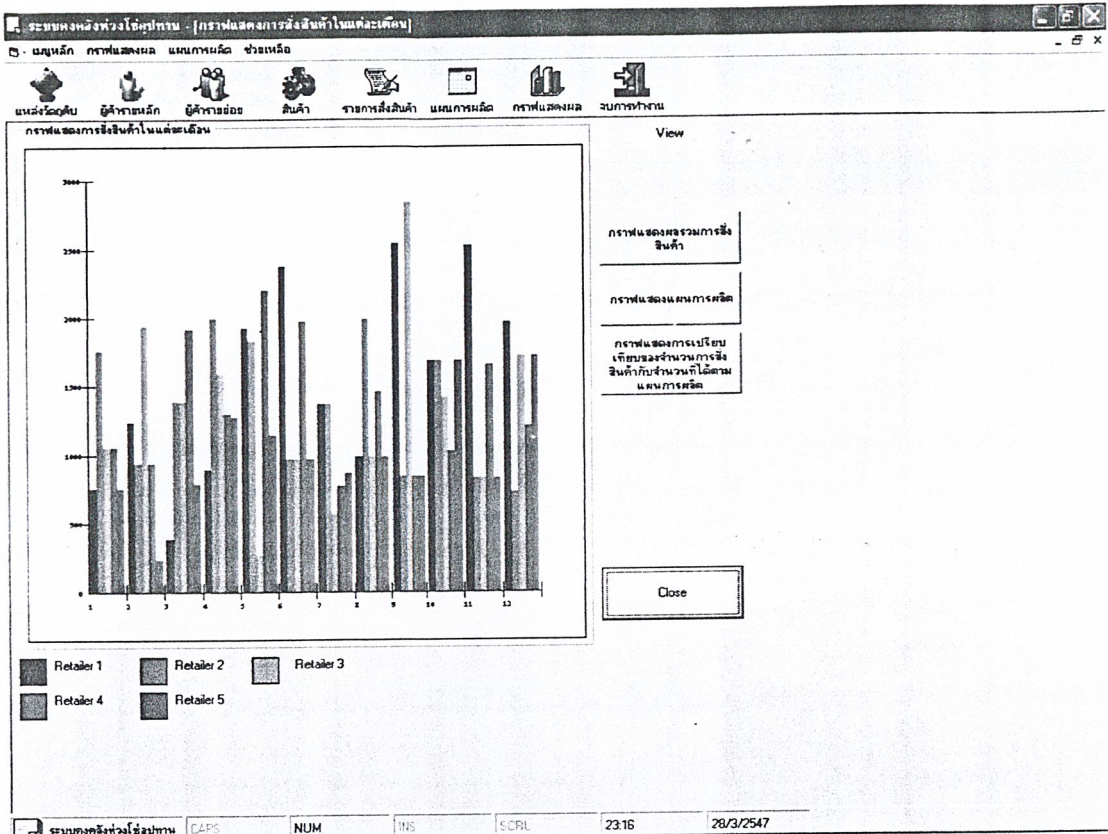
ภาพที่ 4.9 แสดงรายละเอียดของการสั่งซื้อสินค้าแต่ละชนิด



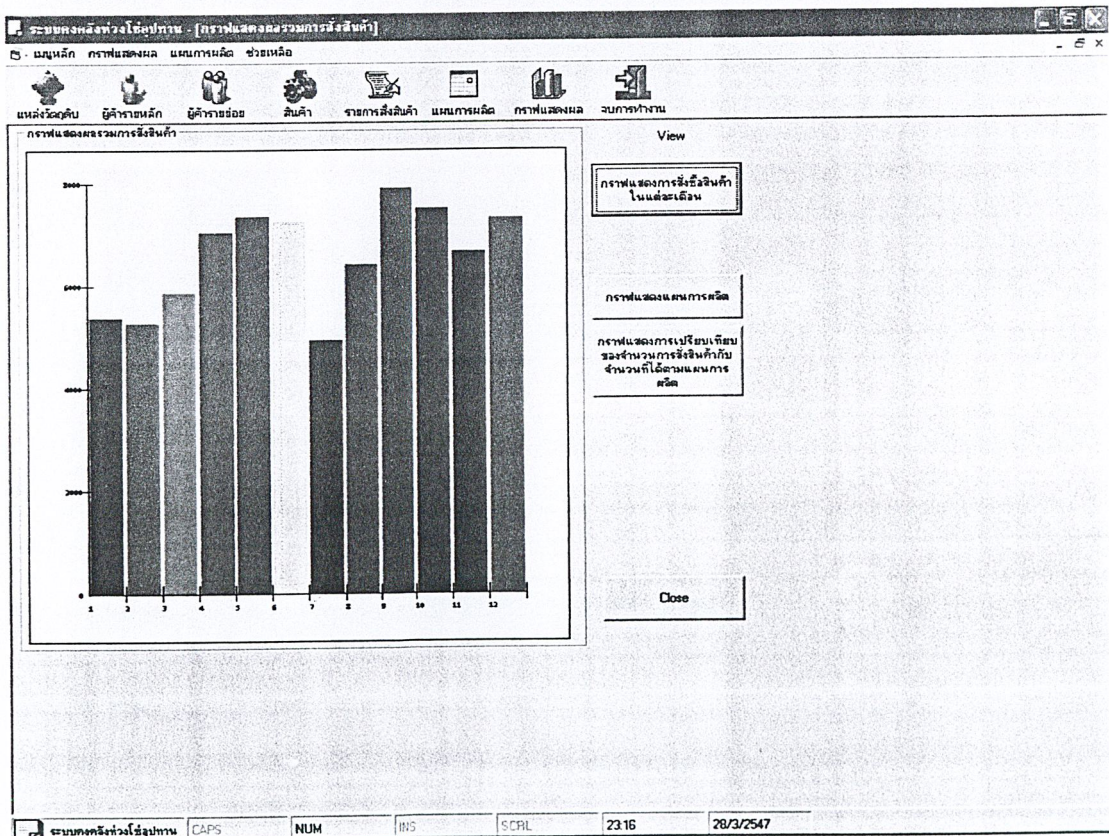
ภาพที่ 4.10 แสดงการขายสินค้าของผู้ค้ารายย่อยแต่ละคน



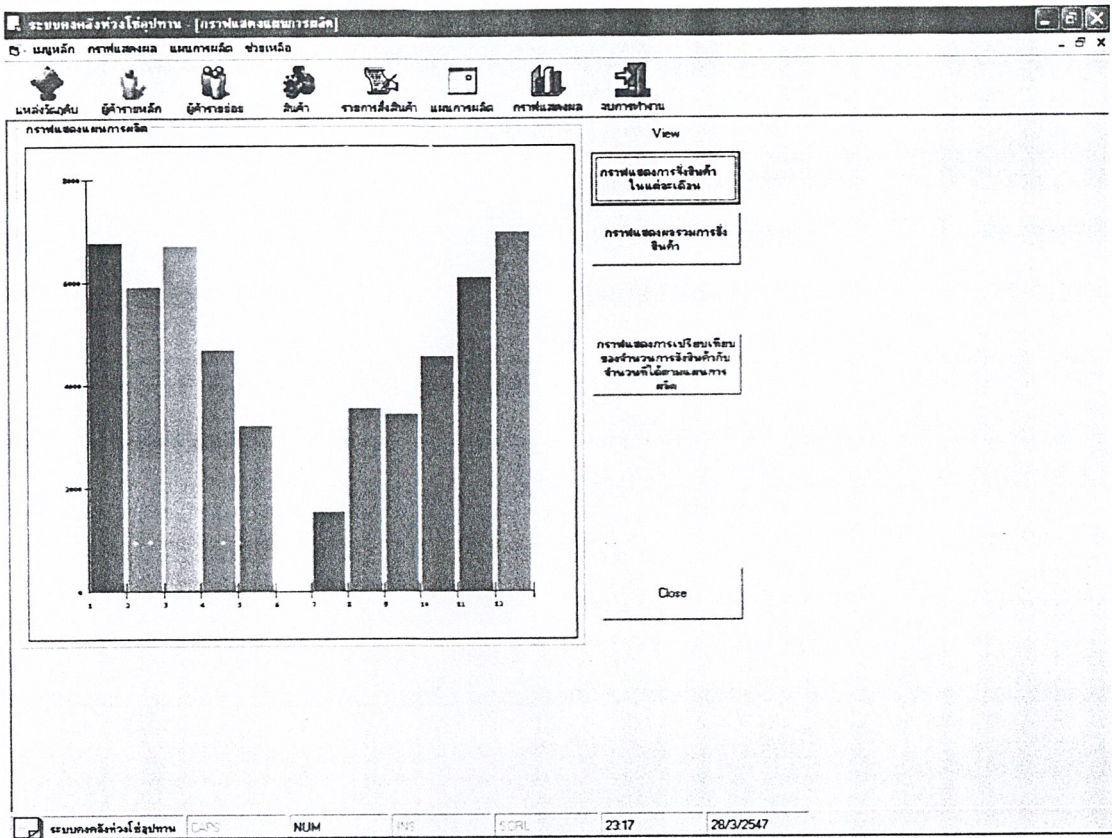
ภาพที่ 4.11 แสดงการรัน โปรแกรมการคำนวณบนหลักการของ Genetic Local Search Algorithm



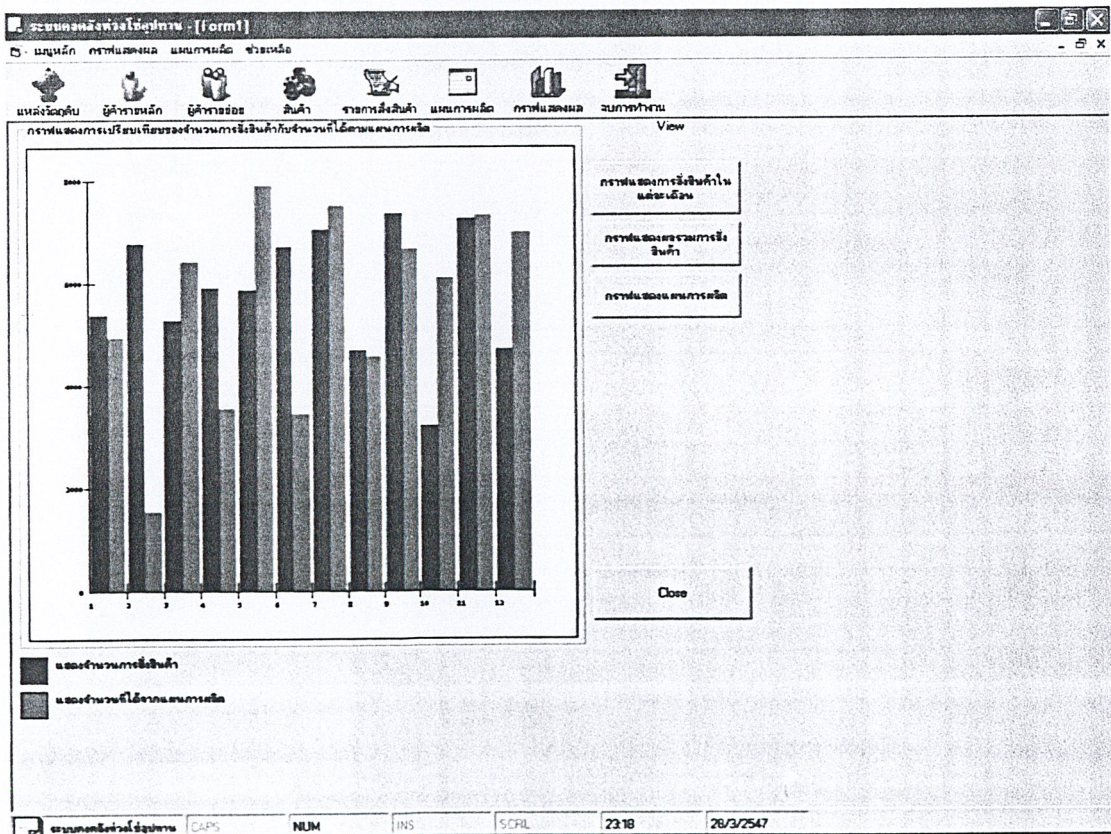
ภาพที่ 4.12 กราฟแสดงการสั่งซื้อสินค้าในแต่ละเดือน



ภาพที่ 4.13 กราฟแสดงผลรวมการสั่งซื้อสินค้าในแต่ละเดือน



ภาพที่ 4.14 กราฟแสดงแผนการผลิตสินค้า



ภาพที่ 4.15 กราฟแสดงการเปรียบเทียบระหว่างปริมาณการสั่งซื้อสินค้าและการได้รับสินค้า

เมื่อทำการรันโปรแกรมแล้วสามารถแก้ไขข้อมูลได้ โดยโปรแกรมจะเก็บรายละเอียดต่างๆ เอาไว้ในฐานข้อมูลโดยอัตโนมัติ แล้วสามารถแสดงจำนวนต่างๆ เป็นกราฟตามการเปลี่ยนแปลงของข้อมูลได้

ในส่วนของ การคำนวณของ Algorithm นั้นสามารถคำนวณได้โดยถ้าอยากได้ผลมีความใกล้เคียงกับค่าที่ดีที่สุด จะต้องกำหนดจำนวนรอบที่มากขึ้น แต่มีข้อเสียคือใช้เวลาในการคำนวณที่นานมากขึ้น ผลของการคำนวณที่ได้นั้นสามารถเก็บอยู่ในฐานข้อมูล แล้วนำมาเปรียบเทียบกับจำนวนที่ถูกค่าตั้งต้นค่าทั้งหมดในแต่ละเดือน ดังกราฟที่แสดงในภาพที่ 4.15

## บทที่ 5

# สรุปการทำงานของโครงการและแนวทางการพัฒนา

### 5.1 สรุปการทำงานของโครงการ

1. เมื่อมีการเปลี่ยนแปลงข้อมูลต่างๆ บน หน้าจอของ Visual Basic ผ่าน Data Control มีการเปลี่ยนแปลงของฐานข้อมูลถูกต้อง
2. การทำงานของการคำนวณบนหลักการ Genetic Local Search Algorithm โดยนำมาจากฐานข้อมูล สามารถคำนวณได้อย่างถูกต้อง
3. สามารถใช้ฐานข้อมูลของระบบ นำไปวางแผนการผลิตที่เหมาะสมได้

### 5.2 แนวทางการพัฒนา

1. เนื่องจากการสั่งซื้อสินค้าโดยการป้อนข้อมูลในการสั่งซื้อ หรือขาย ตลอดเวลา อาจจะนำ Barcode มาใช้ในระบบการผลิตเพื่อที่จะให้อ่านข้อมูลของสินค้าแต่ละตัวได้ในทันที ทำให้ระบบงานมีความรวดเร็วขึ้น
2. เนื่องจากการเขียนโปรแกรมในการคำนวณบนหลักการ Genetic Local Search Algorithm นี้จะใช้เวลานานในการคำนวณ ซึ่งถ้ามีระบบที่ใหญ่ขึ้น เวลาในการคำนวณหาจำนวนการผลิตที่เหมาะสมก็จะใช้เวลานานมากขึ้นอีก
3. จะเป็นการดีหากสามารถนำเอาระบบ Internet มาใช้ดูแลระบบการผลิต และตรวจสอบจำนวนสินค้าคงคลังได้ตลอดเวลา

## บรรณานุกรม

- [1] ดร. พิชิต สุขเจริญพงษ์ “การจัดการวิศวกรรมการผลิต” บริษัท ซีอีคยูเคชั่น จำกัด พ.ศ. 2543
- [2] กิตติ ภัคดีวัฒนะกุล และ จำลอง ทรูอุตสาหะ “Visual Basic 6 ฉบับโปรแกรมเมอร์” บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด พ.ศ.2545
- [3] สุภชัย สมพานิช “สร้างระบบงานฐานข้อมูลด้วย Visual Basic ฉบับโปรแกรมเมอร์” สำนักพิมพ์ อินโฟเพรส พ.ศ. 2545

ภาคผนวก

## 1. frmMain

```
Private Sub mnuExit_Click()
```

```
End
```

```
End Sub
```

```
Private Sub mnuG1_Click()
```

```
frmG1.Show
```

```
End Sub
```

```
Private Sub mnuG2_Click()
```

```
frmG3.Show
```

```
End Sub
```

```
Private Sub mnuG3_Click()
```

```
frmG2.Show
```

```
End Sub
```

```
Private Sub mnuG4_Click()
```

```
frmG4.Show
```

```
End Sub
```

```
Private Sub mnuGenetic_Click()
```

```
frmGenetic.Show
```

```
End Sub
```

```
Private Sub mnuMFG_Click()
```

```
frmMFG.Show
```

```
End Sub
```

```
Private Sub mnuMFGOrder_Click()
```

```
frmMfgOrder.Show
```

```
End Sub
```

```
Private Sub mnuProductMFG_Click()
```

```
frmProductMFG.Show
```

```
End Sub
```

```
Private Sub mnuProductSupplier_Click()
```

```
frmProductSupplier.Show
```

```
End Sub
```

```
Private Sub mnuRetailer_Click()
```

```
frmRetailer.Show
```

```
End Sub
```

```
Private Sub mnuRetailerOrder_Click()
```

```
frmRetailerOrder.Show
```

```
End Sub
```

```
Private Sub mnuRetailerSale_Click()
```

```
frmRetailerSale.Show
```

```
End Sub
```

```
Private Sub mnuSupplier_Click()
```

```
frmSupplier.Show
```

```
End Sub
```

```
Private Sub mnuWeb_Click()
```

```
Shell "Start.Exe http://msdn.microsoft.com", vbHide
```

```
End Sub
```

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
```

```
Select Case Button.Key
```

```
Case "Button1"
```

```
frmSupplier.Show
```

Case "Button2"

frmMFG.Show

Case "Button3"

frmRetailer.Show

Case "Button4"

MsgBox "ข้อมูลสินค้า"

frmProductMFG.Show

Case "Button5"

frmRetailerOrder.Show

Case "Button6"

MsgBox "จำนวนแผนการผลิต"

frmGenetic.Show

Case "Button7"

MsgBox "กราฟแสดงผล"

frmGraph.Show

Case "Button8"

MsgBox "ออกจากโปรแกรม"

End

End Select

End Sub

## 2. frmSupplier

Private Sub cmdAdd\_Click()

DatSupplier.Recordset.AddNew

Text1.SetFocus

End Sub

Private Sub cmdDelete\_Click()

Dim str As String

str = "ต้องการลบรายการนี้?"

If MsgBox(str, vbQuestion + vbYesNo + vbDefaultButton2) = vbYes Then

DatSupplier.Recordset.Delete

```
DatSupplier.Recordset.MoveNext
If DatSupplier.Recordset.EOF = True Then
    DatSupplier.Recordset.MoveLast
End If
End If
End Sub
```

```
Private Sub cmdExit_Click()
    Unload Me
End Sub
```

```
Private Sub cmdMoveFirst_Click()
    DatSupplier.Recordset.MoveFirst
End Sub
```

```
Private Sub cmdMoveLast_Click()
    DatSupplier.Recordset.MoveLast
End Sub
```

```
Private Sub cmdMoveNext_Click()
    With DatSupplier.Recordset
        .MoveNext
        If .EOF Then
            .MoveFirst
        End If
    End With
End Sub
```

```
Private Sub cmdMovePrevious_Click()
    With DatSupplier.Recordset
        .MovePrevious
        If .BOF Then
```

```
        .MoveLast
    End If
End With
End Sub
```

```
Private Sub cmdRefresh_Click()
    DatSupplier.UpdateControls
    If DatSupplier.Recordset.EditMode = dbEditAdd Then
        DatSupplier.Recordset.CancelUpdate
    End If
End Sub
```

```
Private Sub cmdUpdate_Click()
    DatSupplier.UpdateRecord
    DatSupplier.Recordset.Bookmark = DatSupplier.Recordset.LastModified
End Sub
```

```
Private Sub Form_Load()
    With DatSupplier
        .Refresh
        .Recordset.MoveLast
        .Caption = .Caption & ": มีทั้งสิ้น" & .Recordset.RecordCount & "เร็กคอร์ด'"
        .Recordset.MoveFirst
    End With
End Sub
```

### 3. frmRetailerSale

```
Private Sub cmdAdd_Click()
    DatRetailerSale.Recordset.AddNew
    Text1.SetFocus
End Sub
```

```
Private Sub cmdDelete_Click()
    Dim str As String
    str = "ต้องการลบรายการนี้?"
    If MsgBox(str, vbQuestion + vbYesNo + vbDefaultButton2) = vbYes Then
        DatRetailerSale.Recordset.Delete
        DatRetailerSale.Recordset.MoveNext
        If DatRetailerSale.Recordset.EOF = True Then
            DatRetailerSale.Recordset.MoveLast
        End If
    End If
End Sub
```

```
Private Sub cmdExit_Click()
    Unload Me
End Sub
```

```
Private Sub cmdMoveFirst_Click()
    DatRetailerSale.Recordset.MoveFirst
End Sub
```

```
Private Sub cmdMoveLast_Click()
    DatRetailerSale.Recordset.MoveLast
End Sub
```

```
Private Sub cmdMoveNext_Click()
    With DatRetailerSale.Recordset
        .MoveNext
        If .EOF Then
            .MoveFirst
        End If
    End With
End Sub
```

```
Private Sub cmdMovePrevious_Click()
```

```
    With DatRetailerSale.Recordset
```

```
        .MovePrevious
```

```
    If .BOF Then
```

```
        .MoveLast
```

```
    End If
```

```
End With
```

```
End Sub
```

```
Private Sub cmdRefresh_Click()
```

```
    DatRetailerSale.UpdateControls
```

```
    If DatRetailerSale.Recordset.EditMode = dbEditAdd Then
```

```
        DatRetailerSale.Recordset.CancelUpdate
```

```
    End If
```

```
End Sub
```

```
Private Sub cmdUpdate_Click()
```

```
    DatRetailerSale.UpdateRecord
```

```
    DatRetailerSale.Recordset.Bookmark = DatRetailerSale.Recordset.LastModified
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    With DatRetailerSale
```

```
        .Refresh
```

```
        .Recordset.MoveLast
```

```
        .Caption = .Caption & ": มีทั้งสิ้น" & .Recordset.RecordCount & "เรี็กคอร์ด"
```

```
        .Recordset.MoveFirst
```

```
    End With
```

```
End Sub
```

#### 4. frmRetailerOrder

```
Private Sub cmdAdd_Click()
```

```
DatRetailerOrder.Recordset.AddNew
Text1.SetFocus
End Sub
```

```
Private Sub cmdDelete_Click()
    Dim str As String
    str = "ต้องการลบรายการนี้?"
    If MsgBox(str, vbQuestion + vbYesNo + vbDefaultButton2) = vbYes Then
        DatRetailerOrder.Recordset.Delete
        DatRetailerOrder.Recordset.MoveNext
        If DatRetailerOrder.Recordset.EOF = True Then
            DatRetailerOrder.Recordset.MoveLast
        End If
    End If
End Sub
```

```
Private Sub cmdExit_Click()
    Unload Me
End Sub
```

```
Private Sub cmdMoveFirst_Click()
    DatRetailerOrder.Recordset.MoveFirst
End Sub
```

```
Private Sub cmdMoveLast_Click()
    DatRetailerOrder.Recordset.MoveLast
End Sub
```

```
Private Sub cmdMoveNext_Click()
    With DatRetailerOrder.Recordset
        .MoveNext
    End With
    If .EOF Then
```

```
        .MoveFirst
    End If
End With
End Sub
```

```
Private Sub cmdMovePrevious_Click()
    With DatRetailerOrder.Recordset
        .MovePrevious
        If .BOF Then
            .MoveLast
        End If
    End With
End Sub
```

```
Private Sub cmdRefresh_Click()
    DatRetailerOrder.UpdateControls
    If DatRetailerOrder.Recordset.EditMode = dbEditAdd Then
        DatRetailerOrder.Recordset.CancelUpdate
    End If
End Sub
```

```
Private Sub cmdUpdate_Click()
    DatRetailerOrder.UpdateRecord
    DatRetailerOrder.Recordset.Bookmark = DatRetailerOrder.Recordset.LastModified
End Sub
```

```
Private Sub DatRetailerOrder_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    Text12.Text = Val(Text5.Text) - Val(Text6.Text)
    Text8.Text = Val(Text6.Text) * Val(Text7.Text)
    Text10.Text = Val(Text12.Text) * Val(Text9.Text)
End Sub
```

```

Private Sub Form_Load()
    With DatRetailerOrder
        .Refresh
        .Recordset.MoveLast
        .Caption = .Caption & ": มีทั้งสิ้น" & .Recordset.RecordCount & "เรีคคอร์ด'"
        .Recordset.MoveFirst
    End With
End Sub

```

## 5. frmRetailer

```

Private Sub cmdAdd_Click()
    DatRetailer.Recordset.AddNew
    Text1.SetFocus
End Sub

```

```

Private Sub cmdDelete_Click()
    Dim str As String
    str = "ต้องการลบรายการนี้?"

    If MsgBox(str, vbQuestion + vbYesNo + vbDefaultButton2) = vbYes Then
        DatRetailer.Recordset.Delete
        DatRetailer.Recordset.MoveNext
        If DatRetailer.Recordset.EOF = True Then
            DatRetailer.Recordset.MoveLast
        End If
    End If
End Sub

```

```

Private Sub cmdExit_Click()
    Unload Me
End Sub

```

```
Private Sub cmdMoveFirst_Click()  
    DatRetailer.Recordset.MoveFirst  
End Sub
```

```
Private Sub cmdMoveLast_Click()  
    DatRetailer.Recordset.MoveLast  
End Sub
```

```
Private Sub cmdMoveNext_Click()  
    With DatRetailer.Recordset  
        .MoveNext  
        If .EOF Then  
            .MoveFirst  
        End If  
    End With  
End Sub
```

```
Private Sub cmdMovePrevious_Click()  
    With DatRetailer.Recordset  
        .MovePrevious  
        If .BOF Then  
            .MoveLast  
        End If  
    End With  
End Sub
```

```
Private Sub cmdRefresh_Click()  
    DatRetailer.UpdateControls  
    If DatRetailer.Recordset.EditMode = dbEditAdd Then  
        DatRetailer.Recordset.CancelUpdate  
    End If  
End Sub
```

```
Private Sub cmdUpdate_Click()  
    DatRetailer.UpdateRecord  
    DatRetailer.Recordset.Bookmark = DatRetailer.Recordset.LastModified  
End Sub
```

```
Private Sub Form_Load()  
    With DatRetailer  
        .Refresh  
        .Recordset.MoveLast  
        .Caption = .Caption & ": มีทั้งสิ้น" & .Recordset.RecordCount & "เรี็กคอร์ด"  
        .Recordset.MoveFirst  
    End With  
End Sub
```

## 6. frmProductSupplier

```
Private Sub cmdAdd_Click()  
    DatProductSupplier.Recordset.AddNew  
    Text1.SetFocus  
End Sub
```

```
Private Sub cmdDelete_Click()  
    Dim str As String  
    str = "ต้องการลบรายการนี้?"  
  
    If MsgBox(str, vbQuestion + vbYesNo + vbDefaultButton2) = vbYes Then  
        DatProductSupplier.Recordset.Delete  
        DatProductSupplier.Recordset.MoveNext  
        If DatProductSupplier.Recordset.EOF = True Then  
            DatProductSupplier.Recordset.MoveLast  
        End If  
    End If  
End Sub
```

```
Private Sub cmdExit_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub cmdMoveFirst_Click()
```

```
    DatProductSupplier.Recordset.MoveFirst
```

```
End Sub
```

```
Private Sub cmdMoveLast_Click()
```

```
    DatProductSupplier.Recordset.MoveLast
```

```
End Sub
```

```
Private Sub cmdMoveNext_Click()
```

```
    With DatProductSupplier.Recordset
```

```
        .MoveNext
```

```
        If .EOF Then
```

```
            .MoveFirst
```

```
        End If
```

```
    End With
```

```
End Sub
```

```
Private Sub cmdMovePrevious_Click()
```

```
    With DatProductSupplier.Recordset
```

```
        .MovePrevious
```

```
        If .BOF Then
```

```
            .MoveLast
```

```
        End If
```

```
    End With
```

```
End Sub
```

```
Private Sub cmdRefresh_Click()
```

```
    DatProductSupplier.UpdateControls
```

```
If DatProductSupplier.Recordset.EditMode = dbEditAdd Then
    DatProductSupplier.Recordset.CancelUpdate
End If
End Sub
```

```
Private Sub cmdUpdate_Click()
    DatProductSupplier.UpdateRecord
    DatProductSupplier.Recordset.Bookmark = DatProductSupplier.Recordset.LastModified
End Sub
```

```
Private Sub Form_Load()
    With DatProductSupplier
        .Refresh
        .Recordset.MoveLast
        .Caption = .Caption & ": มีทั้งสิ้น" & .Recordset.RecordCount & "เร็กคอร์ด'"
        .Recordset.MoveFirst
    End With
End Sub
```

## 7. frmProductMFG

```
Private Sub cmdAdd_Click()
    DatProductMFG.Recordset.AddNew
    Text1.SetFocus
End Sub
```

```
Private Sub cmdDelete_Click()
    Dim str As String
    str = "ต้องการลบรายการนี้?"
    If MsgBox(str, vbQuestion + vbYesNo + vbDefaultButton2) = vbYes Then
        DatProductMFG.Recordset.Delete
        DatProductMFG.Recordset.MoveNext
        If DatProductMFG.Recordset.EOF = True Then
```

```
DatProductMFG.Recordset.MoveLast
```

```
End If
```

```
End If
```

```
End Sub
```

```
Private Sub cmdExit_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub cmdMoveFirst_Click()
```

```
DatProductMFG.Recordset.MoveFirst
```

```
End Sub
```

```
Private Sub cmdMoveLast_Click()
```

```
DatProductMFG.Recordset.MoveLast
```

```
End Sub
```

```
Private Sub cmdMoveNext_Click()
```

```
With DatProductMFG.Recordset
```

```
.MoveNext
```

```
If .EOF Then
```

```
.MoveFirst
```

```
End If
```

```
End With
```

```
End Sub
```

```
Private Sub cmdMovePrevious_Click()
```

```
With DatProductMFG.Recordset
```

```
.MovePrevious
```

```
If .BOF Then
```

```
.MoveLast
```

```
End If
```

```
End With
End Sub

Private Sub cmdRefresh_Click()
    DatProductMFG.UpdateControls
    If DatProductMFG.Recordset.EditMode = dbEditAdd Then
        DatProductMFG.Recordset.CancelUpdate
    End If
End Sub
```

```
Private Sub cmdUpdate_Click()
    DatProductMFG.UpdateRecord
    DatProductMFG.Recordset.Bookmark = DatProductMFG.Recordset.LastModified
End Sub
```

```
Private Sub Command1_Click()
    frmProductDetail.Show
End Sub
```

```
Private Sub Form_Load()
    With DatProductMFG
        .Refresh
        .Recordset.MoveLast
        .Caption = .Caption & ": มีทั้งหมด" & .Recordset.RecordCount & "เร็กคอร์ด"
        .Recordset.MoveFirst
    End With
End Sub
```

## 8. frmProductDetail

```
Private Sub cmdExit_Click()
    Unload Me
End Sub
```

```
Private Sub Data1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Text7.Text = Val(Text5.Text) * Val(Text8.Text)
End Sub
```

```
Private Sub Form_Load()
    With Data1
        .Refresh
        .Recordset.MoveLast
        .Caption = .Caption & ": มีทั้งสิ้น" & .Recordset.RecordCount & "เรีีกออर्ड'"
        .Recordset.MoveFirst
    End With
End Sub
```

## 9. frmMFGOrder

```
Private Sub cmdAdd_Click()
    DatMFGOrder.Recordset.AddNew
    Text1.SetFocus
End Sub
```

```
Private Sub cmdDelete_Click()
    Dim str As String
    str = "ต้องการลบรายการนี้?"

    If MsgBox(str, vbQuestion + vbYesNo + vbDefaultButton2) = vbYes Then
        DatMFGOrder.Recordset.Delete
        DatMFGOrder.Recordset.MoveNext
        If DatMFGOrder.Recordset.EOF = True Then
            DatMFGOrder.Recordset.MoveLast
        End If
    End If
End Sub
```

```
Private Sub cmdExit_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub cmdMoveFirst_Click()
```

```
    DatMFGOrder.Recordset.MoveFirst
```

```
End Sub
```

```
Private Sub cmdMoveLast_Click()
```

```
    DatMFGOrder.Recordset.MoveLast
```

```
End Sub
```

```
Private Sub cmdMoveNext_Click()
```

```
    With DatMFGOrder.Recordset
```

```
        .MoveNext
```

```
        If .EOF Then
```

```
            .MoveFirst
```

```
        End If
```

```
    End With
```

```
End Sub
```

```
Private Sub cmdMovePrevious_Click()
```

```
    With DatMFGOrder.Recordset
```

```
        .MovePrevious
```

```
        If .BOF Then
```

```
            .MoveLast
```

```
        End If
```

```
    End With
```

```
End Sub
```

```
Private Sub cmdRefresh_Click()
```

```
    DatMFGOrder.UpdateControls
```

```
If DatMFGOrder.Recordset.EditMode = dbEditAdd Then
```

```
    DatMFGOrder.Recordset.CancelUpdate
```

```
End If
```

```
End Sub
```

```
Private Sub cmdUpdate_Click()
```

```
    DatMFGOrder.UpdateRecord
```

```
    DatMFGOrder.Recordset.Bookmark = DatMFGOrder.Recordset.LastModified
```

```
End Sub
```

```
Private Sub DatMFGOrder_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As  
Single)
```

```
    Text8.Text = Val(Text6.Text) * Val(Text7.Text)
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    With DatMFGOrder
```

```
        .Refresh
```

```
        .Recordset.MoveLast
```

```
        .Caption = .Caption & ": มีทั้งหมด" & .Recordset.RecordCount & "เรี็กคอร์ด"
```

```
        .Recordset.MoveFirst
```

```
    End With
```

```
End Sub
```

## 10. frmMFG

```
Private Sub cmdAdd_Click()
```

```
    DatMFG.Recordset.AddNew
```

```
    Text1.SetFocus
```

```
End Sub
```

```
Private Sub cmdDelete_Click()
```

```
    Dim str As String
```

```
str = "ต้องการลบรายการนี้?"
```

```
If MsgBox(str, vbQuestion + vbYesNo + vbDefaultButton2) = vbYes Then
```

```
    DatMFG.Recordset.Delete
```

```
    DatMFG.Recordset.MoveNext
```

```
    If DatMFG.Recordset.EOF = True Then
```

```
        DatMFG.Recordset.MoveLast
```

```
    End If
```

```
End If
```

```
End Sub
```

```
Private Sub cmdMoveFirst_Click()
```

```
    DatMFG.Recordset.MoveFirst
```

```
End Sub
```

```
Private Sub cmdMoveLast_Click()
```

```
    DatMFG.Recordset.MoveLast
```

```
End Sub
```

```
Private Sub cmdMoveNext_Click()
```

```
    With DatMFG.Recordset
```

```
        .MoveNext
```

```
        If .EOF Then
```

```
            .MoveFirst
```

```
        End If
```

```
    End With
```

```
End Sub
```

```
Private Sub cmdMovePrevious_Click()
```

```
    With DatMFG.Recordset
```

```
        .MovePrevious
```

```
        If .BOF Then
```

```
            .MoveLast
```

```
End If
End With
End Sub
```

```
Private Sub cmdRefresh_Click()
    DatMFG.UpdateControls
    If DatMFG.Recordset.EditMode = dbEditAdd Then
        DatMFG.Recordset.CancelUpdate
    End If
End Sub
```

```
Private Sub cmdUpdate_Click()
    DatMFG.UpdateRecord
    DatMFG.Recordset.Bookmark = DatMFG.Recordset.LastModified
End Sub
```

```
Private Sub cmdExit_Click()
    Unload Me
End Sub
```

```
Private Sub Form_Load()
    With DatMFG
        .Refresh
        .Recordset.MoveLast
        .Caption = .Caption & ": มีทั้งสิ้น" & .Recordset.RecordCount & "เร็คคอร์ด'"
        .Recordset.MoveFirst
    End With
End Sub
```

## 11. frmGraph

```
Private Sub Command1_Click()
    frmG1.Show
```

End Sub

Private Sub Command2\_Click()

frmG2.Show

End Sub

Private Sub Command3\_Click()

frmG3.Show

End Sub

Private Sub Command4\_Click()

Unload Me

End Sub

Private Sub Command5\_Click()

frmG4.Show

End Sub

## **12. frmG1**

Private Sub Command1\_Click()

Unload Me

End Sub

Private Sub Command2\_Click()

frmG2.Show

End Sub

Private Sub Command3\_Click()

frmG3.Show

End Sub

Private Sub Command4\_Click()

frmG4.Show

End Sub

Private Sub Form\_Load()

Dim A, X, Y, Z As String

Dim B, C, D, O, R As Integer

Dim E, F, G, H, i, j As Integer

Dim K() As Integer

A = "select RetailerID, count(RetailerID) as count from RetailerOrder "

A = A & "Group by RetailerID"

Data1.RecordSource = A

Data1.Refresh

B = Data1.Recordset.RecordCount

X = "select RetailerID,UnitOnOrder from RetailerOrder "

Data2.RecordSource = X

Data2.Refresh

C = Data2.Recordset.RecordCount

D = C / B

ReDim K(1 To B, 1 To C) As Integer

With Data2.Recordset

    Data2.Refresh

    For i = 1 To B

        .MoveFirst

        R = 0

        For j = 1 To C

            O = .Fields("RetailerID").Value

            If O = i Then

                R = R + 1

                K(i, R) = .Fields("UnitOnOrder").Value

            End If

        .MoveNext

    Next j

Next i

```
End With
Graph1.NumSets = B
Graph1.NumPoints = D
For i = 1 To B
    For j = 1 To D
        Graph1.GraphData = K(i, j)
    Next j
Next i

End Sub
```

### 13. frmG2

```
Private Sub Command1_Click()
Unload Me
End Sub
```

```
Private Sub Command2_Click()
frmG1.Show
End Sub
```

```
Private Sub Command3_Click()
frmG3.Show
End Sub
```

```
Private Sub Command4_Click()
frmG4.Show
End Sub
```

```
Private Sub Form_Load()
Dim A, X, Y, Z As String
Dim B, C, D, O, R As Integer
Dim E, F, G, H, i, j As Integer
```

```

Dim K() As Integer
B = 1
X = "select UnitOnOrder,UnitProduce, UnitInStock,UnitShortage,InventoryCost,ProductID from
ProductDetail "
Data2.RecordSource = X
Data2.Refresh
C = Data2.Recordset.RecordCount
ReDim K(1 To B, 1 To C) As Integer
With Data2.Recordset
    Data2.Refresh
    For i = 1 To B
        .MoveFirst
        R = 0
        For j = 1 To C
            O = .Fields("ProductID").Value
            If O = i Then
                R = R + 1
                K(i, R) = .Fields("UnitProduce").Value
            End If
            .MoveNext
        Next j
    Next i
End With
Graph1.NumSets = B
Graph1.NumPoints = C

    For i = 1 To B
        For j = 1 To C
            Graph1.GraphData = K(i, j)
        Next j
    Next i
End Sub

```

## 14. frmG3

Option Explicit

Private Sub Command1\_Click()

frmG1.Show

End Sub

Private Sub Command2\_Click()

frmG2.Show

End Sub

Private Sub Command3\_Click()

Unload Me

End Sub

Private Sub Command4\_Click()

frmG4.Show

End Sub

Private Sub Form\_Load()

Dim A, X, Y, Z As String

Dim B, C, D, O, R As Integer

Dim E, F, G, H, i, j As Integer

Dim K() As Integer

B = 1

X = "select UnitOnOrder,UnitProduce, UnitInStock,UnitShortage,InventoryCost,ProductID from  
ProductDetail "

Data2.RecordSource = X

Data2.Refresh

C = Data2.Recordset.RecordCount

ReDim K(1 To B, 1 To C) As Integer

With Data2.Recordset

    Data2.Refresh

```

For i = 1 To B
    .MoveFirst
    R = 0
    For j = 1 To C
        O = .Fields("ProductID").Value
        If O = i Then
            R = R + 1
            K(i, R) = .Fields("UnitOnOrder").Value
        End If
    .MoveNext
    Next j
Next i
End With
Graph1.NumSets = B
Graph1.NumPoints = C
    For i = 1 To B
        For j = 1 To C
            Graph1.GraphData = K(i, j)
        Next j
    Next i
End Sub

```

## 15. frmG4

Option Explicit

```
Private Sub Command1_Click()
```

```
frmG1.Show
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
frmG2.Show
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
frmG3.Show
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Dim A, X, Y, Z As String
```

```
Dim B, C, D, O, R As Integer
```

```
Dim E, F, G, H, i, j As Integer
```

```
Dim K() As Integer
```

```
Dim L() As Integer
```

```
B = 1
```

```
X = "select UnitOnOrder,UnitProduce, UnitInStock,UnitShortage,InventoryCost,ProductID from  
ProductDetail "
```

```
Data2.RecordSource = X
```

```
Data2.Refresh
```

```
C = Data2.Recordset.RecordCount
```

```
ReDim K(1 To B, 1 To C) As Integer
```

```
ReDim L(1 To B, 1 To C) As Integer
```

```
With Data2.Recordset
```

```
    Data2.Refresh
```

```
For i = 1 To B
```

```
    .MoveFirst
```

```
    R = 0
```

```
    For j = 1 To C
```

```
        O = .Fields("ProductID").Value
```

```
        If O = i Then
```

```
            R = R + 1
```

```
        K(i, R) = .Fields("UnitOnOrder").Value
```

```

        L(i, R) = .Fields("UnitProduce").Value
    End If
    .MoveNext
Next j
Next i
End With
Graph1.NumSets = 2
Graph1.NumPoints = C
For i = 1 To B
    For j = 1 To C
        Graph1.GraphData = K(i, j)
        Graph1.GraphData = L(i, j)
    Next j
Next i
End Sub

```

## 16. frmGenetic

Option Explicit

Dim Inventory, Shortage, Popu, Popu2, ManC1, Cc, Cycle, NuMa, NuMi As Double

Dim CO, MT As Double

Dim De() As Double

Dim SumDe() As Double

Dim ManuC1 As Double

Dim n() As Double

Dim Inv() As Double

Dim Sht() As Double

Dim Obj() As Double

Dim InvM() As Double

Dim ShtM() As Double

Dim MaxObj As Double

Dim MinObj As Double

Dim SumObj As Double

Dim LastSumObj As Double  
Dim BoObj() As Double  
Dim SumBoObj() As Double  
Dim PopObj() As Double  
Dim SLPop1() As Double  
Dim SLPop2() As Double  
Dim SumRanMan1Pop() As Double  
Dim COPop1() As Double  
Dim COPop2() As Double  
Dim NCOPop1() As Double  
Dim NCOPop2() As Double  
Dim NInv1() As Double  
Dim NSht1() As Double  
Dim NObj1() As Double  
Dim NInv1M() As Double  
Dim NSht1M() As Double  
Dim NInv2() As Double  
Dim NSht2() As Double  
Dim NObj2() As Double  
Dim NInv2M() As Double  
Dim NShtM() As Double  
Dim NCSumDe1() As Double  
Dim NCSumDe2() As Double  
Dim MutNMan() As Double  
Dim MInv() As Double  
Dim MSht() As Double  
Dim MObj() As Double  
Dim MInvM() As Double  
Dim MShtM() As Double  
Dim MSumDe() As Double  
Dim CroBit, CBlock1, CBlock2 As String  
Dim ACBit1, ACBit2 As Double 'ACBit = After CrossCver Bit

Dim CroBit1(), CroBit2() As Variant 'เป็น Variant เพราะว่าก่อนหน้านี้อยู่ในรูปของ Text

Dim FBit, FBit1(), FBit2() As Variant 'FBit= ฝาก bit

Dim A, B, bo, C, D, E, F, H, G, m, O As Double

Dim i, j, K, t, p As Double

Dim m1, m2 As Double

Dim q, R, rc, rm, s, sl1, sl2, X, Y, Z, x1, z1, z2 As Double

Dim s1, s2, t1, t2, Result1, Result2, TestResult As Double

Dim Mut(), MutN() As Variant

Dim BesSumMan, BesSumObj, NumBesSumObj As Double

Dim SS, SM, SI As Double

Dim Ti1, Ti2, Ti3, Ti4, Ti5, Ti6, Ti7, Ti8, Ti9 As Integer

Dim DT1, DT2, DT3, DT4, DT5, DT6, DT7, DT8, DT9, DT10, DT11, alr, r1, o1, il, j1, KFC,

CTC As Double

Dim DT12, DT13, DT14, DT15, DT16, DT17 As Double

Dim PP() As Double

\*\*\*\*\* Main ProGram \*\*\*\*\*

Private Sub ClickGen\_Click()

Text2 = ""

Text3 = ""

Text2.Text = Text2.Text & Format(Time, "Long Time")

Text2.Text = Text2.Text & vbCrLf

Call SumDemand

SumSD = ""

SumSD.Text = SumSD.Text & E

SumSD.Text = SumSD.Text & vbCrLf

'รับค่าจาก TextBox

Popu = Val(PopT.Text)

Popu2 = Popu / 2

CO = Val(CroOT.Text) / 100 'เปอร์เซ็นต์ที่จะทำการ CrossOver

MT = Val(MutT.Text) / 100 'เปอร์เซ็นต์ที่จะทำการ Mutation

Call PopN 'random Manu.

Cycle = 0

Cc = Val(CycleT.Text)

Do While Cycle < Cc

TimG = ""

NumMan = ""

RandMan = ""

ObjG = ""

Cycle = Cycle + 1

TimG.Text = TimG.Text & Cycle

TimG.Text = TimG.Text & vbCrLf

For D = 1 To Popu

NumMan.Text = NumMan.Text & D

NumMan.Text = NumMan.Text & vbCrLf

E = 0

For A = 1 To t

For B = 1 To i

For C = 1 To j

E = E + n(A, B, C, D)

SumRanMan1Pop(D) = E

Next C

Next B

Next A

RandMan.Text = RandMan.Text & E

RandMan.Text = RandMan.Text & vbCrLf

ObjG.Text = ObjG.Text & Obj(D)

ObjG.Text = ObjG.Text & vbCrLf

Next D

Call Selection

Call CrossOver

CrossO1 = ""

CrossO2 = ""

SumManN1T.Text = ""

SumManN2T.Text = ""

NumCroT = ""

For D = 1 To Popu2

SumManN1T.Text = SumManN1T.Text & NCSumDe1(D)

SumManN1T.Text = SumManN1T.Text & vbCrLf

SumManN2T.Text = SumManN2T.Text & NCSumDe2(D)

SumManN2T.Text = SumManN2T.Text & vbCrLf

NumCroT.Text = NumCroT.Text & D

NumCroT.Text = NumCroT.Text & vbCrLf

CrossO1.Text = CrossO1.Text & NObj1(D)

CrossO1.Text = CrossO1.Text & vbCrLf

CrossO2.Text = CrossO2.Text & NObj2(D)

CrossO2.Text = CrossO2.Text & vbCrLf

Next D

Call Mutation

AftMut = ""

ObjMutT = ""

BesSumManT = ""

BesSumObjT = ""

For E = 1 To Popu

AftMut.Text = AftMut.Text & MSumDe(E)

AftMut.Text = AftMut.Text & vbCrLf

ObjMutT.Text = ObjMutT.Text & MObj(E)

ObjMutT.Text = ObjMutT.Text & vbCrLf

If E = 1 Then

BesSumObj = MObj(E)

NumBesSumObj = E

Else

If MObj(E) < BesSumObj Then

BesSumObj = MObj(E)

NumBesSumObj = E

```

        End If
    End If
    BesSumMan = MSumDe(NumBesSumObj)
Next E
    BesSumObjT.Text = BesSumObjT.Text & BesSumObj
    BesSumObjT.Text = BesSumObjT.Text & vbCrLf
    BesSumManT.Text = BesSumManT.Text & BesSumMan
    BesSumManT.Text = BesSumManT.Text & vbCrLf
D = NumBesSumObj
SM = SI = SS = 1
    SumManT = ""
    SumInvT = ""
    SumShoT = ""
SumSManT = ""
SumSInvT = ""
SumSShoT = ""
For A = 1 To t
    For B = 1 To i
        For C = 1 To j
            SumManT.Text = SumManT.Text & MutNMan(A, B, C, D)
            SumInvT.Text = SumInvT.Text & MInv(A, D)
            SumShoT.Text = SumShoT.Text & MSht(A, D)
            SumManT.Text = SumManT.Text & vbCrLf
            SumInvT.Text = SumInvT.Text & vbCrLf
            SumShoT.Text = SumShoT.Text & vbCrLf
            SM = SM + MutNMan(A, B, C, D)
        Next C
    Next B
Next A
    SI = MInvM(D)
    SS = MShtM(D)
SumSManT.Text = SumSManT.Text & SM

```

SumSInvT.Text = SumSInvT.Text & SI

SumSShoT.Text = SumSShoT.Text & SS

SumSManT.Text = SumSManT.Text & vbCrLf

SumSInvT.Text = SumSInvT.Text & vbCrLf

SumSShoT.Text = SumSShoT.Text & vbCrLf

Loop

Call RetailerReceive

Text3.Text = Text3.Text & Format(Time, "Long Time")

Text3.Text = Text3.Text & vbCrLf

End Sub

Private Sub Form\_Load()

i = 1 'จำนวน Manufacturer

j = 1 'จำนวนชนิด Product

SD1.Text = ""

ShtCostT = ""

InvCostT = ""

MonthT.Text = ""

DT5 = "select ProductName,Capacity,InventoryCost,ManuID from ManuProduct "

Data3.RecordSource = DT5

Data3.Refresh

DT6 = Data3.Recordset.RecordCount

With Data3.Recordset

Data3.Refresh

.MoveFirst

DT7 = .Fields("Capacity").Value

DT8 = .Fields("InventoryCost").Value

InvCostT.Text = InvCostT.Text & DT8

End With

DT9 = "select ShortageCost from Retailer "

Data4.RecordSource = DT9

Data4.Refresh

DT10 = Data4.Recordset.RecordCount

With Data4.Recordset

Data4.Refresh

.MoveFirst

DT11 = .Fields("ShortageCost").Value

ShtCostT.Text = ShtCostT.Text & DT11

End With

DT1 = "select RetailerID, count(RetailerID) as count from RetailerOrder "

DT1 = DT1 & "Group by RetailerID"

Data1.RecordSource = DT1

Data1.Refresh

DT2 = Data1.Recordset.RecordCount

DT3 = "select RetailerID,UnitOnOrder from RetailerOrder "

Data2.RecordSource = DT3

Data2.Refresh

DT4 = Data2.Recordset.RecordCount

D = DT4 / DT2

alr = DT4

K = DT2 'จำนวน Retailer

t = D 'จำนวน 12 Month

ReDim De(1 To t, 1 To K) As Double

ReDim ntb(1 To K, 1 To t) As Double

With Data2.Recordset

Data2.Refresh

For i1 = 1 To K

.MoveFirst

r1 = 0

For j1 = 1 To alr

o1 = .Fields("RetailerID").Value

If o1 = i1 Then

r1 = r1 + 1

ntb(i1, r1) = .Fields("UnitOnOrder").Value

End If

```

        .MoveNext
    Next j1
Next i1
End With
For A = 1 To t    'จำนวน 12 Month
MonthT.Text = MonthT.Text & A
MonthT.Text = MonthT.Text & vbCrLf
    For B = 1 To K    'จำนวน Retailer
        For i1 = 1 To K
            For r1 = 1 To t
                If r1 = A And i1 = B Then
                    De(A, B) = ntb(i1, r1)
                    Debug.Print De(A, B)
                    SD1.Text = SD1.Text & De(A, B)
                    SD1.Text = SD1.Text & vbCrLf
                End If
            Next r1
        Next i1
    Next B
Next A
End Sub

```

```

Sub SumDemand()
SD1.Text = ""
    ReDim SumDe(1 To t) As Double
        E = 0
    For A = 1 To t
        For B = 1 To K
            SumDe(A) = SumDe(A) + De(A, B)
        Next B
        SD1.Text = SD1.Text & SumDe(A)
        SD1.Text = SD1.Text & vbCrLf
    Next A
End Sub

```

$E = E + \text{SumDe}(A)$

Next A

End Sub

Sub PopN()

Inventory = Val(InvCostT.Text)

Shortage = 10 \* Val(ShtCostT.Text)

ManuC1 = DT7

ReDim n(1 To t, 1 To i, 1 To j, 1 To Popu)

ReDim Inv(1 To t, 1 To Popu) As Double 'เป็นของแต่ละเดือน

ReDim Sht(1 To t, 1 To Popu) As Double

ReDim InvM(1 To Popu) As Double 'เป็นของแต่ละเดือนตั้งแต่เดือนแรกจนถึง a

ReDim ShtM(1 To Popu) As Double

ReDim Obj(1 To Popu) As Double

ReDim SumRanMan1Pop(1 To Popu) As Double

For D = 1 To Popu

For A = 1 To t           'จำนวน Month

For B = 1 To i           'จำนวน Manufacturer

For C = 1 To j           'จำนวนชนิด Product

\*\*\*\*\*จำนวนที่ Manu จะผลิต\*\*\*\*\*

Randomize

$n(A, B, C, D) = \text{Round}(\text{Rnd} * \text{ManuC1})$

If A = 1 Then

If  $n(A, B, C, D) \geq \text{SumDe}(A)$  Then

$\text{Inv}(A, D) = n(A, B, C, D) - \text{SumDe}(A)$

$\text{InvM}(D) = \text{Inv}(A, D)$

$\text{Sht}(A, D) = 0$

Else

$\text{Sht}(A, D) = \text{SumDe}(A) - n(A, B, C, D)$

$\text{ShtM}(D) = \text{Sht}(A, D)$

$\text{Inv}(A, D) = 0$

End If

Else

If  $(n(A, B, C, D) + \text{Inv}((A - 1), D)) \geq \text{SumDe}(A)$  Then 'N(a, b, c,d) + Inv((a - 1),d)=จำนวนที่ Manu ผลิต+Inv ของเดือนก่อนหน้า<sup>ที่</sup>ผลิตเกิน - sht ของเดือนก่อนหน้า<sup>ที่</sup>ขาด

$\text{Inv}(A, D) = (n(A, B, C, D) + \text{Inv}((A - 1), D)) - \text{SumDe}(A)$

$\text{InvM}(D) = \text{InvM}(D) + \text{Inv}(A, D)$

$\text{Sht}(A, D) = 0$

Else

$\text{Sht}(A, D) = \text{SumDe}(A) - (n(A, B, C, D) + \text{Inv}((A - 1), D))$

$\text{ShtM}(D) = \text{ShtM}(D) + \text{Sht}(A, D)$

$\text{Inv}(A, D) = 0$

End If

End If

Next C

Next B

Next A

$\text{Obj}(D) = \text{Inventory} * \text{InvM}(D) + \text{Shortage} * \text{ShtM}(D)$

Next D

End Sub

Sub Selection()

ReDim BoObj(1 To Popu) As Double

ReDim SumBoObj(1 To Popu) As Double

ReDim PopObj(1 To Popu) As Double

ReDim SLPop1(1 To t, 1 To i, 1 To j, 1 To Popu2) As Double 'ประชากรที่เลือกชุดที่ 1

ReDim SLPop2(1 To t, 1 To i, 1 To j, 1 To Popu2) As Double 'ประชากรที่เลือกชุดที่ 2

bo = 0

\*\*\*\*\* หา Max & Min ของ Obj. \*\*\*\*\*

For D = 1 To Popu

If D = 1 Then

MaxObj = Obj(D)

NuMa = D

Else

If MaxObj < Obj(D) Then

MaxObj = Obj(D)

NuMa = D

End If

End If

Next D

For D = 1 To Popu

If D = 1 Then

MinObj = Obj(D)

NuMi = D

Else

If MinObj > Obj(D) Then

MinObj = Obj(D)

NuMi = D

End If

End If

Next D

MaxObjT = ""

MaxObjT.Text = MaxObjT.Text & MaxObj

MaxObjT.Text = MaxObjT.Text & vbCrLf

MinObjT = ""

MinObjT.Text = MinObjT.Text & MinObj

MinObjT.Text = MinObjT.Text & vbCrLf

MaxManuT = ""

MaxManuT.Text = MaxManuT.Text & SumRanMan1Pop(NuMa) 'บอกค่าที่ของ

Man ที่ผลิตแล้วได้ Obj.มากที่สุด

MaxManuT.Text = MaxManuT.Text & vbCrLf

MinManuT = ""

MinManuT.Text = MinManuT.Text & SumRanMan1Pop(NuMi) 'บอกค่าที่ของ

Man ที่ผลิตแล้วได้ Obj.น้อยที่สุด

MinManuT.Text = MinManuT.Text & vbCrLf

SumObjT = ""

```

Text1 = ""
For D = 1 To Popu
    BoObj(D) = MaxObj / Obj(D)    'BoObj(d) เอาไว้หาค่าที่จะใช้ในการ Random ประชากร
    bo = bo + BoObj(D)
    SumBoObj(D) = bo              'SumBoObj(d) = ผลรวม BoObj. ตั้งแต่ 1 จนถึงที่ d
    Text1.Text = Text1.Text & Format(SumBoObj(D), "0.000")
    Text1.Text = Text1.Text & vbCrLf
    LastSumObj = SumBoObj(Popu)   'LastSumObj = sum obj. ทั้งหมด (ตัวสุดท้าย)
Next D

SumObjT.Text = SumObjT.Text & LastSumObj
SumObjT.Text = SumObjT.Text & vbCrLf
'***** หาก Pop ที่จะ Select Obj. แต่ละตัว *****

PopObjT = ""
For D = 1 To Popu
    PopObj(D) = SumBoObj(D) / LastSumObj
    PopObjT.Text = PopObjT.Text & Format(PopObj(D), "0.0000")
    PopObjT.Text = PopObjT.Text & vbCrLf
Next D

'***** SLPop1 *****

SLPop1T = ""
Ran1 = ""
For E = 1 To Popu2
    Randomize
    R = Rnd
    For D = 1 To Popu
        If D = 1 Then              'ให้ SLPop1 แรกเป็นของสมาชิกของ N แรกทั้งหมดก่อน
            For A = 1 To t
                For B = 1 To i
                    For C = 1 To j
                        SLPop1(A, B, C, E) = n(A, B, C, D)
                        sl1 = D
                    Next C
                Next B
            Next A
        Next D
    Next E

```

Next B

Next A

Else

If  $R > \text{PopObj}(D - 1)$  Then 'ให้เปรียบเทียบว่าถ้า r ที่สุ่มได้มากกว่า PopObj ก่อนหน้า  
ให้ SLPop1 ต่อมาเป็นของ N ที่  $d = \text{Popu}$  ปัจจุบัน

For A = 1 To t

For B = 1 To i

For C = 1 To j

SLPop1(A, B, C, E) = n(A, B, C, D)

sl1 = D

Next C

Next B

Next A

End If

End If

Next D

SLPop1T.Text = SLPop1T.Text & sl1

SLPop1T.Text = SLPop1T.Text & vbCrLf

Ran1.Text = Ran1.Text & Format(R, "0.00")

Ran1.Text = Ran1.Text & vbCrLf

Next E

\*\*\*\*\* SLPop2\*\*\*\*\*

SLPop2T = ""

Ran2 = ""

For E = 1 To Popu2

Randomize

R = Rnd

For D = 1 To Popu

If D = 1 Then 'ให้ SLPop2 แรกเป็นของสมาชิกของ N แรกทั้งหมดก่อน

For A = 1 To t

For B = 1 To i

For C = 1 To j

```

    SLPop2(A, B, C, E) = n(A, B, C, D)
    sl2 = D
Next C
Next B
Next A
Else
    If R > PopObj(D - 1) Then 'ให้เปรียบเทียบว่าถ้า r ที่สุ่มได้มากกว่า PopObj ก่อนหน้า
ให้ SLPop2 ต่อมาเป็นของ N ที่ d = Popu ปัจจุบัน
        For A = 1 To t
        For B = 1 To i
        For C = 1 To j
            SLPop2(A, B, C, E) = n(A, B, C, D)
            sl2 = D
        Next C
        Next B
        Next A
        End If
    End If
Next D
    SLPop2T.Text = SLPop2T.Text & sl2
    SLPop2T.Text = SLPop2T.Text & vbCrLf
    Ran2.Text = Ran2.Text & Format(R, "0.00")
    Ran2.Text = Ran2.Text & vbCrLf
Next E
End Sub

```

Sub CrossOver()

```
ReDim COPop1(1 To t, 1 To i, 1 To j, 1 To Popu2) As Double
```

```
ReDim COPop2(1 To t, 1 To i, 1 To j, 1 To Popu2) As Double
```

```
For E = 1 To Popu2
```

```
    Randomize
```

```
    rc = Rnd
```

```

For A = 1 To t
For B = 1 To i
For C = 1 To j
    If rc < CO Then
        COPop1(A, B, C, E) = SLPop1(A, B, C, E)
        COPop2(A, B, C, E) = SLPop2(A, B, C, E)
    ElseIf rc >= CO Then
        COPop1(A, B, C, E) = SLPop1(A, B, C, E)
        COPop2(A, B, C, E) = SLPop2(A, B, C, E)
        s1 = COPop1(A, B, C, E)
        s2 = COPop2(A, B, C, E)

```

Call ChangeToBits 'ทำการ CrossOver โดยการเปลี่ยนเป็น Bit

\*\*\*\*\* CrossOver \*\*\*\*\*

Randomize

R = Rnd 'Random หาดำแหน่งที่จะทำการ CrossOver เพียง 1 ตำแหน่ง

G = 1 / Y

For m = 1 To Y

If m = 1 Then

H = m

Else

If R > G \* (m - 1) Then

H = m

End If

End If

Next m

FBit = CroBit1(H)

CroBit1(H) = CroBit2(H) 'h = ตำแหน่งที่ CrossOver ; ตำแหน่งอื่นเหมือนเดิม

CroBit2(H) = FBit

Call ChangeToBase10 'ทำการเปลี่ยนเป็นฐาน 10

COPop1(A, B, C, E) = ACBit1

COPop2(A, B, C, E) = ACBit2

End If

Next C

Next B

Next A

Next E

\*\*\*\*\*จำนวนที่ Manu จะผลิตใหม่ จากการที่ CrossOver\*\*\*\*\*

ReDim NCOPop1(1 To t, 1 To i, 1 To j, 1 To Popu2) As Double

ReDim NInv1(1 To t, 1 To Popu) As Double

ReDim NSht1(1 To t, 1 To Popu) As Double

ReDim NObj1(1 To Popu) As Double

ReDim NCSumDe1(Popu2) As Double

ReDim NInv1M(1 To Popu) As Double

ReDim NSht1M(1 To Popu) As Double

ReDim NCOPop2(1 To t, 1 To i, 1 To j, 1 To Popu2) As Double

ReDim NInv2(1 To t, 1 To Popu) As Double

ReDim NSht2(1 To t, 1 To Popu) As Double

ReDim NObj2(1 To Popu) As Double

ReDim NCSumDe2(Popu2) As Double

ReDim NInv2M(1 To Popu) As Double

ReDim NSht2M(1 To Popu) As Double

'SumDe(a) เดิมจาก Retailer

For D = 1 To Popu2

NCSumDe1(D) = 0

NCSumDe2(D) = 0

For A = 1 To t

For B = 1 To i

For C = 1 To j

NCOPop1(A, B, C, D) = COPop1(A, B, C, D)

NCOPop2(A, B, C, D) = COPop2(A, B, C, D)

NCSumDe1(D) = NCSumDe1(D) + NCOPop1(A, B, C, D) 'SumDemand ของการ

CrossOver ทั้ง 12 เดือนของแต่ละ Pop

NCSumDe2(D) = NCSumDe2(D) + NCOPop2(A, B, C, D)

Next C

Next B

Next A

Next D

\*\*\*\*\* ของ Crossover ชุดที่ 1 \*\*\*\*\*

For D = 1 To Popu2

For A = 1 To t

For B = 1 To i

For C = 1 To j

If A = 1 Then

If NCOPop1(A, B, C, D) >= SumDe(A) Then

NInv1(A, D) = n(A, B, C, D) - SumDe(A)

NInv1M(D) = NInv1(A, D)

NSht1(A, D) = 0

Else

NSht1(A, D) = SumDe(A) - NCOPop1(A, B, C, D)

NSht1M(D) = NSht1(A, D)

NInv1(A, D) = 0

End If

Else

If (NCOPop1(A, B, C, D) + NInv1((A - 1), D)) >= SumDe(A) Then 'NCOPop1(a, b, c, d) + NInv((a - 1), d) = จำนวนที่ Manu ผลิต+Inv ของเดือนก่อนหน้าที่ผลิตเกิน

NInv1(A, D) = (NCOPop1(A, B, C, D) + NInv1((A - 1), D)) - SumDe(A)

NInv1M(D) = NInv1M(D) + NInv1(A, D)

NSht1(A, D) = 0

Else

NSht1(A, D) = SumDe(A) - (NCOPop1(A, B, C, D) + NInv1((A - 1), D))

NSht1M(D) = NSht1M(D) + NSht1(A, D)

NInv1(A, D) = 0

End If

End If

Next C

Next B

Next A

$NObj1(D) = Inventory * NInv1M(D) + Shortage * NSht1M(D)$

Next D

\*\*\*\*\* ของ Crossover ชุดที่ 2 \*\*\*\*\*

For D = 1 To Popu2

For A = 1 To t

For B = 1 To i

For C = 1 To j

If A = 1 Then

If  $NCOPop2(A, B, C, D) \geq SumDe(A)$  Then

$NInv2(A, D) = n(A, B, C, D) - SumDe(A)$

$NInv2M(D) = NInv2(A, D)$

$NSht2(A, D) = 0$

Else

$NSht2(A, D) = SumDe(A) - NCOPop2(A, B, C, D)$

$NSht2M(D) = NSht2(A, D)$

$NInv2(A, D) = 0$

End If

Else

If  $(NCOPop2(A, B, C, D) + NInv2((A - 1), D)) \geq SumDe(A)$  Then 'NCOPop1(a, b, c,d) + NInv((a - 1),d)=จำนวนที่ Manu ผลิต+Inv ของเดือนก่อนหน้าที่ผลิตเกิน

$NInv2(A, D) = (NCOPop2(A, B, C, D) + NInv2((A - 1), D)) - SumDe(A)$

$NInv2M(D) = NInv2M(D) + NInv2(A, D)$

$NSht2(A, D) = 0$

Else

$NSht2(A, D) = SumDe(A) - (NCOPop2(A, B, C, D) + NInv2((A - 1), D))$

$NSht2M(D) = NSht2M(D) + NSht2(A, D)$

$NInv2(A, D) = 0$

End If

End If

Next C

Next B

Next A

$NObj2(D) = Inventory * NInv2M(D) + Shortage * NSht2M(D)$

Next D

End Sub

Sub Mutation()

ReDim Mut(1 To t, 1 To i, 1 To j, 1 To Popu) As Variant

ReDim MutN(1 To t, 1 To i, 1 To j, 1 To Popu) As Variant

\*\*\* ทำการเรียง Array ให้เป็นประชากรที่จะนำมา Mutation ให้มีจำนวนเท่ากับ Pop \*\*\*\*\*

For E = 1 To Popu

For A = 1 To t

For B = 1 To i

For C = 1 To j

G = E / 2

H = Int(E / 2)

If G > H Then

Mut(A, B, C, E) = NCOPop1(A, B, C, G + 0.5)

Else

Mut(A, B, C, E) = NCOPop2(A, B, C, G)

End If

Next C

Next B

Next A

Next E

\*\*\*\*\* Mutation ทุกบิตทีละบิต ทุก Pop \*\*\*\*\*

For E = 1 To Popu

For A = 1 To t

For B = 1 To i

For C = 1 To j

s1 = Mut(A, B, C, E)

If s1 = 0 Then

MutN(A, B, C, E) = 0

Else

Call ChangeToBitsM

For m = 1 To Y

Randomize

R = Rnd

If R >= MT Then

If CroBit1(m) = 0 Then

CroBit1(m) = 1

Else

CroBit1(m) = 0

End If

End If

Next m

Call ChangeToBase10M 'ทำการเปลี่ยนเป็นฐาน 10 M

MutN(A, B, C, E) = ACBit1

End If

If MutN(A, B, C, E) > ManuC1 Then

MutN(A, B, C, E) = ManuC1

End If

Next C

Next B

Next A

Next E

\*\*\*\*\* จำนวนที่ Manu จะผลิตใหม่ หลังจากทำการ Mutation \*\*\*\*\*

ReDim MutNMan(1 To t, 1 To i, 1 To j, 1 To Popu) As Double

ReDim MInv(1 To t, 1 To Popu) As Double

ReDim MSht(1 To t, 1 To Popu) As Double

ReDim MInvM(1 To Popu) As Double

ReDim MShtM(1 To Popu) As Double

ReDim MObj(1 To Popu) As Double

ReDim MSumDe(Popu) As Double

'SumDe(a) เดิมจาก Retailer

For D = 1 To Popu

MSumDe(D) = 0

For A = 1 To t

For B = 1 To i

For C = 1 To j

MutNMan(A, B, C, D) = MutN(A, B, C, D)

MSumDe(D) = MSumDe(D) + MutNMan(A, B, C, D) 'SumDemand ของการ

Mutation ทั้ง 12 เดือนของแต่ละ Pop

Next C

Next B

Next A

Next D

For D = 1 To Popu

For A = 1 To t

For B = 1 To i

For C = 1 To j

If A = 1 Then

If MutNMan(A, B, C, D) >= SumDe(A) Then

MInv(A, D) = MutNMan(A, B, C, D) - SumDe(A)

MInvM(D) = MInv(A, D)

MSht(A, D) = 0

Else

MSht(A, D) = SumDe(A) - MutNMan(A, B, C, D)

MShtM(D) = MSht(A, D)

MInv(A, D) = 0

End If

Else

If (MutNMan(A, B, C, D) + MInv((A - 1), D)) >= SumDe(A) Then

MInv(A, D) = (MutN(A, B, C, D) + MInv((A - 1), D)) - SumDe(A)

MInvM(D) = MInvM(D) + MInv(A, D)

```

MSht(A, D) = 0
Else
MSht(A, D) = SumDe(A) - (MutNMan(A, B, C, D) + MInv((A - 1), D))
MShtM(D) = MShtM(D) + MSht(A, D)
MInv(A, D) = 0
End If
End If
Next C
Next B
Next A
MObj(D) = Inventory * MInvM(D) + Shortage * MShtM(D)
Next D
For D = 1 To Popu
SM = SI = SS = 0
SumManT = ""
SumInvT = ""
SumShoT = ""
SumSManT = ""
SumSInvT = ""
SumSShoT = ""
For A = 1 To t
For B = 1 To i
For C = 1 To j
SumManT.Text = SumManT.Text & MutNMan(A, B, C, D)
SumInvT.Text = SumInvT.Text & MInv(A, D)
SumShoT.Text = SumShoT.Text & MSht(A, D)
SumManT.Text = SumManT.Text & vbCrLf
SumInvT.Text = SumInvT.Text & vbCrLf
SumShoT.Text = SumShoT.Text & vbCrLf
SM = SM + MutNMan(A, B, C, D)
Next C
Next B

```

Next A

SI = M<sup>-1</sup>InvM(D)

SS = MShtM(D)

SumSManT.Text = SumSManT.Text & SM

SumSInvT.Text = SumSInvT.Text & SI

SumSShoT.Text = SumSShoT.Text & SS

SumSManT.Text = SumSManT.Text & vbCrLf

SumSInvT.Text = SumSInvT.Text & vbCrLf

SumSShoT.Text = SumSShoT.Text & vbCrLf

Next D

End Sub

Sub ChangeToBase10M() \*\*\*\*\* ทำให้เป็นฐานสิบ \*\*\*\*\*

ReDim FBit1(1 To Y) As Variant

For m = 1 To Y

FBit1(m) = 2 \* CroBit1(m)

Next m

m = 0

Z = 0

z1 = 0

z2 = 0

Do While m < Y - 1

m = m + 1

z1 = z1 + FBit1(m) ^ (Y - m)

ACBit1 = z1

Loop

If FBit1(Y) = 0 Then

ACBit1 = ACBit1 + 0

Else

ACBit1 = ACBit1 + 1

End If

End Sub

Sub ChangeToBitsM()

CBlock1 = "" 'Clear

CBlock2 = ""

Do While s1 > 0

Result1 = s1 / 2 'หารเลขฐานสิบนั้นด้วยสอง

Result2 = Int(s1 / 2) 'หารเลขฐานสิบนั้นด้วยสองโดยไม่สนใจเศษ

TestResult = Result1 - Result2 'นำมาเปรียบเทียบโดยการลบกัน

If TestResult > 0 Then 'ถ้ามากกว่าศูนย์หรือหมายถึงมีเศษจากตัวที่หาร

CroBit = 1 'มีเศษให้เป็นหนึ่ง

Else

CroBit = 0 'ไม่มีเศษให้เป็นศูนย์

End If

CBlock1 = CroBit & CBlock1 'ให้นำ Cbit ต่อด้วยคำตอบที่มีเศษ(1)หรือไม่มีเศษ(0)

s1 = Result2 'ให้ s1 เท่ากับจำนวนเลขที่ไม่มีเศษ

Loop 'วนลูปเพื่อให้หารค่า

Y = Len(CBlock1) 'ความยาวของ CBlock1

ReDim CroBit1(1 To Y) As Variant

For m = 1 To Y 'กำหนดความยาวของ i ซึ่งจะทำการเก็บบิตต่างๆ ไว้ใน Array เท่ากับ y

CroBit1(m) = Mid(CBlock1, m, 1) 'ใส่ค่าต่างๆ ที่ละบิตไว้ใน CBit1 Array โดยการวนลูป

Next m

End Sub

Sub ChangeToBits()

CBlock1 = "" 'Clear

CBlock2 = ""

Do While s1 > 0

Result1 = s1 / 2 'หารเลขฐานสิบนั้นด้วยสอง

Result2 = Int(s1 / 2) 'หารเลขฐานสิบนั้นด้วยสองโดยไม่สนใจเศษ

TestResult = Result1 - Result2 'นำมาเปรียบเทียบโดยการลบกัน

If TestResult > 0 Then 'ถ้ามากกว่าศูนย์หรือหมายถึงมีเศษจากตัวที่หาร

CroBit = 1 'มีเศษให้เป็นหนึ่ง

Else

CroBit = 0 'ไม่มีเศษให้เป็นศูนย์

End If

CBlock1 = CroBit & CBlock1 'ให้นำ Cbit ต่อด้วยคำตอบที่มีเศษ(1)หรือไม่มีเศษ(0)

s1 = Result2 'ให้ s1 เท่ากับจำนวนเลขที่ไม่มีเศษ

Loop 'วนลูปเพื่อให้หารค่า

Do While s2 > 0

Result1 = s2 / 2

Result2 = Int(s2 / 2)

TestResult = Result1 - Result2

If TestResult > 0 Then

CroBit = 1

Else

CroBit = 0

End If

CBlock2 = CroBit & CBlock2

s2 = Result2

Loop

t1 = Len(CBlock1)

t2 = Len(CBlock2)

If t1 > t2 Then

x1 = t1 - t2

For X = 1 To x1

CBlock2 = 0 & CBlock2

Next X

ElseIf t1 < t2 Then

x1 = t2 - t1

For X = 1 To x1

CBlock1 = 0 & CBlock1

Next X

End If

Y = Len(CBlock1) 'ความยาวของ t1 และ t2 จะเท่ากันแล้ว 'CBlock1=CBlock2

ReDim CroBit1(1 To Y) As Variant

ReDim CroBit2(1 To Y) As Variant

ReDim FBit1(1 To Y) As Variant

ReDim FBit2(1 To Y) As Variant

For m = 1 To Y   กำหนดความยาวของ i ซึ่งจะทำให้การเก็บบิตต่างๆ ไว้ใน Array เท่ากับ y

    CroBit1(m) = Mid(CBlock1, m, 1)   ใส่ค่าต่างๆ ที่ละบิตไว้ใน CBit1 Array โดยการวนดู

    CroBit2(m) = Mid(CBlock2, m, 1)

Next m

End Sub

Sub ChangeToBase10()

ReDim FBit1(1 To Y) As Variant

ReDim FBit2(1 To Y) As Variant

\*\*\*\*\* ทำให้เป็นฐานสิบ \*\*\*\*\*

For m = 1 To Y

    FBit1(m) = 2 \* CroBit1(m)

    FBit2(m) = 2 \* CroBit2(m)

Next m

    m = 0

    Z = 0

    z1 = 0

    z2 = 0

Do While m < Y - 1

    m = m + 1

    z1 = z1 + FBit1(m) ^ (Y - m)

    z2 = z2 + FBit2(m) ^ (Y - m)

    ACBit1 = z1

    ACBit2 = z2

Loop

    If FBit1(Y) = 0 Then

        ACBit1 = ACBit1 + 0

    Else

```

ACBit1 = ACBit1 + 1
End If
If FBit2(Y) = 0 Then
ACBit2 = ACBit2 + 0
Else
ACBit2 = ACBit2 + 1
End If

```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
Unload Me
```

```
End Sub
```

```
Sub RetailerReceive()
```

```
Dim NeRe() As Double
```

```
Dim INeRe() As Double
```

```
Dim SNeRe() As Double
```

```
ReDim NeRe(1 To t, 1 To K) As Double 'จำนวนสินค้าที่ให้กับ Retailer ในแต่ละเดือน
```

```
ReDim INeRe(1 To t) As Double 'จำนวนสินค้าที่คงคลังที่ Manu
```

```
ReDim SNeRe(1 To t) As Double 'จำนวนสินค้าที่ขาด
```

```
D = NumBesSumObj
```

```
For A = 1 To t 'จำนวน 12 Month
```

```
If A = 1 Then
```

```
KFC = MutNMan(A, 1, 1, D)
```

```
Else
```

```
KFC = MutNMan(A, 1, 1, D) + INeRe(A - 1)
```

```
End If
```

```
C = 0
```

```
For B = 1 To K 'จำนวน Retailer
```

```
KFC = KFC - De(A, B)
```

```
If KFC > 0 Then
```

```
NeRe(A, B) = De(A, B)
```

```

INeRe(A) = KFC
SNeRe(A) = 0
ElseIf KFC < 0 Then
    If C = 0 Then
        C = C + 1
        NeRe(A, B) = De(A, B) + KFC
        INeRe(A) = 0
        SNeRe(A) = Abs(KFC)
        KFC = 0
    Else
        NeRe(A, B) = 0
        INeRe(A) = 0
        SNeRe(A) = SNeRe(A) + Abs(KFC)
        KFC = 0
    End If
End If

```

Next B

Next A

DT12 = "select ProductID,UnitProduce,UnitInStock,UnitOnOrder,UnitShortage from  
ProductDetail"

Data5.RecordSource = DT12

Data5.Refresh

DT13 = Data5.Recordset.RecordCount

With Data5.Recordset

Data5.Refresh

D = NumBesSumObj

For A = 1 To t

.Edit

.Fields("UnitProduce").Value = MutNMan(A, 1, 1, D)

.Fields("UnitInStock").Value = INeRe(A) 'Inventory <sup>๕</sup> ๖๓ 5 Retailer

.Fields("UnitShortage").Value = SNeRe(A)

.Fields("UnitOnOrder").Value = SumDe(A)

.Update

.MoveNext

Next A

End With

DT16 = "select RetailerID,UnitOnOrder,UnitRecived from RetailerOrder "

Data6.RecordSource = DT16

Data6.Refresh

DT17 = Data6.Recordset.RecordCount

alr = DT4

K = DT2 'b = จำนวน Retailer

t = D 'a = จำนวน 12 Month

With Data6.Recordset

Data6.Refresh

For B = 1 To K

.MoveFirst

A = 0

r1 = 0

For j1 = 1 To alr

o1 = .Fields("RetailerID").Value

If o1 = B Then

A = A + 1

.Edit

.Fields("UnitRecived").Value = NeRe(A, B)

.Update

End If

.MoveNext

Next j1

Next B

End With

End Sub