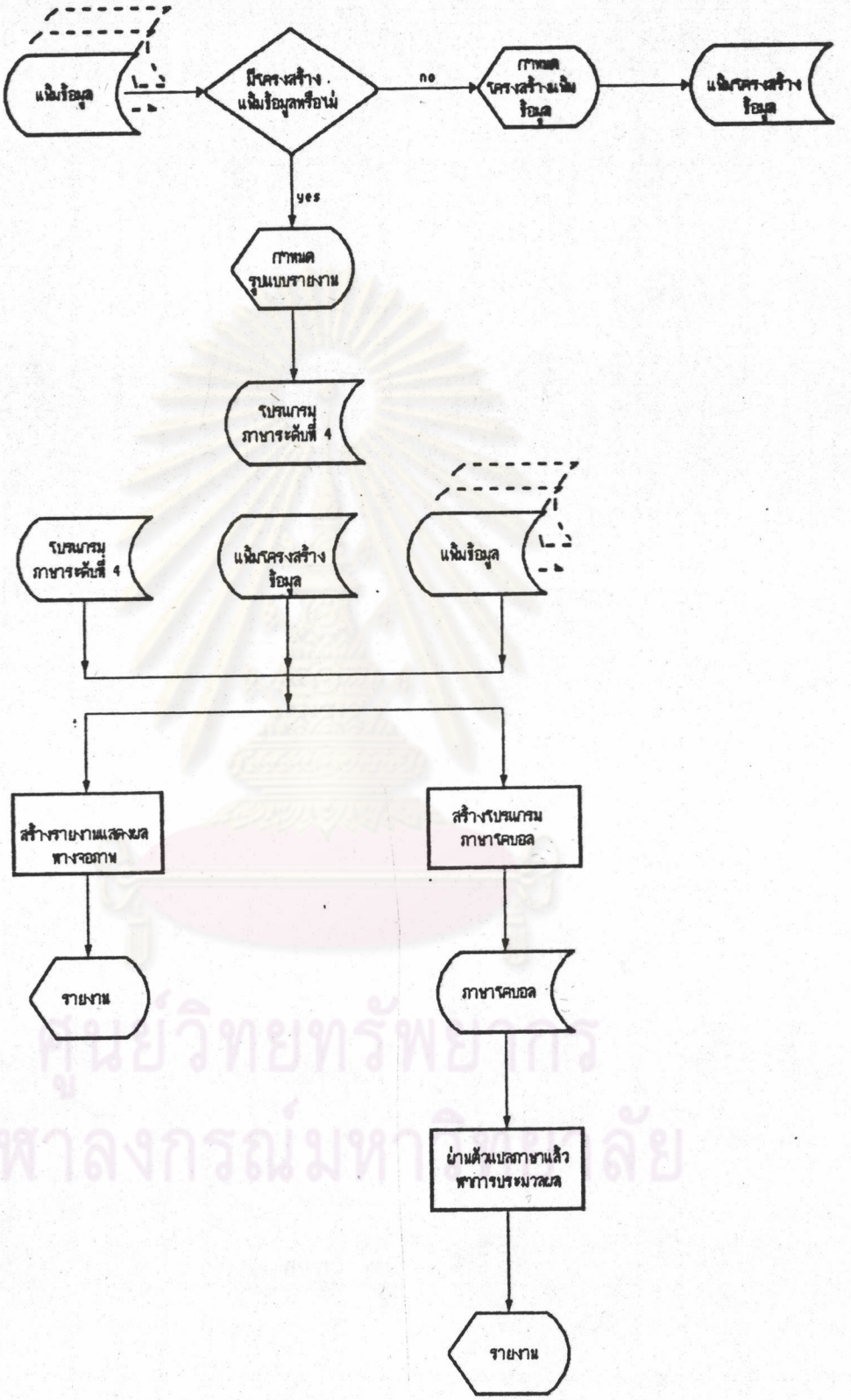
บทที่ 3

การออกแบบระบบ

โครงสร้างของระบบ

เมื่อผู้ใช้มีความต้องการที่จะสร้างรายงานขึ้นมานั้น ผู้ใช้จะต้องกำหนดรูปแบบของรายงานขึ้นมาก่อน พร้อมทั้งจะต้องทราบว่าจะนำข้อมูลจากแฟ้มข้อมูลใดบ้างมาใช้ในการสร้างรายงาน จากนั้นจึงจะทำการเขียนโปรแกรมเพื่อให้ได้รายงานตามรูปแบบที่ต้องการ จากขั้นตอนต่าง ๆ เหล่านี้จะเห็นว่าเมื่อผู้ใช้ต้องการสร้างรายงานจะต้องเขียนโปรแกรมทุกครั้ง ดังนั้นผู้วิจัยจึงได้ทำการออกแบบระบบใหม่ขึ้นมาเพื่อช่วยในการสร้างโปรแกรมเพื่อผลิตรายงาน ซึ่งโปรแกรมที่ถูกสร้างขึ้นมานี้เมื่อนำไปผ่านตัวแปลภาษาแล้วทำการประมวลผลก็จะได้รายงานตามรูปแบบที่ผู้ใช้ต้องการ สำหรับระบบที่ผู้วิจัยได้ทำการออกแบบขึ้นมานั้นมีโครงสร้างของระบบดังรูปที่ 3.1



รูปที่ 3.1 โครงสร้างของระบบ

จากภาพที่ 3.1 จะเห็นว่าเมื่อผู้ใช้ต้องการที่จะสร้างรายงาน ผู้ใช้จะต้องมีแฟ้มข้อมูลต่าง ๆ ที่จะใช้ในการสร้างรายงานอยู่ก่อนแล้ว แต่เนื่องจากแฟ้มข้อมูลประเภท Btrieve ISAM และ Text File ไม่มีการกำหนดโครงสร้างแฟ้มข้อมูลไว้ภายในตัวของมันเองดังเช่นแฟ้มข้อมูล dBASE และ Foxbase ดังนั้นหากผู้ใช้มีความประสงค์จะจัดทำรายงาน โดยนำข้อมูลจากแฟ้มข้อมูลที่ไม่มีการกำหนดโครงสร้างไว้ จึงจำเป็นที่ผู้ใช้จะต้องกำหนดโครงสร้างของแฟ้มข้อมูลเหล่านั้นไว้ก่อนที่จะเริ่มต้นกำหนดรูปแบบของรายงาน เพื่อให้โปรแกรมสามารถอ่านข้อมูลจากเขตข้อมูลต่าง ๆ มาจัดทำรายงานได้อย่างถูกต้อง

ในการกำหนดรูปแบบรายงาน ผู้ใช้ควรจะมีรูปแบบของรายงานที่ต้องการไว้ก่อนแล้วจึงนำมากำหนดรูปแบบที่ต้องการ โดยผ่านโปรแกรมกำหนดรูปแบบรายงาน หรือการเขียนเป็นโปรแกรมภาษาระดับที่ 4 ถ้ากำหนดรูปแบบรายงานโดยผ่านโปรแกรมกำหนดรูปแบบรายงาน ซึ่งโปรแกรมนี้ก็จะทำการแปลงรูปแบบรายงานให้เป็นโปรแกรมภาษาระดับที่ 4 และในขณะเดียวกันก็จัดเก็บรูปแบบรายงานนั้นไว้ในแฟ้มข้อมูลเพื่อความสะดวกแก่การแก้ไขในภายหลัง

สำหรับแฟ้มโปรแกรมภาษาระดับที่ 4 แฟ้มโครงสร้างข้อมูล และ แฟ้มข้อมูลต่าง ๆ จะถูกนำไปเป็นข้อมูลเข้าของโปรแกรมแสดงผลทางจอภาพ เพื่อที่ผู้ใช้จะได้ดูว่าผลลัพธ์ที่ได้นั้นตรงตามความต้องการหรือไม่ ถ้าไม่ตรงตามที่ต้องการก็กลับไปแก้ไขในขั้นตอนการกำหนดรูปแบบรายงานใหม่ เพื่อให้ได้โปรแกรมภาษาระดับที่ 4 ที่ถูกต้องตรงตามรูปแบบรายงานที่ต้องการจริง ๆ

สำหรับวัตถุประสงค์หลักของงานวิจัยนี้ คือการพัฒนาโปรแกรมเพื่อสร้างโปรแกรมภาษาโคบอล โดยมีแฟ้มโปรแกรมภาษาระดับที่ 4 แฟ้มโครงสร้างข้อมูล และ แฟ้มข้อมูลต่าง ๆ เป็น ข้อมูลเข้าเพื่อผลิตโปรแกรมภาษาโคบอล และ เมื่อนำโปรแกรมภาษาโคบอลที่ได้ไปผ่านตัวแปลภาษาแล้วทำการประมวลผล ก็จะได้รายงานตามรูปแบบที่ต้องการ โดยรายงานที่ได้นั้นจะจัดเก็บอยู่ในแฟ้มข้อมูลแบบอักขระ

นิยาม เนื่องจากการออกแบบระบบนี้ เกี่ยวข้องกับการกำหนดรูปแบบรายงานเป็นหลัก ดังนั้นจึงนิยามคำต่าง ๆ ที่เกี่ยวข้องกับรูปแบบรายงานเพื่อใช้อ้างอิงในภายหลัง ดังปรากฏในตารางที่ 3.1

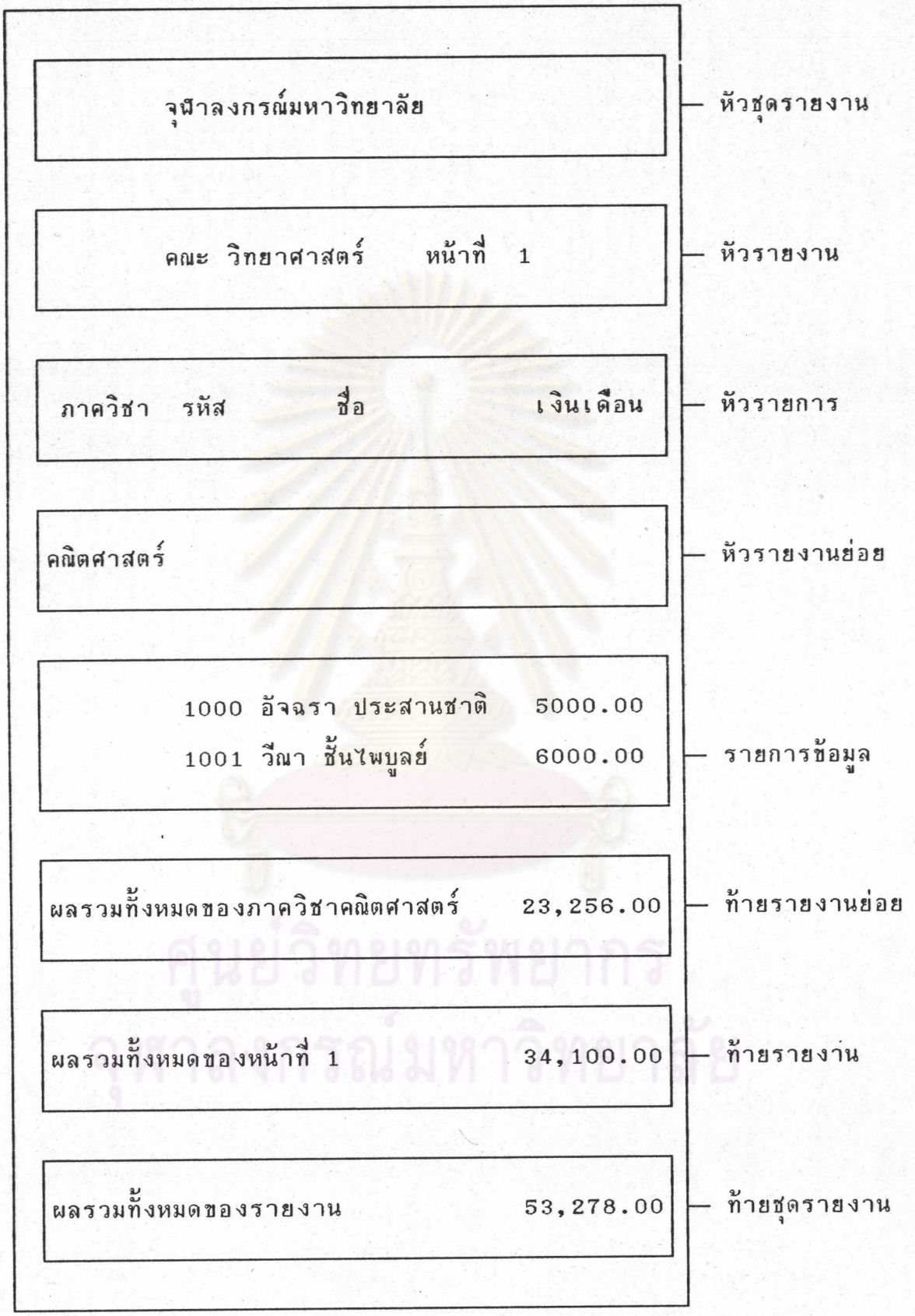
ค่า	ความหมาย
- หัวชุดรายงาน (Report Heading)	รายการที่ต้องการพิมพ์ในส่วนหัวชุดรายงาน ซึ่งจะพิมพ์เพียง 1 ครั้งในหน้าแรกของรายงาน
- หัวรายงาน (Page Heading)	รายการที่ต้องการพิมพ์เป็นส่วนแรกของรายงานแต่ละหน้า โดยถ้าเป็นหน้าแรกจะพิมพ์ถัดจากหัวชุดรายงาน ซึ่งรายการนี้จะพิมพ์ทุกหน้าของรายงาน
- หัวรายการ (Column Heading)	รายการที่ต้องการพิมพ์ในส่วนถัดจากหัวรายงาน ซึ่งจะพิมพ์ทุกหน้าของรายงาน
- รายการข้อมูล (Detail Line)	รายการข้อมูลที่ต้องการพิมพ์ ใน 1 หน้าจะพิมพ์หลายครั้งตามจำนวนที่ผู้ใช้ระบุ
- ท้ายรายงาน (Page Footing)	รายการที่ต้องการพิมพ์เป็นส่วนท้ายของรายงานแต่ละหน้า โดยที่ถ้าเป็นหน้าสุดท้ายจะพิมพ์อยู่ก่อนท้ายชุดรายงาน รายการนี้จะพิมพ์ทุกหน้าของรายงาน
- ท้ายชุดรายงาน (Report Footing)	รายการที่ต้องการพิมพ์เมื่อหมดข้อมูลในแฟ้มข้อมูลหลัก โดยจะพิมพ์เป็นส่วนท้ายของรายงานแต่ละชุด รายการนี้จะพิมพ์เพียง 1 ครั้งในหน้าสุดท้ายของรายงาน
- การเปลี่ยนค่าในเขตข้อมูล หรือ (Control Break)	เขตข้อมูลในแฟ้มข้อมูลหลักที่ถูกกำหนดให้เป็นเขตข้อมูลที่ใช้ในการเปรียบเทียบค่าในระเบียนปัจจุบันกับระเบียนก่อนหน้านี้ หากต่างกันให้พิมพ์ส่วนของหัวรายการย่อย หรือ ท้ายรายการย่อย

ตารางที่ 3.1 คำนิยามที่เกี่ยวข้องกับรายงาน

ค่า	ความหมาย
- หัวรายการย่อย (Control Break Heading)	รายการที่ต้องการพิมพ์เมื่อเกิดการเปลี่ยนค่าในเขตข้อมูล
- ท้ายรายการย่อย (Control Break Footing)	รายการที่ต้องการพิมพ์เมื่อเกิดการเปลี่ยนค่าในเขตข้อมูล

ตารางที่ 3.1 คำนิยามที่เกี่ยวข้องกับรายงาน (ต่อ)

ตัวอย่างรูปแบบรายงานโดยทั่ว ๆ ไป แสดงในรูปที่ 3.2



รูปที่ 3.2 ตัวอย่างรูปแบบรายงาน

การออกแบบระบบ

จากภาพที่ 3.1 พิจารณาพบว่าสิ่งที่ต้องออกแบบประกอบด้วย การออกแบบโปรแกรมภาษาระดับที่ 4 การออกแบบแฟ้มข้อมูล การออกแบบการนำข้อมูลเข้าและ การออกแบบโปรแกรม โดยมีรายละเอียดดังนี้

1. การออกแบบโปรแกรมภาษาระดับที่ 4

การออกแบบโปรแกรมภาษาระดับที่ 4 ประกอบด้วย 2 ส่วนคือ

1.1 ไวยากรณ์ของโปรแกรมของภาษาระดับที่ 4

การออกแบบไวยากรณ์นี้ ใช้หลักการออกแบบไวยากรณ์แบบอีบีเอ็นเอฟ ซึ่งประกอบไปด้วยกลุ่มคำสั่ง 10 กลุ่ม ดังแสดงใน รูปที่ 3.3

```
<4GL File>          ::= <database_command>
                        <pagewidth_command>
                        <pagelong_command>
                        <filename_command>
                        <where_command>
                        <relation_command>
                        <detailgap_command>
                        <lineperpage_command>
                        <ct_command>
                        <level_command>
```

รูปที่ 3.3 กลุ่มคำสั่งโปรแกรมภาษาระดับที่ 4 ที่ออกแบบ

สำหรับรายละเอียดไวยากรณ์ของแต่ละกลุ่มคำสั่งแสดงใน

รูปที่ 3.4

```
<database_command>     ::= database <constant>.
<pagewidth_command>    ::= pagewidth <integernumber>.
<pagelong_command>     ::= pagelong <integernumber>.
<filename_command>     ::= filename <filename1>
                           <filestruorindex]
                            filenumber <integernumber>.
<where_command>        ::= where <condition>.
<relation_command>     ::= relation <relation1>.
<detailgap_command>    ::= detailgap <integernumber>.
<lineperpage_command>::= lineperpage <integernumber>.
<ct_command>           ::= <fieldname>
                         ¦  <fieldname> <ct_command>.
<level_command>        := <levelname><level1>.
<filestruorindex>      ::= [fileindex <filename1>]
                         ¦  [filestru <filename1>]
<condition>            ::= <convarname><relop><convarname>
                         ¦  <convarname><relop><convarname>
                            <logicop><condition>
<relop>                ::= >¦<¦=¦>=¦<=¦<>¦><
<logicop>              ::= and¦or
<convarname>           ::= <number> ¦ <fieldname>
                         ¦  ''''<constant>''''
                         ¦  not <convarname>
```

รูปที่ 3.4 รายละเอียดไวยากรณ์ของแต่ละกลุ่มคำสั่งที่ออกแบบ

```
<relation1>          ::= <fieldname>=<fieldname>
                      |  <fieldname>=<fieldname>
                         <relation1>
<levelname>          ::= rh|ph|cl|ch by <fieldname>|de
                      |  cf by <fieldname>
                      |  pf|rf
<level1>             ::= <position><variablename>
                         [<attribute>][<picture>]
                      |  <position><variablename>
                         [<attribute>][<picture>]<level1>
<position>           ::= col<integernumber>row
                         <integernumber>
                      |  col<integernumber>skip
                         <integernumber>
<attribute>          ::= with <attribute1>
<attribute1>         ::= <attribute2>
                      |  <attribute2><attribute1>
<attribute2>         ::= boldface|enlarge|doubleline
                      |  singleline|italic
<picture>            ::= pic <constant>
<variablename>       ::= source <fieldname>
                      |  value ''''<constant>''''
                      |  compute <expression>
<expression>         ::= <identifier>=<expression1>
                      |  pageno=integernumber
<expression1>        ::= <expression2>|<function>
```

รูปที่ 3.4 รายละเอียดไวยากรณ์ของแต่ละกลุ่มคำสั่งที่ออกแบบ (ต่อ)

```
<expression2>          ::= <term>
                         | <sign><term>
                         | <expression2><addop><term>
<term>                 ::= <term1>
                         | <term1><mulop><term1>
<term1>                ::= <factor>
                         | <term1>^<factor>
<factor>               ::= <facvarname>
                         | '(' <expression2> ')'
<function>             ::= <functionname> '('
                           <facvarname> ')'
<facvarname>           ::= <identifier>|<fieldname>
                         | <number>
<functionname>         ::= sum|avg|std|var
<integernumber>        ::= <digit>
                         | <digit><integernumber>
<number>               ::= <integernumber>
                         | <realnumber>
<realnumber>           ::= <integernumber>.<integernumber>
<fieldname>            ::= <integernumber>.<identifier>
<letter>               ::= A|B|C|...|Z|a|b|c|...|z
<digit>                ::= 0|1|2|...|9
<identifier>           ::= <letter>
                         | <letter><identifier1>
<identifier1>          ::= <identifier2>
                         | <identifier2><identifier1>
```

รูปที่ 3.4 รายละเอียดไวยากรณ์ของแต่ละกลุ่มคำสั่งที่ออกแบบ (ต่อ)

```
<identifier2>          ::= <letter>
                         |  <digit>
<filename1>            ::= <letter>
                         |  <letter><identifier3>
<identifier3>          ::= <identifier2>
                         |  <identifier2><identifier3>
                         |  . <identifier4>
<identifier4>          ::= <identifier2>
                         |  <identifier2><identifier4>
<constant>             ::= all-ascii-value
โดยที่ all-ascii-value หมายถึง ตัวอักษรทุก ๆ ตัว
```

รูปที่ 3.4 รายละเอียดไวยากรณ์ของแต่ละกลุ่มคำสั่งที่ออกแบบ (ต่อ)

1.2 รูปแบบคำสั่งต่าง ๆ ของโปรแกรมภาษาระดับที่ 4

ก่อนที่จะกล่าวถึงรูปแบบของคำสั่งต่าง ๆ จะอธิบายถึงสัญลักษณ์ที่ใช้แทนความหมายของคำสั่ง ซึ่งแสดงไว้ในตารางที่ 3.2

สัญลักษณ์	ความหมาย
กลุ่มตัวอักษรตัวเข้ม	คำสั่งต่าง ๆ ของภาษาระดับที่ 4
กลุ่มตัวอักษรปกติ	พารามิเตอร์ต่าง ๆ
[...]	จะมีคำสั่งในเครื่องหมายนี้หรือไม่มีก็ได้
{...}	ให้เลือกคำสั่งในเครื่องหมายนี้มา 1 คำสั่ง

ตารางที่ 3.2 สัญลักษณ์ที่ใช้แทนความหมายของคำสั่งโปรแกรมภาษาระดับที่ 4

รูปแบบคำสั่งของโปรแกรมภาษาระดับที่ 4 ที่ออกแบบมีทั้งหมด 28 คำสั่งดังนี้

คำสั่ง **database**

รูปแบบ **database** databasetype

โดยที่ databasetype หมายถึง ประเภทของแฟ้มข้อมูล

หน้าที่ เป็นคำสั่งที่ใช้ในการกำหนดประเภทของแฟ้มข้อมูล โดยประเภทของแฟ้มข้อมูลมี 6 ประเภท คือ

- Dbase
- Foxbase
- Btrieve
- ISAM
- COBOLVAR (COBOL Variable Length Record)
- COBOLFIXED (COBOL Fixed Length Record)
- OTHERFIXED (Other Fixed Length Record)

ตัวอย่าง **database** dbase

หมายถึง เป็นฐานข้อมูลประเภท dBASE

คำสั่งที่เกี่ยวข้อง ไม่มี

คำสั่ง **file**

รูปแบบ **file** filename

[**fileindex** filenameindex]

[**filestru** filenamestru]

filenumber filenumber

โดยที่ filename หมายถึง ชื่อแฟ้มข้อมูล

filenameindex หมายถึง ชื่อแฟ้มข้อมูลดัชนี

filenamestru หมายถึง ชื่อแฟ้มโครงสร้างข้อมูล

filenumber หมายถึง หมายเลขแฟ้มข้อมูล

หน้าที่ เป็นคำสั่งที่ใช้ในการกำหนดชื่อแฟ้มข้อมูล พร้อมทั้งหมายเลขแฟ้มที่จะใช้ในการออกรายงาน

- ถ้าเป็นแฟ้มข้อมูลประเภท dBASE Foxbase ในกรณีที่มีแฟ้มข้อมูลดัชนีจะต้องกำหนดชื่อแฟ้มข้อมูลดัชนีให้ด้วย
- ถ้าเป็นแฟ้มข้อมูลประเภท Btrieve ISAM และ Text จะต้องกำหนดชื่อแฟ้มโครงสร้างข้อมูลให้ด้วย

ตัวอย่าง

- **file tran.dbf filenumber 0 .**
  หมายถึง แฟ้มข้อมูล tran.dbf มีหมายเลขแฟ้มข้อมูลเป็นหมายเลข 0
- **file tran.dbf fileindex tran.ndx filenumber 0.**
  หมายถึง แฟ้มข้อมูล tran.dbf มีแฟ้มข้อมูล tran.ndx เป็นแฟ้มข้อมูลดัชนี มีหมายเลขแฟ้มข้อมูลเป็นหมายเลข 0
- **file tran.txt filestru tran.stc filenumber 0.**
  หมายถึง แฟ้มข้อมูล tran.txt มีแฟ้มข้อมูล tran.stc เป็นแฟ้มโครงสร้างข้อมูล มีหมายเลขแฟ้มข้อมูลเป็นหมายเลข 0

คำสั่งที่เกี่ยวข้อง **filestru keynumber filenumber**

คำสั่ง **fileindex**

รูปแบบ **fileindex filenameindex**

โดยที่ filenameindex หมายถึง ชื่อแฟ้มข้อมูลดัชนี

หน้าที่ เป็นคำสั่งที่ใช้ในการกำหนดชื่อแฟ้มข้อมูลดัชนีของแฟ้มข้อมูลประเภท dBASE และ Foxbase

ตัวอย่าง **fileindex master.ndx**

หมายถึง แฟ้มข้อมูลชื่อ master.ndx เป็นแฟ้มข้อมูลดัชนี

คำสั่งที่เกี่ยวข้อง **file**

ค่าสั่ง **filestru**

รูปแบบ **filestru** filenamestru

โดยที่ filenamestru หมายถึง ชื่อแฟ้มโครงสร้างข้อมูล

หน้าที่ เป็นค่าสั่งที่ใช้ในการกำหนดชื่อแฟ้มโครงสร้างข้อมูล โดยแฟ้มนี้ได้มาจากโปรแกรมสร้างแฟ้มโครงสร้างข้อมูล

ตัวอย่าง **filestru** master.stc

หมายถึง แฟ้มข้อมูลชื่อ master.stc เป็นแฟ้มข้อมูลที่จัดเก็บโครงสร้างของข้อมูล

ค่าสั่งที่เกี่ยวข้อง **file**

ค่าสั่ง **filenumber**

รูปแบบ **filenumber** filenumber

โดยที่ filenumber หมายถึง หมายเลขแฟ้มข้อมูล

หน้าที่ เป็นค่าสั่งที่ใช้ในการกำหนดหมายเลขแฟ้มข้อมูลเพื่อจะใช้เป็นตัวแทนชื่อแฟ้มข้อมูลนั้น ๆ ในการเรียกใช้ครั้งต่อ ๆ ไป

- แฟ้มข้อมูลหลักจะต้องกำหนดหมายเลขแฟ้มเป็นหมายเลข 0 เสมอ
- แฟ้มข้อมูลประเภท dBASE Foxbase และ ISAM จะสามารถกำหนดหมายเลขแฟ้มได้จากหมายเลข 0 - 126 หมายความว่าสามารถเปิดแฟ้มข้อมูลได้สูงสุด 127 แฟ้ม เนื่องจากในการเรียกใช้แฟ้มข้อมูลครั้งหนึ่ง ๆ อาจจะมีการเปิดแฟ้มข้อมูลถึง 2 แฟ้ม คือแฟ้มข้อมูลใช้งาน และแฟ้มข้อมูลดัชนี ซึ่งระบบนี้สามารถเปิดแฟ้มข้อมูลทั้งหมดได้สูงสุด 255 แฟ้ม

ตัวอย่าง **filenumber** 0

หมายถึง เป็นแฟ้มข้อมูลหมายเลข 0

ค่าสั่งที่เกี่ยวข้อง **file**

คำสั่ง	**pagelong**
รูปแบบ	**pagelong** integernumber โดยที่ integernumber หมายถึง ตัวเลขจำนวนเต็ม
หน้าที่	เป็นคำสั่งที่ใช้ในการกำหนด จำนวนบรรทัดของรายงานในแต่ละหน้าโดยมี จำนวนสูงสุดไม่เกิน 33 บรรทัด ถ้าผู้ใช้ไม่กำหนด ระบบจะกำหนดให้มีจำนวน 30 บรรทัดต่อหน้า
ตัวอย่าง	**pagelong** 33. หมายถึง ในแต่ละหน้าของรายงานนั้นมีจำนวนบรรทัด 33 บรรทัด
คำสั่งที่เกี่ยวข้อง	ไม่มี

คำสั่ง	**pagewidth**
รูปแบบ	**pagewidth** integernumber โดยที่ integernumber หมายถึง ตัวเลขจำนวนเต็ม
หน้าที่	เป็นคำสั่งที่ใช้ในการกำหนด จำนวนตัวอักษรต่อบรรทัดโดยมีจำนวนตัวอักษรต่อบรรทัดสูงสุดไม่เกิน 132 ตัวอักษร ถ้าผู้ใช้ไม่กำหนด ระบบจะกำหนดให้มีจำนวน 80 ตัวอักษรต่อบรรทัด
ตัวอย่าง	**pagewidth** 132. หมายถึง ในแต่ละบรรทัดของรายงานมีจำนวนตัวอักษร 132 ตัวอักษร
คำสั่งที่เกี่ยวข้อง	ไม่มี

คำสั่ง	**lineperpage**
รูปแบบ	**lineperpage** integernumber โดยที่ integernumber หมายถึง ตัวเลขจำนวนเต็ม
หน้าที่	เป็นคำสั่งที่ใช้ในการกำหนดจำนวนรายการข้อมูลที่ต้องการพิมพ์ในแต่ละหน้าถ้าผู้ใช้ไม่กำหนด ระบบจะกำหนดให้มีค่าเท่ากับ 10 รายการ

ตัวอย่าง lineperpage 20.

หมายถึง ในแต่ละหน้ามีจำนวนรายการข้อมูลเท่ากับ 20 รายการ

คำสั่งที่เกี่ยวข้อง ไม่มี

คำสั่ง detailgap

รูปแบบ detailgap integernumber

โดยที่ integernumber หมายถึง ตัวเลขจำนวนเต็ม

หน้าที่ เป็นคำสั่งที่ใช้ในการกำหนด ระยะห่างระหว่างรายการข้อมูลแต่ละรายการถ้าผู้ใช้ไม่กำหนด ระบบจะกำหนดให้มีค่าเท่ากับ 0

ตัวอย่าง detailgap 1.

หมายถึง กำหนดให้ระยะห่างระหว่างแต่ละรายการข้อมูลมีค่าเท่ากับ 1

คำสั่งที่เกี่ยวข้อง ไม่มี

คำสั่ง where

รูปแบบ where condition

โดยที่ condition หมายถึงเงื่อนไขที่จะใช้ในการทำรายงาน

หน้าที่ เป็นคำสั่งที่ใช้ในการกำหนดเงื่อนไขของการนำข้อมูลจากแฟ้มข้อมูลมาทำรายงาน โดยเขตข้อมูลที่จะใช้ในการกำหนดเงื่อนไขจะต้องเป็นเขตข้อมูลของแฟ้มข้อมูลหลักเท่านั้น

ตัวอย่าง where 0.code > 1000 and 0.code < 9000

หมายถึง ให้นำข้อมูลจากแฟ้มข้อมูลหลัก เฉพาะข้อมูลที่มีค่าในเขตข้อมูลมากกว่า 1000 และ น้อยกว่า 9000 เท่านั้นมาสร้างรายงาน

คำสั่งที่เกี่ยวข้อง ไม่มี

ค่าสั่ง relation

รูปแบบ **relation** filenumber.fieldname=
filenumber.fieldname
[filenumber.fieldname=
filenumber.fieldname...]
โดยที่ filenumber หมายถึง หมายเลขแฟ้มข้อมูล
fieldname หมายถึง ชื่อเขตข้อมูล

หน้าที่ เป็นคำสั่งที่ใช้ในการกำหนด ความสัมพันธ์ระหว่างแฟ้มข้อมูล สำหรับแฟ้มข้อมูลประเภท Btrieve และ ISAM เขตข้อมูลที่ใช้ในการกำหนดความสัมพันธ์ควรเป็นเขตข้อมูลที่เป็นคีย์

ตัวอย่าง **relation** 0.code = 1.id.
หมายถึง เขตข้อมูล code ของแฟ้มข้อมูลหลักมีความสัมพันธ์กับเขตข้อมูล id ของแฟ้มข้อมูลหมายเลข 1

ค่าสั่งที่เกี่ยวข้อง ไม่มี

ค่าสั่ง **ct**

รูปแบบ **ct** filenumber.fieldname
[filenumber.fieldname...]
โดยที่ filenumber หมายถึง หมายเลขแฟ้มข้อมูล
fieldname หมายถึง ชื่อเขตข้อมูล

หน้าที่ เป็นคำสั่งที่ใช้ในการทำคอนโทรลเบรค โดยชื่อเขตข้อมูลจะต้องเป็นชื่อเขตข้อมูลของแฟ้มข้อมูลหลัก และระดับความสำคัญจะเรียงจากซ้ายไปขวา

ตัวอย่าง **ct** 0.uni 0.fac 0.dept
หมายถึง เขตข้อมูล uni fac dept ของแฟ้มข้อมูลหลัก เป็นเขตข้อมูลที่เป็นตัวคอนโทรลเบรค โดยที่เขตข้อมูลชื่อ uni จะมีระดับความสำคัญมากกว่าเขตข้อมูลชื่อ fac และเขตข้อมูลชื่อ fac มีระดับความสำคัญมากกว่าเขตข้อมูลชื่อ dept

ค่าสั่งที่เกี่ยวข้อง **ch cf**

ค่าสั่ง rh

รูปแบบ

```
rh col integernumber
row  ┐
     ├ integernumber
skip ┘
value  'message'
source filenumber.fieldname
          ┌var1=expression
          │
          │                    ┌ var2 ┐
compute   ├var1=functionname( ─┤      ├─ )
          │                    └constanst┘
          │
          └pageno=intergernumber .
[with attribute]
[pic format].
```

โดยที่ integernumber หมายถึง เลขจำนวนเต็ม

message หมายถึง ข้อความที่ต้องการพิมพ์

filenumber หมายถึง หมายเลขแฟ้มข้อมูล

fieldname หมายถึง ชื่อเขตข้อมูล

attribute หมายถึง คุณลักษณะของข้อมูล

format หมายถึง รูปแบบของข้อมูล

var1 หมายถึง ชื่อตัวแปรจัดเก็บผลลัพธ์

expression หมายถึง นิพจน์ทางคณิตศาสตร์

functionname หมายถึง ชื่อฟังก์ชัน

var2 หมายถึง ชื่อตัวแปรที่จะใช้ในการคำนวณ

constanst หมายถึง ค่าคงที่

หน้าที่ เป็นค่าสั่งที่ใช้ในการกำหนด ตำแหน่ง รูปแบบ คุณลักษณะของข้อมูลที่ต้องการพิมพ์เป็นหัวชุดรายงาน

ตัวอย่าง - rh col 1 row 1 value 'report heading'.

หมายถึงให้พิมพ์ข้อความ report heading ณ ตำแหน่ง บรรทัดที่ 1 คอลัมน์ที่ 1 เป็นหัวชุดรายงาน

- **rh col 1 row 1 source 0.code.**
  หมายถึงให้พิมพ์ค่าของเขตข้อมูลชื่อ code ของแฟ้มข้อมูลหลัก ณ ตำแหน่ง บรรทัดที่ 1 คอลัมน์ที่ 1 เป็นหัวชุดรายงาน
- **rh col 1 row 1 source 0.code with italic pic 9(4).**
  หมายถึง ให้พิมพ์ค่าของเขตข้อมูล code ของแฟ้มข้อมูลหลักเป็นหัวชุดรายงาน ณ ตำแหน่ง บรรทัดที่ 1 คอลัมน์ที่ 1 แบบตัวเอียง และ มีรูปแบบของการพิมพ์คือ 9(4)
- **rh col 1 row 1 compute tax = 0.7 * 0.salary with boldface pic z9,999.99.**
  หมายถึง ให้พิมพ์ค่าผลลัพธ์ที่ได้จากการคำนวณค่าของเขตข้อมูลชื่อ salary จากแฟ้มข้อมูลหลักคูณกับ 0.7 และเก็บผลลัพธ์ไว้ที่ตัวแปรชื่อ tax โดยคุณลักษณะการพิมพ์ให้พิมพ์แบบตัวเข้ม และมีรูปแบบการพิมพ์คือ z9,999.99

ค่าสั่งที่เกี่ยวข้อง **col row skip value source compute with pic**

ค่าสั่ง **col**

รูปแบบ **col integernumber**

โดยที่ integernumber หมายถึง เลขจำนวนเต็ม

หน้าที่ เป็นค่าสั่งที่ใช้ในการกำหนดตำแหน่งของคอลัมน์ที่จะพิมพ์

ตัวอย่าง **col 1**

หมายถึง เป็นการกำหนดตำแหน่งที่จะพิมพ์ ณ คอลัมน์ที่ 1

ค่าสั่งที่เกี่ยวข้อง **rh ph cl de pf rf ch และ cf**

ค่าสั่ง row

รูปแบบ row integernumber

โดยที่ integernumber หมายถึง เลขจำนวนเต็ม

หน้าที่ เป็นค่าสั่งที่ใช้ในการกำหนดบรรทัดที่จะพิมพ์

ตัวอย่าง row 1

หมายถึง เป็นการกำหนดบรรทัดที่จะพิมพ์ ณ บรรทัดที่ 1

ค่าสั่งที่เกี่ยวข้อง rh ph cl de pf rf ch และ cf

ค่าสั่ง skip

รูปแบบ skip integernumber

โดยที่ integernumber หมายถึง เลขจำนวนเต็ม

หน้าที่ เป็นค่าสั่งที่ใช้ในการกำหนดตำแหน่งบรรทัด โดยการอ้างอิงกับบรรทัดที่พิมพ์ครั้งล่าสุด

ตัวอย่าง skip 1

หมายถึง ให้ข้ามจากบรรทัดล่าสุดไป 1 บรรทัด

ค่าสั่งที่เกี่ยวข้อง rh ph cl de pf rf ch และ cf

ค่าสั่ง value

รูปแบบ value 'message'

โดยที่ message หมายถึง ข้อความที่ต้องการพิมพ์

หน้าที่ เป็นค่าสั่งที่ใช้ในการกำหนดข้อความที่ต้องการจะพิมพ์ โดยข้อความจะต้องอยู่ในเครื่องหมายอัญญประกาศ (' ')

ตัวอย่าง value 'chulalongkorn university'

หมายถึง ต้องการให้พิมพ์ข้อความ 'chulalongkorn university'

ค่าสั่งที่เกี่ยวข้อง rh ph cl de pf rf ch และ cf

ค่าสั่ง source

รูปแบบ source filenumber.fieldname

โดยที่ filenumber หมายถึง หมายเลขแฟ้มข้อมูล

fieldname หมายถึง ชื่อเขตข้อมูล

หน้าที่ เป็นค่าสั่งที่ใช้ในการกำหนดแหล่งของข้อมูล

ตัวอย่าง source 0.code

หมายถึง ใช้ข้อมูลจากเขตข้อมูลชื่อ code ของแฟ้มข้อมูลหลัก

ค่าสั่งที่เกี่ยวข้อง rh ph cl de pf rf ch และ cf

ค่าสั่ง compute

รูปแบบ

```
                ┌ var1=expression
                │
                │                    ┌ var2      ┐
compute         ├ var1=functionname( ┤           ├ )
                │                    └ constanst ┘
                │
                └ pageno=intergernumber
```

โดยที่ var1 หมายถึง ตัวแปรจัดเก็บผลลัพธ์

expression หมายถึง นิพจน์ทางคณิตศาสตร์

functionname หมายถึง ชื่อฟังก์ชัน

var2 หมายถึง ตัวแปร

constanst หมายถึง ค่าคงที่

integernumber หมายถึง ตัวเลขจำนวนเต็ม

หน้าที่ เป็นค่าสั่งที่ใช้ในการกำหนดให้คำนวณค่าของ นิพจน์ทางคณิตศาสตร์ หาค่าของฟังก์ชันหรือการกำหนดหมายเลขหน้า โดยเครื่องหมายที่ใช้ในนิพจน์ทางคณิตศาสตร์ ประกอบด้วยเครื่องหมายบวก ลบ คูณ หาร ยกกำลังสอง (+ - * / ^) และฟังก์ชันที่ใช้ในการคำนวณมี 4 ฟังก์ชันคือ

- SUM เป็นฟังก์ชันที่ใช้คำนวณหาค่าผลรวมของตัวแปร หรือค่าคงที่
- AVG เป็นฟังก์ชันที่ใช้คำนวณหาค่าเฉลี่ยของตัวแปร หรือ

ค่าคงที่

- STD เป็นฟังก์ชันที่ใช้คำนวณหาค่าความเบี่ยงเบนมาตรฐานของตัวแปรหรือค่าคงที่
- VAR เป็นฟังก์ชันที่ใช้คำนวณหาค่าความแปรปรวน ของตัวแปรหรือค่าคงที่

ตัวอย่าง

- **compute tax = 0.7 * 0.salary**
  หมายถึง ให้คูณค่าของเขตข้อมูลชื่อ salary จากแฟ้มข้อมูลหลักกับ0.7 แล้วเก็บผลลัพธ์ที่ได้ไว้ในตัวแปรชื่อ tax
- **compute sum1 = sum(0.salary)**
  หมายถึงให้คำนวณหาค่าผลรวมของเขตข้อมูลชื่อ salary ของแฟ้มข้อมูลหลัก แล้วเก็บผลลัพธ์ที่ได้ไว้ในตัวแปรชื่อ sum1
- **compute sum1 = sum(tax)**
  หมายถึง ให้คำนวณหาค่าผลรวมของตัวแปรชื่อ tax แล้วเก็บผลลัพธ์ที่ได้ไว้ในตัวแปรชื่อ sum1
- **compute sum1 = sum(100.75)**
  หมายถึง ให้คำนวณหาค่าผลรวมของ 100.75 แล้วเก็บผลลัพธ์ที่ได้ไว้ในตัวแปร ชื่อ sum1
- **compute avg1 = avg(0.salary)**
  หมายถึงให้คำนวณหาค่าเฉลี่ยของเขตข้อมูลชื่อ salary จากแฟ้มข้อมูลหลัก แล้วเก็บผลลัพธ์ที่ได้ไว้ในตัวแปรชื่อ avg1
- **compute pageno = 10**
  หมายถึง กำหนดให้หมายเลขหน้าเริ่มต้นของรายงานเป็นหมายเลข 10

คำสั่งที่เกี่ยวข้อง **rh ph cl de pf rf ch และ cf**

ค่าสั่ง **with**

รูปแบบ **with attribute**

โดยที่ attribute หมายถึง คุณลักษณะของข้อมูล

หน้าที่ เป็นค่าสั่งที่ใช้ในการสั่งให้พิมพ์ข้อมูล ตามคุณลักษณะที่กำหนด โดยคุณลักษณะต่าง ๆ มีดังต่อไปนี้

- singleline ขีดเส้นใต้ 1 เส้น
- doubleline ขีดเส้นใต้ 2 เส้น
- boldface ตัวเข้ม
- enlarge ตัวหนา
- italic ตัวเอียง

ตัวอย่าง **with singleline**

หมายถึง ให้พิมพ์ข้อมูลโดยคุณลักษณะขีดเส้นใต้ 1 เส้น

คำสั่งที่เกี่ยวข้อง **rh ph cl de pf rf ch** และ **cf**

ค่าสั่ง **pic**

รูปแบบ **pic format**

โดยที่ format หมายถึง รูปแบบของข้อมูล

หน้าที่ เป็นค่าสั่งที่ใช้ในการสั่งให้พิมพ์ข้อมูลตามรูปแบบที่กำหนด ซึ่งรูปแบบในการพิมพ์นี้ จะเหมือนกับรูปแบบในการพิมพ์ของภาษาโคบอล

ตัวอย่าง **pic x(10)**

หมายถึง การพิมพ์ข้อมูลตามรูปแบบ x(10)

คำสั่งที่เกี่ยวข้อง **rh ph cl de pf rf ch** และ **cf**

ค่าสั่ง **ph**

รูปแบบ เหมือนคำสั่ง **rh**

หน้าที่ เป็นค่าสั่งที่ใช้ในการกำหนดตำแหน่ง รูปแบบ คุณลักษณะของข้อมูลที่ต้องการจะพิมพ์เป็นหัวรายงาน

ตัวอย่าง **ph col 1 row 2 value 'page heading'.**

หมายถึง ให้พิมพ์ข้อความ page heading เป็นหัวรายงาน ณ ตำแหน่ง บรรทัดที่ 2 คอลัมน์ที่ 1

ค่าสั่งที่เกี่ยวข้อง **rh ph cl de pf rf ch และ cf**

ค่าสั่ง **cl**

รูปแบบ เหมือนค่าสั่ง **rh**

หน้าที่ เป็นค่าสั่งที่ใช้ในการกำหนดตำแหน่ง รูปแบบ คุณลักษณะของข้อมูลที่ต้องการจะพิมพ์เป็นหัวรายการ

ตัวอย่าง **cl col 1 row 3 value 'code' .**

หมายถึง ให้พิมพ์ข้อความ code เป็นหัวรายการ ณ ตำแหน่ง บรรทัดที่ 3 คอลัมน์ที่ 1

ค่าสั่งที่เกี่ยวข้อง **rh ph cl de pf rf ch และ cf**

ค่าสั่ง ch

รูปแบบ

```
ch by filenumber.fieldname
col integernumber
row  ┐
     ├ integernumber
skip ┘
value 'message'
source filenumber.fieldname
compute ┬ var1=expression
        ├ var1=functionname( ┌ var2      ┐ )
        │                    └ constanst ┘
        └ pageno=integernumber

[with attribute]
[pic format].
```

โดยที่ filenumber หมายถึง หมายเลขแฟ้มข้อมูล

fieldname หมายถึง ชื่อเขตข้อมูล

integernumber หมายถึง ตัวเลขจำนวนเต็ม

filenumber หมายถึง หมายเลขแฟ้มข้อมูล

message หมายถึง ข้อความที่ต้องการพิมพ์

attribute หมายถึง คุณลักษณะของข้อมูล

format หมายถึง รูปแบบของข้อมูล

var1 หมายถึง ชื่อตัวแปรจัดเก็บผลลัพธ์

expression หมายถึง นิพจน์ทางคณิตศาสตร์

functionname หมายถึง ชื่อฟังก์ชัน

var2 หมายถึง ชื่อตัวแปรที่จะใช้ในการคำนวณ

constanst หมายถึง ค่าคงที่

หน้าที่ เป็นคำสั่งที่ใช้ในการกำหนดตำแหน่ง รูปแบบ คุณลักษณะของข้อมูลที่ต้องการจะพิมพ์เป็นหัวรายงานย่อย

ตัวอย่าง **ch by 0.code col 1 row 1 value 'control break heading by code'.**

หมายถึง ถ้าค่าของเขตข้อมูลชื่อ code ของแฟ้มข้อมูลหลักเปลี่ยนไปให้พิมพ์ข้อความ control break heading by code เป็นหัวรายงานย่อย ณ ตำแหน่ง บรรทัดที่ 1 คอลัมน์ที่ 1

คำสั่งที่เกี่ยวข้อง **rh ph cl de pf rf ch** และ **cf**

คำสั่ง **de**

รูปแบบ เหมือนคำสั่ง rh

หน้าที่ เป็นคำสั่งที่ใช้ในการกำหนดตำแหน่ง รูปแบบ คุณลักษณะ ของข้อมูลที่ต้องการจะพิมพ์ของรายการข้อมูลแต่ละรายการ

ตัวอย่าง **de col 1 row 4 source 0.code .**

หมายถึง รายการข้อมูลแต่ละรายการให้พิมพ์ค่าของเขตข้อมูลชื่อ code ของแฟ้มข้อมูลหลัก

คำสั่งที่เกี่ยวข้อง **rh ph cl de pf rf ch** และ **cf**

คำสั่ง **cf**

รูปแบบ เหมือนคำสั่ง ch

หน้าที่ เป็นคำสั่งที่ใช้ในการกำหนดตำแหน่ง รูปแบบ คุณลักษณะ ของข้อมูลที่ต้องการจะพิมพ์เป็นท้ายรายงานย่อย

ตัวอย่าง **cf by 0.code col 1 row 25 value 'control break footing by code'.**

หมายถึง ถ้าค่าของเขตข้อมูลชื่อ code ของแฟ้มข้อมูลหลักเปลี่ยนไปให้พิมพ์ข้อความ control break footing by code เป็นท้ายรายงานย่อย ณ ตำแหน่ง บรรทัดที่ 1 คอลัมน์ที่ 1

ค่าสั่ง **pf**

รูปแบบ เหมือนค่าสั่ง **rh**

หน้าที่ เป็นค่าสั่งที่ใช้ในการกำหนดตำแหน่ง รูปแบบ คุณลักษณะ ของข้อมูลที่ต้องการจะพิมพ์เป็นท้ายรายงาน

ตัวอย่าง **pf col 1 row 30 value 'page footing'.**

หมายถึง ให้พิมพ์ข้อความ page footing เป็นท้ายรายงาน ณ ตำแหน่ง บรรทัดที่ 30 คอลัมน์ที่ 1

ค่าสั่งที่เกี่ยวข้อง **rh ph cl de pf rf ch** และ **cf**

ค่าสั่ง **rf**

รูปแบบ เหมือนค่าสั่ง **rh**

หน้าที่ เป็นค่าสั่งที่ใช้ในการกำหนดตำแหน่ง รูปแบบ คุณลักษณะ ของข้อมูลที่ต้องการจะพิมพ์เป็นท้ายชุดรายงาน

ตัวอย่าง **rf col 1 row 32 value 'report footing'.**

หมายถึง ให้พิมพ์ข้อความ report footing เป็นท้ายชุดรายงาน ณ ตำแหน่ง บรรทัดที่ 32 คอลัมน์ที่ 1

ค่าสั่งที่เกี่ยวข้อง **rh ph cl de pf rf ch** และ **cf**

2. การออกแบบแฟ้มข้อมูล

จากโครงสร้างของระบบ พบว่ามีแฟ้มข้อมูลต่าง ๆ ที่ต้องออกแบบ ดังนี้

2.1 แฟ้มโครงสร้างข้อมูล

แฟ้มโครงสร้างข้อมูล เป็นแฟ้มที่เก็บบันทึกโครงสร้างของแฟ้มข้อมูลประเภท Btrive ISAM และ Text ซึ่งผู้ใช้ต้องกำหนดโครงสร้างก่อนที่จะดำเนินการต่าง ๆ แฟ้มข้อมูลนี้มีโครงสร้างดังปรากฏดังตารางที่ 3.3

เขตข้อมูล	ข้อมูลที่จัดเก็บ	ความยาว	หมายเหตุ
1	จำนวนเขตข้อมูลทั้งหมด	2	
2	ชื่อเขตข้อมูล	25	มีจำนวนซ้ำเท่ากับค่าของข้อมูลในเขตข้อมูลที่ 1
3	ประเภทของเขตข้อมูล	1	
4	ความยาวของเขตข้อมูล	2	
5	จำนวนของตัวเลขหลังจุดทศนิยม	2	
6	รหัสสำหรับบอกว่าเป็นเขตข้อมูลดัชนีหรือไม่	1	
7	หมายเลขคีย์	1	

ตารางที่ 3.3 โครงสร้างแฟ้มข้อมูลที่จัดเก็บโครงสร้างของแฟ้มข้อมูลอื่น ๆ

2.2 แฟ้มข้อมูลรูปแบบรายงาน

แฟ้มข้อมูลรูปแบบรายงาน เป็นแฟ้มที่เก็บบันทึกรายละเอียดต่าง ๆ ของแฟ้มข้อมูลที่จะนำมาพิมพ์รายงานซึ่งมีรายละเอียดดังปรากฏดังตารางที่ 3.4 ถ้าความยาวเท่ากับ V (Variable) หมายถึง ความยาวที่จัดเก็บขึ้นอยู่กับจำนวนข้อมูลในเขตข้อมูลนั้น ๆ โดยตำแหน่งการจัดเก็บจะจัดเรียงกันไปตามลำดับ

เขตข้อมูล	ข้อมูลที่เก็บ	ความยาว	หมายเหตุ
1	ประเภทของแฟ้มข้อมูล	1	
2	จำนวนตัวอักขระและคุณลักษณะของข้อมูลทั้งหมด	2	
3	ตัวอักขระและคุณลักษณะของข้อมูล	V	
4	จำนวนเขตข้อมูลที่จะปรากฏในรายงานของบรรทัดที่ 1 ถึงบรรทัดสุดท้ายของแต่ละหน้า	2	ซ้ำเท่ากับจำนวนบรรทัดต่อหน้ารายงาน
5	ความยาวของชื่อแฟ้ม ชื่อเขตข้อมูล สูตรที่ใช้ และรูปแบบการพิมพ์	2	ซ้ำเท่ากับค่าที่จัดเก็บในเขตข้อมูลที่ 4
6	ชื่อแฟ้มข้อมูล	V	
7	ชื่อเขตข้อมูล	V	
8	สูตรที่ใช้ในการคำนวณ		
9	รูปแบบการพิมพ์	V	
10	รหัสคุณลักษณะของข้อมูล	1	
11	ตำแหน่งคอลัมน์ที่จะพิมพ์	1	
12	จำนวนแฟ้มข้อมูล	1	
13	ความยาวของชื่อแฟ้มข้อมูล	1	ซ้ำเท่ากับค่าที่จัดเก็บในเขตข้อมูลที่ 12
14	ชื่อแฟ้มข้อมูล	V	
15	ความยาวของชื่อแฟ้มข้อมูลดัชนี	1	
16	ชื่อแฟ้มข้อมูลดัชนี	V	
17	ความยาวของชื่อแฟ้มโครงสร้างข้อมูล	1	
18	ชื่อแฟ้มโครงสร้างข้อมูล	V	

ตารางที่ 3.4 โครงสร้างแฟ้มข้อมูลรูปแบบรายงาน

เขตข้อมูล	ข้อมูลที่เก็บ	ความยาว	หมายเหตุ
19	ตำแหน่งบรรทัดแรกของหัวชุดรายงาน	1	
20	ตำแหน่งบรรทัดสุดท้ายของหัวชุดรายงาน	1	
21	ตำแหน่งบรรทัดแรกของหัวรายงาน	1	
22	ตำแหน่งบรรทัดสุดท้ายของหัวรายงาน	1	
23	ตำแหน่งบรรทัดแรกของหัวรายการ	1	
24	ตำแหน่งบรรทัดสุดท้ายของหัวรายการ	1	
25	ตำแหน่งบรรทัดแรกของท้ายรายงาน	1	
26	ตำแหน่งบรรทัดสุดท้ายของท้ายรายงาน	1	
27	ตำแหน่งบรรทัดแรกของท้ายชุดรายงาน	1	
28	ตำแหน่งบรรทัดสุดท้ายของท้ายชุดรายงาน	1	
29	ตำแหน่งบรรทัดแรกของรายการข้อมูล	1	
30	ตำแหน่งคอลัมน์แรกของรายการข้อมูล	1	
31	ตำแหน่งบรรทัดสุดท้ายของรายการข้อมูล	1	

ตารางที่ 3.4 โครงสร้างแฟ้มข้อมูลรูปแบบรายงาน (ต่อ)

เขตข้อมูล	ข้อมูลที่เก็บ	ความยาว	หมายเหตุ
32	จำนวนเขตข้อมูลคอนโทรลเบรค	2	
33	ความยาวของชื่อเขตข้อมูล	2	ซ้ำเท่ากับค่าที่จัดเก็บในเขตข้อมูลที่ 32
34	ชื่อเขตข้อมูล	V	
35	รหัสขึ้นหน้าใหม่	1	
36	จำนวนเขตข้อมูลของหัวรายการย่อย	2	
37	ความยาวของชื่อเขตข้อมูล	1	ซ้ำเท่ากับค่าที่จัดเก็บในเขตข้อมูลที่ 36
38	ชื่อเขตข้อมูล	V	
39	ตำแหน่งบรรทัดแรกของหัวรายการย่อย	1	
40	ตำแหน่งคอลัมน์แรกของหัวรายงานย่อย	1	
41	ตำแหน่งบรรทัดสุดท้ายของหัวรายงานย่อย	1	
42	ตำแหน่งคอลัมน์สุดท้ายของหัวรายงานย่อย	1	
43	จำนวนเขตข้อมูลของท้ายรายงานย่อย	2	

ตารางที่ 3.4 โครงสร้างแฟ้มข้อมูลรูปแบบรายงาน (ต่อ)

เขตข้อมูล	ข้อมูลที่เก็บ	ความยาว	หมายเหตุ
44	ความยาวของชื่อเขตข้อมูล	1	
45	ชื่อเขตข้อมูล	V	
46	ตำแหน่งบรรทัดแรกของท้ายรายงานย่อย	V	
47	ตำแหน่งคอลัมน์แรกของท้ายรายงานย่อย	1	ซ้ำเท่ากับค่าที่จัดเก็บอยู่ในเขตข้อมูลที่ 43
48	ตำแหน่งบรรทัดสุดท้ายของท้ายรายงานย่อย	1	
49	ตำแหน่งคอลัมน์สุดท้ายของท้ายรายงานย่อย	1	
50	จำนวนความสัมพันธ์ระหว่างแฟ้มข้อมูล	2	
51	ความยาวของชื่อแฟ้มข้อมูลที่ 1	1	
52	ชื่อแฟ้มข้อมูลที่ 1	V	
53	ความยาวของชื่อเขตข้อมูลของแฟ้มข้อมูลที่ 1	1	
54	ชื่อเขตข้อมูล ของแฟ้มข้อมูลที่ 1	V	ซ้ำเท่ากับค่าที่จัดเก็บอยู่ในเขตข้อมูลที่ 50
55	ความยาวของชื่อแฟ้มข้อมูลที่ 2	1	
56	ชื่อแฟ้มข้อมูลที่ 2	V	
57	ความยาวของชื่อเขตข้อมูลของแฟ้มข้อมูลที่ 2	1	
58	ชื่อเขตข้อมูล ของแฟ้มข้อมูลที่ 2	V	
59	จำนวนบรรทัดต่อหน้าของรายงาน	1	
60	จำนวนตัวอักษรต่อบรรทัด	1	

ตารางที่ 3.4 โครงสร้างแฟ้มข้อมูลรูปแบบรายงาน (ต่อ)

เขตข้อมูล	ข้อมูลที่เก็บ	ความยาว	หมายเหตุ
61	ความยาวของชื่อสารบบหลัก	1	
62	ชื่อของสารบบหลัก	V	
63	ความยาวของเงื่อนไขที่ใช้ในการออกรายงาน	2	
64	เงื่อนไขที่ใช้ในการออกรายงาน	V	

ตารางที่ 3.4 โครงสร้างแฟ้มข้อมูลรูปแบบรายงาน (ต่อ)

2.3 แฟ้มข้อมูลภาษาโปรแกรมระดับที่ 4

แฟ้มข้อมูลภาษาโปรแกรมระดับที่ 4 เป็นแฟ้มที่เก็บบันทึกโปรแกรมภาษาระดับที่ 4 ซึ่งออกแบบให้เก็บเป็นแฟ้มข้อมูลอักขระ (Text File)

2.4 แฟ้มข้อมูลโปรแกรมภาษาโคบอล

แฟ้มข้อมูลโปรแกรมภาษาโคบอล เป็นแฟ้มที่เก็บบันทึกโปรแกรมภาษาโคบอล ซึ่งออกแบบให้เก็บเป็นแฟ้มข้อมูลอักขระ เพื่อที่ผู้ใช้สามารถนำไปประมวลผลต่อไปได้

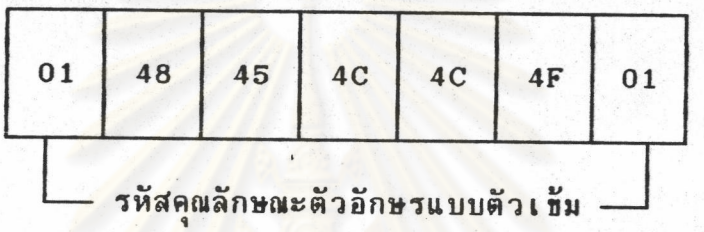
2.5 แฟ้มข้อมูลรายงานที่ได้จากโปรแกรมภาษาโคบอล เป็นแฟ้มที่จัดเก็บรายงานที่ได้จากการนำโปรแกรมภาษาโคบอลที่ถูกสร้างขึ้น ไปผ่านตัวแปลภาษาแล้วทำการประมวลผล ซึ่งผู้วิจัยได้ออกแบบให้แฟ้มข้อมูลรายงานที่ได้นี้จัดเก็บข้อมูลแบบอักขระ เมื่อผู้ใช้ต้องการดูรายงานที่ได้ก็สามารถนำแฟ้มข้อมูลนี้ไปแสดงผลทางจอภาพหรือทางเครื่องพิมพ์ได้

ถ้ารายงานที่ได้นั้นเป็นภาษาอังกฤษ และคุณลักษณะตัวอักษรเป็นแบบธรรมดาก็สามารถแสดงผลทางจอภาพหรือเครื่องพิมพ์ได้ทันที แต่เนื่องจากผู้วิจัยต้องการให้ระบบนี้สามารถสร้างรายงานเป็นภาษาไทย และสามารถพิมพ์ตัวอักษรตามลักษณะตัวอักษรแบบต่าง ๆ ได้ จึงได้ออกแบบให้ใช้รหัสภาษาไทย สมอ. และ ให้ใช้รหัสแอสกี (ASCII) แทนคุณลักษณะตัวอักษรแบบต่าง ๆ ต่อไปนี้

- ตัวอักษรแบบตัวเข้ม ใช้รหัส 01

- ตัวอักษรแบบตัวใหญ่ ใช้รหัส 05
- ตัวอักษรแบบขีดเส้นใต้ 2 เส้น ใช้รหัส 12
- ตัวอักษรแบบขีดเส้นใต้ 1 เส้น ใช้รหัส 13
- ตัวอักษรแบบตัวเอียง ใช้รหัส 17

ในการใช้รหัสแอสกีแทนคุณลักษณะตัวอักษรแบบต่าง ๆ นั้น ผู้วิจัยได้ออกแบบให้ข้อมูลที่ต้องการพิมพ์ตามคุณลักษณะตัวอักษรแบบต่าง ๆ นั้น อยู่ระหว่างกลางระหว่างรหัสแอสกีเหล่านี้ เช่นถ้าต้องการพิมพ์ข้อความ HELLO แบบตัวเข้ม แฟ้มข้อมูลรายงานจะจัดเก็บข้อมูลดังภาพที่ 3.5



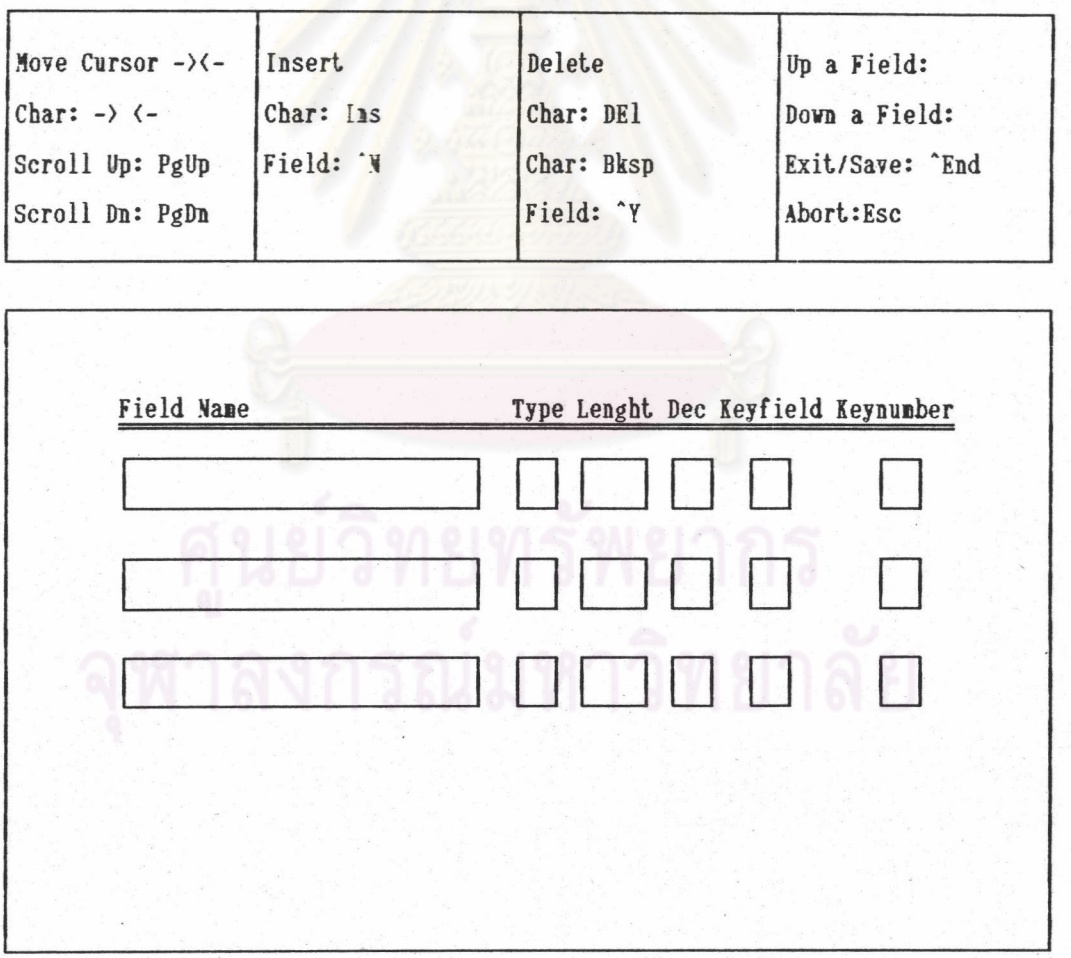
ภาพที่ 3.5 ตัวอย่างข้อมูลที่จัดเก็บอยู่ในแฟ้มข้อมูลรายงานแบบตัวเข้ม

ดังนั้นถ้ารายงานที่ได้นั้นเป็นภาษาไทย หรือ มีการพิมพ์ตัวอักษรตามคุณลักษณะตัวอักษรแบบต่าง ๆ จะต้องแสดงผลโดยผ่านตัวขับ (Driver) ที่ใช้รหัสภาษาไทย สมอ. และใช้รหัสแอสกี แทนคุณลักษณะตัวอักษรชุดเดียวกันกับที่ผู้วิจัยได้ออกแบบไว้ ซึ่งในที่นี้ผู้วิจัยได้ใช้ซอฟต์แวร์ CU Writer สำหรับการแสดงผล

3. <u>การออกแบบการนำข้อมูลเข้า</u>

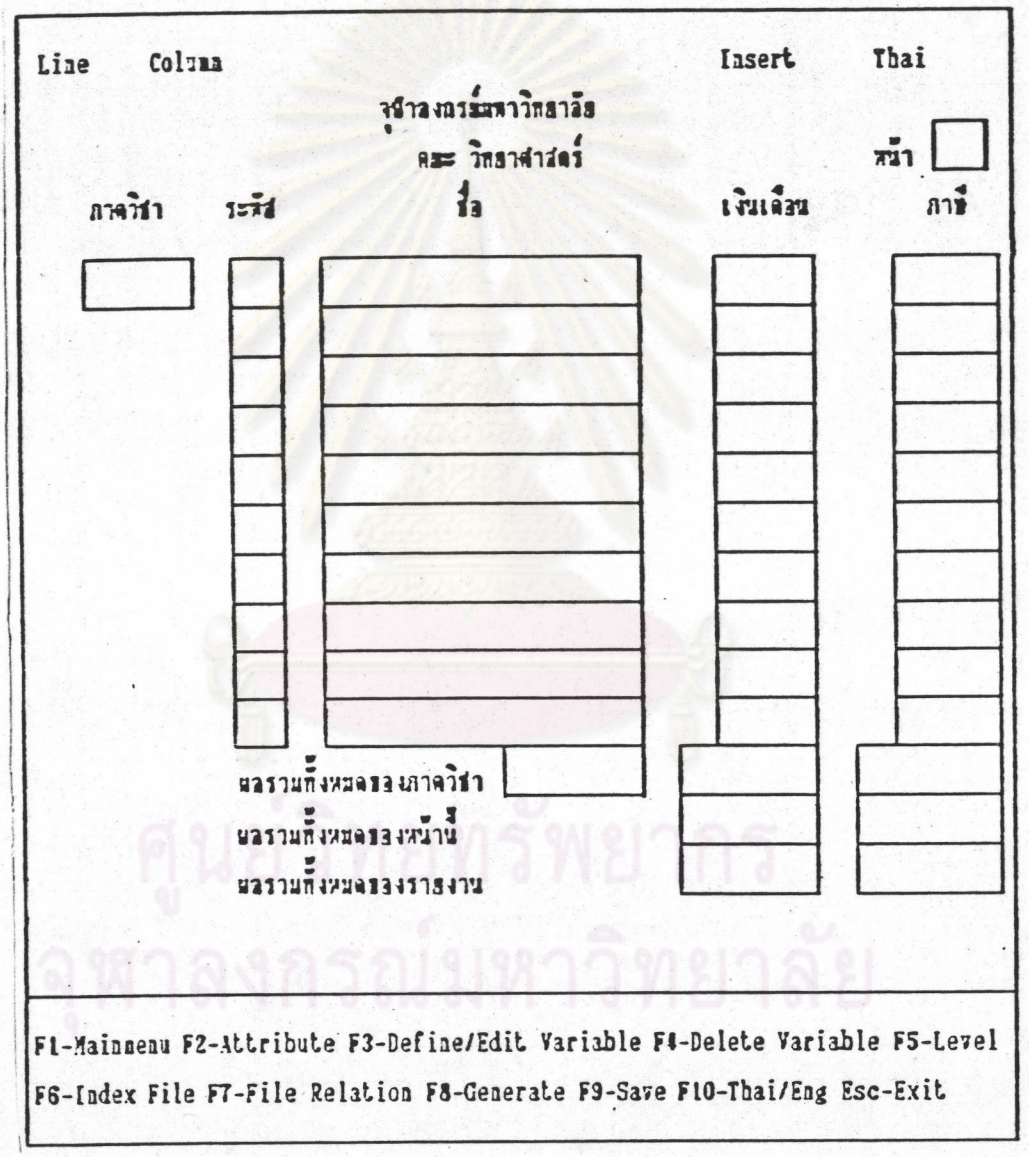
การออกแบบการนำข้อมูลเข้าที่จำเป็นสำหรับระบบนี้ ประกอบไปด้วย

3.1 การกำหนดโครงสร้างแฟ้มข้อมูลทางจอภาพ เป็นการกำหนดโครงสร้างของแฟ้มข้อมูลประเภท ISAM Btrieve และ Text เพื่อเก็บไว้ในแฟ้มข้อมูลสำหรับการเรียกใช้ต่อไป ซึ่งรูปแบบของการจัดเก็บนั้นได้กล่าวไว้แล้วในหัวข้อที่ 2.1 ซึ่งการออกแบบการบันทึกข้อมูลทางจอภาพมีรายละเอียดดังปรากฏใน รูปที่ 3.6



รูปที่ 3.6 จอภาพของการกำหนดโครงสร้างแฟ้มข้อมูล

3.2 การกำหนดรูปแบบรายงานทางจอภาพ เป็นการกำหนดรูปแบบรายงานและรายละเอียดต่าง ๆ ดังปรากฏในภาพที่ 3.7 ซึ่งรายละเอียดต่าง ๆ นี้จะจัดเก็บไว้ในแฟ้มข้อมูลรูปแบบรายงานเพื่อความสะดวกในการเรียกใช้ในภายหลัง ซึ่งรูปแบบการจัดเก็บนั้นได้กล่าวไว้แล้วในหัวข้อที่ 2.2



รูปที่ 3.7 จอภาพของการกำหนดรูปแบบรายงาน

3.3 การสร้างโปรแกรมภาษาระดับที่ 4

การสร้างโปรแกรมภาษาระดับที่ 4 เป็นการสร้างแฟ้มข้อมูลเพื่อเก็บโปรแกรมภาษาระดับที่ 4 ที่สร้างขึ้นโดยผู้ใช้ที่มีความคุ้นเคยกับภาษาโปรแกรมระดับที่ 4 ที่ออกแบบไว้ และ มีความประสงค์จะสร้างโปรแกรมภาษาระดับที่ 4 ขึ้นใช้เอง โดยไม่ต้องการกำหนดรูปแบบรายงานทางจอภาพก็สามารถทำได้ โดยการเขียนโปรแกรมผ่านตัวบรรณาธิกรต่าง ๆ บันทึกโปรแกรมที่ต้องการลงแฟ้มข้อมูล แล้วจึงนำแฟ้มข้อมูลนั้นไปเป็นข้อมูลเข้าของระบบต่อไป

4. <u>การออกแบบโปรแกรม</u>

โปรแกรมที่เกี่ยวข้องกับระบบประกอบไปด้วย 5 โปรแกรมดังนี้

4.1 โปรแกรมกำหนดโครงสร้างแฟ้มข้อมูล (Createst.exe)

โปรแกรมกำหนดโครงสร้างแฟ้มข้อมูล เป็นโปรแกรมที่ใช้สำหรับกำหนดโครงสร้างแฟ้มข้อมูลของแฟ้มข้อมูลประเภท Btrieve ISAM และ Text File เนื่องจากแฟ้มข้อมูลประเภทนี้ไม่มีการจัดเก็บโครงสร้างไว้ ส่วนแฟ้มข้อมูลประเภท dBASE และ Foxbase มีโครงสร้างแฟ้มข้อมูลจัดเก็บอยู่แล้วจึงไม่ต้องเรียกใช้โปรแกรมนี้ ผลของโปรแกรมนี้จะได้แฟ้มข้อมูลที่เก็บโครงสร้างแฟ้มข้อมูลซึ่งมีลักษณะดังที่กล่าวไว้ในหัวข้อที่ 2.1

4.2 โปรแกรมกำหนดรูปแบบรายงานทางจอภาพ (User.exe)

โปรแกรมกำหนดรูปแบบรายงานทางจอภาพเป็นโปรแกรมที่อนุญาตให้ผู้ใช้กำหนดรูปแบบรายงานผ่านจอภาพ โดยใช้เทคนิคแบบพูลดาวน์เมนู (Pull Down Menu) และ การบรรณาธิกรข้อมูลทางจอภาพ (Screen Editor) ผลของโปรแกรมนี้จะได้แฟ้มข้อมูลของภาษาโปรแกรมระดับที่ 4 (4GL File) และ แฟ้มข้อมูลที่เก็บข้อมูลเกี่ยวกับรูปแบบรายงานที่ผู้ใช้ต้องการ ซึ่งกล่าวไว้ในหัวข้อ 2.2 และ 2.3 ตามลำดับ

4.3 โปรแกรมแสดงผลทางจอภาพ (Monitor.exe)

โปรแกรมแสดงผลทางจอภาพ เป็นโปรแกรมที่นำเอาข้อมูลในแฟ้มข้อมูลโครงสร้าง ข้อมูลในแฟ้มข้อมูลที่จะพิมพ์รายงาน และ แฟ้มข้อมูลภาษาโปรแกรมระดับที่ 4 มาประมวลผลเพื่อออกรายงานทางจอภาพ

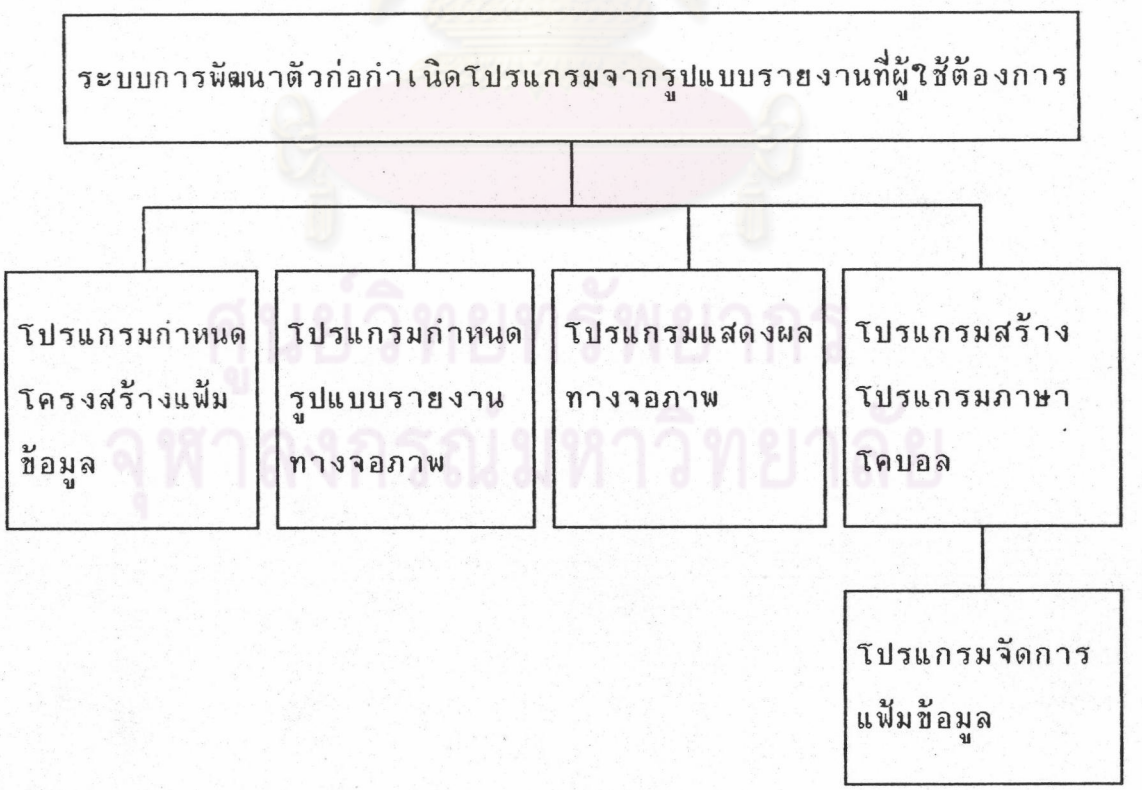
4.4 โปรแกรมสร้างโปรแกรมภาษาโคบอล (Program.exe)

โปรแกรมสร้างโปรแกรมภาษาโคบอล เป็นโปรแกรมที่นำข้อมูลในแฟ้มข้อมูลโครงสร้าง ข้อมูลแฟ้มข้อมูลที่จะพิมพ์รายงาน แฟ้มข้อมูลภาษาโปรแกรมระดับที่ 4 มาประมวลผลเพื่อสร้างเป็นแฟ้มข้อมูลภาษาโคบอลซึ่งผู้ใช้สามารถนำโปรแกรมภาษาโคบอลนี้ไปประมวลผลเพื่อออกรายงานตามต้องการได้

4.5 โปรแกรมจัดการแฟ้มข้อมูล

เนื่องจากภาษาโคบอล ไม่สามารถอ่านข้อมูลจากแฟ้มข้อมูล dBASE Foxbase และ Text File บางประเภทได้ ดังนั้นจึงจำเป็นต้องทำการพัฒนาโปรแกรมจัดการแฟ้มข้อมูลขึ้นมา เพื่อให้โปรแกรมภาษาโคบอลที่สร้างขึ้นมานั้นสามารถเรียกใช้ข้อมูลจากแฟ้มข้อมูลประเภทต่าง ๆ ดังกล่าวข้างต้นได้ ซึ่งโปรแกรมจัดการแฟ้มข้อมูลนี้พัฒนาขึ้นโดยใช้ภาษาแอสเซมบลี 8088/8086 จากนั้นนำไปผ่านตัวแอสเซมเบลอร์ผลลัพธ์ที่ได้คือโปรแกรมภาษาจุดหมาย (Object Program) ทำให้สามารถเชื่อมโยง (Link) เข้ากับภาษาโคบอลที่ถูกสร้างขึ้นได้

ผังโครงสร้างโปรแกรมของระบบแสดงได้ดังรูปที่ 3.8 สำหรับรายละเอียดของแต่ละโปรแกรมจะกล่าวถึงในบทที่ 4



รูปที่ 3.8 ผังโครงสร้างโปรแกรมของระบบ