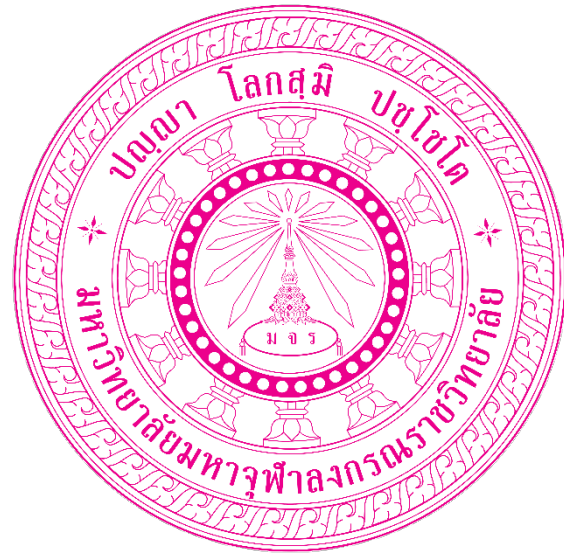


210 305

# การวิเคราะห์และออกแบบระบบ

Systems Analysis and Design



# การประมวลผลข้อมูล



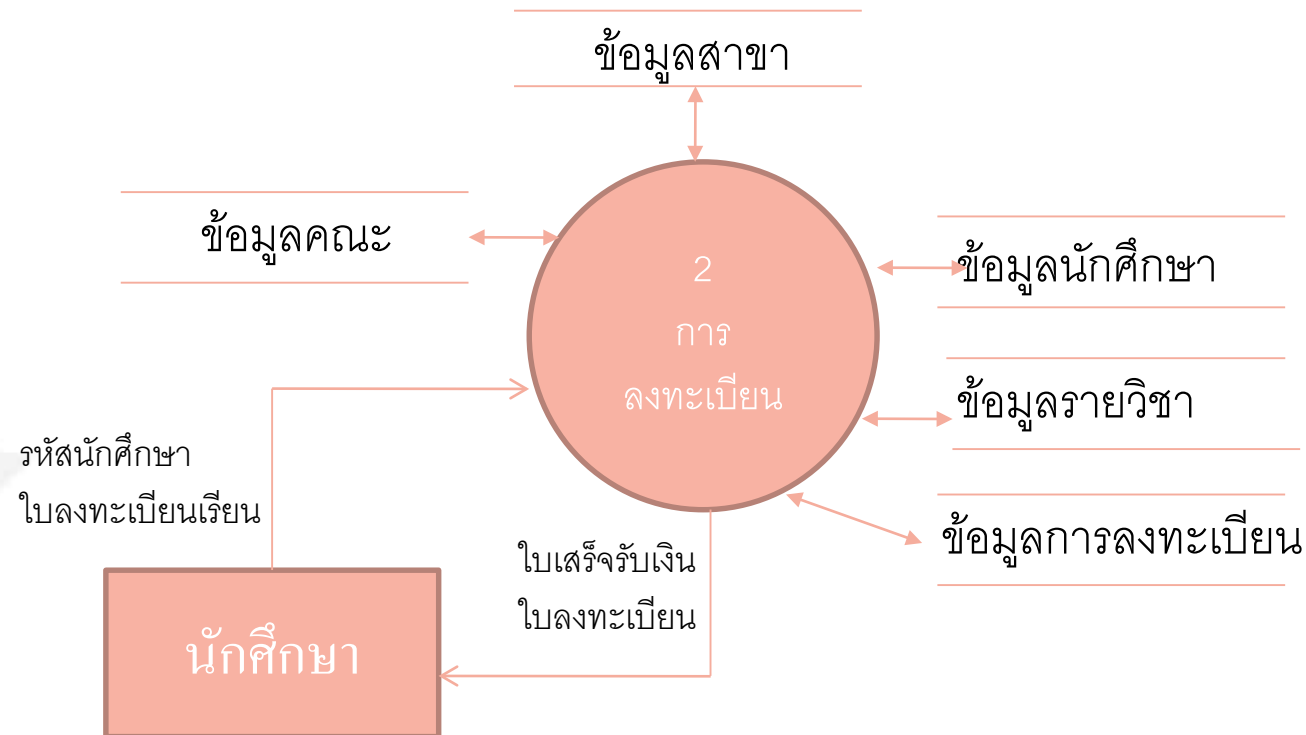
# ความต้องการของผู้ใช้



- แผนภาพกระแสข้อมูล สามารถบรรยายภาพรวมของระบบในลักษณะของแผนภาพ
- การประมวลผลข้อมูลเป็นรายละเอียดหรือเป็นแนวทางหรืออัลกอริทึมการทำงานในแต่ละโปรเซส จึงอธิบายได้ดีกว่าแผนภาพ
- การประมวลผลจะมากหรือน้อยขึ้นอยู่กับจำนวนรายละเอียดที่ใช้งานแต่ละระดับของแผนภาพ

# การประมวลผลระบบทะเบียนนักศึกษา

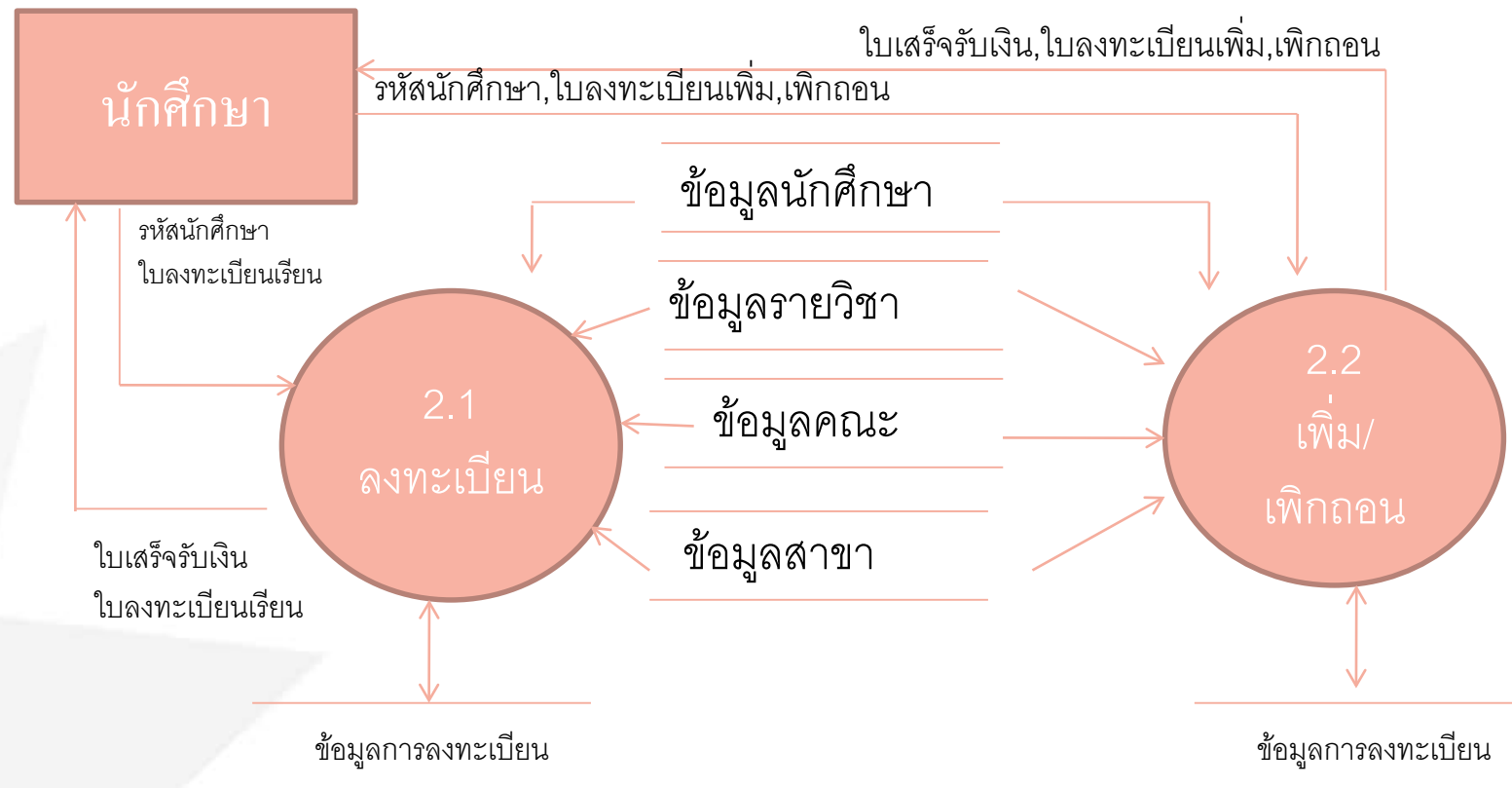
- ทุกๆ กระบวนการหรือทุกๆ โพรเซสในแผนภาพกระแสข้อมูลต้องมีการอธิบายการประมวลผลข้อมูลเสมอ รายละเอียดการประมวลผลจะมีมากน้อยเพียงใดขึ้นอยู่กับการประมวลผล เช่น โพรเซสการลงทะเบียน



# การประมวลผลระบบทะเบียนนักศึกษา



➤ โพรเซสมีการแตกย่อยออกไปอีกคือ



# การประมวลผลระบบทะเบียนนักศึกษา



- สามารถเขียนคำอธิบายการประมวลผลลงทะเบียน

System : ระบบทะเบียนนักเรียน

Process : 2,ลงทะเบียน

Date : dd/mm/yy

Task of Activities 2.1 ลงทะเบียนเรียน

2.2 เพิ่ม/เพิกถอนรายวิชา

# การประมวลผลระบบทะเบียนนักศึกษา



- สามารถเขียนคำอธิบายการประมวลผลลงทะเบียนในรูปแบบภาษาเขียนเพื่ออธิบายข้อมูลต่างๆ

**System** : ระบบทะเบียนนักเรียน

**Process** : ระบบทะเบียนนักศึกษา

**Date** : dd/mm/yy

## Task of Activities

เป็นการลงทะเบียนรายวิชาเรียน ซึ่งนักศึกษาสามารถลงทะเบียนได้สูงสุดไม่เกิน 21 หน่วยกิตสำหรับภาคการศึกษาปกติ และลงทะเบียนเรียนสูงสุดไม่เกิน 9 หน่วยกิตสำหรับภาคฤดูร้อน

# การประมวลผลระบบทะเบียนนักศึกษา



การประมวลผลข้อมูลสามารถเขียนอธิบายในลักษณะอื่นๆได้อีก ดังนี้

1. Structured English Logic
2. Decision Tree
3. Decision Table



# Structured English Logic



## โครงสร้างภาษาอังกฤษ (Structured English Logic)

เป็นโครงสร้างภาษาที่มีลักษณะคล้ายอัลกอริทึม โดยมีคำสั่งเป็นภาษาอังกฤษ

1. รูปแบบของการเขียน Structured English จะมีลักษณะคล้ายกับรูปแบบของการเขียนโปรแกรมแบบโครงสร้าง (Structured Programming) ที่จำแนกมาจากการทำงานของโปรแกรม ซึ่งมี 3 ลักษณะดังนี้
  - แบบตามลำดับ (Sequence)
  - แบบมีเงื่อนไข (Conditional หรือ Decision Structure)
  - แบบการทำซ้ำ (Iteration หรือ Repetition)

# Structured English Logic



โครงสร้างภาษาอังกฤษ (Structured English Logic)  
ประกอบไปด้วยคำสั่งดังนี้

BEGIN REPEAT IF  
END UNTIL THEN  
CASE WHILE ELSE  
OF DO FOR

# Structured English Logic



## 1. แบบตามลำดับ (Sequence)

- มีลักษณะการทำงานเป็นไปตามลำดับขั้นตอนหรือกิจกรรม ไม่มีการกระโดดข้ามไปทำขั้นตอนหรือกิจกรรมอื่นก่อน ดังนั้นในการเขียนคำอธิบาย Process ด้วยการใช้ Structured English แบบตามลำดับนี้ควรมีลักษณะที่เป็นประโยคที่แสดงให้เห็นถึงการทำงานเดียวอย่างชัดเจน
- ตัวอย่างการเขียนแบบตามลำดับ เช่น

```
Read GRADE,CREDIT
```

```
SUM=CREDIT*GRADE
```

```
Print SUM
```

# Structured English Logic



## 2. แบบมีเงื่อนไข (Conditional /Decision Structure)

เป็นการทำงานที่มีการกำหนดการกระทำหรือกิจกรรมการทำงานแตกต่างกันไปตามแต่ละเงื่อนไข ถ้าข้อมูลที่เข้าสู่ Process นั้นเป็นไปตามเงื่อนไขได้ ให้ทำงานภายใต้สิ่งที่เงื่อนไขนั้นกำหนดไว้ โดยรูปแบบการเขียนคำอธิบาย Process โดยใช้ Structured English แบบมีเงื่อนไข แบ่งออกได้ 2 ลักษณะ

# Structured English Logic



## If-then-else

- เป็นโครงสร้างการเขียนแบบมีเงื่อนไขโดยใช้ประโยค IF-THEN-ELSE มาช่วยในการอธิบายลักษณะการทำงานที่จะเกิดการกระทำกิจกรรม (Action) ใดๆ ที่กำหนดไว้ได้ ก็ต่อเมื่อเงื่อนไขที่ระบุไว้นั้นเป็นจริง แต่ถ้าเงื่อนไขนั้นเป็นเท็จจะต้องกระทำกิจกรรมอื่นที่กำหนดไว้ ภายใต้เงื่อนไขที่เป็นเท็จนั้น ดังตัวอย่างต่อไปนี้

- If Normal semester then
- Max\_Credit=21
- Else
- Max\_Credit=9
- End If

# Structured English Logic



## CASE

- เป็นโครงสร้างการเขียนแบบมีเงื่อนไข ซึ่งจะใช้ในกรณีที่มีการกระทำกิจกรรมที่เป็นไปได้มากกว่าสองทาง โดยใช้คำว่า CASE เพื่อตรวจสอบแต่ละเงื่อนไขที่เป็นไปได้เหล่านั้น ด้วยรูปแบบที่ดูง่ายกว่าการใช้ IF-Then-Else If-Then-Else If-Then-Else ...หลายๆครั้ง ดังตัวอย่างต่อไปนี้

```
Select Case Item
```

```
Case 1 : if normal_semester<21 then
```

```
    No Submit_Register
```

```
Case 2 : if normal_semester> 21 then
```

```
    Submit_Register
```

```
Case 3 : If normal_semester = 21 then
```

```
    Submit_Register
```

```
End Select
```

# Structured English Logic



## 3. แบบการทำซ้ำ (Iteration/Repetition)

- เป็นโครงสร้างของการเขียนที่มีลักษณะการกระทำกิจกรรมซ้ำไปเรื่อยๆ ภายใต้เงื่อนไขที่กำหนด ลักษณะการทำซ้ำสามารถเขียนคำอธิบาย Process ด้วย Structured English ได้ดังนี้
- Do-While  
เป็นการกระทำกิจกรรมภายใต้เงื่อนไขที่เป็นจริงเท่านั้น จึงทำกิจกรรมเหล่านั้นซ้ำ จนกระทั่งเงื่อนไขเป็นเท็จ จึงหยุดประมวลผล ดังตัวอย่าง

```
Read Credit
```

```
Do Credit while
```

```
    Credit =21
```

```
End Do
```

# Structured English Logic



- Do-Until

เป็นการกระทำกิจกรรมใดๆ ซ้ำจนกระทั่งเงื่อนไขนั้นเป็นจริง ดังตัวอย่างต่อไปนี้

Do

    Read Credit

        Print Credit

Until Credit =21



## การตัดสินใจแบบต้นไม้ (Decision Tree)

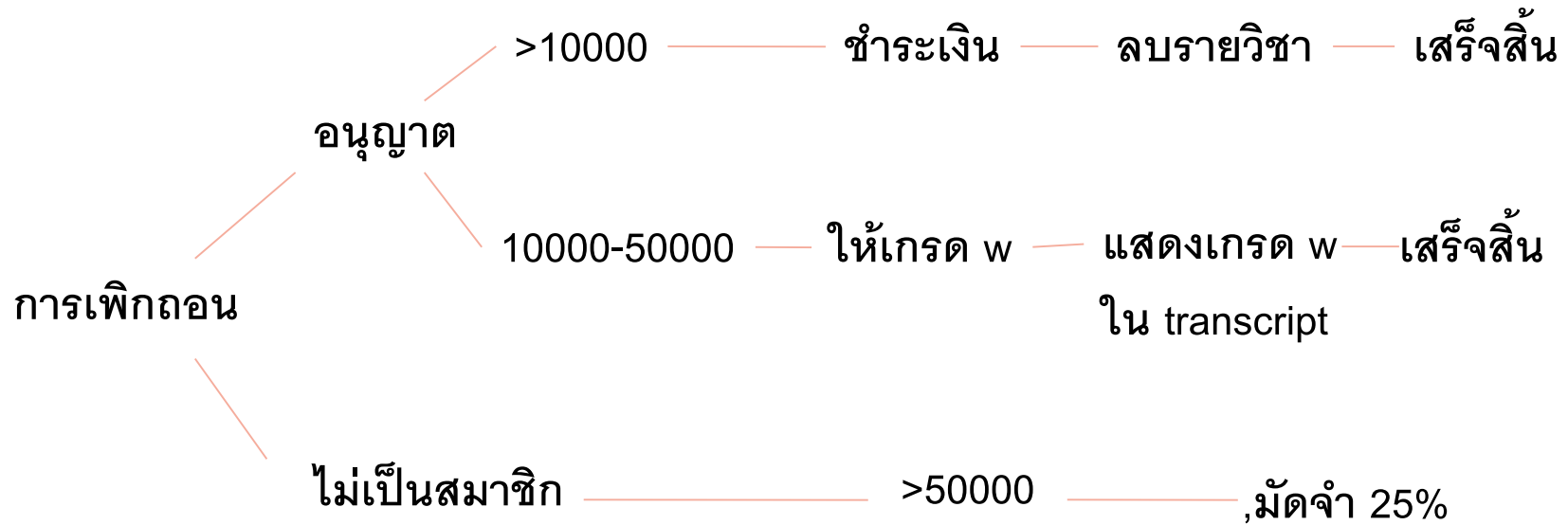
เทคนิคที่ใช้ในการอธิบายการทำงานของ Process โดยการสร้างเงื่อนไขที่มีลักษณะคล้ายต้นไม้ที่แบ่งเงื่อนไขออกเป็นส่วนๆ ให้เห็นโครงสร้างชัดเจน



# Decision Tree



การตัดสินใจแบบต้นไม้ (Decision Tree) การเพิกถอนการลงทะเบียน



# Decision Tree



## การสร้างตารางการตัดสินใจ โดยแบ่งตารางออกเป็นส่วนๆ

- Conditions คือ เงื่อนไขต่างๆ ที่กำหนดขึ้น
- Action คือ ผลของเงื่อนไข ซึ่งได้จากเงื่อนไขต่างๆ มาประมวลจนได้ผลลัพธ์
- Rule คือ กฎเกณฑ์ เป็นการรวมกันของเงื่อนไขและการกระทำอันใดอันหนึ่งที่ระบุว่ากิจกรรมใดที่จะต้องกระทำตามเงื่อนไขใด

Conditions	Rules 1. 2. 3. 4. 5. 6. 7. 8. Etc.
1.... 2.... 3.... 4.... 5... etc.	
Actions	
1.... 2.... 3... etc.	

# Decision Tree



อธิบายการประมวลผลในลักษณะของ Decision Table ของการอนุมัติบัตรเครดิตโดย Conditions คือ เงื่อนไขต่างๆ ซึ่งถ้าเงื่อนไขเป็น Y จะหมายถึงถูกตรงเงื่อนไข ถ้าเป็น N จะไม่ตรงกับเงื่อนไข ส่วน Action คือ การกระทำหรือผลที่เกิดจาก Condition ต่างๆ ว่าจะได้ผลอย่างไร

## Conditions

- |                   |  |
|-------------------|--|
| 1. เป็นสมาชิก     | Y ,N   |
| 2. ยอดการสั่งซื้อ | 1. <10000 บาท<br>2. 10,000 – 50,000 บาท<br>3. >=50,000 บาท |

## Action

- การจ่ายเงินสด
- มัดจำ 25 %
- ให้เครดิต 1 เดือน

# Decision Tree



การสร้างตารางการตัดสินใจ โดยแบ่งตารางออกเป็นส่วนๆ

	1	2	3	4	5	6
Conditions เป็นสมาชิก	Y	Y	Y	N	N	N
ยอดการสั่งซื้อ	<10000	10,000 – 50,000	>=50,000	<10000	10,000 – 50,000	>=50,000
Actions	การจ่ายเงินสด	มัดจำ 25 %	เครดิต 1 เดือน	จ่ายเงินสด	จ่ายเงินสด	มัดจำ 25 %

Y = Yes, condition true

N = No, condition not true



- เป็นการใช้เทคนิคในการเขียนคำอธิบายขั้นตอนการทำงานของระบบ (Process) ที่ปรากฏอยู่บนแผนภาพกระแสข้อมูล (Data Flow Diagram:DFD) ซึ่งในส่วนของ DFD เองไม่สามารถแสดงรายละเอียดลักษณะการทำงานของ Process ที่ต้องมีการตัดสินใจได้ จึงต้องอาศัยเทคนิคในการอธิบายลักษณะการทำงานดังกล่าวได้แก่
1. ภาษาอังกฤษแบบโครงสร้าง (Structured English)
  2. การตัดสินใจแบบต้นไม้ (Decision Tree)
  3. และตารางการตัดสินใจ (Decision Table)

# สรุปผล

- **Structured English** เป็นการเขียนเงื่อนไขและการกระทำด้วยภาษาอังกฤษในลักษณะที่คล้ายกับการเขียนโปรแกรมด้วยภาษาชนิดต่างๆ ซึ่งจะทำให้โปรแกรมเมอร์สามารถอ่านการทำงานได้ง่ายขึ้น
- **Decision Tree** เป็นการแสดงเงื่อนไขการตัดสินใจในรูปแบบโหนด แสดงการกระทำที่เป็นผลลัพธ์จากการทดสอบเงื่อนไขต่างๆ และเชื่อมต่อกันด้วยเส้นตรงเป็นเส้นทางที่แตกแขนงคล้ายกับต้นไม้ เทคนิคชนิดนี้ทำให้ผู้ใช้ระบบสามารถเข้าใจการทำงานได้โดยง่าย
- **Decision Table** เป็นการแสดงเงื่อนไขการตัดสินใจ (Conditions) การกระทำ (Actions) และกฎเกณฑ์ในการตัดสินใจ (Rules) ในรูปแบบตาราง เป็นเทคนิคที่รองรับการทำงานที่มีเงื่อนไขซับซ้อนได้โดยไม่ทำให้ผู้อ่านสับสน



Q & A

Thank you