

CHAPTER V

IMPLEMENTATION AND TESTING

5.1 Implementation

5.1.1. Java Code

Function of Java Code to organize action or data display appear when the application is working. Code below is core of each application.

- Simulator Class

Function of this code is to access other class.

```
Intent hash = new Intent(getApplicationContext(),GrowthPage.class);
startActivity(hash);
```

And function of this code to exit program.

```
Intent exit = new Intent(Intent.ACTION_MAIN);
exit.addCategory(Intent.CATEGORY_HOME);
exit.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
startActivity(exit);
```

- Input Page Class

Function of this code is to input data (Transaksi and Item) to SQLite.

```
db.execSQL("INSERT INTO tblRinciTransaksi
VALUES ('"+t+"','"+i+"')");
```

- View Page Class

Function of this code is to show data from SQLite.

```
db.rawQuery("SELECT notransaksi, group_concat(item) AS
'item' FROM tblRinciTransaksi group by notransaksi",
null);
```

- Hash Page Class

Function of this code is to determine Hash Value and input itemset 2 item to each indeks.

```
Cursor qq = db.rawQuery("SELECT
(((C1.noitem*100)+C2.noitem)%"+prima+) AS 'nomor',
C1.item AS 'item1', C2.item AS 'item2', (SELECT
```

```

        count(*)FROM tblRinciTransaksi AS tblC INNER JOIN
tblRinciTransaksi AS tblD ON (tblC.notransaksi =
tblD.notransaksi)WHERE tblC.item = C1.item AND
tblD.item = C2.item) AS 'frekuensi' from tblItem AS
C1, tblItem AS C2 WHERE C2.noitem > C1.noitem AND
C1.frekuensi >= '"+m+"' AND C2.frekuensi>='"+m+"'",
null);
if(qq.moveToFirst()) {
    do{
        String itemset =
"+qq.getString(qq.getColumnIndex("item1"))
+""+qq.getString(qq.getColumnIndex("item2"));
        int hashvalue =
qq.getInt(qq.getColumnIndex("nomor"));
        if(bucket[hashvalue+1][1] == kosong) {
            bucket[hashvalue+1][1] =
qq.getString(qq.getColumnIndex("nomor"));
            bucket[hashvalue+1][2] = itemset;
            bucket[hashvalue+1][3] =
qq.getString(qq.getColumnIndex("frekuensi"));
        }
        else {
            for(int i=0;i<prima;i++) {
                if(bucket[i+1][1] == kosong) {
                    bucket[i+1][1] =
qq.getString(qq.getColumnIndex("nomor"));
                    bucket[i+1][2] = itemset;
                    bucket[i+1][3] =
qq.getString(qq.getColumnIndex("frekuensi"));
                    String ss = "+i;
                    break;
                }
            }
        }
    }
    } while(qq.moveToNext());
}qq.close();

```

- Growth Page Classs

This code is to determine Items in FP-Tree Simulation.

```

Cursor d = db.rawQuery("SELECT *FROM tblGrowth WHERE
kolom = '"+col+"'", null);
d.moveToFirst();
hasil += ""+d.getString(d.getColumnIndex("item"));
db.execSQL("INSERT INTO tblGrowthDua
VALUES('"+d.getString(d.getColumnIndex("item"))
+"','"++d.getString(d.getColumnIndex("baris"))
+"','"++hasil+"')");
if(d != null) {
    while(d.moveToNext()) {
        hasil +=
"-"+d.getString(d.getColumnIndex("item"));
        db.execSQL("INSERT INTO tblGrowthDua
VALUES('"+d.getString(d.getColumnIndex("item"))
+"','"++d.getString(d.getColumnIndex("baris"))
+"','"++hasil+"')");
    }
} d.close();

```

- Growth Two Page Class
Show the result from FP-Growth Algorithm Process.
`db.rawQuery("Select indeks, sum(frekuensi) AS frekuensi, unik from tblGrowthFix group by unik having sum(frekuensi) >= "+min, null);`
- Bantuan Class
Show text about program and code below to show text to explain about View Page.
`o2.setText("Halaman Lihat Data Bertujuan untuk Melihat Data-Data yang sudah diInputkan dalam bentuk tabel.");`

5.1.2. XML Code

Function of XML Code to design organize or organize display application. In this Code we can organize background, Button, EditText, and other widget. Code below is one of XML Code to be used in each class buliding.

- Code to show Button

```
<Button
    android:id="@+id/input"
    android:layout_gravity="right"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:background="@drawable/selector_gr
een"
    android:text="Input Data" />
```

- Code to show TextView

```
<TextView
    android:id="@+id/text_c1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Transaksi : (Masukan No atau
Nama Transaksi)"
    android:textAppearance="?"
    android:attr/textAppearanceMedium" />
```

- Code to show EditText

```

<EditText
    android:id="@+id/transaksi"
    android:layout_width="match_parent"
    android:hint="Ex: T100"
    android:layout_height="wrap_content"
    android:ems="10" >

    <requestFocus />
</EditText>
```
- Code to show TableLayout (Table)

```

<TableLayout
    android:id="@+id/table4"
    android:layout_width="wrap_content"
    android:layout_gravity="center"
    android:layout_height="wrap_content" >

    </TableLayout>
```
- Code to show scrollview

```

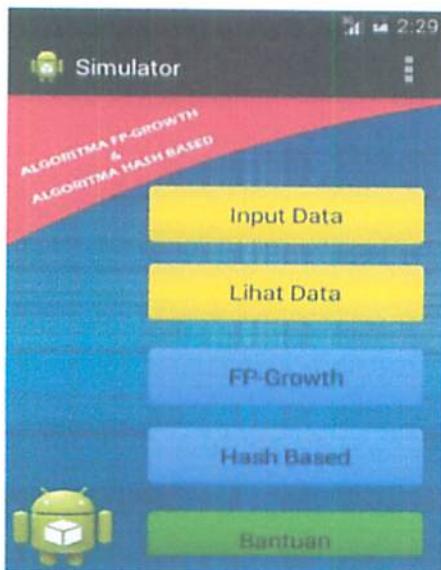
<ScrollView
    android:id="@+id/scrollView8"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="60dp" >
```

5.2 Testing

The pictures below is step application working. From Input Data, Show Data, until FP-Growth Algorithm and Hash Based Algorithm result. This step is to testing the application and go to show FP-Growth Algorithm and Hash Based Algorithm application result equals with result on CHAPTER II Literature Study.

- Simulator Class

This is first screen from Simulation Application.



Picture 5.1 Simulator Page

- Input Page Class

First, Writer must input data (Transaksi and Item) in Input Page class. Writer data input is equal to the existing data in chapter 2 (Table 2.1). Picture below only show one of data input.



Picture 5.2 Input Page

- View Page Class

Data already input include in SQLite. And the View Page Class will display data from SQLite in table included in the data that we input in Picture 5.2.



The screenshot shows a mobile application interface titled "Simulator" with a subtitle "ALGORITMA FP-TREE & ALGORITMA HASH SALTO". It features a yellow "Menu" button, a red "Clear All" button, and a green "Input Data" button. Below these buttons is a table titled "Tampilan Setiap Transaksi dan Item yang sudah dimposkan". The table has two columns: "No Transaksi" and "Item". The data is as follows:

No Transaksi	Item
T100	A1,A2,A5
T200	A2,A4
T300	A2,A3
T400	A1,A2,A4
T500	A1,A3
T600	A3,A2
T700	A1,A3
T800	A1,A2,A3,A5
T900	A1,A2,A3

At the bottom left is a green Android robot icon holding a white cube.

Picture 5.3 View Page

- Growth Page Class

Existing Data will be processed using the FP-Growth Algorithm. The Pictures below will show every process from first Iteration until FP-Tree Simulation (Picture 5.6) such as Picture 2.3. Next Step entry on Growth Two Page. Picture 5.4 show items and input min support value equals with Table 2.2, and Picture 5.5 show items that been order in each transaction equals with Table 2.4.

Simulator
ALGORITMA FP-GROWTH & ALGORITMA HASH BASED

2 Min Support

Tabel C1(Candidate 1 Itemset)

No Item	Item	Frekuensi
0	A1	6
1	A2	7
2	A3	6
3	A4	2
4	A5	2

Picture 5.4 Table Item

Simulator
ALGORITMA FP-GROWTH & ALGORITMA HASH BASED

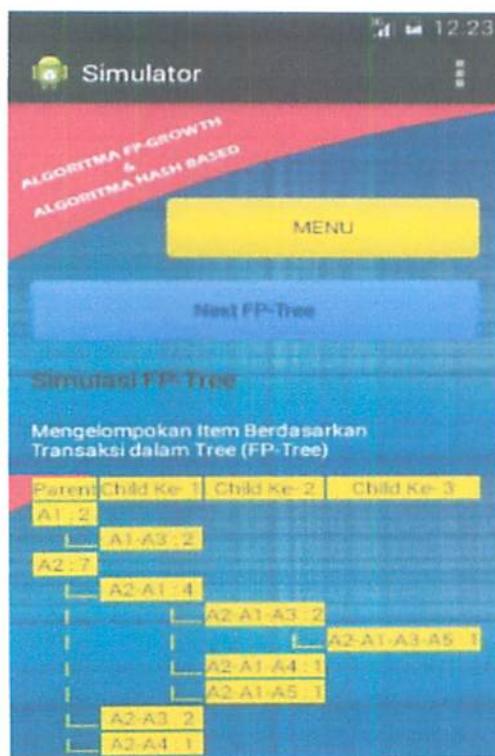
MENU

First FP-Tree

Menampilkan Item Yang Memiliki Frekuensi == 2 dan Mengurutkan Item dalam setiap transaksi berdasarkan jumlah frekuensi terbesar.

Kolom	No Transaksi	Itemset	Baris
0	T900	A2,A1,A3	0,1,2
1	T800	A2,A1,A3,A5	0,1,2,3
2	T400	A2,A1,A4	0,1,2
3	T100	A2,A1,A5	0,1,2
4	T300	A2,A3	0,1
5	T600	A2,A3	0,1
6	T200	A2,A4	0,1
7	T500	A1,A3	0,1
8	T700	A1,A3	0,1

Picture 5.5 First Iteration



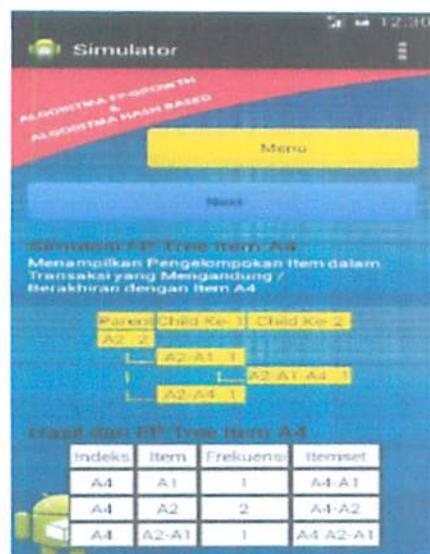
Picture 5.6 FP-Tree Simulation

- Growth Two Page Class

This class complete the FP-Growth Algorithm from Growth Page Class. Pictures below show FP-Tree Simulation that have small frequent item accord with Picture 2.4 to Picture 2.6.



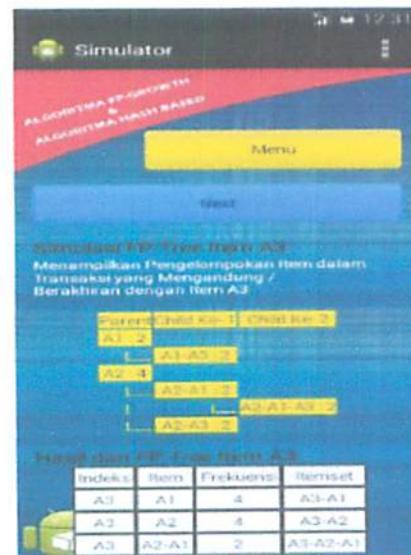
Picture 5.7



Picture 5.8

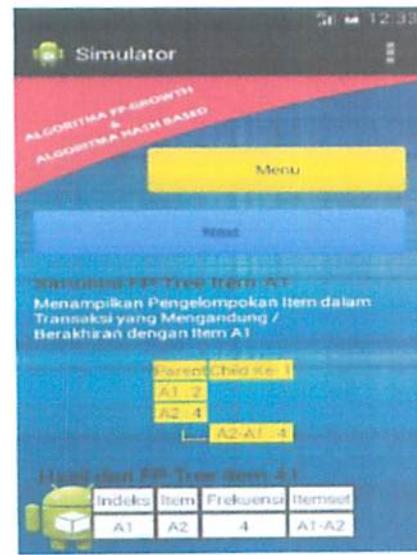
FP-Tree End Line in A5

FP-Tree End Line in A4



Picture 5.9

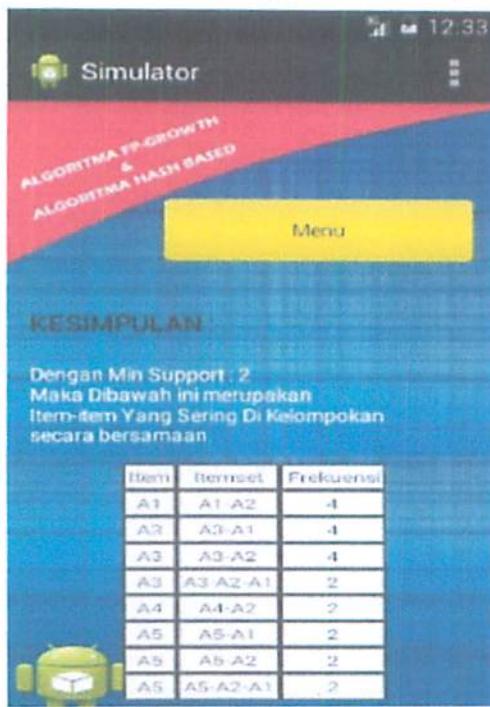
FP-Tree End Line in A3



Picture 5.10

FP-Tree End Line in A1

And from Growth Two Page Class, we will get the result where the result from FP-Growth Algorithm Application (Picture 5.11) equals with FP-Growth Algorithm in Chapter II.



Picture 5.11 Result FP-Growth Algorithm

- Hash Page Class

Pictures below show the Hash Based Algorithm process from first Iteration to third Iteration. Purpose of these pictures is to show that the result of the Hash Based Algorithm Application is equal to the Hash Based Algorithm in Chapter II. Picture 5.12 is equal to Table 2.5, Picture 5.13 is equal to Table 2.6, Picture 5.14 is equal to Table 2.8, and Picture 5.16 that shows the third iteration from the Hash Based Algorithm is equal to Table 2.9.

Simulator
ALGORITMA FP-GROWTH & ALGORITMA HASH BASED

MENU

2 Min Support

Tabel C1 (Candidate 1 Itemset)

No Item	Item	Frekuensi
1	A1	6
2	A2	7
3	A3	6
4	A4	2
5	A5	2

Picture 5.12 Table Item

Simulator
ALGORITMA FP-GROWTH & ALGORITMA HASH BASED

MENU

Next Hash Table - C2

ITERASI 1

Menampilkan Item yang Memiliki Frekuensi >= 2

No Item	Item	Frekuensi
1	A1	6
2	A2	7
3	A3	6
4	A4	2
5	A5	2

Picture 5.13 First Iteration

Simulator
ALGORITMA FP-GROWTH & ALGORITMA HASH BASED

MENU

Next Hash Table - C2

Tabel C2 (Candidate 2 Itemset)

Item yang Memenuhi Syarat Iterasi 1 digabungkan menjadi itemset 2 item dan dengan menggunakan rumus Hash Bucket, itemset tersebut dimasukkan dalam Hash Table berdasarkan indeks yang didapat dari hasil Hash Bucket.

Indeks	Nilai Hash	Item	Frekuensi
0	5	A2-A3	4
1	6	A2-A4	2
2	7	A3-A4	0
3	3	A1-A2	4
4	4	A1-A3	4
5	5	A1-A4	1
6	6	A1-A5	2
7	7	A2-A5	2
8	8	A3-A5	1
9	9	A4-A5	0
10	-	-	-

Picture 5.14 Table C2 (Hash Table)

Simulator
ALGORITMA FP-GROWTH & ALGORITMA HASH BASED

MENU

Next Hash Table - C2

ITERASI 2

Menampilkan Itemset 2 item yang memiliki Frekuensi >= 2

Indeks	Nilai Hash	Item	Frekuensi
0	5	A2-A3	4
1	6	A2-A4	2
2	-	-	-
3	3	A1-A2	4
4	4	A1-A3	4
5	-	-	-
6	6	A1-A5	2
7	7	A2-A5	2
8	-	-	-
9	-	-	-
10	-	-	-

Picture 5.15 Second Iteration



Picture 5.16 Third Iteration