

## บทที่ 2

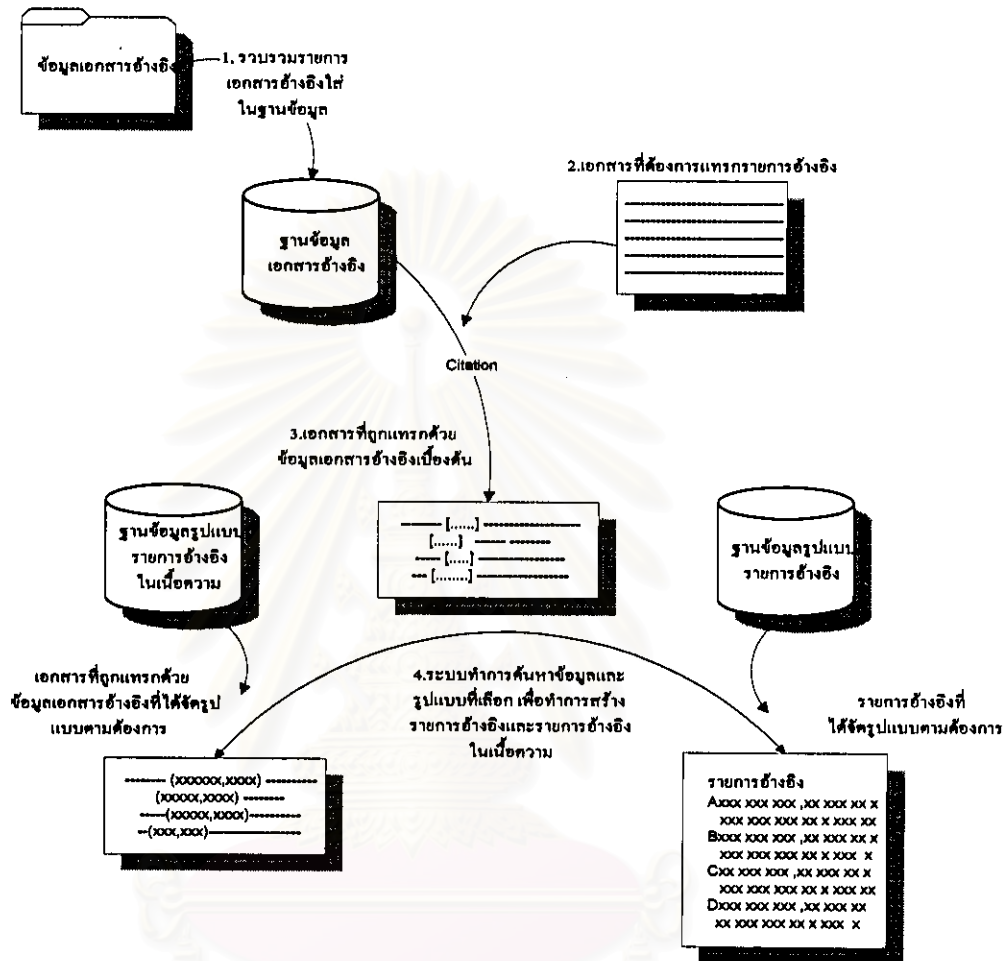
### แนวคิดและทฤษฎีที่เกี่ยวข้อง

บทนี้จะกล่าวถึงแนวคิดและทฤษฎีที่ใช้ในการพัฒนาระบบจัดสร้างฐานข้อมูลเอกสารอ้างอิงและบรรณานุกรม โดยจะแบ่งสิ่งที่กล่าวต่อไปนี้ออกเป็น 2 ส่วนคือ

- 1) ขั้นตอนโดยทั่วไปในการทำงานของการจัดเก็บข้อมูลเอกสารอ้างอิง และการนำข้อมูลเอกสารอ้างอิงนั้นมาใช้อ้างอิงในเอกสาร
- 2) ทฤษฎีที่เกี่ยวข้อง
  - 2.1) โครงสร้างเพิ่มข้อมูลของเวิร์คดูพาบวินโดวส์ (รุ่นจุฬารีก 78)
  - 2.2) โครงสร้างเพิ่มข้อมูลสำหรับฐานข้อมูลเอกสารอ้างอิง
  - 2.3) การนำข้อมูลเข้าหรือออกจากโปรแกรม
  - 2.4) การค้นหาค่า
  - 2.5) การเชื่อมโยงกับเวิร์คดูพา
  - 2.6) การจัดรูปแบบของการอ้างอิง

## 2.1 ขั้นตอนโดยทั่วไปในการทำงาน

โดยทั่วไปไม่ว่าจะใช้คอมพิวเตอร์หรือไม่จะมีขั้นตอนในการทำงานอยู่ 4 ขั้นตอนดังในรูปที่ 1

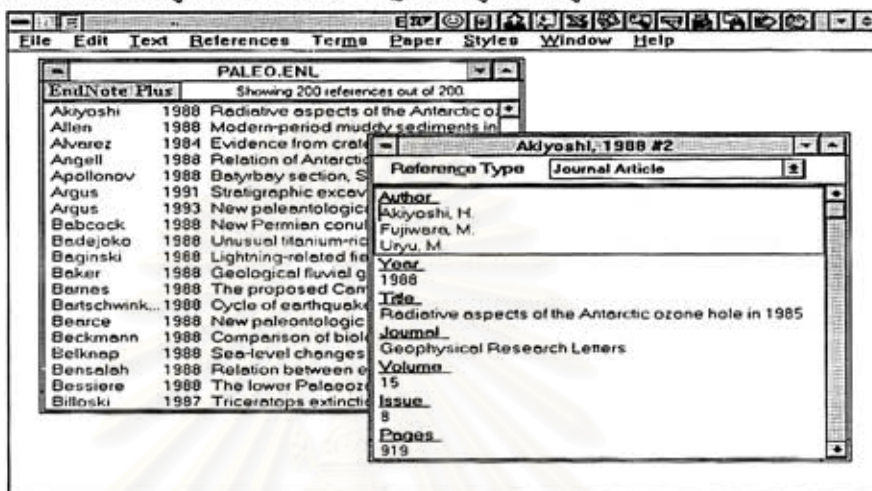


รูปที่ 1 ขั้นตอนการในการอ้างอิงเอกสารอ้างอิง และการจัดสร้างรายการเอกสารอ้างอิง  
จากรูปที่แสดงข้างบนนี้มีขั้นตอนในการทำงานดังต่อไปนี้คือ

- 1) จัดเก็บข้อมูลเอกสารอ้างอิงลงในฐานข้อมูล ซึ่งอาจเป็นบัตรครรชนิ ที่เรียงตามชื่อผู้แต่งหรือหัวเรื่อง หรืออาจเก็บลงในฐานข้อมูลในคอมพิวเตอร์
- 2) ผู้ประพันธ์จัดพิมพ์เนื้อหาความ โดยใช้โปรแกรมประมวลคำ
- 3) ผู้ประพันธ์ทำการอ้างอิงข้อความที่พิมพ์ (In-Text Citation) โดยใช้ข้อมูลจากฐานข้อมูลในรูปแบบที่กำหนด
- 4) ทำการจัดสร้างรายการเอกสารอ้างอิง (Bibliographic Reference) ตามที่ได้อ้างอิงไว้ในข้อความ ในรูปแบบที่กำหนด

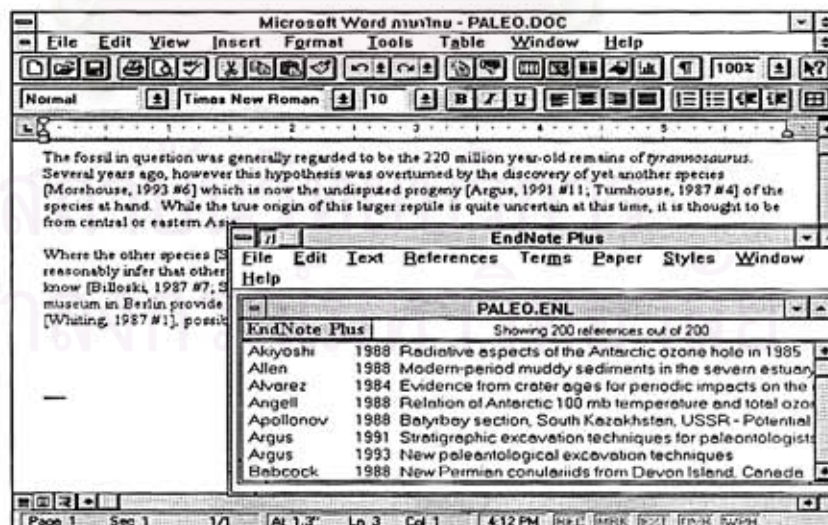
ถ้าเป็นการทำงานโดยการใช้คอมพิวเตอร์โปรแกรม ร่วมกับโปรแกรมประมวลคำ โดยรูปต่อไปนี้เป็นการทำงานของโปรแกรมที่มีอยู่ในต่างประเทศคือ EndNote โดยจะมีลักษณะการทำงานดังต่อไปนี้คือ

- 1) ทำการจัดเก็บข้อมูลเอกสารอ้างอิงลงฐานข้อมูล ตามรูปที่ 2



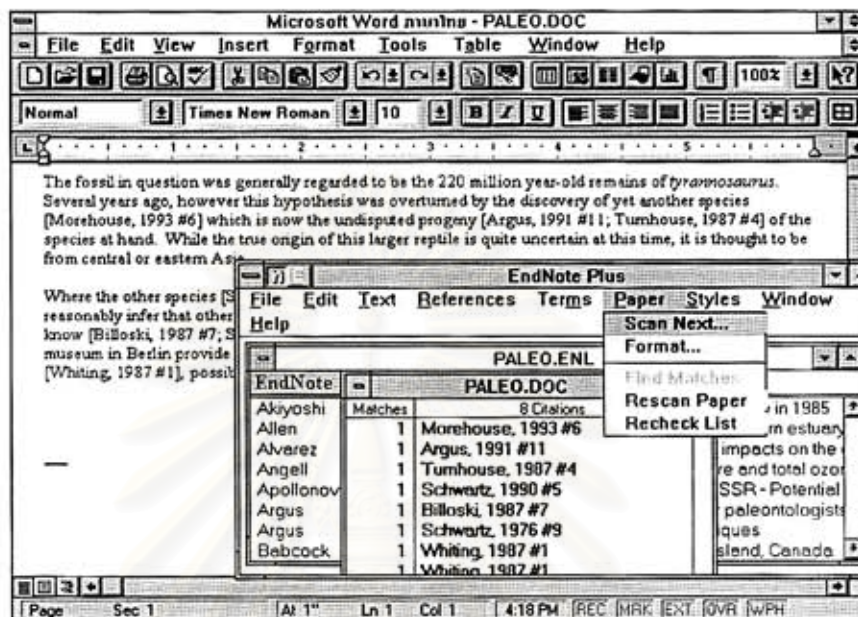
รูปที่ 2 การจัดเก็บข้อมูลเอกสารอ้างอิงลงฐานข้อมูล

- 2) ทำการอ้างอิงได้โดยการเลือกเอกสารอ้างอิงที่ต้องการจากฐานข้อมูลที่เก็บไว้ แล้ววางไว้ที่ตำแหน่งของเอกสารที่ต้องการ ซึ่งโปรแกรมจะทำการแทรกเครื่องหมายอ้างอิง (Citation Marker) และข้อมูลของเอกสารอ้างอิงบางส่วนลงไปในเอกสาร ตามรูปที่ 3



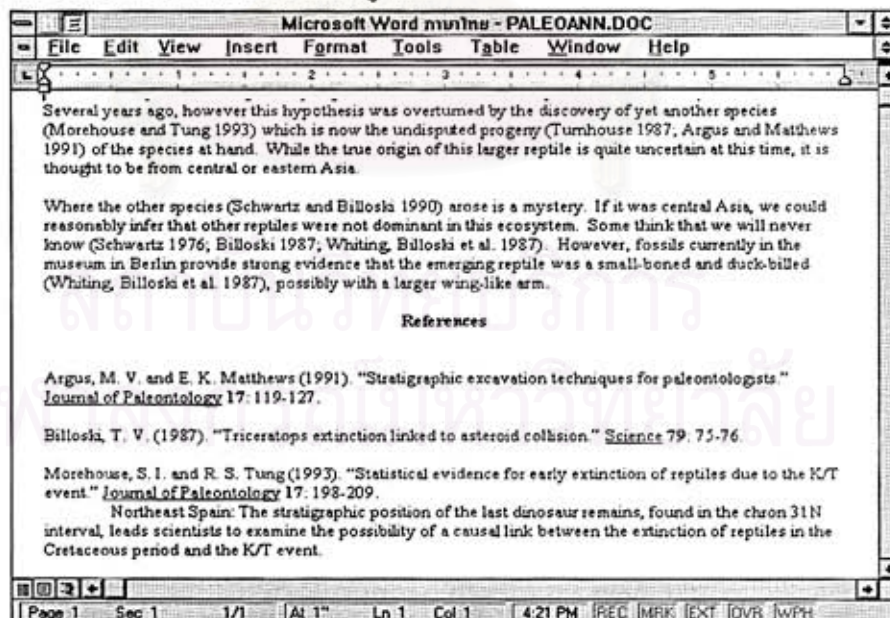
รูปที่ 3 การแทรกการอ้างอิงเอกสารในเนื้อความ

หลังจากนั้นโปรแกรมจะตรวจเอกสารที่ทำเครื่องหมายไว้เพื่อสร้างรายการเอกสารอ้างอิงที่ได้แทรกไว้ในเอกสารต้นฉบับ ตามรูปที่ 4 เพื่อรวบรวมข้อมูลหาเอกสารอ้างอิงที่แทรกไว้ในเอกสาร



รูปที่ 4 การตรวจเอกสารและทำรายการเอกสารอ้างอิง

- 3) จากนั้นทำการจัดรูปแบบเอกสารใหม่ เพื่อจัดสร้างการอ้างอิงในเนื้อความ และรายการอ้างอิงตามแบบที่ได้เลือกไว้ ตามรูปที่ 5



รูปที่ 5 ผลลัพธ์การจัดรูปแบบเอกสารใหม่

### 2.1.1 ส่วนประกอบของระบบ

จากขั้นตอนการทำงานการอ้างอิงเอกสารข้างต้น จะเห็นได้ว่าต้องมียกประกอบที่จำเป็นดังต่อไปนี้

- 1) ฐานข้อมูลเอกสารอ้างอิง
- 2) การสร้างการอ้างอิงและรายการอ้างอิงโดยใช้ข้อมูลจากฐานข้อมูลอ้างอิง เป็นการนำเอาข้อมูลจากฐานข้อมูลที่เก็บไว้มาใช้งานซึ่งมีอยู่ 2 รูปแบบคือ
  - 2.1) การอ้างอิงแบบแทรกในเนื้อหา (In-Text Citation)
  - 2.2) รายการอ้างอิง (References) หรือบรรณานุกรม (Bibliography) เป็นการบอกที่มาของแหล่งข้อมูลที่มีการใช้อยู่ 2 แบบคือ
    - 2.2.1) แบบท้ายบทความ (Endnote Bibliography) เป็นรายการเอกสารอ้างอิงที่เรียบเรียงไว้ด้านท้ายบทความในเอกสารทางวิชาการ ซึ่งอาจเรียงตามชื่อผู้แต่ง หรือลำดับของเอกสารอ้างอิงที่เก็บ
    - 2.2.2) แบบเชิงอรรถ (Footnote Bibliography) เป็นรายการเอกสารอ้างอิงที่เรียบเรียงไว้ด้านล่างเอกสารแต่ละหน้า

#### 2.1.1.1 ฐานข้อมูลเอกสารอ้างอิง

เป็นที่รวมข้อมูลที่เกี่ยวข้องกับเอกสารอ้างอิง ซึ่งจะใช้อ้างอิงถึงในเอกสารทางวิชาการ หรือรายงาน เช่นใน วิทยานิพนธ์ หนังสือ บทความ เป็นต้น ซึ่งสามารถเก็บ โดยใช้กระดาษเก็บข้อมูลของเอกสารอ้างอิงหนึ่งชิ้นต่อหนึ่งแผ่น แล้วนำมาเรียงลำดับตามชื่อผู้แต่ง ปีที่พิมพ์ หรือตามหัวเรื่องซึ่งไม่สะดวกนักในการค้นหา หรือเรียงลำดับ ซึ่งถ้าใช้ฐานข้อมูลที่เก็บในคอมพิวเตอร์ที่ออกแบบให้เหมาะสมสำหรับการค้นหา และเรียกใช้ จะทำให้ใช้งานได้สะดวกกว่าการใช้กระดาษเก็บข้อมูล

ฐานข้อมูลที่เหมาะสมกับงานวิจัยนี้ควรมีลักษณะดังต่อไปนี้

- 1) เก็บข้อมูลเอกสารอ้างอิงได้หลายประเภท

เนื่องจากเอกสารอ้างอิงนั้นมีอยู่หลายประเภทเช่น หนังสือ วารสาร บทความ นิตยสาร หนังสือพิมพ์ วิทยานิพนธ์ เป็นต้น ซึ่งข้อมูลที่เก็บและใช้จากเอกสารอ้างอิงเหล่านั้นก็แตกต่างกันไป ทำให้ฐานข้อมูลนั้นจะต้องมีการเก็บข้อมูลที่ยืดหยุ่นพอที่จะเก็บข้อมูลเหล่านั้นได้ครบถ้วน

## 2) สามารถเรียงลำดับข้อมูลถูกต้องตามหลักภาษาและข้อกำหนด

การเรียงลำดับข้อมูลจาก พิลด์เดียว หรือหลายพิลด์ ต้องสามารถเรียงลำดับภาษาไทย-อังกฤษได้อย่างถูกต้องตามหลักภาษา และยังมีกฎเกณฑ์บางอย่างของการอ้างเอกสารอ้างอิง ที่ใช้ในการเรียงลำดับเช่น ถ้าต้องการเรียงลำดับตามชื่อผู้แต่ง แต่เอกสารอ้างอิงบางชิ้นไม่ปรากฏชื่อผู้แต่ง ให้เรียงลำดับตามชื่อเรื่องแทน และให้เข้ากับข้อกำหนดของสถาบันที่จะรับงานพิมพ์นั้น

## 3) สามารถค้นหาข้อมูลได้ทั้งไทย-อังกฤษ ด้วยคีย์ที่ไม่ใช่ คีย์หลัก

การค้นหาข้อมูลที่ไม่ใช่คีย์หลักเช่นชื่อผู้แต่ง, คำสำคัญ (Keyword), ชื่อวารสาร เป็นต้น

### 2.1.1.2 การสร้างการอ้างอิงและรายการอ้างอิง

นอกจากข้อมูลจากฐานข้อมูลอ้างอิงที่เก็บไว้แล้ว การสร้างการอ้างอิงและรายการอ้างอิงในรูปแบบที่ต้องการในเอกสารก็มีความสำคัญเช่นกัน โดยสิ่งที่ระบบควรมีต่อไปคือ

- 1) มีฐานข้อมูลที่เก็บรูปแบบของการอ้างอิงที่กำหนดของต่างประเทศ ในประเทศ และที่ผู้ใช้กำหนดได้
- 2) ความสามารถที่จะสร้างรายการอ้างอิงในเอกสารที่สร้างด้วยโปรแกรมประมวลคำที่ได้กำหนดไว้ ในที่นี้คือเวิร์ดจูปา

## 2.2 ทฤษฎีที่เกี่ยวข้อง

จากหลักการทำงานโดยทั่วไปที่กล่าวมา จนวิเคราะห์ได้สิ่งที่ควรจะมีในระบบแล้วสิ่งที่จะกล่าวต่อไปคือทฤษฎีที่เกี่ยวข้องกับการวิจัย ออกแบบและการสร้างระบบจากความต้องการ และขอบเขตการวิจัยที่ได้กล่าวมาแล้ว มีดังต่อไปนี้

- 1) โครงสร้างเพิ่มข้อมูลของเวิร์ดจูปาบนวินโดวส์ (รุ่นจูปาจาริก 78)
- 2) โครงสร้างเพิ่มข้อมูลสำหรับฐานข้อมูลเอกสารอ้างอิง
- 3) การนำข้อมูลเข้าหรือออกจากโปรแกรม
- 4) การค้นหาค่า
- 5) การเชื่อมโยงกับเวิร์ดจูปา
- 6) การจัดรูปแบบของการอ้างอิง

## 2.2.1 โครงสร้างเพิ่มข้อมูลของเวิร์ดจุฬานวินโดวส์ (รุ่นจุฬจารึก 78)

โครงสร้างเพิ่มข้อมูลจุฬจารึก 78 (เขาวลัทธิ, 1995) แบ่งเป็น 10 ส่วนคือ header, Style, Section, Font, Picture, Text, Index, Attribute, Overlay Drawing และ TTF Embedded โดยแต่ละส่วนมีรายละเอียดต่อไปนี้

MAXDOCINFO	= 81
MAXPASSWORD	= 9
MAXSTYLENAME	= 20
MAXTAB	= 10
MAXPAGETITLE	= 20

### 2.2.1.1 ส่วน Header มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
DWORD	dwIndent
WORD	dwSize
DWORD	pgStyle
DWORD	pgSection
DWORD	pgText
DWORD	pgIdx
DWORD	pgAttr
DWORD	pgFont
DWORD	pgFont
DWORD	pgPicture
DWORD	pgDrawing
DWORD	pgTTFEmbed
int	ttfembed
int	fitzoom
int	deftab
int	viewallchar
int	autosavemin
int	compressencrypt
int	bReadOnly
char	Title[MAXDOCINFO]
char	Subject[MAXDOCINFO]
char	Author[MAXDOCINFO]
char	Keywords[MAXDOCINFO]
char	Password[MAXDOCINFO]
int	bSnapTo
int	xGrid
int	yGrid
DWORD	unused1
DWORD	unused2
DWORD	unused3
DWORD	unused4

ตารางที่ 1 โครงสร้างข้อมูลเวิร์ดจุฬาส่วน Header

### 2.2.1.2 ส่วน Style มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
WORD	id
DWORD	size
DWORD	numstyle
STYLEINFO	styleinfo[]

ตารางที่ 2 โครงสร้างข้อมูลเวิร์คดูพาส่วน Style

#### 1) STYLEINFO มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
char	stylename[MAXSTYLENAME]
int	leftindent
int	rightindent
int	firstindent
int	spacebefore
int	spaceafter
int	linespacing
BYTE	alignment
TABINFO	tabinf[MAXTAB]
BYTE	textcolor
BYTE	textattribute
BYTE	floating
BYTE	bullet
BYTE	bulletcode
BYTE	bulletcolor
int	bullettotext
int	indenttotext
WORD	bullefontidx
UINT	fontindex
FONTSTYLEINFO	textfont
FONTSTYLEINFO	bulletfont

ตารางที่ 3 โครงสร้างข้อมูลเวิร์คดูพาส่วน StyleInfo

#### 2) TABINFO มีโครงสร้างข้อมูลดังนี้

Data Type	Variable name
BYTE	type
INT	tabpos

ตารางที่ 4 โครงสร้างข้อมูลเวิร์คดูพาส่วน TabInfo

#### 3) FONTSTYLEINFO มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
char	fontname[LF_FACESIZE]
int	point
BYTE	bold
BYTE	italic
int	npoint

ตารางที่ 5 โครงสร้างข้อมูลเวิร์คดูพาส่วน FontStyleInfo



### 2.2.1.3 ส่วน Section มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
WORD	id
DWORD	size
DWORD	numsection
WORD	papertype
WORD	papersource
WORD	papaerwidth
WORD	paperheight
WORD	orientation
WORD	leftmargin
WORD	rightmargin
WORD	topmargin
WORD	bottommargin
WORD	headermargin
WORD	footermargin
int	pagewidth
int	pageheight
int	pageno
int	startpage
BYTE	pagetype
int	pagexpos
int	pageypos
char	pagetitle[MAXPAGETITLE]
BYTE	pagelang
UINT	pagestyleidx
DWORD	numheaderpara
PARAGRAPHINFO	headerpargraphinfo[]
DWORD	numbodypara
PARAGRAPHINFO	bodyparagraphinfo[]
DWORD	numfooterpara[]
PARAGRAPHINFO	footerparagraphinfo[]

ตารางที่ 6 โครงสร้างข้อมูลเวิร์ดจุฬาส่วน Section

1) โดย PARAGRAPHINFO มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
int	leftindent
int	rightindent
int	firstindent
int	spacebefore
int	spaceafter
int	linespacing
BYTE	bullet
BYTE	bulletcode
BYTE	bulletcolor
WORD	bulletfontidx
int	bullettotext
int	indenttotext
BYTE	alignment
TABINFO	tabinf[MAXTAB]
BYTE	textcolor
BYTE	texattribute
BYTE	floating
UINT	styleidx
UINT	fontindex
WORD	numdrawinfo
DWORD	unused1
DWORD	unused2

ตารางที่ 7 โครงสร้างข้อมูลเวิร์คดูพาส่วน ParagraphInfo

2.2.1.4 ส่วน Font มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
WORD	id
DWORD	size
DWORD	numfont
FONTINFO	fontinfo[]

ตารางที่ 8 โครงสร้างข้อมูลเวิร์คดูพาส่วน Font

1) โดย FONTINFO มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
char	fontname[LF_FACESIZE]
int	point
BYTE	bold
BYTE	italic
int	npoint
BYTE	pitchandfamily
BYTE	owner
int	suboffset
int	superoffset

ตารางที่ 9 โครงสร้างข้อมูลเวิร์คดูพาส่วน FontInfo

### 2.2.1.5 ส่วน Picture มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
WORD	id
DWORD	size
DWORD	numpicture
PICTUREINFO	pictureinfo[]

ตารางที่ 10 โครงสร้างข้อมูลเวิร์คบุคส่วน Picture

#### 1) โดย PICTUREINFO มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
int	width
int	height
int	orgwidth
int	orgheight
int	vertadj
BYTE	type
int	scalex
int	scaley
int	cropleft
int	croptop
int	cropright
int	cropbottom
DWORD	picturesize

ตารางที่ 11 โครงสร้างข้อมูลเวิร์คบุคส่วน PictureInfo

### 2.2.1.6 ส่วน Text มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
WORD	id
DWORD	size
BYTE	text[]

ตารางที่ 12 โครงสร้างข้อมูลเวิร์คบุคส่วน Text

### 2.2.1.7 ส่วน Index มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
WORD	id
DWORD	size
INDEXINFO	indexinfo[]

ตารางที่ 13 โครงสร้างข้อมูลเวิร์คบุคส่วน Index

#### 1) โดย INDEXINFO มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
WORD	index
WORD	numuseindex

ตารางที่ 14 โครงสร้างข้อมูลเวิร์คบุคส่วน IndexInfo

### 2.2.1.8 ส่วน Attribute มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
WORD	id
DWORD	size
ATTRIBUTEINFO	attributeinfo[]

ตารางที่ 15 โครงสร้างข้อมูลเวิร์คดูพาส่วน Attribute

#### 1) ATTRIBUTEINFO มีโครงสร้างข้อมูลดังนี้

Data Type	Variable name
BYTE	attribute
WORD	numusedattribute

ตารางที่ 16 โครงสร้างข้อมูลเวิร์คดูพาส่วน AttributeInfo

### 2.2.1.9 ส่วน Overlay มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
WORD	id
DWORD	size
DWORD	number
BYTE	type
BYTE	FillColor
BYTE	FillPattern
BYTE	PatColor
BYTE	LineStyleWidth
BYTE	LineColor
BYTE	LineFlag
BYTE	ArrowHead
BYTE	HVPos
int	xPos
int	yPos
int	width
int	height
int	numpoint

ตารางที่ 17 โครงสร้างข้อมูลเวิร์คดูพาส่วน Overlay

### 2.2.1.10 ส่วน TTF Embedded มีโครงสร้างข้อมูลดังนี้

Data Type	Variable Name
WORD	id
DWORD	size
DWORD	number

ตารางที่ 18 โครงสร้างข้อมูลเวิร์คดูพาส่วน TTFEmbedded

## 2.2.2 โครงสร้างเพิ่มข้อมูลสำหรับฐานข้อมูลเอกสารอ้างอิง

การจัดระเบียบเพิ่มข้อมูล (File Organization) (Miller, 1987) ที่จะใช้เก็บข้อมูลเอกสารอ้างอิงนั้นมีได้หลายแบบเช่น

- 1) แบบเรียงลำดับ (Sequential)
- 2) แบบสุ่ม (Random)
- 3) แบบลำดับครรรชนี (Indexed Sequential)
- 4) แบบมัลติเปิด-คีย์ (Multiple-Key)

### 2.2.2.1 การจัดระเบียบเพิ่มข้อมูลแบบเรียงลำดับ

เป็นการบันทึกระเบียบข้อมูลแบบเรียงตามลำดับ ในการเข้าถึงข้อมูลก็ต้องทำตามลำดับการบันทึกเช่นเดียวกัน

### 2.2.2.2 การจัดระเบียบเพิ่มข้อมูลแบบสุ่ม

เป็นการจัดเพิ่มข้อมูลที่มีคีย์ฟิลด์ (Key Field) มีความสัมพันธ์กับตำแหน่งของระเบียบในเพิ่มข้อมูล ทำให้การเข้าถึงระเบียบข้อมูลไม่ต้องผ่านตามลำดับการจัดเก็บ โดยสามารถไปยังระเบียบใดๆ ก็ได้โดยการคำนวณจากคีย์ ที่นำมาใช้กับเพิ่มข้อมูลแบบสัมพัทธ์ (Relative File) เป็นรูปแบบเพิ่มข้อมูลที่ใช้กับโครงสร้างเพิ่มข้อมูลแบบสุ่ม

ซึ่งเพิ่มข้อมูลแบบสุ่มประกอบด้วย 2 ส่วนคือ

- 1) การจัดการครรรชนี
- 2) การจัดเก็บข้อมูลในทางกายภาพ

ซึ่งการทำครรรชนี มีหลายวิธี

- 1) การกำหนดเลขที่อยู่โดยตรง (Direct Addressing) ซึ่งมีทั้งแบบสัมบูรณ์ และแบบสัมพัทธ์ (Absolute Addressing and Relative Addressing)
- 2) แฮชชิง (Hashing)
- 3) ต้นไม้ค้นหาแบบทวิภาค (Binary Search Trees)
- 4) ต้นไม้แบบบี (B-Trees)
- 5) การทำครรรชนีแบบมัลติเปิด-คีย์ (Multiple-Key Indexing)

### 2.2.2.3 การจัดระเบียบแฟ้มข้อมูลแบบลำดับครรรชนี

รวมการจัดระเบียบโครงสร้างแฟ้มข้อมูลแบบเรียงลำดับ และแบบสุ่มเข้าด้วยกัน โดยมี อยู่ 2 ส่วนคือ

- 1) กลุ่มของระเบียบที่อยู่เรียงลำดับตามคีย์ฟิลด์ อยู่ในบล็อก (Blocks) ของ แฟ้มข้อมูลแบบสัมพันธ์
- 2) ครรรชนีของคีย์ระเบียบและ เลขบล็อกที่ชี้ไปยังแฟ้มข้อมูลแบบสัมพันธ์ นั้น

การเข้าถึงข้อมูลต้องค้นหาจากครรรชนี โดยคีย์ที่ได้มาก่อน จากนั้นก็จะได้เลขบล็อกที่ ระเบียบนั้นควรจะอยู่ จากนั้นก็ตรงไปที่บล็อกนั้นทำการค้นหาตามลำดับ ว่ามีระเบียบที่ต้องการหรือไม่

### 2.2.2.4 การจัดระเบียบแฟ้มข้อมูลแบบมัลติเปิด-คีย์

สามารถค้นหาข้อมูลโดยใช้หลายคีย์ฟิลด์ได้ ปกติจะมีฟิลด์หนึ่งที่เป็น ยูนิคคีย์ (Unique Key) ที่สามารถแยกระเบียนจากกันได้เรียกว่าฟิลด์คีย์หลัก ถ้าในการเข้าถึงข้อมูลโดยใช้ฟิลด์อื่นที่ไม่ใช่คีย์หลัก เรียกว่าฟิลด์คีย์รอง (Secondary Key Fields)

โดยการจัดระเบียบแฟ้มข้อมูลที่ได้กล่าวมาแล้ว แบบเรียงลำดับ แบบสุ่ม และ แบบลำดับครรรชนี เป็นการจัดระเบียบแฟ้มข้อมูลแบบที่ใช้คีย์เดียวซึ่งใช้คีย์หลักตัวเดียวในการเข้าถึงข้อมูล ซึ่งถ้าต้องการหาระเบียนทั้งหมดโดยใช้ฟิลด์อื่นที่ไม่ใช่คีย์หลัก ต้องค้นหาข้อมูลแบบเรียงลำดับ ซึ่งเสียเวลานาน

การจัดระเบียบแฟ้มข้อมูลแบบมัลติเปิด-คีย์ จะสร้างความสัมพันธ์ของข้อมูลในระเบียบกับคีย์หลัก และยอมให้มีการเข้าถึงข้อมูลโดยการใช้ฟิลด์อื่นในการค้นหาจากคีย์หลัก คล้ายกับครรรชนีของหนังสือที่สามารถจะค้นหาข้อมูลได้โดยค้นหาจากคำที่สนใจ แทนที่จะเป็นเลขหน้าหนังสือ ซึ่งในที่นี้เลขหน้าหนังสือเป็นคีย์หลักที่ไม่ซ้ำเลขกัน และคำที่ใช้ค้นหาเป็นคีย์รอง ซึ่งให้เลขหน้าหนังสือ โดยอาจมีเพียงหน้าเดียวหรือหลายหน้าก็ได้ ซึ่งเป็นความสัมพันธ์ของข้อมูลกับ คีย์หลัก

ความสำคัญของครรรชนีของคีย์รอง คือเป็นโครงสร้างอีกอย่างหนึ่งที่เพิ่มเติมเข้ามาบนโครงสร้างแฟ้มข้อมูลที่มีอยู่แล้วไม่ว่าจะเป็นแฟ้มข้อมูลแบบเรียงลำดับ หรือสัมพันธ์ ซึ่งใช้คีย์หลักในการเข้าถึงข้อมูล ซึ่งเมื่อใช้คีย์รองในการค้นหาจากครรรชนีนั้นจะได้กลุ่มของคีย์หลัก ซึ่งสามารถนำไปใช้ในการหาระเบียนของข้อมูลต่อไปได้

ซึ่งมีการจัดเพิ่มข้อมูลแบบมัลติเปิล-คีย์ อยู่ 2 แบบคือ

- 1) การจัดเพิ่มข้อมูลแบบผกผัน (Inverted File Organization)
- 2) การจัดเพิ่มข้อมูลแบบมัลติลิสต์ (Multilist File Organization)

ซึ่งทั้งสองแบบจะสร้างดัชนีสำหรับแต่ละฟิลด์คีย์รองขึ้นมา คล้ายกับกรณีของหนังสือ ความแตกต่างของสองวิธีคือ วิธีการในการแสดงลิสต์ของคีย์หลักสำหรับคีย์รองแต่ละตัวที่เก็บ

#### 2.2.2.4.1 การจัดเพิ่มข้อมูลแบบผกผัน

โดยการจัดเพิ่มข้อมูลแบบผกผัน จะสร้างดัชนี โดยการสร้างลิงค์ลิสต์ (Linked List) ของคีย์หลักที่สัมพันธ์กับคีย์รองที่เกี่ยวข้อง ดังแสดงในตัวอย่างต่อไปนี้

Primary Key	Secondary Key	
Ref. Id	Author	Journal
1	Birnstein ML	CELL
2	Ciliberto G	CURRENT TOPICS IN DEVELOPMENTAL BIOLOGY
3	Parker CS	CELL
4	Parker CS	CELL
5	Kadonaga JT	TRENDS IN BIOCHEMICAL SCIENCES
6	DasGupta C	CELL
7	DasGupta C	CELL
8	Gonda DK	CELL
9	Julin DA	J BIOL CHEM
10	Lin VK	J CELL BIOCHEM
11	Lin VK	J CELL BIOCHEM
12	Gross DS	TRENDS IN BIOCHEMICAL SCIENCES

ตารางที่ 19 ข้อมูลตัวอย่างข้อมูลเอกสารอ้างอิง

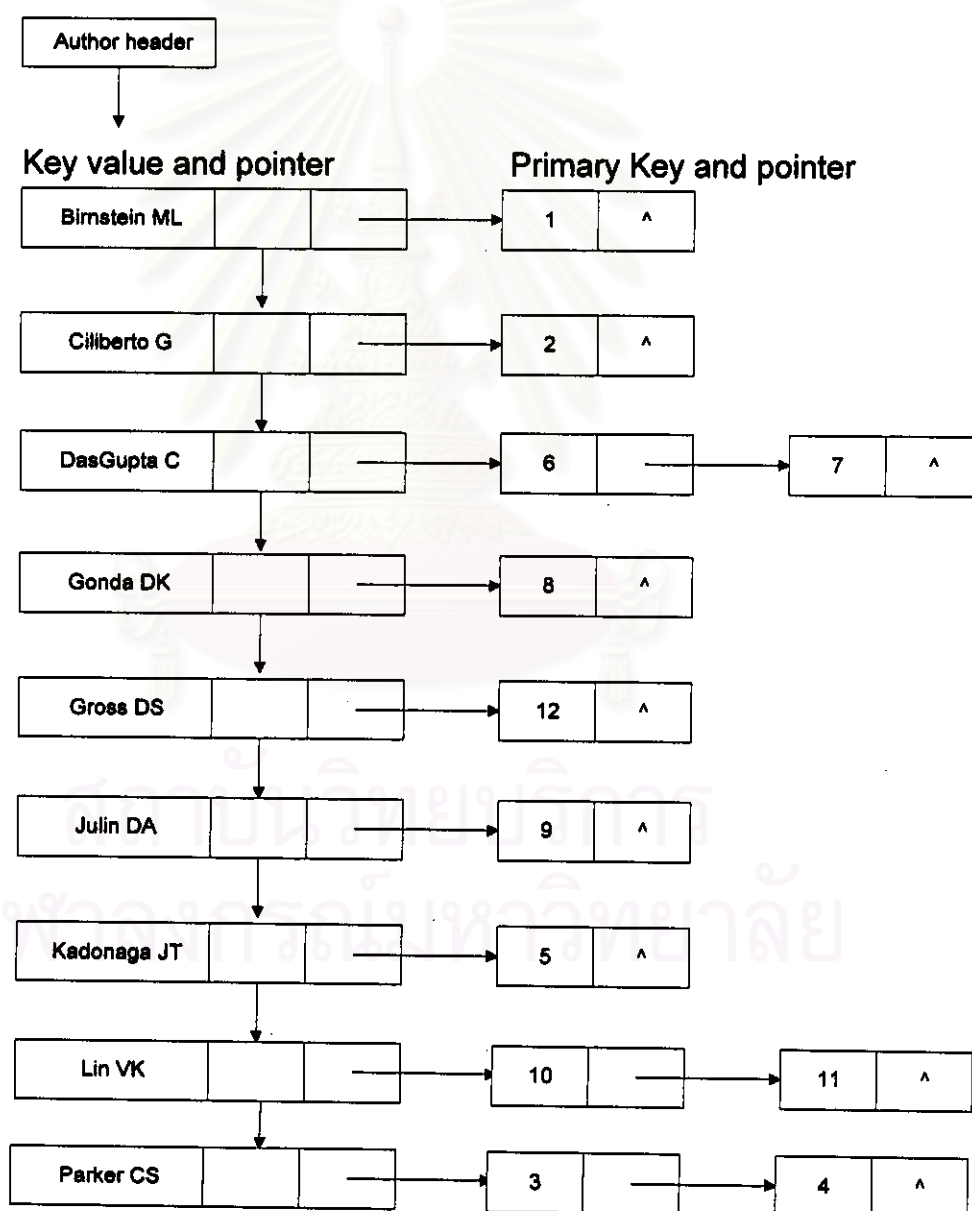
Author	Primary KeyRef. Id
Birnstein ML	1
Ciliberto G	2
Parker CS	3,4
Kadonaga JT	5
DasGupta C	6,7
Gonda DK	8
Julin DA	9
Lin VK	10,11
Gross DS	12

ตารางที่ 20 ลิสต์ผกผันของฟิลด์ Author

Journal	Primary Key Ref. Id
CELL	1,3,4,6,7,8
CURRENT TOPICS IN DEVELOPMENTAL BIOLOGY	2
TRENDS IN BIOCHEMICAL SCIENCES	5,12
J BIOL CHEM	9
J CELL BIOCHEM	10,11

ตารางที่ 21 ลิขสิทธิ์ของฟิลด์ Journal

รายการผูกพันของฟิลด์ Author ในโครงสร้างที่โยงกัน



รูปที่ 6 รูปแบบการเก็บข้อมูลแบบโยงของรายการผูกพัน



จากรูปที่ 6 จะเห็นได้ว่ามีคีย์อยู่สองประเภทคือ

- 1) รูปแบบในการเก็บค่าคีย์ และตัวชี้ไปยังคีย์หลักในรายการที่อยู่

Key value	Next value	First key
-----------	------------	-----------

ตารางที่ 22 รูปแบบการเก็บค่าคีย์ และตัวชี้

- 2) รูปแบบในการเก็บค่าคีย์หลัก และตัวชี้ไปยังคีย์หลักตัวอื่นในรายการที่อยู่

Primary key	Next value
-------------	------------

ตารางที่ 23 รูปแบบการเก็บค่าคีย์หลัก

#### 2.2.2.4.2 การจัดเพิ่มข้อมูลแบบมัลติลิสต์

มัลติลิสต์จะสร้างกรณีคล้ายกับเพิ่มผกผันแต่ที่ต่างกันคือเก็บเพียงค่าของคีย์หลักตัวแรกของรายการเชื่อมโยงและความยาวของลิสต์ ส่วนเพิ่มข้อมูลก็เพิ่มฟิลด์ของตัวโยง สำหรับแต่ละคีย์รอง ดังตัวอย่าง

เพิ่มมัลติลิสต์ที่มีฟิลด์โยงสำหรับข้อมูลของ reference สำหรับฟิลด์รองคือ Author กับ Journal มีลักษณะดังต่อไปนี้

- 1) Author

Author	Primary Key Ref. Id	Length
Birnstein ML	1	1
Ciliberto G	2	1
Parker CS	3	2
Kadonaga JT	5	1
DasGupta C	6	2
Gonda DK	8	1
Julin DA	9	1
Lin VK	10	2
Gross DS	12	1

ตารางที่ 24 ตารางเพิ่มข้อมูลมัลติลิสต์ของ Author

- 2) Journal

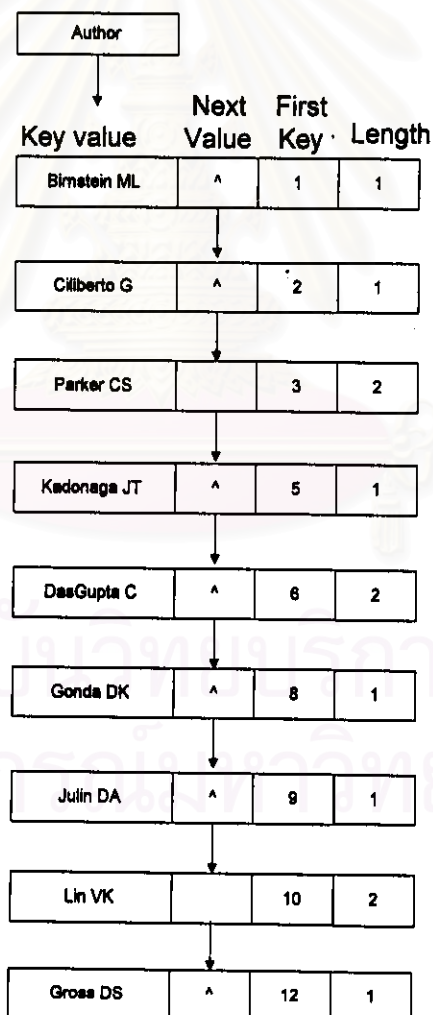
Journal	Primary Key Ref. Id	Length
CELL	1	6
CURRENT TOPICS IN DEVELOPMENTAL BIOLOGY	2	1
TRENDS IN BIOCHEMICAL SCIENCES	5	2
J BIOL CHEM	9	1
J CELL BIOCHEM	10	2

ตารางที่ 25 ตารางเพิ่มข้อมูลมัลติลิสต์ของ Journal

ตารางต่อไปนี้เป็นตารางเพิ่มข้อมูลในแบบมัลติลิสต์

Primary Key	Secondary Key			
Ref. Id	Author	Link	Journal	Link
1	Birnstein ML	^	CELL	3
2	Ciliberto G	^	CURRENT TOPICS IN DEVELOPMENTAL BIOLOGY	^
3	Parker CS	4	CELL	4
4	Parker CS	^	CELL	6
5	Kadonaga JT	^	TRENDS IN BIOCHEMICAL SCIENCES	12
6	DasGupta C	7	CELL	7
7	DasGupta C	^	CELL	8
8	Gonda DK	^	CELL	^
9	Julin DA	^	J BIOL CHEM	^
10	Lin VK	11	J CELL BIOCHEM	11
11	Lin VK	^	J CELL BIOCHEM	^
12	Gross DS	^	TRENDS IN BIOCHEMICAL SCIENCES	^

ตารางที่ 26 เพิ่มข้อมูลในแบบมัลติลิสต์



รูปที่ 7 โครงสร้างแบบโยงสำหรับฟิลด์ของ Author

## โครงสร้างโยงที่ใช้มีโครงสร้างดังตารางที่ 27

Key value	Next value	First key	Length
-----------	------------	-----------	--------

### ตารางที่ 27 รูปแบบสำหรับโครงสร้างโยง

ข้อดีของเพิ่มข้อมูลแบบมัลติลิสต์ ที่เหนือกว่าเพิ่มข้อมูลแบบผกผัน คือใช้ความยาวระเบียบแบบคงที่ในการเก็บครรรชนี แต่มีข้อเสียคือเสียเวลามากขึ้นในการเข้าถึงข้อมูล

### โครงสร้างเพิ่มข้อมูลที่เลือกใช้

โครงสร้างเพิ่มที่น่าจะเหมาะสมกับการจัดเก็บ และค้นหาข้อมูลเอกสารอ้างอิงคือ เพิ่มแบบมัลติเปิด-คีย์ ที่จัดเพิ่มในรูปแบบมัลติลิสต์ (Miller, 1987) เนื่องจากการใช้งาน จะมีการค้นหาข้อมูลจากฟิลด์ที่ไม่ใช่คีย์หลักในที่นี้คือ Reference ID เช่นค้นหาจาก ชื่อผู้แต่ง, ปี, คำสำคัญ, หัวเรื่อง เป็นต้น ซึ่งจำเป็นต้องมีการทำครรรชนีฟิลด์รองที่สำคัญไว้ ซึ่งมัลติลิสต์จะเหมาะสมกว่าเพิ่มผกผัน เพราะไม่ต้องใช้โครงสร้าง ระเบียบแบบความยาวแปรในการเก็บลิสต์ของคีย์หลักที่คู่กัน

ส่วนโครงสร้างเพิ่มลำดับ เนื่องจากลักษณะการเข้าถึงข้อมูล ในการอ้างอิงข้อมูลนั้นไม่ได้เรียงตามลำดับก่อนหลังในการเก็บ ซึ่งในการทำงานกับเพิ่มข้อมูล ยังมีการปรับปรุงเพิ่ม เช่น การลบ, การแทรก, การเรียงลำดับ หรือ การจัดทำครรรณีอยู่บ่อยๆ ซึ่งการจัดเพิ่มแบบเพิ่มลำดับไม่เหมาะกับการปรับปรุงเพิ่มบ่อยๆ แต่มีข้อดีคือสามารถใช้กับระเบียบที่มีขนาดความยาวไม่แน่นอนได้ และการเข้าถึงก็เป็นแบบการเข้าถึงแบบสุ่ม

ซึ่งการทำจริงอาจใช้คลังโปรแกรม หรือใช้วัตถุดิบข้อมูล ของบริษัทไมโครซอฟต์ที่อยู่ใน วิชาดซีพลัสพลัส (Visual C++) หรือ วิชาดเบสิก (Visual Basic) ทำการเก็บเพื่อง่ายแก่การนำข้อมูลเข้า และการนำข้อมูลออก จากโปรแกรมฐานข้อมูลอื่นเช่น dBASE, FoxPro, Access ซึ่ง วัตถุดิบข้อมูล ของ ไมโครซอฟต์ยังสนับสนุน โอดีบีซี (ODBC: Open Database Connectivity) ซึ่งทำให้สามารถเข้าถึงข้อมูลของเอกสารอ้างอิงเป็นไปได้นบนเครือข่ายคอมพิวเตอร์

### 2.2.3 การนำข้อมูลเข้าหรือการนำข้อมูลออกจากโปรแกรม

มีอยู่ 2 รูปแบบที่นิยมกันคือ

- 1) รีเฟอร์ (Refer) หรือ บิบิกซ์ (BibIX) ซึ่งมาจากโปรแกรมฐานข้อมูลอ้างอิงในระบบยูนิกซ์
- 2) โปร-ไซท์ (Pro-Cite) จากโปรแกรมจัดการข้อมูลอ้างอิงโปร-ไซท์ (Pro-Cite)

### 2.2.3.1 รูปแบบของเพิ่มข้อมูลแบบรีเฟอร์ (Niles, 1991)

เป็นเพิ่มข้อมูลข้อความธรรมดา ซึ่งสามารถใช้โปรแกรมบรรณาธิกรณข้อความแก้ไขได้ โดยเพิ่มข้อความนี้ประกอบด้วยรายการเอกสารอ้างอิงต่างๆ ซึ่งบรรยายถึงข้อมูลของเอกสารอ้างอิง เช่น บทความในวารสาร หรือ หนังสือ เป็นต้น รายการเอกสารอ้างอิงต่างๆ ประกอบด้วยข้อมูลที่เป็นฟิลด์ต่างๆ เช่น ชื่อผู้แต่ง วันที่ที่ตีพิมพ์ ชื่อเรื่อง เป็นต้น และแต่ละรายการเอกสารอ้างอิง จะแยกด้วยบรรทัดว่างหนึ่งบรรทัดหรือมากกว่า แต่ละรายการเอกสารอ้างอิงจะขึ้นต้นด้วยเครื่องหมายเปอร์เซ็นต์ ตามด้วยอักษรภาษาอังกฤษตัวเดียวที่บอกชนิดของข้อมูลที่ตามหลังดังตัวอย่างต่อไปนี้

%A Geoffrey Chaucere  
 %D 1957  
 %T The Works of Geoffrey Chaucer  
 %E F. N. Robinson  
 %I Houghton  
 %C Boston  
 %N 2nd  
 %O An "Edition"

%A Herber H. Clark  
 %A Thomas H. Carlson  
 %D 1982  
 %T Hearers an Speech Acts  
 %J Language  
 %V 58  
 %P 332-373  
 %A Feodor Dostoevsky  
 %D 1964  
 %T Crime and Punishment  
 %I Norton  
 %C New York  
 %O Translation

สถาบันวิทยบริการ  
 จุฬาลงกรณ์มหาวิทยาลัย

ตัวภาษาอังกฤษที่แยกประเภทของข้อมูลมีความหมายดังต่อไปนี้ตามตารางที่ 28

Tag	Refer Meaning
A	Author Name
B	Title of Book
C	City of Publication
D	Date of Publication
E	Editor of Book
F	Citation to Insert Into Text
G	Gvnerment (NTIS) Number
H	Header
I	Issuer(Publisher)
J	Journal Name
K	Keyword (for searching)
N	Issue Number
O	Other Information
P	Selected page(s) of document
Q	Corportate Author
R	Technical report number
S	Series title
T	Title
U	Volume Number in a Series
V	Volume Number
W	Where it can be found
X	Abstract
Y	Series Editor of Book
Z	Total Length of Document

ตารางที่ 28 แทกที่ใช้ในรูปแบบรีเฟอร์/ บิบลิส์

### 2.2.3.2 รูปแบบของเพิ่มข้อมูลแบบ โปร-ไซท์ (Niles, 1991)

เป็นเพิ่มข้อมูลแบบข้อความที่ใช้เครื่องหมายอัญญาประกาศ (“ ”) กั้นปิดหัวท้ายโดยแต่ละเอกสารอ้างอิงจะอยู่บนบรรทัดเดียวแยกจากเอกสารอ้างอิงอื่น โดยรหัสปิดแถว (Carriage Return) แต่ละฟิลด์จะอยู่ในเครื่องหมายอัญญาประกาศ และกั้นแต่ละฟิลด์ด้วยเครื่องหมายจุลภาค (.)

ซึ่งลำดับของฟิลด์ต่างๆ นั้นต้องเรียงตามลำดับ ถ้าฟิลด์ใดไม่มี ต้องใส่เครื่องหมายอัญญาประกาศที่ไม่มีข้อมูลอยู่ และลำดับของฟิลด์นั้นขึ้นกับชนิดของเอกสารอ้างอิง เช่น ลำดับของหนังสือจะแตกต่างจากลำดับของบทความวารสาร ซึ่งฟิลด์แรกจะเป็นตัวบอกว่าเป็นเอกสารอ้างอิงชนิดใด โดยใช้อักษรภาษาอังกฤษตัวใหญ่หนึ่งตัว ที่อยู่ในเครื่องหมายอัญญาประกาศ ซึ่งความหมายนั้นดูได้จากตารางที่ 29 นี้

Code	Pro-Cite Meaning
A	Book, Long Form
B	Book, Short Form
C	Journal, Long Form
D	Journal, Short Form
E	Report
F	Newspaper
G	Dissertation
H	Trade Catalog
I	Letter
J	Manuscript
K	Conference Proceedings
L	Map
M	Musical Score
N	Sound Recording
O	Motion Picture
P	Audiovisual Material
Q	Video Recording
R	Art Work
S	Computer Program

ตารางที่ 29 รหัสบอกชนิดของเอกสารอ้างอิงในรูปแบบแฟ้มโปร-ไซท์

ต่อไปนี้เป็นรายการรูปแบบลำดับของฟิลด์ต่างๆ ที่ปรากฏในเอกสารอ้างอิงชนิดต่างๆ

หนังสือ รูปแบบเต็ม (Book, Long Form)

“A”, -,Author, -,Title, -,Secondary Author, -,Secondary Title, -,,-,-,-  
 ,Edition,Subsidiary author, -,Place Published, Publisher,Year,Date,Volume, -  
 ,Number,Pages, -,,-,-,Tertiary Author, -,Tertiary Title-,-,-,-,-,-,ISBN,Notes,Abstract, -  
 ,Keywords

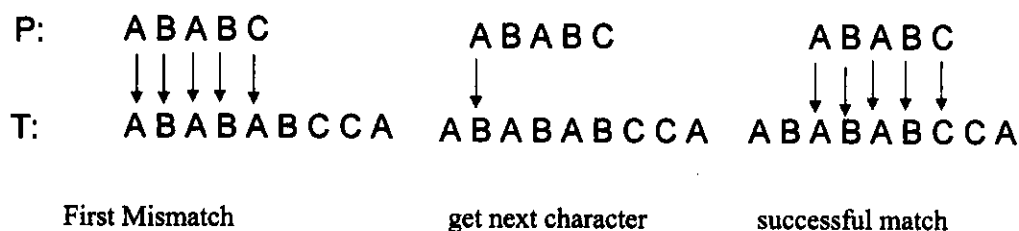
หนังสือ รูปแบบย่อ (Book, Short Form)

“B”, -,,-,-,-,-, Author,-,Title,-,-,-,-,-,Place Published, Publisher,Year, -,,-,-,-,-  
 ,,-,-,-,-,-,-,-,-,ISBN,Notes,Abstract, -,Keywords

วารสาร รูปแบบเต็ม (Journal, Long Form)

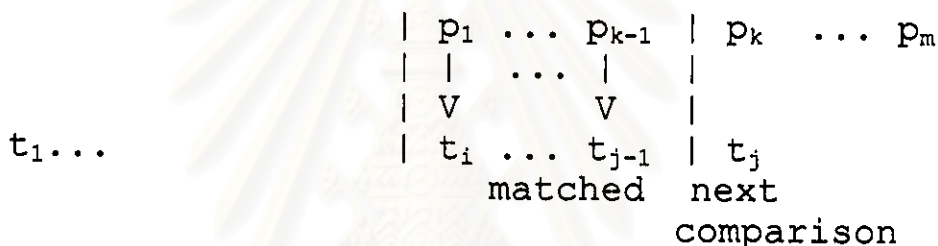
“C”, -,Author, -,Title, -,Secondary Author, -,Secondary Title, -,,-,-,-,Edition,Subsidiary  
 author, -,Place Published, Publisher,Year,Date,Volume, -,Number,Pages, -,,-,-,Tertiary  
 Author, -,Tertiary Title-,-,-,-,-,-,ISBN,Notes,Abstract, -,Keywords





### รูปที่ 8 การเปรียบเทียบตารางอักษรแบบตรงไปตรงมา

จะมีการเปรียบเทียบเริ่มจากอักษรตัวแรกในแบบ (P) กับอักษรตัวแรกในสายอักขระข้อความ (T) แต่ละอักษรแล้วพบว่า อักษรใดตรงกันก็จะเปรียบเทียบอักษรของ P ตัวต่อไป กับอักษรตัวต่อไปของ T ถ้าไม่ตรงกัน (Mismatch) ก็จะเลื่อน P ไปทางขวาอีกหนึ่งตำแหน่ง แล้วเริ่มเปรียบเทียบ ตั้งแต่อักษรตัวแรกของ P กับอักษรตัวที่ไม่ตรงกันของ T เปรียบเทียบกันจนกว่าอักษรจะตรงกันทุกตัว (Successful Match) หรือ จนกว่า ตำแหน่งอักษรของ T ที่ทำการเปรียบเทียบเกินกว่าความยาวของ T



### รูปที่ 9 การเลื่อนตัวอักษรและการเปรียบเทียบ

P และ T เป็น Pattern และ Text string

โดย  $P.length = m$  และ  $T.length = n$  ความยาวของสายอักขระ (String) ทั้งสองนั้นรู้มาก่อน  
i เป็น index ตำแหน่งใน T ที่ อักษรใน P ตัวที่ 1 เริ่มทำการเปรียบเทียบ

j เป็น index ตำแหน่งปัจจุบันของอักษรใน T

k เป็น index ตำแหน่งปัจจุบันของอักษรใน P

จากวิธีการเปรียบเทียบอักษรกันโดยวิธีดังกล่าว ในกรณีที่แย่มากที่สุด (Worst Case) จะเป็น  $mn$  หรือ  $O(mn)$  ซึ่งวิธีที่เหลืออีก 2 วิธีจะดีกว่าวิธีนี้คือเป็น  $O(m+n)$

#### 2.2.4.2 ขั้นตอนวิธีคนูท-มอริส-แพท

แนวกิด (Sedgewick, 1992) คือถ้ามีการเปรียบเทียบของแบบ กับข้อความจากซ้ายไปขวา แล้วพบว่า อักษรตัวใดที่ไม่เหมือนกัน และจากข้อมูลที่อยู่ก่อนว่า อักษรที่อยู่ในแบบส่วนแรกที่ไม่เหมือนกันนั้น ไม่มีทางที่จะปรากฏให้เห็นอีก ก็สามารถเลื่อนแบบได้เท่าจำนวนอักษรที่ไม่ตรงคู่กัน ลบหนึ่ง (Mismatch - 1) แทนที่จะขยับไปทีละตัว ทำให้การเปรียบเทียบได้เร็วขึ้น เช่น จากรูปที่ 10



แบบที่เป็นสายอักขระแบบทวิภาค (Binary String) คือ 10 000 000 ถ้าเกิดไม่ตรงกันที่อักษร 0 ที่ตำแหน่งที่ 6 ก็สามารถเลื่อนแบบ ไปที่ตำแหน่งที่ 6 ด้วยบนข้อความแทนที่จะเป็นที่ละตัว เพราะรู้ว่าไม่มีเลข 1 ในข้อความจะปรากฏให้เห็นอีกจนกว่าจะถึงลำดับที่ 6

P:	1	0	0	0	0	0	0	0
T:	1	0	0	0	0	1	0	0
Position	1	2	3	4	5	6	7	8
						^		
						mismatch		

P:	move	.	.	.	.	.	1	0	0	0	0	0	0	0	0
T:		1	0	0	0	0	1	0	0	.	.	.	.	.	.
Position		1	2	3	4	5	6	7	8						

ถ้าให้		p <sub>1</sub>	...	p <sub>k-1</sub>		p <sub>k</sub>	...	p <sub>m</sub>
			...					
		V		V				
t <sub>1</sub> ...		t <sub>i</sub>	...	t <sub>j-1</sub>		t <sub>j</sub>		
				matched		next		
						comparison		

T = Text string;           Length = m; pointer = j  
P = Pattern string;       Length = n; pointer = k

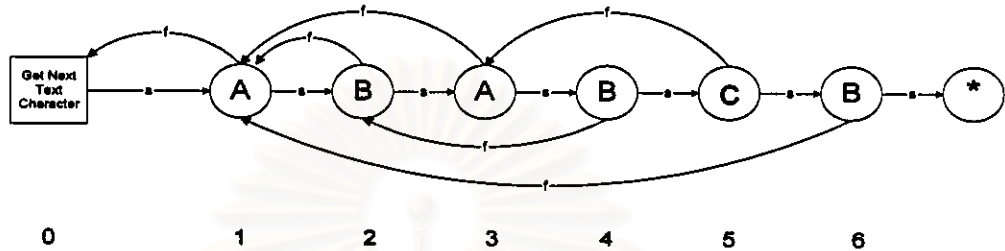
รูปที่ 10 การเปรียบเทียบและการเลื่อนตัวแบบเพื่อไปเปรียบเทียบกับสายอักขระแบบ เคเอ็มพี

จากรูปที่ 10 จะเห็นว่า p<sub>i</sub> ซึ่ที่ t<sub>j</sub> และสมมติว่าไม่ตรงคู่อักษรที่ตำแหน่ง k ในแบบ และ j ในข้อความดังนั้นแทนที่จะขยับตัวชี้ที่แบบ (Pattern Pointer) กลับไปที่ 1 และขยับตัวชี้ j กลับไปที่ j - k + 1 ก็สามารถไปที่ j ได้เลย

แต่จริงๆ แล้วการข้ามแบบไปดังข้างต้นไม่สามารถทำได้ในกรณีที่แบบมีตัวอักษรที่ตรงคู่กันเองที่จุดไม่ตรงคู่กับข้อความ เช่นการหาว่าแบบ 10100111 มีในข้อความ 1010100111 ซึ่งตอนแรกพบว่าไม่ตรงคู่กัน ในตำแหน่งที่ 5 แต่ไม่สามารถจะเลื่อนแบบไปที่ตำแหน่งที่ 5 บนข้อความได้ เพราะมีรูปแบบซ้ำกัน คือ 1010 จึงเลื่อนไปอย่างมากที่สุดไปที่ตำแหน่ง 3

ซึ่งการที่จะรู้ว่าเมื่อไม่ตรงคู่กัน แล้วจะเลื่อนแบบไปทางขวาโดยให้อักษรที่เคยตรงคู่กันมาแล้วตัวใดมาเปรียบเทียบตรงตำแหน่งที่ไม่ตรงคู่กันของข้อความ ซึ่งต้องทำการสร้างไฟไนท์สเตทแมชชีน (Finite State Machine) ของแบบเรียกว่า ฟังงานเคเอ็มพี (KMP flowchart) ขึ้นมาเพื่อเป็นข้อมูลที่ใช้ในตอนทำการเปรียบเทียบ (Baase, 1988) โดยที่

- 1) จุดต่อ (Node) คือ อักษรในแบบกลุ่มอักษรที่ทำกรอ่านเข้ามา แล้วทำการเปรียบเทียบกับอักษรในข้อความ
- 2) เส้นลูกศรที่ชี้ตรงๆ จากจุดต่อหนึ่ง ไปยังอีกจุดต่อหนึ่ง เรียกว่า เส้น โยงสำเร็จ (Success link (s))
- 3) เส้นลูกศรที่เป็นเส้น โค้งชี้ย้อนกลับๆ จากจุดต่อหลัง ไปยัง จุดต่อข้างหน้าเรียกว่า เส้น โยงล้มเหลว (Failure Link (f))



The KMP flowchart for P= 'ABABCB'

รูปที่ 11 ผังงาน เคเอ็มพี ที่ใช้ในการเปรียบเทียบสายอักขระ

ความหมายของผังงานเคเอ็มพี (KMP flowchart) คือจะทำการเปรียบเทียบอักษรตัวต่อไป ก็ต่อเมื่ออักษรของแบบนั้น ตรงคู่กับอักษรในข้อความ ก็มีเส้นโยงสำเร็จ (Success Link) แต่ถ้าไม่ตรงคู่กันคือเส้นโยงล้มเหลว (Failure Link) ก็ให้เลื่อนแบบให้อักษร ตามที่เส้นโยงล้มเหลว (Failure Link) ชี้ไว้ไปที่ตำแหน่งของอักษรในข้อความ ที่ไม่ตรงคู่กัน แล้วทำการเปรียบเทียบใหม่ ถ้ายังไม่ตรงคู่กันก็ให้ทำแบบเดิม จนกว่าจะตรงคู่กัน หรือ ไปยังจุดต่อพิเศษที่ให้เริ่มใหม่ โดยขยับตัวชี้ของข้อความไปอีกหนึ่ง แล้วทำการเปรียบเทียบกับอักษรตัวแรกในแบบ

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

KMP cell number	Text character being scanned		Success(s) or failure (f)
	Index	Character	
1	1	A	s
2	2	C	s
1	2	C	f
0	2	C	f
1	3	A	get next char
2	4	B	s
3	5	A	s
4	6	A	s
2	6	A	f
1	6	A	f
2	7	B	s
3	8	A	s
4	9	B	s
5	10	A	f
3	10	A	s
4	11	none	failure

ตารางที่ 30 ผลของการทำงานตามผังงาน เคเอ็มพี

จากการเปรียบเทียบแบบ 'ABABCB' กับข้อความ 'ACABAABABA' โดยใช้ผังงาน เคเอ็มพี ข้างต้น จะเห็นว่าต้องมี 2 ขั้นตอนวิธี (Algorithms) คือ

- 1) การสร้างผังงานเคเอ็มพี (KMP flowchart)
- 2) การกวาดผังงานเคเอ็มพี (KMP Scan)

โดยขั้นตอนวิธีแรกจะทำการสร้างแถวลำดับ (Array) ของเส้นโยงสัมพันธ์ของอักษรทุกตัวในแบบว่าจะชี้ไปที่อักษรใดในแบบเมื่อเกิดไม่ตรงคู่ขึ้น ส่วนขั้นตอนวิธีที่สองจะทำการเปรียบเทียบสายอักขระของแบบ และข้อความ โดยจะตั้งค่าดัชนีตามแถวลำดับที่สัมพันธ์ (Fail Array) ที่ได้มาเมื่อมีการไม่ตรงคู่เกิดขึ้น

ซึ่งประสิทธิภาพในการเปรียบเทียบนี้ได้เป็น  $O(m+n)$  เพราะเคเอ็มพีไม่เคยถอยดัชนีของข้อความกลับไปเมื่อไม่มีการตรงคู่เกิดขึ้น

#### 2.2.4.3 ขั้นตอนวิธีบอเซอร์-มัวร์

เมื่อให้  $T = \text{Text string}$ ; Length =  $m$ ; และ  $P = \text{Pattern string}$ ; Length =  $n$ ; จากวิธีการค้นหากลุ่มคำของเคเอ็มพี (KMP) ถ้า  $P$  เปรียบเทียบกับ  $T$  ที่  $t_i$  ต้องเปรียบเทียบตั้งแต่  $t_i, t_{i+1}, \dots, t_{i+m-1}$  ซึ่งในขั้นตอนวิธีบอเซอร์-มัวร์อาจจะข้ามการเปรียบเทียบตั้งแต่  $t_i$  เป็นต้นไป ถ้าไม่มีอักษรตัวนั้นใน

P และถ้า P ยืงยาว และมีขั้นตอนวิธีที่ดีจะทำให้สามารถข้ามตำแหน่งการเปรียบเทียบใน T ได้ไกลมากขึ้นวิธีบอเซอร์-มัวร์ใช้ 2 วิทยาการศึกษาศำนึก (Heuristics) ในการตัดสินใจว่าจะข้ามอักษรใน T ไปกี่ตัว และใช้วิธีการกวาดแบบจากขวาไปซ้าย

#### 2.2.4.3.1 วิทยาการศึกษาศำนึกอย่างแรกของบอเซอร์-มัวร์

จากรูปที่ 12 จะเห็นว่าหาคำว่า must มีอยู่ในประโยค "If you wish to understand others you must..." โดยครั้งแรก P จะวางตรงให้ตำแหน่งแรกของอักษรตรงกับอักษรตัวแรกใน T แล้วทำการเปรียบเทียบจากขวามาซ้ายของ P โดยครั้งแรกจะเปรียบเทียบ t กับ y ซึ่งจะเห็นว่าไม่ตรงคู่กันและอักษร y นั้นไม่มีอยู่ใน P ดังนั้นจึงเลื่อน P ไปทั้งแบบไปทางขวา โดยวางไว้ที่ตำแหน่งเริ่มต้นจาก y หนึ่งตัวอักษร แล้วทำการเปรียบเทียบใหม่ โดยเปรียบเทียบ t กับ w ซึ่งก็ไม่ตรงคู่กัน และ w ไม่มีอยู่ในแบบที่ซ่อนอยู่บนข้อความก่อน w ดังนั้นจึงเลื่อน P ไปทางขวาอีก 4 ตัวอักษรพบว่าช่องว่างไม่มีใน must และเลื่อน P ออกไปอีก 4 ตัวอักษรพบ u ซึ่งไม่ตรงคู่กัน แต่ u เป็นอักษรใน must จึงเลื่อนให้ u ใน P ตรงกับ u ใน T

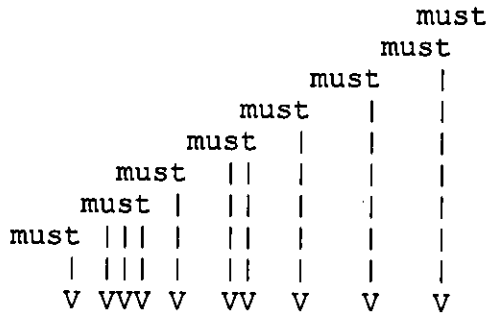
			must
		must	
	must		
must			
V	V	V	V

If you wish to understand others you must..

รูปที่ 12 ตัวอย่างการเปรียบเทียบสายอักขระแบบบอเซอร์-มัวร์อย่างแรก ในช่วงแรก

จากรูปที่ 12 เมื่อทำการเปรียบเทียบต่อไป ก็จะพบว่ามีคู่ที่ไม่ตรงกันที่ T คืออักษร u และ r ใน understand จึงเลื่อนแบบต่อไปอีกจนผ่านอักษร r ไป ในทำนองเดียวกันต่อมาพบว่า s ใน must กับ o ใน others นั้นไม่ตรงคู่กัน จึงเลื่อนแบบผ่านอักษร และทำอย่างนี้จนตรงคู่กับ must จากรูปที่ 13

จากตัวอย่างนี้ทำการเปรียบเทียบเพียง 18 ตัวอักษร เพราะ must ใน T อยู่ที่ตำแหน่ง 38 ถ้าเป็น ขั้นตอนวิธีอื่นอย่างน้อยจะต้องเปรียบเทียบถึง 41 ครั้ง



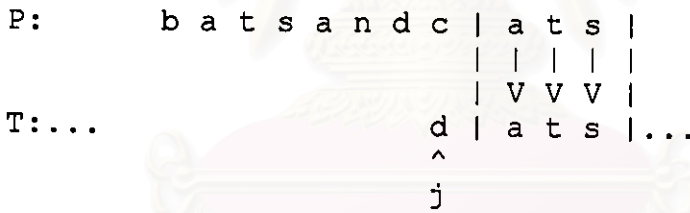
If you wish to understand others you must

รูปที่ 13 ตัวอย่างการเปรียบเทียบสายอักขระแบบบอเซอร์-มัวร์อย่างแรก ในช่วงหลัง

### 2.2.4.3.2 วิชยการศึกษาสำนึกอย่างที่สองของบอเซอร์-มัวร์

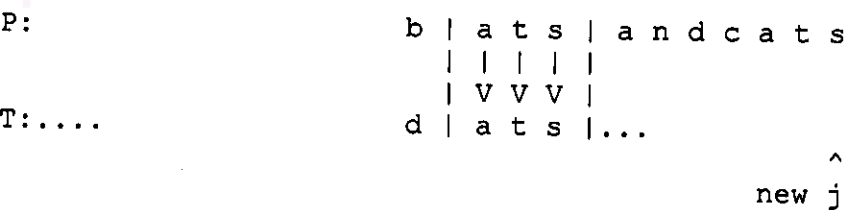
การใช้วิชยการศึกษาสำนึก อย่างแรกทำให้การทำงานเร็วกว่าเคเอ็มพี ในหลายกรณี แต่ ถ้าวรวมแนวคิดเรื่องเส้นโยงข้ามเหลวของขั้นตอนวิธีเคเอ็มพีจะทำให้การทำงานดีขึ้นไปอีก

จากรูปที่ 14 ถ้าสมมติทางขวาของแบบ P คือ a t s นั้นตรงกับส่วนหนึ่งของ T ก่อนที่จะไม่ตรงคู่กัน



รูปที่ 14 ตัวอย่างการเปรียบเทียบสายอักขระแบบบอเซอร์-มัวร์อย่างที่สอง ในช่วงแรก

ตอนนี้ครรชนีของ T ซึ่งที่อักษร d ปกติแบบจะเลื่อนไปทางขวา 1 ตัวอักษรคือ d สมมติว่า แบบ P นั้นไม่มีสายอักขระ ats อื่นอีกก็สามารถเลื่อน P ผ่าน ats ใน T ได้เลย แต่จริงๆ แล้วตอนนี้มี ats อีกตัวอยู่ตอนต้นของ P ดังนั้นจึงต้องเลื่อน ats ใน P มาตรงกับ ats ใน T ดังรูปที่ 15



รูปที่ 15 ตัวอย่างการเปรียบเทียบสายอักขระแบบบอเซอร์-มัวร์อย่างที่สอง ในช่วงหลัง

จากวิธีของบอเซอร์-มัวร์ หรือบีเอ็ม (BM) ดังกล่าวทำให้บีเอ็มทำงานได้เร็วกว่าเคเอ็มพี และถ้าแบบ มีค่าที่ยาวจะทำให้การเปรียบเทียบทำได้เร็วขึ้น

ขั้นตอนวิธีในการค้นหาโดยใช้การเปรียบเทียบสายอักขระที่เลือกใช้

ขั้นตอนวิธีของบอเซอร์-มัวร์ (Baase, 1988) น่าจะเป็นวิธีที่ดี เนื่องจากประสิทธิภาพในการเปรียบเทียบสายอักขระ ทำได้ดีกว่าวิธีอื่นๆ และรวมเอาข้อดีของขั้นตอนวิธีของ คนูท-มอริส-แพรทเข้ามาไว้ด้วยกัน

### 2.2.5 การเชื่อมโยงกับเวิร์คดูพา

ใช้ดีดีอี (DDE: Dynamic Data Exchange) หรือ โอแอลอี (OLE: Object Link Embedded) ของ ไมโครซอฟต์วินโดวส์ในการส่งข้อมูลจากฐานข้อมูล ไปทำการสร้างการอ้างอิงในเนื้อความ หรือสร้างรายการอ้างอิงในเวิร์คดูพา ขึ้นอยู่กับว่าเวิร์คดูพานั้นสนับสนุนการติดต่อภายในโปรเซส (IPC: Interprocess Communication) แบบใด ซึ่งถ้าไม่สามารถทำได้ทั้งดีดีอี (DDE) หรือ โอแอลอี (OLE) ก็จะทำแบบไม่เชื่อมโยงโดยตรง (Off-line) คือสร้างเอกสารที่ถูกร้างอ้างอิงในรูปแบบที่กำหนด ในแฟ้มข้อมูลแบบเวิร์คดูพาขึ้นอีกฉบับ

### 2.2.6 การจัดรูปแบบการอ้างอิง (Niles, 1991)

รูปแบบการอ้างอิงนั้นขึ้นกับประเภทของเอกสารอ้างอิงที่นำมาใช้ องค์กร สมาคม หน่วยงาน หรือมหาวิทยาลัยที่งานวิจัยนั้นจะส่งไปตีพิมพ์ ซึ่งมีรูปแบบแตกต่างกันไปโดยที่นี้จะแยกเป็นสามประเภทคือ

- 1) แบบแผนมาตรฐานของต่างประเทศ
- 2) แบบแผนของมหาวิทยาลัยในประเทศไทย
- 3) แบบแผนที่ผู้ใช้กำหนดเอง (User Defined Style)

#### 2.2.6.1 แบบแผนมาตรฐานของต่างประเทศ

เป็นแบบที่สมาคม หรือวารสารทางวิชาการเป็นผู้กำหนดขึ้นมาเช่น

- 1) แบบเอพีเอ (APA Style) เป็นแบบของสมาคมจิตวิทยาอเมริกัน (American Psychological Association)
- 2) แบบผู้แต่ง-วันที่ (Author -Date Style) เป็นแบบใช้ทั่วไปไม่เจาะจง  
แบบเอ็มแอลเอ (MLA Style) เป็นแบบของสมาคมภาษาสมัยใหม่ของอเมริกา (The Modern Language Association of America)

ตัวอย่างเช่น

1) แบบเอพีเอ (APA Style)

แบบอ้างอิงแทรกในข้อความ (In-Text Citation):

(Hartley, Harker, & Walsh, 1990; Pavio, 1975; Strunk, & White, 1979)

บทความวารสาร (Journal Article)

Pavio, A. (1975). Perceptual comparisons through the mind's eye. Memory & Cognition, 3, 635-647.

หนังสือ (Book)

Strunk, W., Jr., & White, E. B. (1979). The elements of style (3rd ed.). New York: Macmillan.

บทความในหนังสือ

Hartley, J. T., Harker, J. O., & Walsh, D. A. (1980). Contemporary issues and new directions in adult development of learning and memory. In L. W. Poon (Ed.), Adding in the 1980s: Psychological issues (pp. 239-252). Washington, DC: American Psychological Association.

#### 2.2.6.2 แบบแผนของมหาวิทยาลัยในประเทศไทย

เป็นแบบแผนที่กำหนดให้ใช้ในงานเขียนวิทยานิพนธ์ของมหาวิทยาลัยในประเทศไทย ตัวอย่างเช่น

จุฬาลงกรณ์มหาวิทยาลัย คู่มือจากคู่มือการเสนอวิทยานิพนธ์ของบัณฑิตวิทยาลัย ปี 2539 บทที่ 4 หน้าที่ 15 (บัณฑิตวิทยาลัย, 2539)

การอ้างอิงแบบแทรกในเนื้อหา เช่น

- 1) ให้ระบุนามผู้แต่ง และปีที่พิมพ์เอกสารนั้นในวงเล็บเช่น  
(สุจิตรา บรมินهنทร์ 2509)
- 2) ถ้ามีนามผู้แต่งในข้อความ ให้อ้างปีที่พิมพ์ในวงเล็บเช่น  
สุจิตรา บรมินهنทร์ (2509)

หนังสือ

แบบแผนที่ 1. ผู้แต่ง. ชื่อหนังสือ. เล่มที่หรือจำนวนเล่ม (ถ้ามี). ครั้งที่พิมพ์ (ถ้ามี). ชื่อชุดหนังสือและลำดับที่ (ถ้ามี). สถานที่พิมพ์: สำนักพิมพ์, ปีที่พิมพ์.

ตัวอย่างเช่น

ขจร สุขพานิช. ฐานันดรไพร่. กรุงเทพมหานคร: ภาควิชาประวัติศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ ประสานมิตร, 2519.

แบบแผนที่ 2. ผู้แต่ง. ปีที่พิมพ์. ชื่อหนังสือ. เล่มที่หรือจำนวนเล่ม (ถ้ามี). ครั้งที่พิมพ์ (ถ้ามี). ชื่อชุดหนังสือและลำดับที่ (ถ้ามี). สถานที่พิมพ์: สำนักพิมพ์.

ตัวอย่างเช่น

ขจร สุขพานิช. 2519. ฐานันดรไพร่. กรุงเทพมหานคร: ภาควิชาประวัติศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ ประสานมิตร.

บทความในหนังสือ

แบบแผนที่ 1. ผู้เขียนบทความ. ชื่อบทความ. ใน ชื่อบรรณาธิการ (ถ้ามี), ชื่อเรื่อง, เลขหน้า. สถานที่พิมพ์: สำนักพิมพ์, ปีที่พิมพ์.

ตัวอย่างเช่น

ชัยพร วิชาวุธ. การสอนในระดับอุดมศึกษา. ใน การสอนและการวัดผลการศึกษา, หน้า 1-30. พระนคร: ฝ่ายวิชาการ จุฬาลงกรณ์มหาวิทยาลัย, 2518.

แบบแผนที่ 2. ผู้เขียนบทความ. ปีที่พิมพ์. ชื่อบทความ. ใน ชื่อบรรณาธิการ (ถ้ามี), ชื่อเรื่อง, เลขหน้า. สถานที่พิมพ์: สำนักพิมพ์.

ตัวอย่างเช่น

ชัยพร วิชาวุธ. 2518. การสอนในระดับอุดมศึกษา. ใน การสอนและการวัดผลการศึกษา, หน้า 1-30. พระนคร: ฝ่ายวิชาการ จุฬาลงกรณ์มหาวิทยาลัย.

บทความในวารสาร

แบบแผนที่ 1. ผู้เขียนบทความ. ชื่อบทความ. ชื่อวารสาร ปีที่หรือเล่มที่ (เดือน ปี): เลขหน้า.



ตัวอย่างเช่น

จักรกฤษณ์ นรนิติผดุงการ และ ทวี สนวนมาลี. ความสามารถในการเงินของเทศบาล:  
กรณีของเทศบาลนครกรุงเทพก่อนเปลี่ยนแปลงเป็นกรุงเทพมหานคร. วารสาร  
พัฒนบริหารศาสตร์ 16 (เมษายน 2519): 231-254.

แบบแผนที่ 2. ผู้เขียนบทความ. ปีที่พิมพ์. ชื่อบทความ. ชื่อวารสาร ปีที่หรือเล่มที่:  
เลขหน้า.

ตัวอย่างเช่น

จักรกฤษณ์ นรนิติผดุงการ และ ทวี สนวนมาลี. 2519. ความสามารถในการเงินของ  
เทศบาล: กรณีของเทศบาลนครกรุงเทพก่อนเปลี่ยนแปลงเป็นกรุงเทพมหานคร.  
นคร. วารสารพัฒนบริหารศาสตร์ 16: 231-254.

### 2.2.6.3 แบบแผนที่ใช้กำหนดเอง (User Defined Style)

ในบางครั้งแบบที่กำหนดไว้ให้นั้นอาจไม่เหมาะสม มีการเปลี่ยนแปลง หรือมีแบบเกิด  
ขึ้นมาใหม่ ทำให้ผู้ใช้จำเป็นต้องกำหนดแบบขึ้นมาใช้ใหม่ ระบบที่จะสร้างนี้ต้องเปิดโอกาสให้  
ผู้ใช้ดัดแปลงแบบที่มีอยู่แล้ว หรือสร้างขึ้นมาใหม่ได้

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย