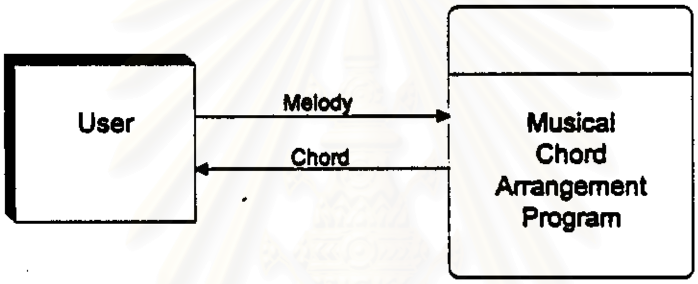
บทที่ 5

การทำงานของโปรแกรม

ในบทนี้จะกล่าวถึงการทำงานของโปรแกรม โดยจะใช้แผนภาพกระแสข้อมูล (Data Flow Diagram) และคำสั่งเทียม (Pseduo Instruction) เป็นสื่อในการอธิบาย รวมถึงจะใช้ความรู้ของบทที่ 3 ทฤษฎีดนตรี และบทที่ 4 ระเบียบวิธีการจัดคอร์ดเพลง ประกอบด้วย

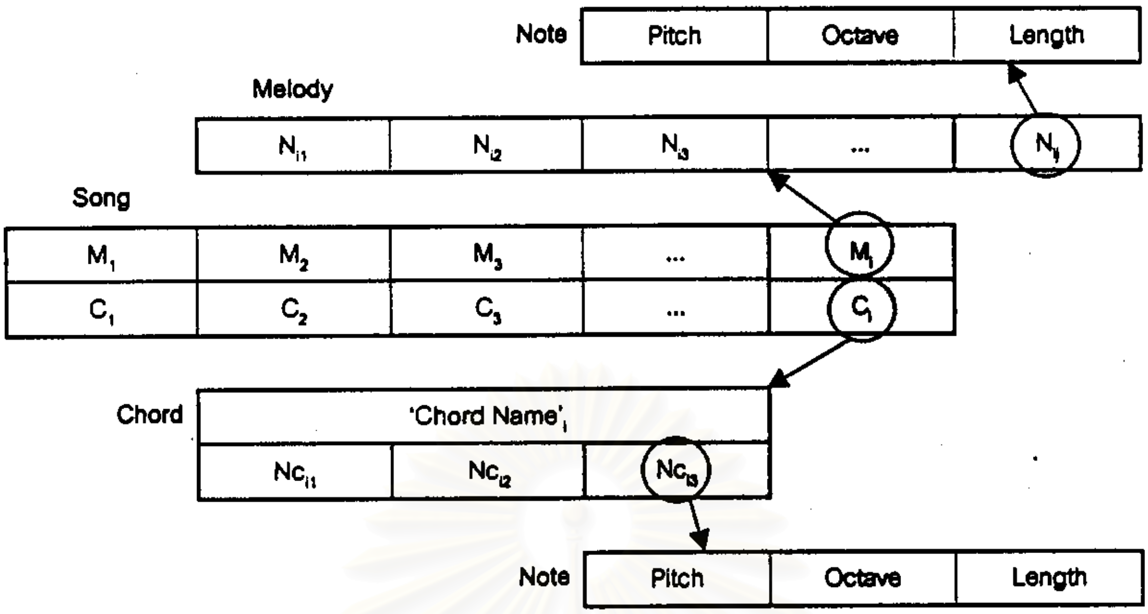
5.1 ภาพรวมการทำงานของโปรแกรม



รูปภาพที่ 5.1 ภาพรวมการทำงานของโปรแกรมจัดคอร์ดเพลง (Context Diagram)

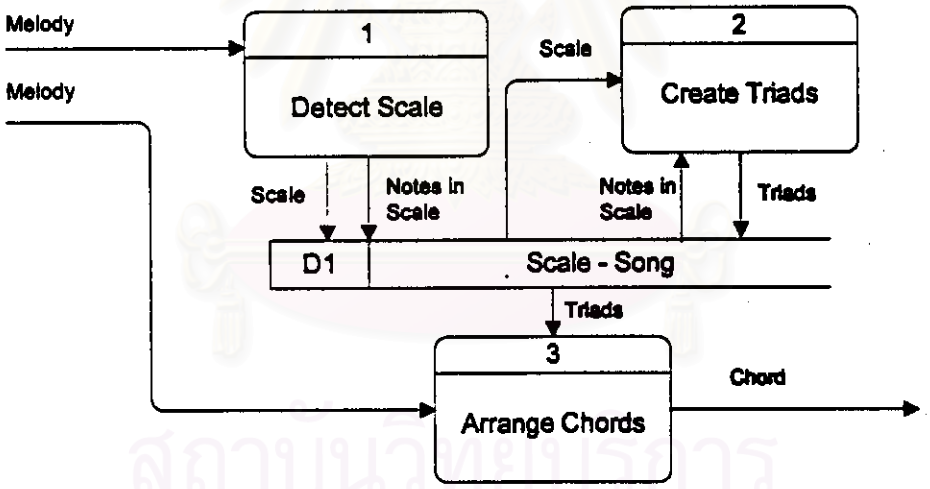
จากรูปภาพที่ 5.1 จะแสดงภาพรวมการทำงานของโปรแกรม ซึ่ง *"ผู้ใช้" (User)* จะเป็นผู้ป้อน *"ทำนองเพลง" (Melody)* ผ่านหน้าจอป้อนข้อมูลเข้าสู่ *"โปรแกรม" (Musical Chord Arrangement Program)* เพื่อให้โปรแกรมประมวลผล เมื่อเสร็จสิ้น โปรแกรมจะส่ง *"คอร์ดเพลง" (Chord)* ที่ได้คำนวณว่าเหมาะสมกับทำนองเพลงให้กับผู้ใช้

โครงสร้างข้อมูล *"เพลง" (Song)* ซึ่งประกอบด้วย *"ทำนองเพลง"* และ *"คอร์ดเพลง"* แสดงได้ดังรูปภาพที่ 5.2 โดยกำหนดให้ i แทนหมายเลขห้องเพลง และ j แทนหมายเลขโน้ตในห้องเพลงที่ i ดังนั้น M_i จะแทนทำนองเพลงในห้องเพลงที่ i N_{ij} แทนโน้ตเพลงที่ j ในห้องเพลงที่ i และภายในโน้ต N_{ij} แต่ละตัวจะประกอบด้วยระดับเสียง (Pitch) ขั้นคู่แปด (Octave) และความยาวโน้ต (Length) ส่วน C_i แทนคอร์ดเพลง ที่ใช้ประสานเสียงในห้องเพลงที่ i ภายในคอร์ดเพลงประกอบด้วย ชื่อคอร์ด ('Chord Name') และโน้ตในคอร์ด Nc_{i1} ถึง Nc_{i3} จำนวน 3 ตัว และภายในโน้ตแต่ละตัวจะประกอบด้วยระดับเสียง ขั้นคู่แปด และความยาวโน้ตเช่นเดียวกับกับโน้ตในทำนองเพลง



รูปภาพที่ 5.2 โครงสร้างข้อมูลของทำนองเพลงและคอร์ดเพลง

5.2 ส่วนประกอบของโปรแกรม



รูปภาพที่ 5.3 แผนภาพกระแสข้อมูลระดับที่ 1 (DFD Level 1)

จากรูปภาพที่ 5.3 โปรแกรมจะประกอบด้วยส่วนการทำงาน 3 ส่วนคือ

ส่วนแรก *"ส่วนตรวจสอบบันไดเสียง" ("Detect Scale" Process)* ทำหน้าที่ตรวจสอบว่าในทำนองเพลงที่ผู้ใช้ป้อนข้อมูลเข้ามามีบันไดเสียงใดเป็นบันไดเสียงของเพลง

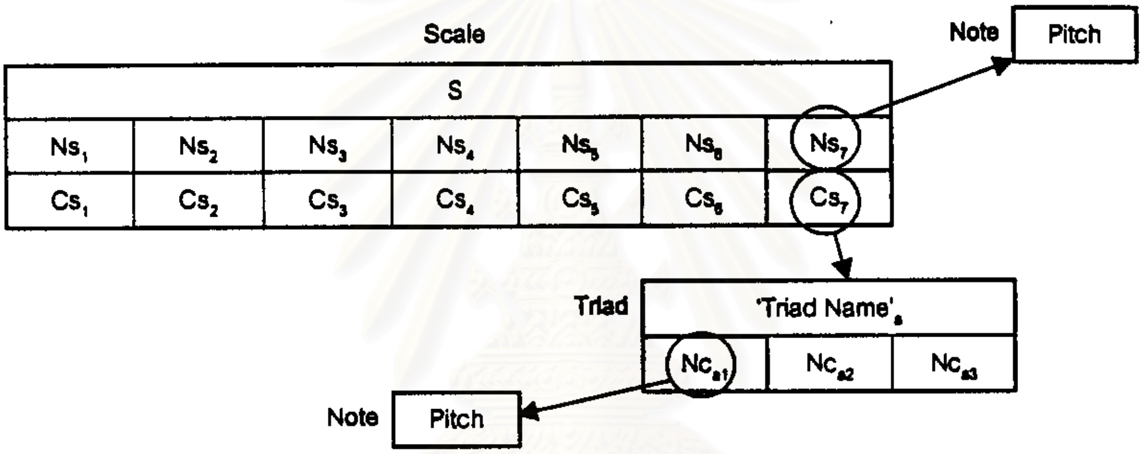
ส่วนที่สอง *"ส่วนสร้างตรัยเอ็ด" ("Create Triad" Process)* ทำหน้าที่สร้างคลังคอร์ดเพลงแบบตรัยเอ็ดจากบันไดเสียงเพื่อเตรียมนำไปจัดให้กับทำนองเพลง

ส่วนที่สาม *"ส่วนจัดคอร์ดเพลง" ("Arrange Chords" Process)* ทำหน้าที่เลือกคอร์ดเพลงที่เหมาะสมจากคลังคอร์ดเพลงเพื่อใช้ประสานเสียงกับทำนองเพลง โดยจะใส่คอร์ดเพลงหนึ่งคอร์ดต่อหนึ่งห้องเพลง

ใน *"แหล่งเก็บข้อมูลบันไดเสียงของเพลง" ("Scale – Song" Data Store)* จะเป็นที่เก็บข้อมูลของ *"บันไดเสียง" (Scale)* *"โน้ตในบันไดเสียง" (Notes in Scale)* และ *"ตรัยเอ็ด" (Triads)* ในบันไดเสียง ซึ่งแหล่งเก็บข้อมูลนี้จะเป็นส่วนที่ให้บริการข้อมูลคอร์ดเพลงกับส่วนจัดคอร์ดเพลง

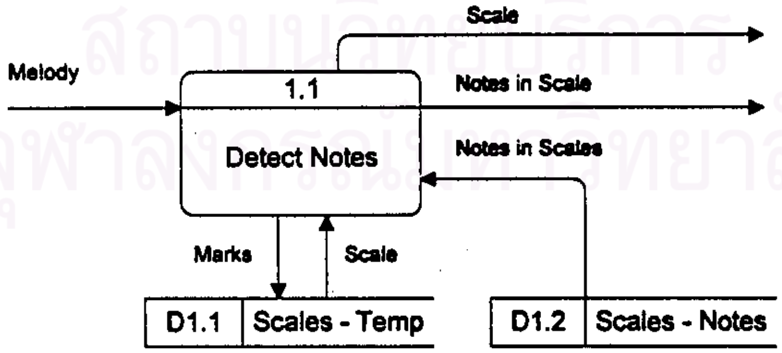
โครงสร้างข้อมูลของ *"แหล่งเก็บข้อมูลบันไดเสียงของเพลง"* จะแสดงได้ดังรูปภาพที่ 5.4 โดย S แทนบันไดเสียง Ns_1 ถึง Ns_7 แทนโน้ตในบันไดเสียง 7 ตัว ซึ่งโน้ตแต่ละตัวจะมีระดับเสียงเป็นองค์ประกอบ ส่วน Cs_1 ถึง Cs_7 แทนคอร์ดเพลงพื้นฐานในบันไดเสียงหรือตรัยเอ็ด 7 คอร์ด ซึ่งในแต่ละตรัยเอ็ดจะประกอบด้วย ชื่อตรัยเอ็ด ('Triad Name'$_a$) และโน้ตในตรัยเอ็ด 3 ตัว คือ Nc_{a1} ถึง Nc_{a3} เมื่อ a แทนหมายเลขตรัยเอ็ด โดยโน้ตในตรัยเอ็ดจะมีระดับเสียงเป็นองค์ประกอบ

โน้ตในบันไดเสียงทั้ง 7 ตัวคือ Ns_1 ถึง Ns_7 จะเป็นที่มาของโน้ตในทำนองเพลง และตรัยเอ็ดทั้ง 7 คอร์ดคือ Cs_1 ถึง Cs_7 จะเป็นที่มาของคอร์ดเพลงที่ประสานเสียงกับทำนองเพลง



รูปภาพที่ 5.4 โครงสร้างข้อมูลของแหล่งเก็บข้อมูลบันไดเสียงของเพลง

5.3 ส่วนตรวจสอบบันไดเสียง ("Detect Scale" Process)



รูปภาพที่ 5.5 แผนภาพกระแสข้อมูลระดับที่ 2 ของส่วนตรวจสอบบันไดเสียง

(DFD Level 2 – "Detect Scale" Process)

จากรูปภาพที่ 5.5 *"ส่วนตรวจสอบโน้ตในทำนองเพลง" ("Detect Notes" Process)* จะใช้ *"ข้อมูลโน้ตในบันไดเสียงต่างๆ" (Notes in Scales)* ที่เก็บอยู่ใน *"แหล่งเก็บข้อมูลโน้ตในบันไดเสียง" ("Scales – Notes" Data Store)* ดังแสดงในตารางที่ 5.1 มาประมวลผลตรวจสอบกับโน้ตใน *"ทำนองเพลง" (Melody)* เพื่อหา *"บันไดเสียง" (Scale)* ของทำนองเพลง เมื่อตรวจสอบบันไดเสียงได้ จะส่งข้อมูลพร้อมกับข้อมูล *"โน้ตในบันไดเสียง" (Notes in Scale)* ไปเก็บไว้ใน *"แหล่งเก็บข้อมูลบันไดเสียงของเพลง" ("Scale – Song" Data Store)* เพื่อเตรียมให้บริการข้อมูลกับส่วนการทำงานอื่นๆในขั้นถัดไป

ทั้งนี้ เมื่อโปรแกรมทำงาน *"ส่วนตรวจสอบโน้ตในทำนองเพลง"* จะมีการทดข้อมูล *"คะแนน" (Marks)* ของบันไดเสียงต่างๆไว้ใน *"แหล่งเก็บข้อมูลตารางทดคะแนนบันไดเสียง" ("Scales – Temp" Data Store)* เพื่อหา *"บันไดเสียง"* ที่ได้รับคะแนนมากที่สุดมาใช้เป็นบันไดเสียงของทำนองเพลง

ลำดับที่	บันไดเสียง		โน้ตในบันไดเสียง เรียงจาก C ถึง B	โน้ตที่ติดชาร์ป (#) หรือแฟล็ต (b) ในบันไดเสียง
	บันไดเสียงเมเจอร์	บันไดเสียงไมเนอร์		
1	C Major (C)	A minor (Am)	C D E F G A B	-
2	G Major (G)	E minor (Em)	C D E $F^\#$ G A B	$F^\#$
3	D Major (D)	B minor (Bm)	$C^\#$ D E $F^\#$ G A B	$F^\#$ $C^\#$
4	A Major (A)	$F^\#$ minor ($F^\#$m)	$C^\#$ D E $F^\#$ $G^\#$ A B	$F^\#$ $C^\#$ $G^\#$
5	E Major (E)	$C^\#$ minor ($C^\#$m)	$C^\#$ $D^\#$ E $F^\#$ $G^\#$ A B	$F^\#$ $C^\#$ $G^\#$ $D^\#$
6	B Major (B)	$G^\#$ minor ($G^\#$m)	$C^\#$ $D^\#$ E $F^\#$ $G^\#$ $A^\#$ B	$F^\#$ $C^\#$ $G^\#$ $D^\#$ $A^\#$
7	$F^\#$ Major ($F^\#$)	$D^\#$ minor ($D^\#$m)	$C^\#$ $D^\#$ $E^\#$ $F^\#$ $G^\#$ $A^\#$ B	$F^\#$ $C^\#$ $G^\#$ $D^\#$ $A^\#$ $E^\#$
8	$C^\#$ Major ($C^\#$)	$A^\#$ minor ($A^\#$m)	$C^\#$ $D^\#$ $E^\#$ $F^\#$ $G^\#$ $A^\#$ $B^\#$	$F^\#$ $C^\#$ $G^\#$ $D^\#$ $A^\#$ $E^\#$ $B^\#$
9	F Major (F)	D minor (Dm)	C D E F G A B^b	B^b
10	B^b Major (B^b)	G minor (Gm)	C D E^b F G A B^b	B^b E^b
11	E^b Major (E^b)	C minor (Cm)	C D E^b F G A^b B^b	B^b E^b A^b
12	A^b Major (A^b)	F minor (Fm)	C D^b E^b F G A^b B^b	B^b E^b A^b D^b
13	D^b Major (D^b)	B^b minor (B^bm)	C D^b E^b F G^b A^b B^b	B^b E^b A^b D^b G^b
14	G^b Major (G^b)	E^b minor (E^bm)	C^b D^b E^b F G^b A^b B^b	B^b E^b A^b D^b G^b C^b
15	C^b Major (C^b)	A^b minor (A^bm)	C^b D^b E^b F^b G^b A^b B^b	B^b E^b A^b D^b G^b C^b F^b

ตารางที่ 5.1 แหล่งเก็บข้อมูลโน้ตในบันไดเสียง ("D1.2 : Scales – Notes" Data Store)

หมายเหตุ : ตารางนี้เป็นตารางเดียวกันกับตารางที่ 3.4 ในบทที่ 3

"ส่วนตรวจสอบโน้ตในทำนองเพลง" จะแบ่งการทำงานเป็น 3 ขั้น คือ

1. เก็บความยาวทั้งหมดของโน้ตในทำนองเพลง แยกตามระดับเสียงต่างๆ

2. เปรียบเทียบระดับเสียงของโน้ตในทำนองเพลงทุกตัว กับโน้ตในแต่ละบันไดเสียงจากตารางที่ 5.1 และหากพบว่าตรงกัน จะให้คะแนนกับบันไดเสียงนั้นตามความยาวของโน้ตที่เก็บไว้ในข้อ 1 ทั้งนี้เพื่อค้นหาว่าโน้ตในทำนองเพลงที่ผู้ใช้ป้อนเข้ามาสามารถเป็นบันไดเสียงใดได้บ้าง ซึ่งหากบันไดเสียงใดได้รับคะแนนมากที่สุด โปรแกรมจะเลือกใช้บันไดเสียงนั้นเป็นบันไดเสียงของทำนองเพลง

3. ใช้โน้ตตัวสุดท้ายของเพลงเป็นตัวบ่งชี้ในการเลือกใช้บันไดเสียงระบบเมเจอร์หรือระบบไมเนอร์ โดยโน้ตตัวสุดท้ายของเพลงควรจะตรงกับโน้ตตัวแรกของบันไดเสียง

ยกตัวอย่างเช่น ประโยคหนึ่งของทำนองเพลงจิงเกิลเบลส์ (Jingle Bell) ประกอบด้วยโน้ตเพลงดังตารางที่ 5.2 ซึ่งในแต่ละคอลัมภ์คือโน้ตเพลงแต่ละตัว โดยตัวเลขในแถวแรกคือ ลำดับที่ของห้องเพลง ตัวอักษรในแถวที่สองคือ ระดับเสียง ตัวเลขในแถวที่สามคือขั้นคู่แปด ตัวเลขในแถวที่สี่คือความยาวโน้ต

1			2			3				4	5			6			7				8
E	E	E	E	E	E	E	G	C	D	E	F	F	F	F	E	E	G	F	E	D	C
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	8	4	4	8	4	4	4	4	16	4	4	8	4	4	8	4	4	4	4	16

ตารางที่ 5.2 ประโยคหนึ่งของทำนองเพลงจิงเกิลเบลส์ (Jingle Bell)

การทำงานของโปรแกรมจะมีขั้นตอนดังนี้

1. เก็บความยาวทั้งหมดของโน้ตในทำนองเพลง แยกตามระดับเสียงต่างๆ ซึ่งในทำนองเพลงนี้ประกอบด้วยโน้ต 5 ระดับเสียงคือ C D E F G และแต่ละระดับเสียงมีความยาวรวมทั้งประโยคเพลงคือ 20, 8, 68, 24 และ 8 เช่น ในทั้งประโยคเพลง มีโน้ต C 2 ตัวความยาว 4 และ 16 ตามลำดับจึงรวมความยาวโน้ตเพลงทั้งหมดเป็น 20 เป็นต้น

2. เปรียบเทียบระดับเสียงของโน้ตในทำนองเพลงทุกตัว กับโน้ตในแต่ละบันไดเสียงจากตารางที่ 5.1 และหากพบว่าตรงกัน จะให้คะแนนกับบันไดเสียงนั้นตามความยาวของโน้ตที่เก็บไว้ในข้อ 1 เช่น บันไดเสียงซีเมเจอร์ (C Major) มีระดับเสียงเป็น C D E F G A B ซึ่งตรงกับระดับเสียงของโน้ตในทำนองเพลงทั้งหมดคือ C D E F G โปรแกรมจะให้คะแนนรวมเป็น 20+8+68+24+8 = 128 คะแนน ในบันไดเสียงจีเมเจอร์ (G Major) ซึ่งมีระดับเสียงเป็น G A B C D E F# จะมีระดับเสียงที่ตรงกับทำนองเพลงเพียง 4 ระดับเสียงคือ C D E และ G โดยที่ขาดระดับเสียง F ไป โปรแกรมจะให้คะแนนรวมเป็น 20+8+68+8 = 104 คะแนน เป็นต้น ทั้งนี้ โปรแกรมจะทำการทดข้อมูลคะแนนของบันไดเสียงต่างๆลงใน *"แหล่งเก็บข้อมูลตารางทดคะแนนบันไดเสียง"* ซึ่งแสดงได้ดังตารางที่ 5.3 และเมื่อรวมคะแนนแล้ว โปรแกรมจะดึงข้อมูลบันไดเสียงของเพลงที่ได้รับคะแนนมากที่สุดออกมา คือ บันไดเสียงซีเมเจอร์ (C Major) หรือบันไดเสียงเอไมเนอร์ (A minor) และบันไดเสียงเอฟเมเจอร์ (F Major) หรือบันไดเสียงดีไมเนอร์ (D minor)

3. ใช้โน้ตตัวสุดท้ายของเพลงเป็นตัวบ่งชี้ในการเลือกใช้บันไดเสียงระบบเมเจอร์หรือระบบไมเนอร์ โดยโน้ตตัวสุดท้ายของเพลงควรจะตรงกับโน้ตตัวแรกของบันไดเสียง ซึ่งเมื่อนำโน้ตตัวสุดท้ายของเพลงคือ โน้ต C มา

พิจารณาเปรียบเทียบกับโน้ตตัวแรกของบันไดเสียงทั้งหมดคือ C A F และ D จะพบว่าตรงกับโน้ต C ในบันไดเสียงซีเมเจอร์ ดังนั้นโปรแกรมจะเลือกใช้บันไดเสียงซีเมเจอร์สำหรับเพลงจิงเกิลเบลส์

หลังจากการทำงานในส่วนนี้เสร็จเรียบร้อย โปรแกรมจะส่งข้อมูลไปเก็บไว้ใน *"แหล่งเก็บข้อมูลบันไดเสียงของเพลง"* ซึ่งจะประกอบด้วยข้อมูลคือ *"บันไดเสียง"* และ *"โน้ตในบันไดเสียง"*

ลำดับที่	บันไดเสียงเมเจอร์	บันไดเสียงไมเนอร์	จำนวนระดับเสียงที่ตรงกับทำนองเพลง	คะแนนที่แต่ละบันไดเสียงได้รับ
1	C Major (C)	A minor (Am)	5	128
2	G Major (G)	E minor (Em)	4	104
3	D Major (D)	B minor (Bm)	3	84
4	A Major (A)	$F^{\#}$ minor ($F^{\#}$m)	2	78
5	E Major (E)	$C^{\#}$ minor ($C^{\#}$m)	1	68
6	B Major (B)	$G^{\#}$ minor ($G^{\#}$m)	1	68
7	$F^{\#}$ Major ($F^{\#}$)	$D^{\#}$ minor ($D^{\#}$m)	0	0
8	$C^{\#}$ Major ($C^{\#}$)	$A^{\#}$ minor ($A^{\#}$m)	0	0
9	F Major (F)	D minor (Dm)	5	128
10	B^{b} Major (B^{b})	G minor (Gm)	4	60
11	E^{b} Major (E^{b})	C minor (Cm)	4	60
12	A^{b} Major (A^{b})	F minor (Fm)	3	52
13	D^{b} Major (D^{b})	B^{b} minor (B^{b}m)	2	44
14	G^{b} Major (G^{b})	E^{b} minor (E^{b}m)	1	24
15	C^{b} Major (C^{b})	A^{b} minor (A^{b}m)	0	0

ตารางที่ 5.3 ตัวอย่างข้อมูลในแหล่งเก็บข้อมูลตารางทดคะแนนบันไดเสียง
("D1.1 : Scala – Temp" Data Store)

สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

การใช้วิธีคิดคะแนนในการเลือกบันไดเสียงดังที่กล่าวมา เป็นการป้องกันการเลือกบันไดเสียงผิดพลาด ในกรณีที่มีโน้ตนอกบันไดเสียงรวมอยู่ด้วย ซึ่งการคิดคะแนนเช่นนี้จะทำให้โน้ตนอกบันไดเสียงที่มีจำนวนเพียงเล็กน้อย ได้คะแนนไม่มากและมีความสำคัญต่ำกว่าโน้ตในบันไดเสียงซึ่งมีจำนวนมากและได้คะแนนสูงกว่า ดังนั้น เมื่อโปรแกรมนำคะแนนรวมไปใช้ในการเลือกบันไดเสียงของทำนองเพลงจะได้บันไดเสียงที่ถูกต้องออกมา ทั้งนี้ ในบทเพลงเดียวกัน ที่มีการใช้บันไดเสียงแตกต่างกัน จะไม่ทำให้การทำงานของโปรแกรมผิดพลาด เนื่องจากการทำงานของโปรแกรมจะไม่ขึ้นอยู่กับบทเพลง แต่จะขึ้นกับโน้ตในทำนองเพลงนั้น

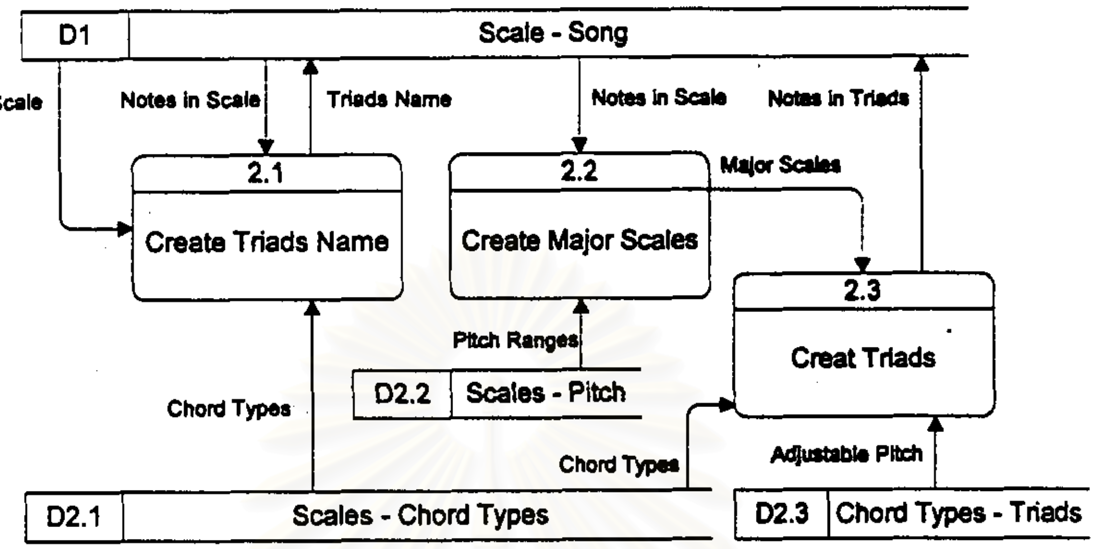
คำสั่งเทียมที่แสดงการทำงานของ *"ส่วนตรวจสอบบันไดเสียง"* แสดงได้ดังรูปภาพที่ 5.6

```
Detect Scale ()
{
        /* Get note length of all note in melody. */
        do {
                Read data in "Melody";
                Check note pitch;
                Increse all note length with current note length for current note pitch;
        } until end of "Melody";

        /* Check scale. */
        for  (n=1; n<=15; n++)                          {Maximum scales is 15.}
                if ("Notes in Scales" =note in "Melody")
                        Increase scales marks in "Song - Temp1" Data Store with note length;
                Select scale that there is maximum marks;

        /* Check last note. */
        for (n=1; n<= num of selected scale; n++)
                if ((first note in "Scale" no. n)=(last note in "Melody"))
                        Send "Scale" and "Notes in Scale" to "Scele-Song" Data Store;
}
```

รูปภาพที่ 5.6 คำสั่งเทียมแสดงการทำงานของส่วนตรวจสอบบันไดเสียง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

5.4 ส่วนสร้างตรัยเอ็ด ("Create Triads" Process)



รูปภาพที่ 5.7 แผนภาพกระแสข้อมูลระดับที่ 2 ของส่วนสร้างตรัยเอ็ด

(DFD Level 2 – "Create Triad" Process)

ตรัยเอ็ด คือ คอร์ดพื้นฐานในบันไดเสียง เป็นคอร์ดแบบ 3 ตัวโน้ต กำหนดให้ *"ตรัยเอ็ด" (Triads)* ประกอบด้วย *"ชื่อตรัยเอ็ด" (Triads Name)* และ *"โน้ตในตรัยเอ็ด" (Notes in Triads)* ซึ่งจากรูปภาพที่ 5.7 *"ส่วนสร้างตรัยเอ็ด" ("Create Triads" Process)* จะประกอบด้วยส่วนการทำงาน 3 ส่วน ในส่วนแรกจะสร้าง *"ชื่อตรัยเอ็ด"* ในส่วนที่สองและสามจะสร้าง *"โน้ตในตรัยเอ็ด"*

ส่วนแรก *"ส่วนสร้างชื่อตรัยเอ็ด" ("Create Triad Name" Process)* ทำหน้าที่สร้าง *"ชื่อตรัยเอ็ด"* 7 คอร์ดในบันไดเสียง โดยใช้ข้อมูล 3 อย่างคือข้อมูล *"บันไดเสียง" (Scale)* และ *"โน้ตในบันไดเสียง" (Notes in Scale)* จาก *"แหล่งเก็บข้อมูลบันไดเสียงของเพลง" ("Scale – Song" Data Store)* และข้อมูล *"ระบบคอร์ดเพลง" (Chord Types)* ต่างๆจาก *"แหล่งเก็บข้อมูลบันไดเสียงและระบบคอร์ด" ("Scales – Chord Types" Data Store)* เป็นข้อกำหนดว่าตรัยเอ็ดลำดับต่างๆต้องสร้างเป็นระบบคอร์ดแบบใด

ส่วนที่สอง *"ส่วนสร้างบันไดเสียงเมเจอร์" ("Create Major Scales" Process)* ทำหน้าที่สร้าง *"บันไดเสียงระบบเมเจอร์ของโน้ตเพลงทั้ง 7 ตัว" (Major Scales Data)* เพื่อเตรียมสร้างเป็น *"โน้ตในตรัยเอ็ด"* ในส่วนถัดไป การสร้างบันไดเสียงเมเจอร์จะใช้ข้อมูล 2 อย่างคือข้อมูล *"โน้ตในบันไดเสียง"* และข้อมูล *"ระยะห่างของเสียง" (Pitch Ranges)* จาก *"แหล่งเก็บข้อมูลบันไดเสียงและระยะห่างของเสียง" ("Scales – Pitch" Data Store)*

ส่วนที่สาม *"ส่วนสร้างโน้ตในตรัยเอ็ด" ("Create Triad" Process)* ทำหน้าที่นำโน้ตลำดับที่ 1, 3 และ 5 ใน *"บันไดเสียงระบบเมเจอร์ของโน้ตเพลงทั้ง 7 ตัว"* ที่สร้างขึ้นใหม่จากส่วนที่สอง มาปรับเปลี่ยนระดับเสียงตามข้อมูล *"ค่าปรับแต่งระยะเสียง" (Adjustable Pitch)* จาก *"แหล่งเก็บข้อมูลคอร์ดเพลงและค่าปรับแต่งเสียง" ("Chord Types – Triads" Data Store)* และข้อมูล *"ระบบคอร์ดเพลง"* เพื่อสร้างเป็น *"โน้ตในตรัยเอ็ด" (Notes in Triads)* แต่ละตัว

"แหล่งเก็บข้อมูลบันไดเสียงและระบบคอร์ด" แสดงได้ดังตารางที่ 5.4 จะเป็นที่เก็บข้อมูลระบบคอร์ดเพลง 7 คอร์ดในบันไดเสียงเมเจอร์และบันไดเสียงไมเนอร์ *"แหล่งเก็บข้อมูลบันไดเสียงและระดับเสียง"* แสดงได้ดังตารางที่ 5.5 จะเก็บข้อมูลระยะห่างของระดับเสียงของโน้ต 7 ตัวในบันไดเสียงเมเจอร์และบันไดเสียงไมเนอร์ และ *"แหล่งเก็บข้อมูลคอร์ดเพลงและค่าปรับแต่งเสียง"* แสดงได้ดังตารางที่ 5.6 จะเก็บข้อมูลที่กำหนดว่าในคอร์ดแต่ละระบบซึ่งประกอบด้วยโน้ตลำดับที่ 1, 3 และ 5 ของโน้ตในบันไดเสียงเมเจอร์ที่สร้างขึ้นใหม่จะต้องมีการปรับเปลี่ยนระดับเสียงอย่างไร

ระบบบันไดเสียง	ระบบคอร์ดสำหรับโน้ตทั้ง 7 ตัวในบันไดเสียง						
	คอร์ดที่ 1	คอร์ดที่ 2	คอร์ดที่ 3	คอร์ดที่ 4	คอร์ดที่ 5	คอร์ดที่ 6	คอร์ดที่ 7
บันไดเสียงเมเจอร์	เมเจอร์	ไมเนอร์	ไมเนอร์	เมเจอร์	เมเจอร์	ไมเนอร์	ดิมินิชด์
บันไดเสียงไมเนอร์	ไมเนอร์	ดิมินิชด์	เมเจอร์	ไมเนอร์	ไมเนอร์	เมเจอร์	เมเจอร์

ตารางที่ 5.4 แหล่งเก็บข้อมูลบันไดเสียงและระบบคอร์ด ("D2.1 : Scales - Chord Types" Data Store)

หมายเหตุ : ตารางนี้เป็นตารางเดียวกันกับตารางที่ 3.6 ในบทที่ 3

ระยะห่างระหว่างเสียง	ระบบบันไดเสียงเมเจอร์	ระบบบันไดเสียงไมเนอร์
โน้ตที่ 1 และโน้ตที่ 2	เต็มเสียง	เต็มเสียง
โน้ตที่ 2 และโน้ตที่ 3	เต็มเสียง	ครึ่งเสียง
โน้ตที่ 3 และโน้ตที่ 4	ครึ่งเสียง	เต็มเสียง
โน้ตที่ 4 และโน้ตที่ 5	เต็มเสียง	เต็มเสียง
โน้ตที่ 5 และโน้ตที่ 6	เต็มเสียง	ครึ่งเสียง
โน้ตที่ 6 และโน้ตที่ 7	เต็มเสียง	เต็มเสียง
โน้ตที่ 7 และโน้ตที่ 1 ในอีกขั้นคู่แปดถัดไป	ครึ่งเสียง	เต็มเสียง

ตารางที่ 5.5 แหล่งเก็บข้อมูลบันไดเสียงและระยะห่างของเสียง ("D2.2 : Scales – Pitch" Data Store)

หมายเหตุ : ตารางนี้เป็นตารางเดียวกันกับตารางที่ 3.3 ในบทที่ 3

ระบบคอร์ด	โน้ตในคอร์ด		
	โน้ตในคอร์ดตัวที่ 1	โน้ตในคอร์ดตัวที่ 2	โน้ตในคอร์ดตัวที่ 3
คอร์ดเมเจอร์ (Major)	โน้ตลำดับที่ 1	โน้ตลำดับที่ 3	โน้ตลำดับที่ 5
คอร์ดไมเนอร์ (Minor)	โน้ตลำดับที่ 1	โน้ตลำดับที่ 3 ลดลงครึ่งเสียง	โน้ตลำดับที่ 5
คอร์ดดิมินิชด์ (Diminish)	โน้ตลำดับที่ 1	โน้ตลำดับที่ 3 ลดลงครึ่งเสียง	โน้ตลำดับที่ 5 ลดลงครึ่งเสียง
คอร์ดอ็อกเม็นเตท (Augmented)	โน้ตลำดับที่ 1 เพิ่มครึ่งเสียง	โน้ตลำดับที่ 3	โน้ตลำดับที่ 5

ตารางที่ 5.6 แหล่งเก็บข้อมูลคอร์ดเพลงและค่าปรับแต่งเสียง ("D2.3 : Chord Types – Triads" Data Store)

หมายเหตุ : ตารางนี้เป็นตารางเดียวกันกับตารางที่ 3.5 ในบทที่ 3

5.4.1 ส่วนสร้างชื่อตรัยเอ็ด ("Create Triads Name" Process)

การเรียกชื่อตรัยเอ็ดจะแบ่งเป็น 2 ส่วน โดยเรียกชื่อระดับเสียงก่อนและตามด้วยชื่อระบบคอร์ด เช่น คอร์ด G อ่านว่า คอร์ด จี - เมเจอร์ เป็นต้น ดังนั้น เมื่อจะสร้างชื่อคอร์ด โปรแกรมจะใช้ข้อมูล *"ระบบคอร์ด"* ใน *"แหล่งเก็บข้อมูลบันไดเสียงและระบบคอร์ด"* ตารางที่ 5.4 พิจารณากับข้อมูล *"บันไดเสียง"* ใน *"แหล่งเก็บข้อมูลบันไดเสียงของเพลง"* เพื่อกำหนดว่าว่าจะใช้คอร์ดระบบใดบ้างกับโน้ตทั้ง 7 ตัวในบันไดเสียง ซึ่งหากเป็นบันไดเสียงระบบเมเจอร์ โน้ตตัวที่ 1 ถึง 7 ในบันไดเสียงจะถูกสร้างคอร์ดในระบบ เมเจอร์, ไมเนอร์, ไมเนอร์, เมเจอร์, เมเจอร์, ไมเนอร์ และดิมินิชด์ ตามลำดับ แต่หากเป็นบันไดเสียงระบบไมเนอร์ โน้ตตัวที่ 1 ถึง 7 ในบันไดเสียงจะถูกสร้างคอร์ดในระบบ ไมเนอร์ , ดิมินิชด์, เมเจอร์, ไมเนอร์, ไมเนอร์, เมเจอร์ และเมเจอร์ ตามลำดับ ยกตัวอย่างเช่น

ในบันไดเสียงจีเมเจอร์ (G Major) เป็นบันไดเสียงระบบเมเจอร์ประกอบด้วยโน้ตที่มีระดับเสียงเป็น G A B C D E F# เมื่อนำมาสร้างคอร์ดในบันไดเสียงจะได้เป็น G Am Bm C D Em และ F#dim ในบันไดเสียง A minor เป็นบันไดเสียงระบบไมเนอร์ ประกอบด้วยโน้ตที่มีระดับเสียงเป็น A B C D E F G เมื่อนำมาสร้างคอร์ดในบันไดเสียงจะได้เป็น Am Bdim C Dm Em F และ G เป็นต้น

ทั้งนี้เมื่อโปรแกรมประมวลผลเสร็จ โปรแกรมจะนำ *"ชื่อตรัยเอ็ด"* ไปเก็บไว้ใน *"แหล่งเก็บข้อมูลบันไดเสียงของเพลง"*

คำสั่งเทียมที่แสดงการทำงานของ *"ส่วนสร้างชื่อตรัยเอ็ด"* แสดงได้ดัง

```
Create Triad Name ()
{
    for (i=1;i<= 7;i++) {          /* i is "Notes in Scale" sequence.*/
        Lookup "Chord Types" in "Scale – Chord Type" data storage with "Scale" and i ;
        Select ("Chord Types") {
            Major      :  "Triads Name" = "Notes in Scale";            /* Ex. C */
            minor      :  "Triads Name" = "Notes in Scale" & 'm';      /* Ex. Am */
            Diminish   :  "Triads Name" = "Notes in Scale" & 'dim';    /* Ex. F#dim */
        }
        Put "Triads Name" to "Scale – Song" data storage;
    }
}
```

รูปภาพที่ 5.8 คำสั่งเทียมแสดงการทำงานของส่วนสร้างชื่อตรัยเอ็ด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

5.4.2 สร้างบันไดเสียงเมเจอร์ ("Create Major Scales" Process)

ในส่วนนี้จะสร้างบันไดเสียงเมเจอร์ 7 บันไดเสียงของโน้ต 7 ตัวในบันไดเสียงของเพลง เพื่อเป็นข้อมูลให้กับส่วนถัดไปในการสร้าง *"โน้ตในตรัยเอ็ด"*

การสร้างบันไดเสียงเมเจอร์จะสร้างจากข้อมูล *"ระยะห่างของเสียง"* ใน *"แหล่งเก็บข้อมูลบันไดเสียงและระยะห่างของเสียง"* ตารางที่ 5.5 โดยเริ่มสร้างบันไดเสียงเมเจอร์ครั้งละ 1 บันไดเสียงเรียงไปจนครบ 7 บันไดเสียง เช่น ในบันไดเสียงจีเมเจอร์ (G Major) ประกอบด้วยโน้ตที่มีระดับเสียงเป็น G A B C D E F# เมื่อนำโน้ตแต่ละตัวมา สร้างเป็นบันไดเสียงเมเจอร์จะได้ดังตารางที่ 5.7 ซึ่งเป็นข้อมูลขาออกของการทำงานส่วนนี้

บันไดเสียง	โน้ตที่ 1	โน้ตที่ 2	โน้ตที่ 3	โน้ตที่ 4	โน้ตที่ 5	โน้ตที่ 6	โน้ตที่ 7
G Major	G	A	B	C	D	E	F#
G Major	G	A	B	C	D	E	F#
A Major	A	B	C#	D	E	F#	G#
B Major	B	C#	D#	E	F#	G#	A#
C Major	C	D	E	F	G	A	B
D Major	D	E	F#	G	A	B	C#
E Major	E	F#	G#	A	B	C#	D#
F# Major	F#	G#	A#	B	C#	E#	F#

ตารางที่ 5.7 ตัวอย่างบันไดเสียงเมเจอร์ของโน้ตทั้ง 7 ในบันไดเสียงจีเมเจอร์ (G Major)

คำสั่งเทียมที่แสดงการทำงานของ *"ส่วนสร้างบันไดเสียงเมเจอร์"* แสดงได้ดังรูปภาพที่ 5.9

```
Create Major Scales()
{
        for (i=1;i<=7;i++) {                    /* i is "Notes in Scale" Sequence. */
            for (j=1;j<=7;j++) {                /* j is note in "Major scales" sequence.*/
                Lookup "Pitch Ranges" in "Scales – Pitch" data storage;
                Create note[j] in major scales[i] from "Pitch Ranges";
            }
}
```

รูปภาพที่ 5.9 คำสั่งเทียมแสดงการทำงานของส่วนสร้างบันไดเสียงเมเจอร์

5.4.3 ส่วนสร้างโน้ตในตรัยเอ็ด (Create Triad Process)

การทำงานส่วนนี้จะสร้าง *"โน้ตในตรัยเอ็ด"* โดยใช้ข้อมูลโน้ตในบันไดเสียงเมเจอร์ที่สร้างขึ้นในส่วนที่แล้ว โดยโปรแกรมจะสร้างคอร์ดเพลงครั้งละ 1 คอร์ดด้วยการนำโน้ตในลำดับที่ 1, 3 และ 5 ของบันไดเสียงเมเจอร์มาปรับเปลี่ยนระดับเสียงตามระบบคอร์ดโดยใช้ข้อมูล *"ระบบคอร์ดเพลง"* จาก *"แหล่งเก็บข้อมูลบันไดเสียงและระบบคอร์ด"* ตารางที่ 5.4 และข้อมูล *"ค่าปรับแต่งเสียง"* จาก *"แหล่งเก็บข้อมูลคอร์ดเพลงและค่าปรับแต่งเสียง"* ตารางที่ 5.6 เพื่อสร้างเป็นคอร์ดเพลง ยกตัวอย่างช่น

ในบันไดเสียงจีเมเจอร์ โน้ตลำดับที่ 1 คือโน้ต G เมื่อนำมาตรวจสอบกับตารางที่ 5.4 จะพบว่า ในระบบบันไดเสียงเมเจอร์ โน้ตลำดับที่ 1 จะสร้างคอร์ดเป็นคอร์ดระบบเมเจอร์ จากนั้น โปรแกรมจะตรวจสอบกับตารางที่ 5.6 เพื่อหาข้อมูลว่าคอร์ดระบบเมเจอร์ต้องมีการปรับเปลี่ยนระดับเสียงอย่างไร ซึ่งจะพบว่าไม่ต้องปรับเปลี่ยนระดับเสียงใดๆ ดังนั้น จากตารางที่ 5.7 คอร์ด G คือโน้ตลำดับที่ 1, 3 และ 5 ของบันไดเสียงจีเมเจอร์ซึ่งไม่ต้องปรับแต่งระดับเสียง อันได้แก่โน้ต G B และ D

ในบันไดเสียงจีเมเจอร์ โน้ตลำดับที่ 2 คือโน้ต A เมื่อนำมาตรวจสอบกับตารางที่ 5.4 จะพบว่า ในระบบบันไดเสียงเมเจอร์ โน้ตลำดับที่ 2 จะสร้างคอร์ดเป็นคอร์ดระบบไมเนอร์ จากนั้น โปรแกรมจะตรวจสอบกับตารางที่ 5.6 เพื่อหาข้อมูลว่าคอร์ดระบบไมเนอร์ต้องมีการปรับเปลี่ยนระดับเสียงอย่างไร ซึ่งจะพบว่าต้องปรับเปลี่ยนระดับเสียงของโน้ตลำดับที่ 3 ลงครึ่งเสียง ดังนั้น จากตารางที่ 5.7 คอร์ด Am คือโน้ตลำดับที่ 1, 3 ลดลงครึ่งเสียง และ 5 ของบันไดเสียงเอเมเจอร์ อันได้แก่โน้ต A C และ E

ทั้งนี้เมื่อโปรแกรมประมวลผลเสร็จ โปรแกรมจะนำ *"โน้ตในตรัยเอ็ด"* ไปเก็บไว้ใน *"แหล่งเก็บข้อมูลบันไดเสียงของเพลง"* ซึ่งในตารางที่ 5.8 จะแสดงตัวอย่างข้อมูลโน้ตในตรัยเอ็ดของบันไดเสียงจีเมเจอร์

ชื่อตรัยเอ็ด	โน้ตในตรัยเอ็ดลำดับที่ 1 (โน้ตในบันไดเสียงลำดับที่ 1)	โน้ตในตรัยเอ็ดลำดับที่ 2 (โน้ตในบันไดเสียงลำดับที่ 3)	โน้ตในตรัยเอ็ดลำดับที่ 3 (โน้ตในบันไดเสียงลำดับที่ 5)
G	G	B	D
Am	A	C	E
Bm	B	D	F#
C	C	E	G
D	D	F#	A
Em	E	G	B
F#dim	F#	A	C

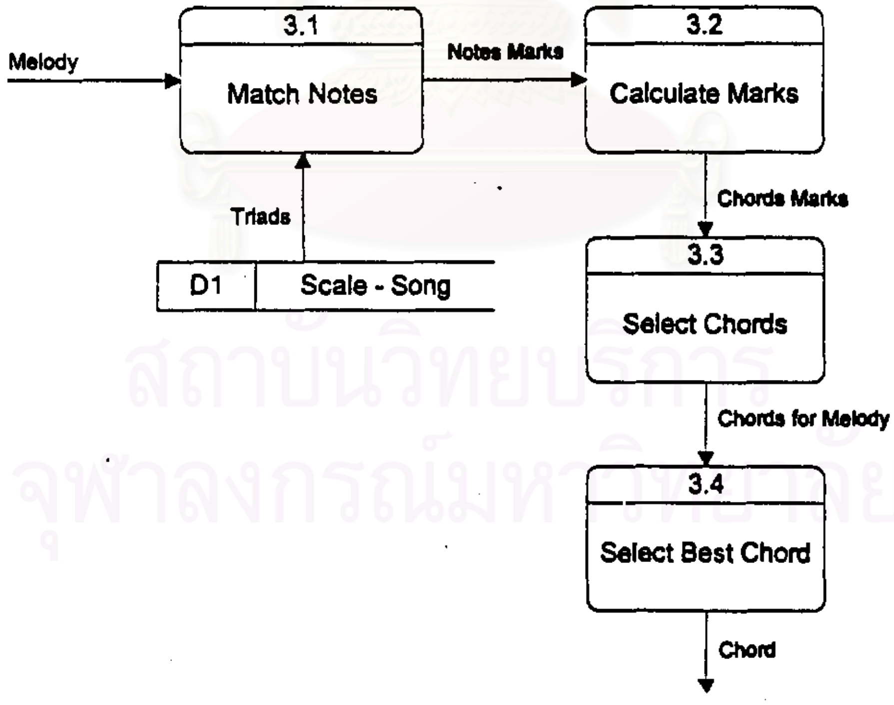
ตารางที่ 5.8 ตัวอย่างข้อมูลโน้ตในตรัยเอ็ดของบันไดเสียงจีเมเจอร์

คำสั่งเทียมที่แสดงการทำงานของ *"ส่วนสร้างโน้ตในตรัยเอ็ด"* แสดงได้ดังรูปภาพที่ 5.10

```
Create Triad ()
{
        for (i=1;i<=7;i++) {                    /* i is scale sequence in "Major Scales".*/
                Create "Notes in Triads" from note no. 1, 3 and 5 in "Major Scales";
                Lookup "Chord Types" in "Scales – Chord Types" data storage with i;
                Lookup "Adjustable Pitch" in "Chord Types – Triads" data storage with
                        "Chord Types";
                Adjust "Notes in Triads" pitch with "Adjustable Pitch";
                Put "Notes in Triads" to "Scale – Song" data storage;
        }
}
```

รูปภาพที่ 5.10 คำสั่งเทียมแสดงการทำงานของส่วนสร้างโน้ตในตรัยเอ็ด

5.5 ส่วนจัดคอร์ดเพลง (*"Arrange Chords" Process*)



รูปภาพที่ 5.11 แผนภาพกระแสข้อมูลระดับที่ 2 ของส่วนจัดคอร์ดเพลง

(DFD Level 2 – "Arrange Chords" Process)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

"ส่วนจัดคอร์ดเพลง" ("Arrange Chord" Process) จะประกอบด้วยส่วนการทำงาน 4 ส่วน คือ

ส่วนแรก *"ส่วนจับคู่โน้ต" ("Match Notes" Process)* ทำหน้าที่จับคู่ *"โน้ตในตรัยเอ็ด" (Notes in Triads)* แต่ละคอร์ด กับโน้ตใน *"ทำนองเพลง" (Melody)* แต่ละตัว และเป็นคิดคะแนนตามความยาวของโน้ตในทำนองเพลง หากโน้ตทั้งคู่มีระดับเสียงตรงกัน

ส่วนที่สอง *"ส่วนคำนวณคะแนน" ("Calculate Merks" Process)* ทำหน้าที่รวมคะแนนของคอร์ดเพลงแต่ละคอร์ดในทุกๆห้องเพลง

ส่วนที่สาม *"ส่วนเลือกคอร์ดเพลงครั้งที่ 1" ("Select Chords" Process)* ทำหน้าที่คัดเลือกคอร์ดเพลงที่ได้คะแนนมากที่สุดออกมา ซึ่งในแต่ละห้องเพลง อาจมีคอร์ดเพลงได้มากกว่า 1 คอร์ดที่ได้รับการคัดเลือก

ส่วนที่สี่ *"ส่วนเลือกคอร์ดเพลงครั้งที่ 2" ("Select Best Chord" Process)* ทำหน้าที่คัดเลือกคอร์ดเพลงที่ดีที่สุดเพียงคอร์ดเดียว จากคอร์ดที่ผ่านจากการคัดเลือกในขั้นที่ 1

5.5.1 ส่วนจับคู่โน้ต ("Match Notes" Process)

ในส่วนนี้ โปรแกรมจะรับข้อมูลขาเข้าเป็นข้อมูล *"ทำนองเพลง" (Melody)* และข้อมูล *"ตรัยเอ็ด" (Triads)* อันประกอบด้วย *"ชื่อตรัยเอ็ด" (Triads Name)* และ *"โน้ตในตรัยเอ็ด" (Notes in Triads)* ซึ่งโปรแกรมจะประมวลผลครั้งละ 1 โน้ตเพลงจนครบโน้ตในทำนองเพลงทุกตัว โดยจะไล่จับคู่โน้ตในทำนองเพลงกับโน้ตในตรัยเอ็ดทุกคอร์ด หากตรงกันโปรแกรมจะนำค่าความยาวของโน้ตในทำนองเพลงบันทึกเป็นคะแนนเก็บไว้ ข้อมูลขาออกของส่วนการทำงานนี้คือ *"ตารางคะแนนโน้ตเพลง" (Notes Merks)* ซึ่งมีรูปแบบดังตารางที่ 5.9 โดยกำหนดให้ทำนองเพลงมี 3 ห้องเพลง อยู่ในบันไดเสียงจีเมเจอร์ (G Major) และมีค่าดังตาราง

ห้องเพลงที่	1			2				3			
โน้ต	G	C	C	G	C	F#	F#	G	C	B	F#
ความยาว	8	4	4	4	4	4	4	4	4	4	4
คอร์ด G	8	0	0	4	0	0	0	4	0	4	0
คอร์ด Am	0	4	4	0	4	0	0	0	4	0	0
คอร์ด Bm	0	0	0	0	0	4	4	0	0	4	4
คอร์ด C	8	4	4	4	4	0	0	4	4	0	0
คอร์ด D	0	0	0	0	0	4	4	0	0	0	4
คอร์ด Em	8	0	0	4	0	0	0	4	0	4	0
คอร์ด F#dim	0	4	4	0	4	4	4	0	4	0	4

ตารางที่ 5.9 ตัวอย่างตารางคะแนนโน้ตเพลง

จากตารางที่ 5.9 โปรแกรมจะเริ่มทำงานโดยพิจารณาโน้ต G ซึ่งเป็นโน้ตตัวแรกในทำนองเพลง กับตรัยเอ็ดในตารางที่ 5.8 พบว่าโน้ต G เป็นโน้ตตัวหนึ่งในตรัยเอ็ดต่อไปนี้ คอร์ด G C และ Em ดังนั้นโปรแกรมจะบันทึกค่าความยาวโน้ต 8 ลงในช่องเก็บข้อมูลของคอร์ด G C และ Em และบันทึกค่า 0 ลงในตรัยเอ็ดอื่นที่ไม่ปรากฏโน้ต G อยู่ ถัดมาในโน้ตตัวที่สองคือโน้ต C จะพบว่าโน้ต C เป็นโน้ตตัวหนึ่งในตรัยเอ็ดต่อไปนี้ Am C

และ $F^{\#}$dim ดังนั้นโปรแกรมจะบันทึกค่าความยาวโน้ต 4 ลงในช่องเก็บข้อมูลของคอร์ด Am C และ $F^{\#}$dim และบันทึกค่า 0 ลงในตรัยเอ็ดอื่นที่ไม่ปรากฏโน้ต C อยู่ ในโน้ตตัวที่สามเป็นโน้ต C เช่นกัน ความยาวเท่ากัน ดังนั้นโปรแกรมจะประมวลผลได้คะแนนเหมือนกันกับโน้ตตัวที่สอง

เมื่อโปรแกรมจะทำงานจนครบโน้ตในทำนองเพลงทุกตัวซึ่งจะได้ผลลัพธ์ดังตารางที่ 5.9

คำสั่งเทียมที่แสดงการทำงานของ *"ส่วนจับคู่โน้ต"* แสดงได้ดังรูปภาพที่ 5.12

```
Match Note()
{
    for (i=1; i<=end of "Melody"; i++)          /* i is note sequence in "Melody" */
        for (j=1; j<= 7; j++)                   /* j is triad sequence in "Triads" */
            for (k=1; k<=3; k++)                /* k is note sequence in "Notes in Triads" */
                If (note in "Melody" = "Notes in Triads") {
                    Put note length to "Notes Marks";
                    break;
                }
                else
                    Put value 0 to "Notes Marks";
}
```

รูปภาพที่ 5.12 คำสั่งเทียมแสดงการทำงานของส่วนจับคู่โน้ต

5.5.2 ส่วนคำนวณคะแนน ("Calculate Merks" Process)

ในส่วนนี้ โปรแกรมจะรับ *"ตารางคะแนนโน้ตเพลง" (Notes Marks)* เป็นข้อมูลขาเข้า เพื่อนำมาประมวลผล รวมคะแนนของโน้ตเพลงทุกตัวในแต่ละห้องเพลง เพื่อเป็นคะแนนสรุปสำหรับตรัยเอ็ดแต่ละคอร์ด และส่งข้อมูลขาออกเป็น *"ตารางคะแนนคอร์ดเพลง" (Chords Marks)* ซึ่งแสดงได้ดังตารางที่ 5.10 โดยข้อมูลภายในตารางเป็นตัวอย่างต่อเนื่องจากตารางที่ 5.9

ห้องเพลงที่	1	2	3
คอร์ด G	8	4	8
คอร์ด Am	8	4	4
คอร์ด Bm	0	8	8
คอร์ด C	16	8	8
คอร์ด D	0	8	4
คอร์ด Em	8	4	8
คอร์ด $F^{\#}$dim	8	12	8

ตารางที่ 5.10 ตารางคะแนนคอร์ดเพลง

ในห้องเพลงแรก ประกอบด้วยโน้ต G C และ C

คอร์ดแรก คอร์ด G จะได้คะแนนรวมเท่ากับ 8 จากโน้ต G

คอร์ดที่สอง คอร์ด Am จะได้คะแนนรวมเท่ากับ 8 จากโน้ต C 2 ตัวๆละ 4 คะแนน

คอร์ดที่สาม คอร์ด Bm ได้คะแนนรวมเท่ากับ 0 เนื่องจากไม่มีโน้ตในห้องเพลงนี้ตัวใดที่เหมือนกับโน้ตในคอร์ด Bm ซึ่งประกอบด้วยโน้ต B, D และ F#

ในคอร์ดต่อมาคือคอร์ดที่สี่ถึงเจ็ด รวมถึงในห้องเพลงถัดๆไป คือห้องเพลงที่ 2 และ 3 จะใช้ประมวลผลด้วยวิธีเดียวกันซึ่งได้ผลลัพธ์ดังแสดงในตารางที่ 5.10

คำสั่งเทียมที่แสดงการทำงานของ *"ส่วนคำนวณคะแนน"* แสดงได้ดังรูปภาพที่ 5.13

```
Calculate Marks ()
{
    for (i=1;i<= num of measure in "Melody"; i++) {            /* i is measure sequence.*/
        for (j=1; j<=7; j++) {                                  /* j is triad sequence.*/
            for (k=1; k<=max note in measure [i]; k++) /* k is note in measure sequence.*/
                Summary marks and put in "Chord Marks";
}
```

รูปภาพที่ 5.13 คำสั่งเทียมแสดงการทำงานของส่วนคำนวณคะแนน

5.5.3 ส่วนคัดเลือกคอร์ดเพลงครั้งที่ 1 ("Select Chords" Process)

ในส่วนนี้ โปรแกรมจะรับ *"ตารางคะแนนคอร์ดเพลง" (Chords Marks)* เป็นข้อมูลขาเข้า ซึ่งโปรแกรมจะนำมาประมวลผลหาตรัยเอ็ดที่มีคะแนนมากที่สุดในแต่ละห้องเพลง และคัดเลือกไว้เพื่อเป็นตรัยเอ็ดที่ใช้ประสานเสียงกับทำนองเพลง หากพบว่ามีตรัยเอ็ดที่มีคะแนนมากที่สุดมากกว่า 1 คอร์ด โปรแกรมจะคัดเลือกไว้ทุกคอร์ด และจะส่งเป็นข้อมูลขาออกเป็น *"ตารางคอร์ดเพลง" (Chords for Melody)* ซึ่งแสดงได้ดังรูปภาพที่ 5.13 โดยข้อมูลภายในตารางเป็นตัวอย่างต่อเนื่องจากรูปภาพที่ 5.12

ห้องเพลงที่	1	2	3
คอร์ดที่ 1 (คอร์ด G)			คอร์ด G
คอร์ดที่ 2 (คอร์ด Am)			
คอร์ดที่ 3 (คอร์ด Bm)			คอร์ด Bm
คอร์ดที่ 4 (คอร์ด C)	คอร์ด C		คอร์ด C
คอร์ดที่ 5 (คอร์ด D)			
คอร์ดที่ 6 (คอร์ด Em)			คอร์ด Em
คอร์ดที่ 7 (คอร์ด F#dim)		คอร์ด F#dim	คอร์ด F#dim

ตารางที่ 5.11 ตารางคอร์ดเพลง

คำสั่งเทียมที่แสดงการทำงานของ *"ส่วนคัดเลือกคอร์ดเพลงครั้งที่ 1"* แสดงได้ดังรูปภาพที่ 5.14 และในส่วนถัดไปจะเรียก *"ตรัยเอ็ด"* ว่า *"คอร์ดเพลง"*

```
Selected Chords ()
{
        for (i=1; i<= num of measure in "Melody"; i++) {          /* i is measure sequence. */
                max_marks = 0;
                for (j=1; j<=7; j++)                              /* j is triad sequence. */
                        If ("Chords Marks" > max_marks) {   /* Find maximum marks. */
                                max_marks = "Chord Marks";
                for (j=1; j<=7; j++)
                        If ("Chord Marks" = max_marks)       /* Select triads. */
                                Select this triad and put to "Chords for Melody";
        }
}
```

รูปภาพที่ 5.14 คำสั่งเทียมแสดงการทำงานของส่วนคัดเลือกคอร์ดเพลงครั้งที่ 1

5.5.4 ส่วนคัดเลือกคอร์ดเพลงครั้งที่ 2 ("Select Best Chord" Process)

ในส่วนนี้ โปรแกรมจะรับ *"ตารางคอร์ดเพลง" (Chords for Melody)* เป็นข้อมูลขาเข้า เพื่อนำมาประมวลผลในกรณีที่มีคอร์ดเพลงมากกว่า 1 คอร์ดได้รับคัดเลือก โดยโปรแกรมจะใช้ทฤษฎีดนตรีของคอร์ดหลัก (Primary Chord) และคอร์ดรอง (Secondary Chord) เป็นข้อตัดสินเป็นอันดับแรก โดยคอร์ดหลักคือคอร์ดลำดับที่ 1, 4 และ 5 ของบันไดเสียงทั้งระบบเมเจอร์และระบบไมเนอร์ ส่วนคอร์ดรองคือคอร์ดลำดับที่ 2, 3, 6 และ 7 ของบันไดเสียง โปรแกรมจะตัดสินใจเลือกใช้คอร์ดหลักก่อนคอร์ดรอง และหากมีคอร์ดหลักมากกว่า 1 คอร์ด โปรแกรมจะเลือกใช้คอร์ดหลักที่ไม่ซ้ำกับคอร์ดในห้องเพลงที่ผ่านมา และจะเลือกใช้คอร์ดเรียงตามลำดับก่อนหลังในบันไดเสียง ยกตัวอย่างเช่น

จากตารางที่ 5.11 ในห้องเพลงที่ 1 และ 2 มีคอร์ดที่ได้รับคะแนนสูงสุดเพียงคอร์ดเดียว คือ คอร์ด C และ F#dim ดังนั้นโปรแกรมจะเลือกใช้คอร์ด C และ F#dim สำหรับห้องเพลงที่ 1 และ 2 ตามลำดับ ส่วนห้องเพลงที่ 3 มีคอร์ดที่ได้รับคะแนนสูงสุดเท่ากัน 5 คอร์ดคือ คอร์ด G, Bm, C, Em และ F#dim เมื่อพิจารณาจะพบว่าคอร์ดหลักคือคอร์ด G และ C ซึ่งเป็นคอร์ดที่ 1 และ 4 ดังนั้น คอร์ดเพลงที่โปรแกรมจะคัดเลือก เหลือสองคอร์ดคือคอร์ด G และ C ถัดมาโปรแกรมจะตรวจสอบคอร์ดของห้องเพลงที่ผ่านมา คือห้องเพลงที่ 2 ซึ่งมีคอร์ดเป็นคอร์ด F#dim พบว่าไม่ซ้ำกับคอร์ด G หรือ C เลย โปรแกรมจึงสามารถเลือกใช้คอร์ดได้ทั้งคอร์ด G และ C แต่จะเลือกใช้คอร์ดเรียงตามลำดับก่อนหลัง ซึ่งคอร์ด G เป็นคอร์ดที่ 1 และคอร์ด C เป็นคอร์ดที่ 4 ดังนั้น โปรแกรมจะคัดเลือกคอร์ด G สำหรับห้องเพลงที่ 3

ข้อยกเว้นประการหนึ่ง ก็คือ ในห้องเพลงสุดท้ายของบทเพลง โปรแกรมจะเลือกใช้คอร์ดที่ 1 ของบันไดเสียงเสมอ เพื่อความไพเราะและการจบเพลงอย่างสมบูรณ์

ข้อมูลขาออกของการทำงานส่วนนี้คือ ข้อมูล *"คอร์ดเพลง" (Chord)* ที่โปรแกรมได้คำนวณว่าเหมาะสมกับทำนองเพลงมากที่สุดในแต่ละห้องเพลง ซึ่งแสดงได้ดังตารางที่ 5.12

ห้องเพลงที่	1	2	3
ชื่อตรัยเอ็ด (Triads Name)	คอร์ด C	คอร์ด F#dim	คอร์ด G
โน้ตเพลงในตรัยเอ็ด (Notes in Triads)	C E G	F# A C	G B D

ตารางที่ 5.12 ข้อมูลคอร์ดเพลงที่ได้รับคัดเลือก

คำสั่งเทียมที่แสดงการทำงานของ *"ส่วนคัดเลือกคอร์ดเพลงครั้งที่ 1"* แสดงได้ดังรูปภาพที่ 5.15

```
Select Best Chord ()
{
    for (i=1; i<=num of measure in "Melody"; i++)      /* i is measure sequence. */
        If (selected chords>1) {
            Select primary chord before secondary chord;
            If (selected chords>1) {
                Select primary chord that different from chord of previous measure;
                If (selected chord>1)
                    Select chord by sequence in scale;
            }
        }
    Select first chord in scele at last measure;
}
```

รูปภาพที่ 5.15 คำสั่งเทียมแสดงการทำงานของส่วนคัดเลือกคอร์ดเพลงครั้งที่ 2

5.6 สรุป

โปรแกรมจะรับข้อมูลเข้าคือทำนองเพลง และประมวลผลสร้างคอร์ดเพลงที่สามารถประสานเสียงกับทำนองเพลงได้ไพเราะให้กับผู้ใช้ โดยมีขั้นตอนการทำงานคือ

1. ตรวจสอบบันไดเสียงของทำนองเพลง
    1.1 เก็บความยาวทั้งหมดของโน้ตในทำนองเพลง แยกตามระดับเสียงต่างๆ

1.2 เปรียบเทียบระดับเสียงของโน้ตในทำนองเพลงทุกตัว กับโน้ตในแต่ละบันไดเสียง และหากตรงกันจะให้คะแนนกับบันไดเสียงนั้นตามความยาวของโน้ตที่เก็บไว้ เมื่อทำครบทุกบันไดเสียงจะเลือกบันไดเสียงที่ได้รับคะแนนมากที่สุดเป็นบันไดเสียงของทำนองเพลง

1.3 ใช้โน้ตตัวสุดท้ายของเพลงเป็นตัวบ่งชี้ในการเลือกใช้บันไดเสียงระบบเมเจอร์หรือระบบไมเนอร์ โดยโน้ตตัวสุดท้ายของเพลงควรจะตรงกับโน้ตตัวแรกของบันไดเสียง

2. สร้างตรัยเอ็ดในบันไดเสียง

2.1 สร้างชื่อตรัยเอ็ด

2.2 สร้างบันไดเสียงเมเจอร์ เพื่อเตรียมไว้สร้างโน้ตในตรัยเอ็ด

2.3 สร้างโน้ตในตรัยเอ็ด เพื่อเตรียมไว้สำหรับจัดคอร์ดเพลง

3. จัดคอร์ดเพลงให้กับทำนองเพลง

3.1 จับคู่โน้ตในทำนองเพลงกับโน้ตในตรัยเอ็ด และให้คะแนนตามความยาวโน้ต หากพบว่าตรงกัน

3.2 คำนวณคะแนนรวมของทุกคอร์ดในทุกห้องเพลง

3.3 คัดเลือกคอร์ดเพลงครั้งที่ 1 โดยเลือกคอร์ดเพลงที่ได้คะแนนสูงสุดในแต่ละห้องเพลงออกมา

3.4 คัดเลือกคอร์ดเพลงครั้งที่ 2 คือการเลือกคอร์ดเพลงที่ดีที่สุดให้เหลือเพียงคอร์ดเดียวในแต่ละห้องเพลงในกรณีที่มีคอร์ดเพลงได้รับเลือกมากกว่า 1 คอร์ดจากการคัดเลือกในครั้งที่ 1

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย