

CHAPTER V

IMPLEMENTATION AND TESTING

5.1 Implementation

5.1.1 Preprocessing

First, we need sqlite already installed and prepared the browser that will be used. And then, prepare jquery-1.11.1.min.js stored in the application folder. The next stage we create a file page1_1.html, iterasi.js, and stored in the application folder as early in the process of running the application. In page1_1.html there is a process to call jquery-1.11.1.min.js and iterasi.js.

```
<DOCTYPE html>
<html>
<title>Algoritma apriori</title>
<head>
<link rel="stylesheet" type="text/css" href="myCSS.css">
<script type="text/javascript" src="./jquery-1.11.1.min.js">
</script>
<script type="text/javascript" src="./iterasi.js">
</script>
```

Figure 5.1.1 JQuery calling process

And then, we create and open the database in the file iterasi.js:

```
$(document).ready(function(){
//open database
var db = openDatabase('mydb', '1.0', 'item DB', 2 * 1024 * 1024);
```

Figure 5.1.2 Open Database

explained that create a database in the browser with the name mydb with sqlite version 1.0 and database size per bits is $2 * 1024 * 1024$.

In the database table we insert option is tblBuah, tblMinuman, and tblMakanan. There is also a table for storing the data to be processed is tblTransaksi, tblNamaitem and tblTrans.

```

Andrio-MacBook-Pro:~ andriambocx$ sqlite3 ~/Library/Safari/Databases/file_0/6668888888888888.db
SQLite version 3.6.3
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .databases
ceq  name      file
-----
0  main      /Users/andriambocx/Library/Safari/Databases/file_0/666888
sqlite> .tables
tblInfoDatabaseInfoTable__  tblTrans
tblBusch                    tblTransakul
tblMcaitec

```

Figure 5.1.3 View Table

Table option is used to specify the tables and data to be used as items to be compared.

5.1.2 Transaction process

The first transaction process by determining amount of the transaction is determined from the value of the index is added in each new transaction.

```

var getForm = function() {
  return $('
    <form name="myForm" id="myForm">\
      <input type="button" href="#" id="add'+[index]+' " value="Transaksi'+ [index] +' ">\
    </form>\
  ');
}

addItem.hide();

//***** add Transaction *****
addItem.on("click", function() {
  if(index>12){
    alert("Transaction overload");
    return false;
  }
  index++;
  var form = getForm(index);
  menuItem.append(form).delay(980);
  form.find("#add"+index).on("click", function() {
    addBut(index);
  });
});

```

Figure 5.1.4 New Transaction

The next process, count the number of items that will appear on every transaction by selecting a random number. Value of the random the number of

will determine the number of items will be saved in the variable numRand then send it to RandomData function (index, numRand).

```
function addBut(index) {  
  var numLow = 1;  
  var numHigh = 5;  
  var adjustedHigh = (parseFloat(numHigh) - parseFloat(numLow)) + 1;  
  var numRand = Math.floor(Math.random()*adjustedHigh) + parseFloat(numLow);  
  //jml(index, numRand);  
  RandomData(index, numRand);  
}
```

Figure 5.1.5 Random process

In RandomData function, the value of the index is taken to fill in the variables valTrans and enter into tblTransaksi. Then there is a second random to be repeated according to the value of numRand to get a random value that is stored in variable idRand.

It used to retrieve the id on tblBuah or other tables, and displays the name of the corresponding item id. The next process dataRand and idRand will be put into tblNamaItem, and then valTrans and dataRand on insert into tblTrans as a foreign key.

```

function RandoData(index, numRand) {
    pros1.empty();
    pros1.append(pros);
    if(index < 2) {
        frekuensi();
    }
    $(addInsert).hide();
    $(addInsert).append("<br/>");
    var form3 = getForm(index);
    var valTrans = form3.find('#add'+index).val();
    $(addInsert).append(valTrans + " : ");
    var valDatabases = $("databases").val();

    db.transaction(function (tx) {
        var query1 = "INSERT INTO tblTransaksi (trans , jmlRandom) VALUES ('"+valTrans+"', '"+numRand+"')";
        tx.executeSql(query1);
    });

    j=0;
    while (j<numRand) {
        var idLow = 1;
        var idHigh = 10;
        var idRand = [];
        db.transaction(function (tx) {
            var adjHigh = (parseFloat(idHigh) - parseFloat(idLow)) + 1;
            var idRand = Math.Floor(Math.random()*adjHigh) + parseFloat(idLow);
            tx.executeSql('SELECT* FROM '+valDatabases+' where id = '+[idRand]+';', [], function (tx, result) {
                dataset = result.rows;
                for (var i = 0, item = null; i < dataset.length; i++) {
                    var item = dataset.item(i);
                    var dataRand = item['nama'];
                    var query2 = "INSERT INTO tblRandom (random , idRandom) VALUES ('"+dataRand+"', '"+idRand+"')";
                    var query3 = "INSERT INTO tblTrans (trans , random) VALUES ('"+valTrans+"', '"+dataRand+"')";
                    tx.executeSql(query2);
                    tx.executeSql(query3);
                    $(addInsert).append("<b>"+dataRand+"</b> &nbsp;");
                    $(addInsert).show('slow');
                }
            });
        });
        j++;
    }
}

```

Figure 5.1.6 Insert Data

5.1.3 Iterations Process

A. Iterations 1

The first iteration 1 process apriori algorithm with frequency display items that have been through the selection process of the value of the minimum support. The previous there should have been a minimum support value for reference in a minimal amount of items that frequently appear.

```

//***** Iterasi 1 *****
var minSupport = $("#it1");
var c1 = 1;

function iterasi1(mins) {
var minSupport = '<b>Minimal support - ' + mins + '</b>';
var textIterasi1 = '<h3 align="center">iterasi 1</h3>';
showIterasi1.append(textIterasi1);
var allFrekuensi = 'SELECT A1.linkimg as Namaitem, (select count(tblTrans.trans) FROM tblTrans where tblTrans.linkimg =
A1.linkimg) as Frekuensi from tblNamaitem as A1 where (select count(tblTrans.trans) FROM tblTrans where
tblTrans.linkimg = A1.linkimg) >= ' + mins + ' group by A1.linkimg';
do.transaction(function(tx) {
tx.executeSql( allFrekuensi , [], function (tx, result) {
dataset = result.rows;
var e1 = '<table style="width:300px;"><tr><td style="width:200px">Name</td><td align="center"
style="width:100px"> Frekuensi </td></tr>';
showIterasi1.append(e1);
for (var i = 0, item = null; i < dataset.length; i++) {
var item = dataset.item(i);
var img1 = item['Namaitem'];
var h1 = "<tr><td style='width:200px'><img src= " + img1 + " alt='algo face' width='30px'
height='30px'></td><td align='center' style='width:100px;'>" + item['Frekuensi'] + "</td></tr>";
showIterasi1.append(h1);
}
if(img1 == undefined) {
showIterasi1.hide();
} else {
showIterasi1.delay(200);
showIterasi1.show('slow').delay(2000).hide('slow');
showIterasi2.delay(3000);
}
}
var e2 = '</table>';
showIterasi1.append(e2).fadeIn("slow");
});
});
$("#minsup1").empty();
minsup1.append(minSupport);
}

```

Figure 5.1.7 process iteration 1

B. Iterations 2

Iteration 2 process is done by using the results of the itemset and frequency at iteration 1. Itemset in iteration 2 to 2 itemset experience a combination that uses the join step $L1 > L2$. The combination will receive the number of frequencies, and the frequencies that are not included in the minimum support will be deleted.

```

//***** iterasi 2 *****

var showiterasi2 = $('#It2');

function iterasi2(mins) {
var Iterasi2 = '<h3 align="center">Iterasi 2</h3>';
showiterasi2.append(Iterasi2);
var alliterasi2 = 'SELECT A1.linkimg as Item1, A2.linkimg as Item2, (select count(*) from tblTrans as tblA inner join
tblTrans as tblB on (tblA.trans = tblB.trans) where tblA.linkimg = A1.linkimg and tblB.linkimg = A2.linkimg) as
Frekuensi from tblNamaitem as A1, tblNamaitem as A2 where A2.linkimg > A1.linkimg and (select count(*) from tblTrans as
tblA inner join tblTrans as tblB on (tblA.trans = tblB.trans) where tblA.linkimg = A1.linkimg and tblB.linkimg =
A2.linkimg) >' + mins + ' group by A1.linkimg, A2.linkimg';
db.transaction(function (tx) {
tx.executeSql(alliterasi2, [], function (tx, result) {
dataset = result.rows;
var e3 = '<table style="width:300px"><tr><td style="width:200px">Nama</td><td align="center"
style="width:100px"> Frekuensi </td></tr>';
showiterasi2.append(e3);
for (var i = 0, item = null; i < dataset.length; i++) {
var item = dataset.item(i);
var img1 = item['Item1'];
var img2 = item['Item2'];
var hal = "<tr><td style='width:200px'> <img src= " + img1 + " alt='algo face' width='30px'
height='30px'>&nbsp;&nbsp;<img src= " + img2 + " alt='algo face' width='30px' height='30px'> </td><td align='center'
style='width:100px'>" + item['Frekuensi'] + "</td></tr>";
showiterasi2.append(hal);
}
if(img2 == undefined) {
showiterasi2.hide();
butfinish.delay(3000);
butfinish.show('slow');
} else { showiterasi2.show('slow').delay(2000).hide('slow');
showiterasi3.delay(6000);
}
var e4 = '</table>';
showiterasi2.append(e4);
});
});
}

```

Figure 5.1.8 process iteration 2

C. Iterations 3

Iteration 3 process is done by using the results of the itemset and frequency at iteration 2. Itemset in iteration 3 to 3 itemset experience a combination that uses the join step $L1 \times L2 \times L3$. The combination will receive the number of frequencies, and the frequencies that are not included in the minimum support will be deleted.

```

//***** Iterasi 3 *****

var showiterasi3 - $("#It3");

function iterasi3(mins) {
var Iterasi3 - '<h3 align="center">Iterasi 3</h3>';
showiterasi3.append(iterasi3);
var alliterasi3 - 'SELECT A1.linkimg as Item1, A2.linkimg as Item2, A3.linkimg as Item3, (select count(*) from tblTrans
as tblA inner join tblTrans as tblB on (tblA.trans - tblB.trans) inner join tblTrans as tblC on (tblA.trans -
tblC.trans) where tblA.linkimg = A1.linkimg and tblB.linkimg = A2.linkimg and tblC.linkimg = A3.linkimg) as Frekuensi
from tblNamaitem as A1, tblNamaitem as A2, tblNamaitem as A3 where A2.linkimg > A1.linkimg and A3.linkimg > A2.linkimg
and (select count(*) from tblTrans as tblA inner join tblTrans as tblB on (tblA.trans - tblB.trans) inner join tblTrans
as tblC on (tblA.trans - tblC.trans) where tblA.linkimg = A1.linkimg and tblB.linkimg = A2.linkimg and tblC.linkimg =
A3.linkimg) >= 'mins+' group by A1.linkimg, A2.linkimg, A3.linkimg';
db.transaction(function (tx) {
tx.executeSql(alliterasi3, [], function (tx, result) {
dataset = result.rows;
var e5 - '<table style="width:350px;"><tr><td style="width:250px">Nama</td><td align="center"
style="width:100px"> Frekuensi </td></tr>';
showiterasi3.append(e5);
for (var i = 0, item = null; i < dataset.length; i++) {
var item = dataset.item(i);
var img1 = item['item1'];
var img2 = item['item2'];
var img3 = item['item3'];
var hal - '<tr><td style="width:250px"> <img src= " + img1 + " alt="algo face" width="30px"
height="30px">&nbsp;&nbsp;&nbsp;<img src= " + img2 + " alt="algo face" width="30px" height="30px">&nbsp;&nbsp;&nbsp;<img src= " + img3 + "
alt="algo face" width="30px" height="30px"> </td><td align="center" style="width:100px"> ' + item['Frekuensi'] + '</td>
</tr>';
showiterasi3.append(hal);
}
if(img3 == undefined) {
showiterasi1.hide();
butfinish.delay(6000);
butfinish.show('slow');
} else { showiterasi3.show('slow').delay(2000).hide('slow');
showiterasi4.delay(9000);
}
var e6 - '</table>';
showiterasi3.append(e6);
}};

```

Figure 5.1.9 process iteration 3

D. Iterations 4

Iteration 4 process is done by using the results of the itemset and frequency at iteration 3. Itemset in iteration 4 to 4 itemset experience a combination that uses the join step $L1 \times L2 \times L3 \times L4$. The combination will receive the number of frequencies, and the frequencies that are not included in the minimum support will be deleted.

```

function iterasi4(mins) |
var Iterasi4 = '<h3 align="center">Iterasi 4</h3>';
showiterasi4.append(Iterasi4);
var alliterasi4 = 'SELECT A1.linkimg as Item1, A2.linkimg as Item2, A3.linkimg as Item3, A4.linkimg as Item4, (select
count(*) from tblTrans as tblA inner join tblTrans as tblB on (tblA.trans = tblB.trans) inner join tblTrans as tblC on
(tblA.trans = tblC.trans) inner join tblTrans as tblD on (tblA.trans = tblD.trans) where tblA.linkimg = A1.linkimg and
tblB.linkimg = A2.linkimg and tblC.linkimg = A3.linkimg and tblD.linkimg = A4.linkimg) as Frekuensi from tblNamaitem as
A1, tblNamaitem as A2, tblNamaitem as A3, tblNamaitem as A4 where A2.linkimg > A1.linkimg and A3.linkimg > A2.linkimg
and A4.linkimg > A3.linkimg and (select count(*) from tblTrans as tblA inner join tblTrans as tblB on (tblA.trans =
tblB.trans) inner join tblTrans as tblC on (tblA.trans = tblC.trans) inner join tblTrans as tblD on (tblA.trans =
tblD.trans) where tblA.linkimg = A1.linkimg and tblB.linkimg = A2.linkimg and tblC.linkimg = A3.linkimg and
tblD.linkimg = A4.linkimg) >- 'mins' group by A1.linkimg, A2.linkimg, A3.linkimg, A4.linkimg';
db.transaction(function(tx) {
tx.executeSql(alliterasi4, [], function(tx, result) {
dataset = result.rows;
var e7 = '<table style="width:350px;"><tr><td style="width:250px">Name</td><td align="center"
style="width:100px"> Frekuensi </td></tr>';
showiterasi4.append(e7);
for (var i = 0, item = null; i < dataset.length; i++) {
var item = dataset.item(i);
var img1 = item['Item1'];
var img2 = item['Item2'];
var img3 = item['Item3'];
var img4 = item['Item4'];
var hal = '<tr><td style="width:250px"> <img src= " + img1 + " alt="algo face" width="30px"
height="30px">&nbsp;<img src= " + img2 + " alt="algo face" width="30px" height="30px">&nbsp;<img src= " + img3 + "
alt="algo face" width="30px" height="30px">&nbsp;<img src= " + img4 + " alt="algo face" width="30px" height="30px">
</td><td align="center" style="width:100px"> + item['Frekuensi'] + "</td></tr>";
showiterasi4.append(hal);
}
if(img4 == undefined) {
showiterasi4.hide();
butfinish.delay(9000);
butfinish.show('slow');
} else {
showiterasi4.show('slow').delay(2000).hide('slow');
showiterasi5.delay(12000);
}
var e8 = '</table>';
showiterasi4.append(e8);
});

```

Figure 5.1.10 process iteration 4

E. Iterations 5

Iteration 5 process is done by using the results of the itemset and frequency at iteration 4. Itemset in iteration 5 to 5 itemset experience a combination that uses the join step $L1 \times L2 \times L3 \times L4 \times L5$. The combination will receive the number of frequencies, and the frequencies that are not included in the minimum support will be deleted.


```

function iterasi5(mins) {
var Iterasi5 = '<h3 align="center">Iterasi 5</h3>';
showiterasi5.append(Iterasi5);
var alliterasi5 = 'SELECT A1.linkimg as Item1, A2.linkimg as Item2, A3.linkimg as Item3, A4.linkimg as Item4, A5.linkimg
as Item5, (select count(*) from tblTrans as tblA inner join tblTrans as tblB on (tblA.trans = tblB.trans) inner join
tblTrans as tblC on (tblA.trans = tblC.trans) inner join tblTrans as tblD on (tblA.trans = tblD.trans) inner join
tblTrans as tblE on (tblA.trans = tblE.trans) where tblA.linkimg = A1.linkimg and tblB.linkimg = A2.linkimg and
tblC.linkimg = A3.linkimg and tblD.linkimg = A4.linkimg and tblE.linkimg = A5.linkimg) as Frekuensi from tblNamaitem as
A1, tblNamaitem as A2, tblNamaitem as A3, tblNamaitem as A4, tblNamaitem as A5 where A2.linkimg > A1.linkimg and
A3.linkimg > A2.linkimg and A4.linkimg > A3.linkimg and A5.linkimg > A4.linkimg and (select count(*) from tblTrans as
tblA inner join tblTrans as tblB on (tblA.trans = tblB.trans) inner join tblTrans as tblC on (tblA.trans = tblC.trans)
inner join tblTrans as tblD on (tblA.trans = tblD.trans) inner join tblTrans as tblE on (tblA.trans = tblE.trans) where
tblA.linkimg = A1.linkimg and tblB.linkimg = A2.linkimg and tblC.linkimg = A3.linkimg and tblD.linkimg = A4.linkimg and
tblE.linkimg = A5.linkimg) >= '+mins+' group by A1.linkimg, A2.linkimg, A3.linkimg, A4.linkimg, A5.linkimg';
db.transaction(function (tx) {
tx.executeSql(alliterasi5, [], function (tx, result) {
dataset = result.rows;
var e9 = '<table style="width:350px;"><tr><td style="width:250px">Name</td><td align="center"
style="width:100px"> Frekuensi </td></tr>';
showiterasi5.append(e9);
for (var i = 0, item = null; i < dataset.length; i++) {
var item = dataset.item(i);
var img1 = item['Item1'];
var img2 = item['Item2'];
var img3 = item['Item3'];
var img4 = item['Item4'];
var img5 = item['Item5'];
var frek = item['Frekuensi'];
var hsl = "<tr><td style='width:250px'> <img src= " + img1 + " alt='algo face' width='30px'
height='30px'>&nbsp;&nbsp;&nbsp;<img src= " + img2 + " alt='algo face' width='30px' height='30px'>&nbsp;&nbsp;&nbsp;<img src= " + img3 + "
alt='algo face' width='30px' height='30px'>&nbsp;&nbsp;&nbsp;<img src= " + img4 + " alt='algo face' width='30px'
height='30px'>&nbsp;&nbsp;&nbsp;<img src= " + img5 + " alt='algo face' width='30px' height='30px'> </td><td align='center'
style='width:100px'>" + frek + "</td></tr>";
showiterasi5.append(hsl);
}
if(img5 == undefined) {
showiterasi5.hide();
butfinish.delay(12000);
butfinish.show('slow');
} else {
butfinish.delay(13000);
butfinish.show('slow');
}
showiterasi5.show('slow').delay(2000).hide('slow');
}
}

```

Figure 5.1.11 process iteration 5

5.2 Testing

A. step-1

Enter a title and select the database to be used.

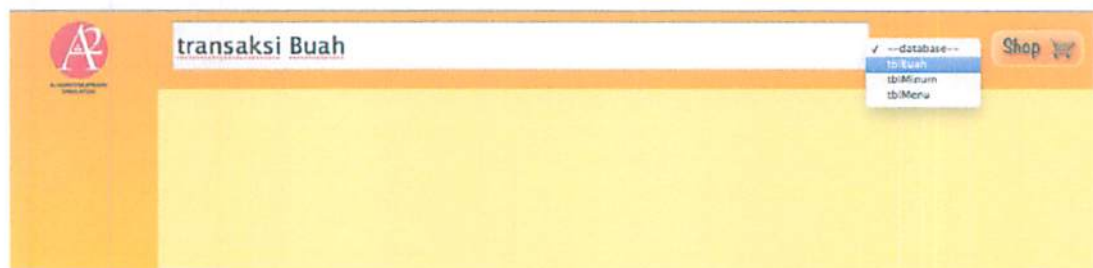


Figure 5.2.1 Testing database selection

B. Step-2

Determine the number of transactions to be compared. And determine itemset that has been selected at random

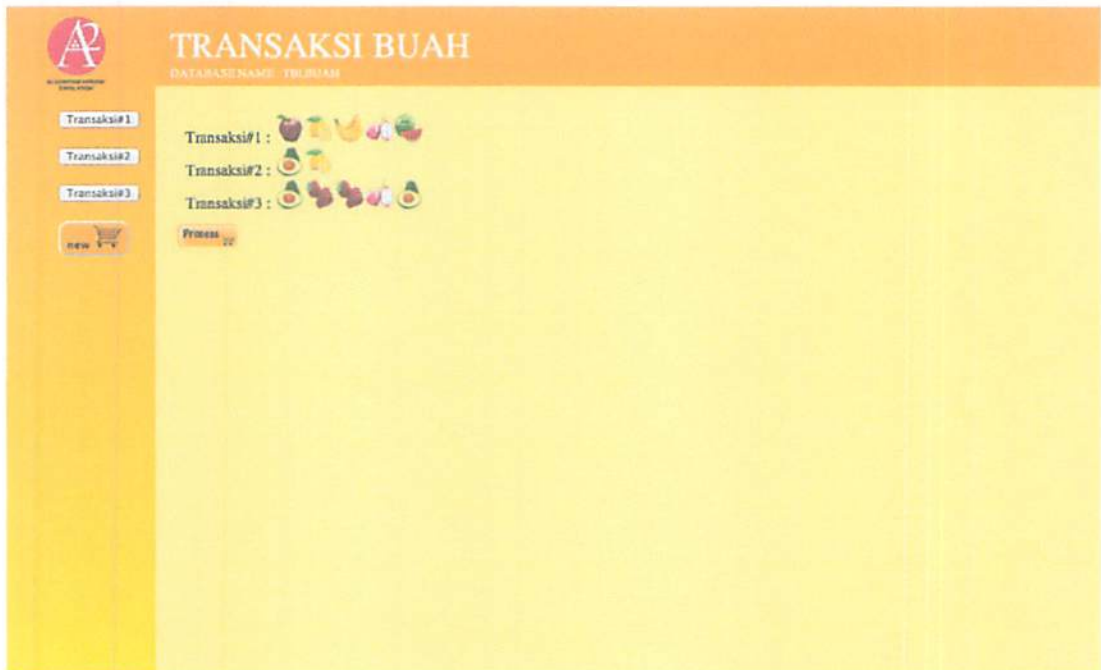


Figure 5.2.2 Transaction testing

C. Step-3

Calculate the frequency of the selected transactions and enter the minimum support value

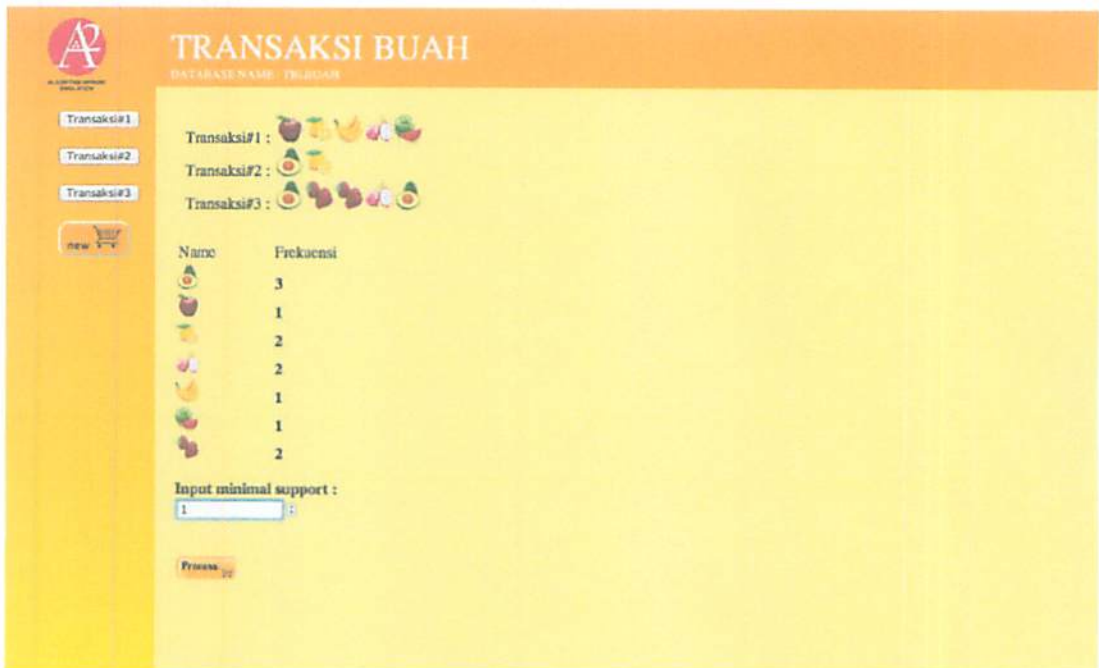


Figure 5.2.3 *Transaction testing*

D. Step-4

after the data is processed with a ratio of minimum support, it will display the final results of iterations until it finds frequent data.

a. Iterations 1

The first iteration process with minimal support comparison.

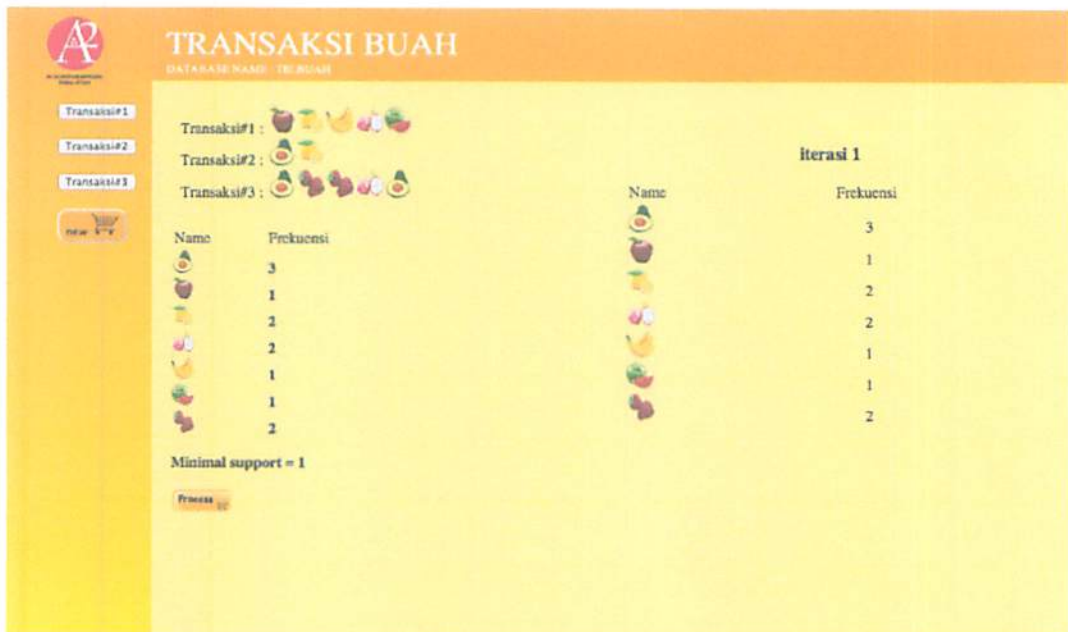


Figure 5.2.4 View iteration 1

b. Iterations 2

The second iteration process with minimal support comparison.

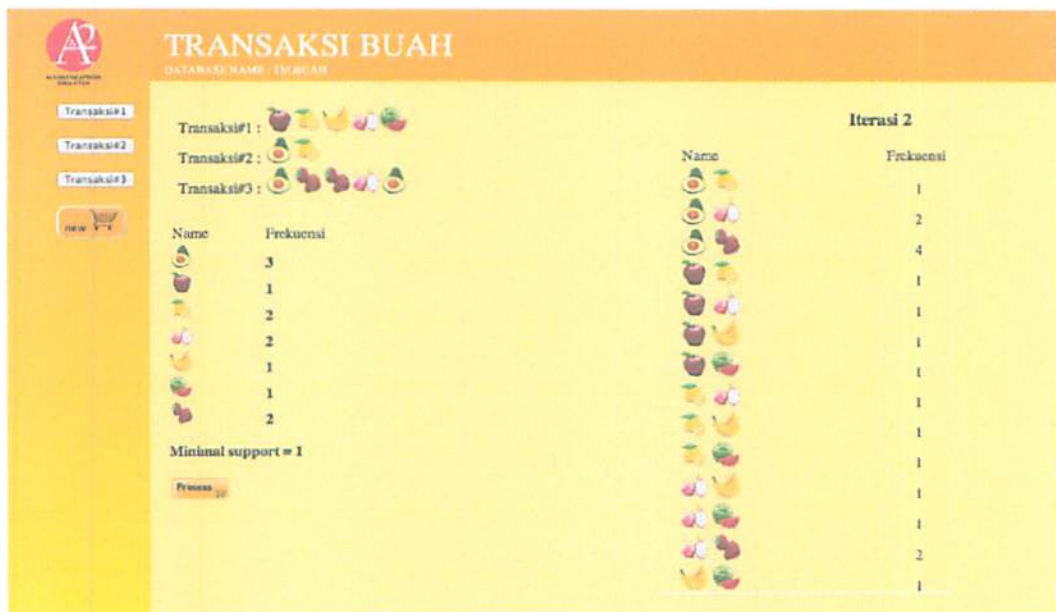


Figure 5.2.5 View iteration 2

c. Iterations 3

The third iteration process with minimal support comparison.

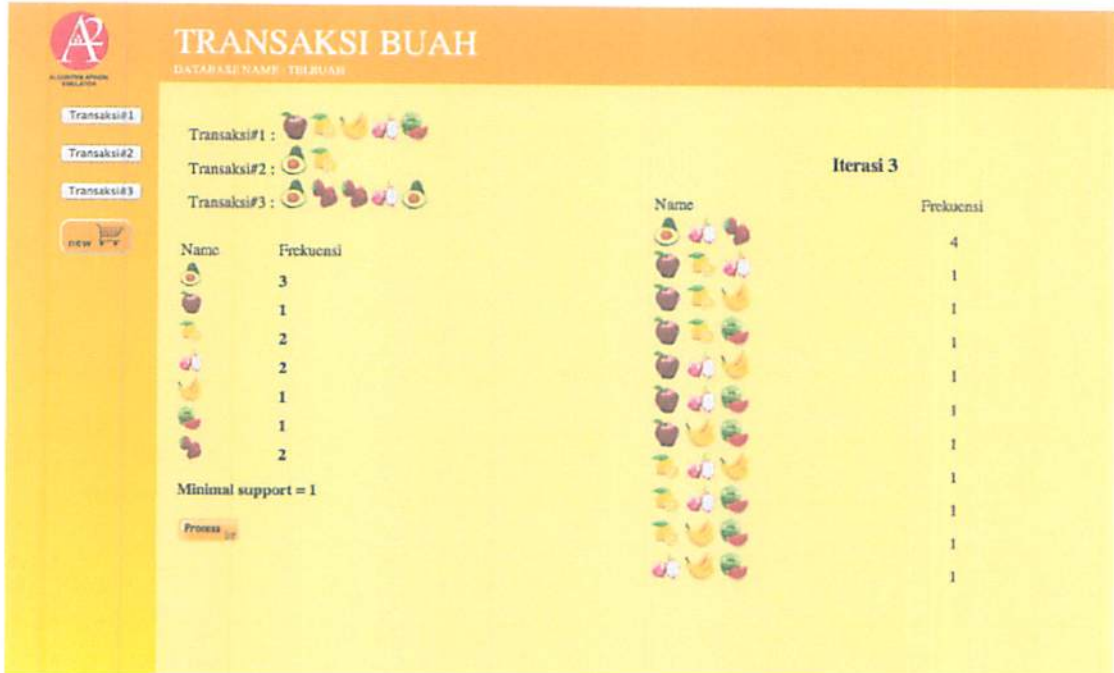


Figure 5.2.6 View iteration 3

d. Iterations 4

The fourth iteration process with minimal support comparison



Figure 5.2.7 view iteration 4

e. Iteration 5

The fifth iteration process with minimal support comparison.

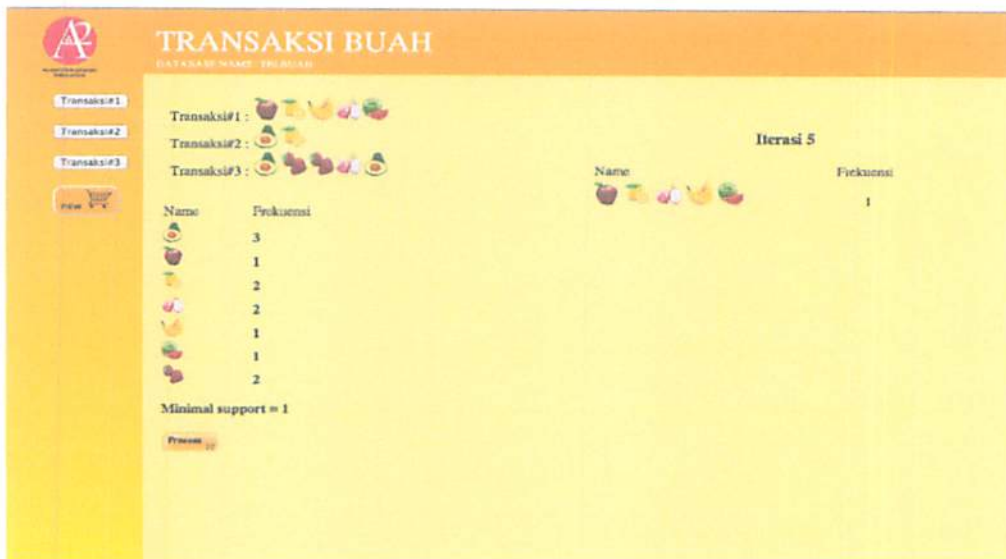


Figure 5.2.8 view iteration 5

5.3 Interface

A. Before process

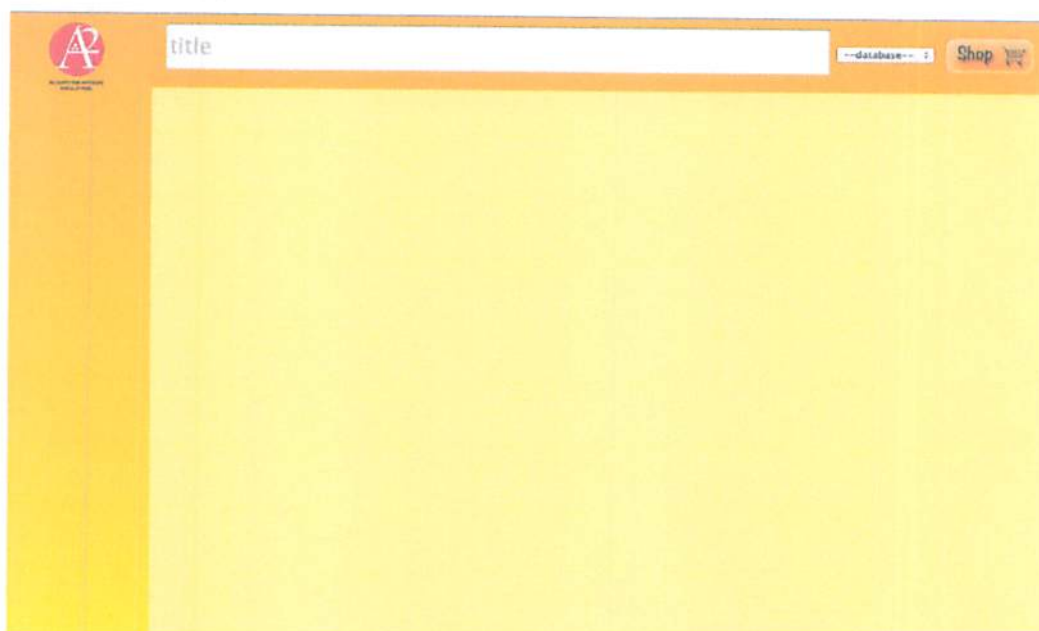


Figure 5.3.1 Before process

B. After Process



TRANSAKSI BUAH
DATABASE NAME: TELURIAH

Transaksi#1 : 
Transaksi#2 : 
Transaksi#3 : 

Name	Frekuensi
	3
	1
	2
	2
	1
	1
	2

Minimal support = 1

Process 

Figure 5.3.2 After Process