CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation

This is tables that are used in this project.

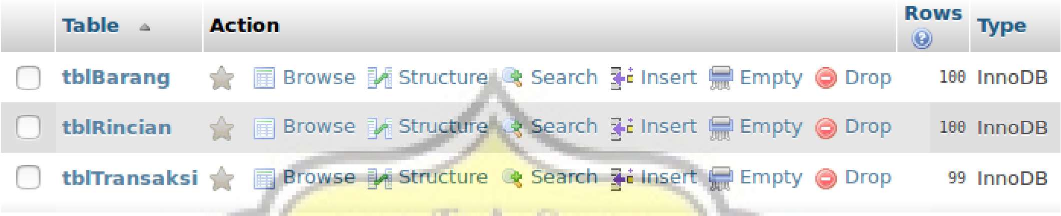


	Table ▲	Action	Rows	Type
□	tblBarang	☆ Browse Structure Search Insert Empty Drop	100	InnoDB
□	tblRincian	☆ Browse Structure Search Insert Empty Drop	100	InnoDB
□	tblTransaksi	☆ Browse Structure Search Insert Empty Drop	99	InnoDB

Illustration 5.1: Tables on Database

After creating database and tables, data is inputted into database.

```
1. LOAD DATA INFILE
   '/opt/lampp/htdocs/__perjuanganAkhir/data/Barang.csv' INTO
   TABLE tblBarang
2. FIELDS TERMINATED BY ',' ENCLOSED BY '"'
3. LINES TERMINATED BY '\n'
4. IGNORE 1 LINES;
```

Code above is used to inputting data to the database. Line 1 is for loading data in file from CSV data (given by data location) into any table name where to be inputted. Line 2 and 3 is for terminated data. And last line is to ignore how many lines from top. Top line / 1 lines is a header of table which is no need to inserted to database.

After data has been inputted, researcher can’t directly enter to main program which is in index.php, minimum support must be inputted first.

```
1. <?php
2.   include('koneksi.php');
3.   session_start();
4.
5.   if(isset($_POST['username']))
6.       $_SESSION['minimSu']= $_POST['username'];
7.
8.   if(!isset($_SESSION['minimSu']))
9.       header("Location: isiDulu.php");
10.
11.      $minimumSupport = $_SESSION['minimSu'];
12.  ?>
```

That is the code from index.php to input minimum support first before loading the system program. On the fifth line, there's IF statement that if $_POST['username'] has been set, the value given to $_SESSION['minimSu']. Then on eighth line, IF statement that if $_SESSION['minimSu'] is not set, that means $_POST['username'] is not se too, and minimum support must be inputted.

After minimum support is inputted, The program system is entered. Next, when about to enter FP-Growth process, a button must be clicked. When button is clicked, the process starts. Early the process starts, time is recorded.

```
$waktuAwal = microtime(true);
```

Because for system analysis, time is recorded while system is working. The function is program time can be achieved. This using microtime(true) to record the time.

```
$waktuAkhir = microtime(true);
$waktuProses = $waktuAkhir - $waktuAwal;
```

At the end of program, still using microtime(true) but placed in different variable. Next, two variables are eliminated and system process can be achieved.

Then, here are the codes of creating Conditional Pattern Base.

```
<h5>d.Membuat Temporary tabel untuk menjadikan Conditional Pattern Base</h5>
<br>
<?php
//pindah variabel biar isa diubah"
$wKB     = $_SESSION['listKodeBarang'];
$lString = $_SESSION['listString'];
//install temporary table
$tabel = "CREATE TEMPORARY TABLE conditionalPattern(
        kodeBarang VARCHAR(25),
        listTransaksi VARCHAR(255));";
$temporary = mysqli_query($db_link,$tabel);

//loop
foreach ($wKB as $noKB => $valueKodeBarang)
{
    $c=0;
    foreach ($lString as $iii => $valueListString)
    {
        $checker = strpos($valueListString, $wKB[$noKB]);
        if($checker !== FALSE)
        {
```

```
22.             /*echo  "Transaksi  no.".$iii."  ada  kode  barang  ".
  $wKB[$noKB]." ";
23.             echo "<br>";
24.             $c++;
25.             */ //--dipotong Stringnya
26.             $lString[$iii]=str_replace($wKB[$noKB],'',
  $lString[$iii]);
27.             $lString[$iii]=trim($lString[$iii],',');
28.             $lString[$iii]=str_replace(',,',',',
  $lString[$iii]);
29.             //dimasukkan Table temporer
30.             if($lString[$iii] != NULL)
31.             {
32.             $sql = "INSERT INTO conditionalPattern VALUES
33.                 ('".$wKB[$noKB]."','".$lString[$iii]."');";
34.             $hasil = mysqli_query($db_link,$sql);
35.             }
36.         }
37.     }
38. }
39.  ?>
```

On lines 4 and 5, value from both Session is transferred into other variables. Then lines 8 until 10 is new temporary table called conditionalPattern declaration. After that, started from line 14, double looping is done to inputted data Item and data Transaction into Conditional Pattern. Line 26 using variable $lString[$iii] to save array of modification of transaction data temporary. Then, $lString[$iii] is trimmed if there’s commas on begin and last sentence. On line 32, query is inserted from data item to modification of data transaction. For next step using temporary table, the process is alike the code above (making Conditional FP-Tree and Frequent Item Set)

After Frequent Pattern 2-Set is found, a code needed to modify a Frequent Pattern 2-Set to Frequent Pattern 3-Set

```
1. foreach ($arrayFrequentItem as $key => $value)
2.         {
3.             $tabel3 = "SELECT * FROM frequentItemSet;";
4.             $itabel3 = mysqli_query($db_link,$tabel3);
5.             while($row=mysqli_fetch_array($itabel3))
6.             {
7.                 if($arrayKodeBarang[$i]==$row['kodeBarang']  &&
  $arrayJumlahItem[$i]== $row['jumlahItem'])
8.                 {
9.                   if($value != $row['frequentItem'])
10.                      {
```

```
11.                         $fp3set         =         $value.",".
   $row['frequentItem'];
12.                         $fp3set        =        ltrim($fp3set,
   $row['kodeBarang']);
13.                         $fp3set = trim($fp3set,',');
14.
15.                         $cmp3set=strcmp($fp3set,$wadahf3S);
16.                         if($cmp3set==1) //gatau pokoke beda 1
   wae
17.                         {
18.                             echo    $fp3set."    sebanyak    ".
   $row['jumlahItem']." Transaksi <br>";
19.
20.                         }
21.                         $wadahf3S=$fp3set;
22.                       }
23.                     }
24.                 }
25.                 $i++;
26.             }
```

Double looping is done to modify Frequent Item 2-Set (shown on line 1 and 5). Then on line 15 using strcmp() to compare two Frequent Item 2-Set that can be used to Frequent Item 3-Set. If the result of strcmp (saved in variable $cmp3set) is 1, the result is printed wih its count. So, Frequent Item 3-Set can be generated.

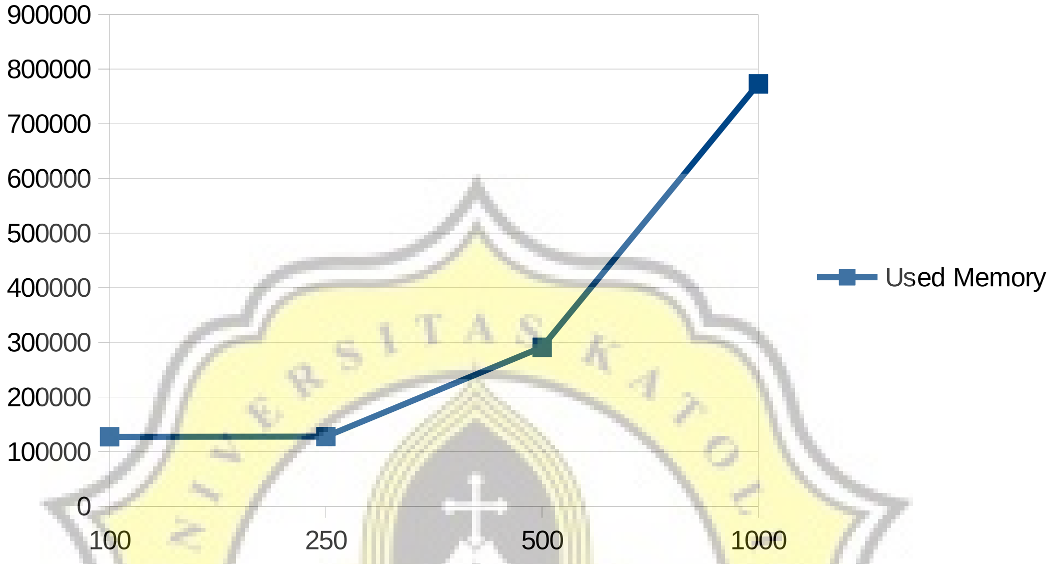
5.2 Testing

For user, this system is done when Frequent Item 2-Set and Frequent Item 3-Set is generated. But, time process and memory usage of the program that are important to analyzed.

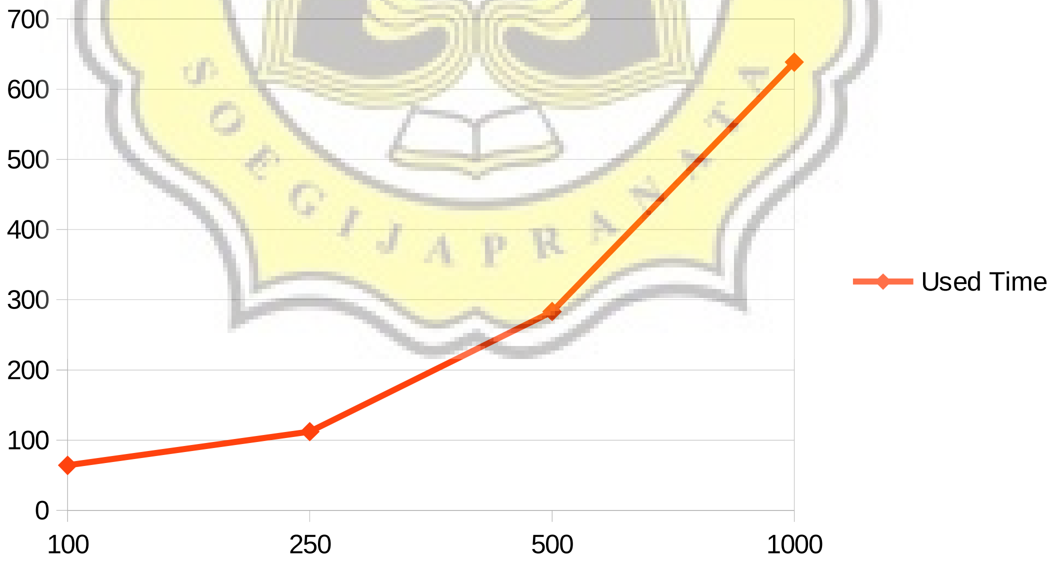
Table 5.1: Table Report System

Data	Used Memory (bytes)	Used Time
100 data	127096	64,078
250 data	127472	112,059
500 data	291108	283,172
1000 data	773216	638,61

As shown on table above, both memory and time are increased when data is bigger. This is when Used Memory shown as chart



And this is Used time table shown as chart.



From table and charts above, it can be concluded that increasing of memory usage and used time is not a linear progression. The more data are processed, memory and time that used are bigger and bigger. From 100 data to 250 data, still there are little changes in time and memory usage. When 250 data is multiplided by 2,to 500 data, time and memory increased by 2 times too. When 500 data is multiplied 2 times, become 1000 data, time and memory usage increase 3 times bigger than before.