The Implementation of Association Rules in Analyzing the Sales of Amigo Group

Tugas Akhir

Diajukan kepada Fakultas Teknik Informatika
Universitas Kristen Duta Wacana
sebagai salah satu syarat dalam memperoleh gelar
Sarjana Komputer

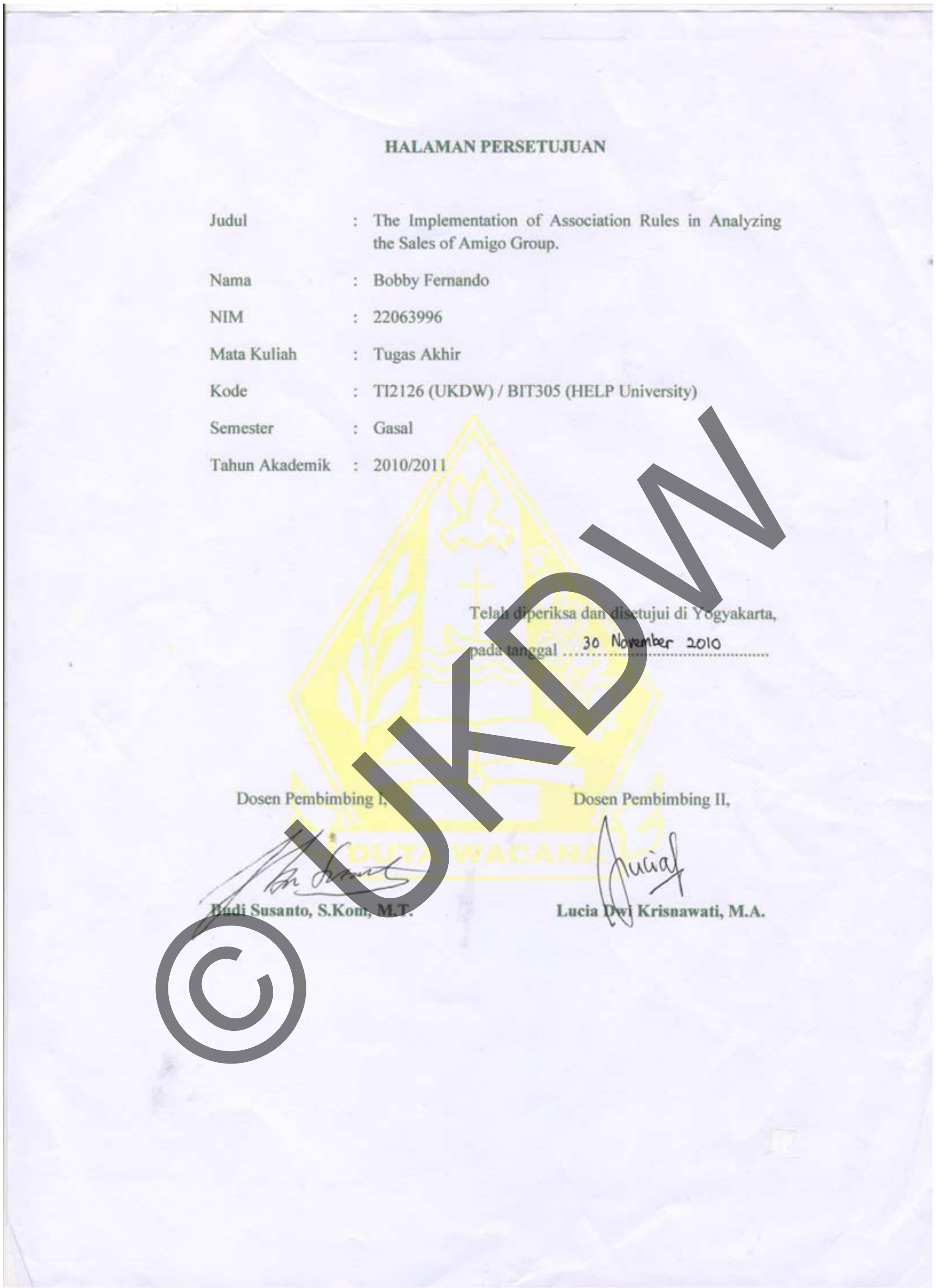
Disusun oleh:
Bobby Fernando
22.06.3996

Program Studi Teknik Informatika Fakultas Teknik
Universitas Kristen Duta Wacana
2010

Abstract

This report discusses the completion of a web-based system to assist final-year Business Information Management undergraduates in working on their Final Year Project II subject. The motivation to learn and work on a data mining project for the first time is underlying the student to do the best on this project. This project is a data mining project for a company called Amigo Group. The problem underlying this project is large numbers of sales transactions data which are always increasing but yet, there is no application that have been utilized to generate valuable information from these data. Another problem is the lack of human resources ability in Amigo's managerial staff to do manual analysis on the sales transactions. Based on these problems, Amigo Group managerial staff needs a data mining system to help their managers understand about customers' buying behavior and pattern of items which are usually purchased together based on sales transactions (data). Hence, the system will produce association rule discovery to help managers do analysis about sales decision and marketing strategy. The system will run on the Local Area Network and it is used internally by the managers of Amigo Group's branches. User requirements are obtained through constant communications with the end users. Technologies used include PHP for the programming language, MySQL for the RDBMS, and CSS for the layout.

In preparation to do this project, the interface design, database design, and also all of the use cases and diagrams which are already completed during the Final Year Project I are already finalized. Therefore, this report focuses more on the implementation and testing phases of the project. However, there are some changes made during the implementation of this Final year Project II. Further explanation about changes and the implementation of the system will be the main focus in this report.



HALAMAN PERSETUJUAN

Judul	:	The Implementation of Association Rules in Analyzing the Sales of Amigo Group.
Nama	:	Bobby Fernando
NIM	:	22063996
Mata Kuliah	:	Tugas Akhir
Kode	:	TI2126 (UKDW) / BIT305 (HELP University)
Semester	:	Gasal
Tahun Akademik	:	2010/2011

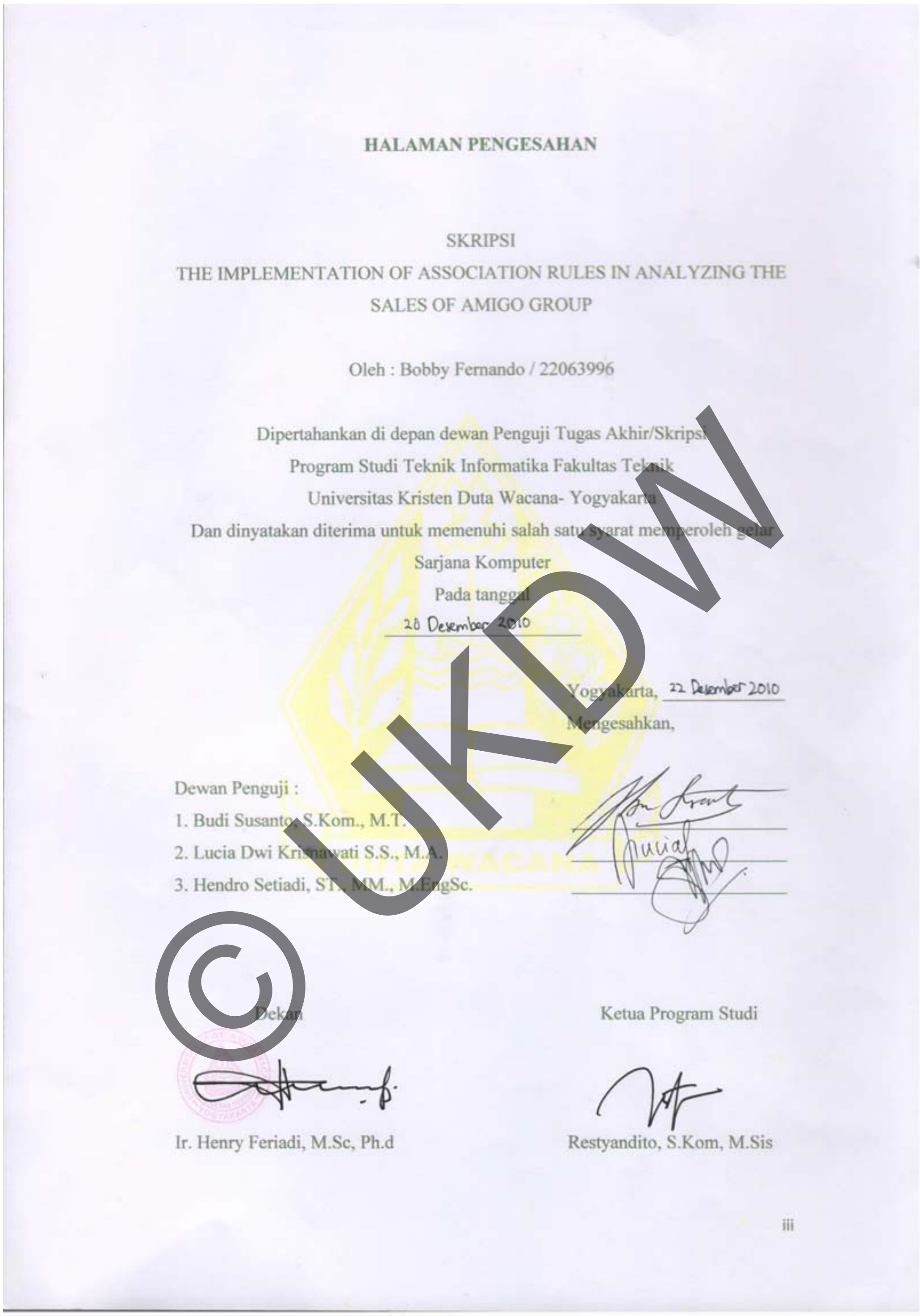
Telah diperiksa dan disetujui di Yogyakarta,
pada tanggal ……30 November 2010……

Dosen Pembimbing I,

Budi Susanto, S.Kom, M.T.

Dosen Pembimbing II,

Lucia Dwi Krisnawati, M.A.



HALAMAN PENGESAHAN

SKRIPSI
THE IMPLEMENTATION OF ASSOCIATION RULES IN ANALYZING THE
SALES OF AMIGO GROUP

Oleh : Bobby Fernando / 22063996

Dipertahankan di depan dewan Penguji Tugas Akhir/Skripsi
Program Studi Teknik Informatika Fakultas Teknik
Universitas Kristen Duta Wacana- Yogyakarta
Dan dinyatakan diterima untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Komputer
Pada tanggal
28 Desember 2010

Yogyakarta, 22 Desember 2010
Mengesahkan,

Dewan Penguji :
1. Budi Susanto, S.Kom., M.T.
2. Lucia Dwi Krisnawati S.S., M.A.
3. Hendro Setiadi, ST., MM., M.EngSc.

Dekan

Ir. Henry Feriadi, M.Sc, Ph.d

Ketua Program Studi

Restyandito, S.Kom, M.Sis

iii

Acknowledgement

This project will not be possible without the help of a number of people. I would like to express my deepest gratitude to my lecturers, Mr Budi Susanto and Mrs Lucia Dwi Krisnawati who have devoted much of their time to supervise me in this project. I also owe my gratitude to Mr Willy Sudiarto Raharjo, Mr Antonius Rachmat and all ‘kaskuser’ who have helped me in problem solving and testing during the implementation of this project.

Special thanks to Jesus Christ, my parents and my girlfriend who always support me with love and kindness in order to finish my Final Year Project.

Thank You.

Table of Contents

List of Table and Figures

Figure

Table

List of Symbols and Abbreviations

AJAX	Asynchronous JavaScript and XML
CEO	Chief Executive Officer
CPU	Central Processing Unit
CS	Creative Suite
CSS	Cascading Style Sheets
CSV	Comma Separated Value
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
GUI	Graphical User Interface
HTML	Hypertext Mark-up Language
IT	Information Technology
MS	Microsoft
PHP	PHP Hypertext Pre-processor
RDBMS	Relational Database Management System
SQL	Structured Query Language
TID	Transaction ID

Abstract

This report discusses the completion of a web-based system to assist final-year Business Information Management undergraduates in working on their Final Year Project II subject. The motivation to learn and work on a data mining project for the first time is underlying the student to do the best on this project. This project is a data mining project for a company called Amigo Group. The problem underlying this project is large numbers of sales transactions data which are always increasing but yet, there is no application that have been utilized to generate valuable information from these data. Another problem is the lack of human resources ability in Amigo's managerial staff to do manual analysis on the sales transactions. Based on these problems, Amigo Group managerial staff needs a data mining system to help their managers understand about customers' buying behavior and pattern of items which are usually purchased together based on sales transactions (data). Hence, the system will produce association rule discovery to help managers do analysis about sales decision and marketing strategy. The system will run on the Local Area Network and it is used internally by the managers of Amigo Group's branches. User requirements are obtained through constant communications with the end users. Technologies used include PHP for the programming language, MySQL for the RDBMS, and CSS for the layout.

In preparation to do this project, the interface design, database design, and also all of the use cases and diagrams which are already completed during the Final Year Project I are already finalized. Therefore, this report focuses more on the implementation and testing phases of the project. However, there are some changes made during the implementation of this Final year Project II. Further explanation about changes and the implementation of the system will be the main focus in this report.

Chapter 1: Proposal

1.1 Project Summary

The project is concerned with the development of a web-based application that can assist a company named Amigo Group to run market basket analysis. The analysis will be based on association rule method that applies to Amigo Group's system. The algorithm used is FP-Growth Algorithm and implemented using PHP script language. This application will be used by the IT administrators of Amigo Group as the administrator of the web and the operational managers of Amigo Group as the web users. The administrator will do user management and has privilege for data cleaning operation from the database. The users will draw analysis by finishing a wizard that will help them go through step by step operation to produce / generate the association result of the transaction data. This web-based application will be accessed internally via Local Area Network and non-privileges users do not have access to enter this system.

1.2 Company Background

This application is specifically developed for the use of Amigo Group's management team. Amigo Group has started their retail business since around 25 years ago. They sell things like clothes, shoes, accessories, caps, belts, and many other daily needs for men, women, and also kids. Their first shop, Bimbo was built in 1976 with only two employees. By the time goes on, their business has grown and new branches were built in several places around Central Java. Nowadays, they have nine stores and located their headquarter on a city called Delanggu, Klaten in Central Java.

Figure 1.1 Amigo Group Logo

After Bimbo was built, there are eight more shops built until 2010, namely Granada, Amigo Klaten, Dinasty Prambanan, Amigo Pedan, Amigo Sukoharjo, Amigo Boyolali, Amigo Wonosari, Amigo Karanganyar. Their business operation is like ordinary retail business and they already have Information System for transactions management and stocks. In order to keep their consistency in retail business, they are also developing systems as they need to manage their transactions. At the first time, Clipper was used as basic of the application until our seniors from Duta Wacana Christian University changed it into a new web-based Information System. Now, Amigo Group's IT staffs are making a data centralization project and it is already on the testing phase.

Amigo Group's organization structure is lead by the General Leader (CEO) Mr Edi Sulistyanto. There are five departments directly under the CEO, Accounting department, Stock department, IT department, Human Resource department and Marketing department. Mr Rubi Prasetyo is the head of IT department who is also the person in charge for this project. He was recommended by Mr Budi Susanto, the IT consultant of Amigo Group which is also one of the writer's supervisor in this Final year Project II.

1.3 Issues with Current System

The problem with the existing system is that there is no data mining techniques implemented in Amigo Group's system yet. Amigo Group's managers and assistants still decide the price and make assumption based on the previous transaction manually (manual system). Their capabilities to do analysis are not very good because most of them are graduates only from senior high school and do not continue to higher education level. If this happen all the time, the analysis won't be accurate enough and the manager will easily forget about their analysis in two or three weeks later, even sometimes, two or three days later. So, they need a data mining system with affinity analysis capabilities.

The needs of analysis tool based on the system are required to help their shop managers decide point of sales of the items, especially when it comes to association analysis. This system needs to give them list of rules that reflect relations of the bought items and the system must be easy to used by the manager.

According to Han & Kamber , association analysis is the discovery of association rules showing attribute-value conditions that occur frequently together in a given set of data [HK01]. Meanwhile, affinity analysis is used to determine the likelihood that a set of items will be bought together is [Gui06].

1.4 Benefits and Constraints of Proposed System

This application is beneficial for administrator and the users (managers). For the managers, they can easily generate the association rules between items by just clicking. They can choose to select event-based result or custom result. Based on the association rules generated by the system, the manager is able to decide what steps should be taken next. To use the program, the managers will be trained by Mr Rubi Prasetyo as the administrator of the system. If there is new manager, the administrator will have to add new account for him/her via the administrator page and train him/her again to operate the application.

For administrator, this application provides a good user management function so the administrator can add, update, and delete the user’s account information directly through the administration page. Because the consideration of this application will be used for local only and for security concerns, new user account must be created manually with agreement from the administrator.

The existing analysis method (manual method) will be replaced with this application. Hopefully, the process will be easier to do and the result will be more accurate. The managers can dramatically reduce time to analyze thousands of transaction notes by entering the data in the database, instead of analyzing it one by one manually. They can also see the result as a graphic, convert it into a CSV format and then print it as they need it so they can easily bring the result anywhere they want. Moreover, this system can be considered as Amigo Group’s competitive advantage among other retail business company.

1.5 Project Description

The end users of the system consist of administrator and the user (manager). The system will be used in Local Area Network only and each user must have an account registered by the administrator to log into the system. This project will be titled as ‘Amigo Data Mining System’ and should be implemented using PHP scripting language for the server side and utilize HTML + JavaScript + CSS for the client side. The Relational Database Management System for this system is MySQL. Algorithm used is the FP-Growth algorithm. The total number of functionalities in the system is 16.

Nine functionalities for the managers (users) are as follows:

- Log into the system
- Edit profile
- Run Wizard
- Display Dashboard
- Display Result
- Search result
- Sort result
- Filter result
- Export result to CSV format

Seven functionalities for the administrator are as follows:

- Log into the system
- Add user
- Delete user
- Edit user
- Edit profile
- Data Cleaning
- Display cleaned data

Amigo groups provide three years data from 2006 - 2008 to be analyzed. For the future, the administrator can add managers via the ‘add user’ function so if there is new manager hired that will be no problem for the administrator to registered him to gain access to the system. The database should be maintained by the administrator too in order to keep the system stable and produce a valid result. In addition, this system will use Bahasa Indonesia (Indonesia Language) in the Graphical User Interface because this system will be used in real system by Amigo’s staff.

Outputs of this project can be shown by the managers (users) through the dashboard page. One of the outputs is a table containing list of association rule between items. The motivation of this project is to create data mining software which can help Amigo Group to rebuild their sales and pricing system by determining association between the items.

The application diagram of the systems are illustrated on the Figure below:

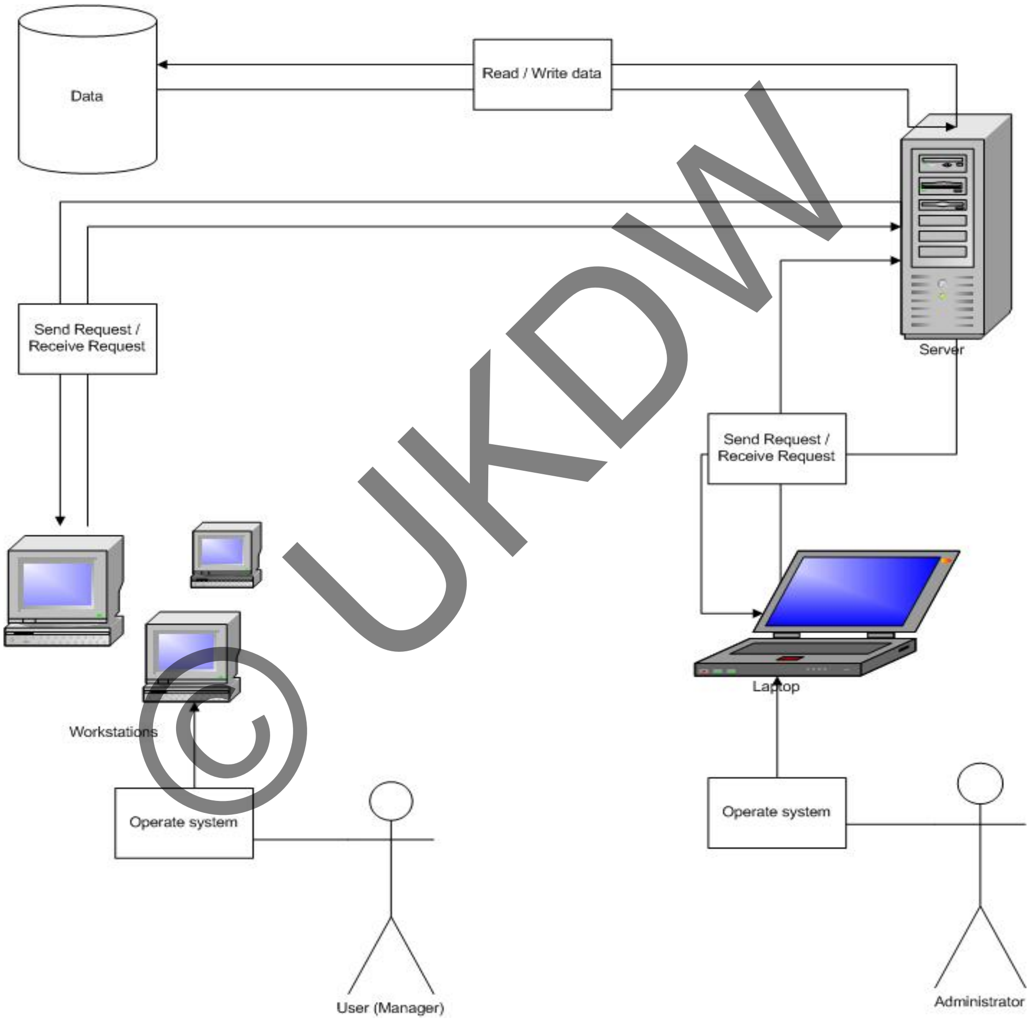


Figure 1.2 Application diagram of the system

1.6 Project Aims

The project's aim is to create a web-based system to help Amigo Group's manager come to the decision about sales by using association rule's application. This project will help the manager gives clear insight about relation between items on the stores. In addition, this project also aims to enlarge the Amigo Group’s IT system by implementing data mining methodologies.

1.7 Project Objectives

The objectives of this project are as follows:

- To help the managers make analysis about sales decision.
- To help the managers combine items in order to increase the sales.
- To help the managers preparing items for special events like Idul Fitri Day (Ramadhan Feast) and Christmas Day.
- To provide on customer shopping pattern analysis for Amigo Group.
- To deliver event-based analysis timely in order to help managers prioritize items sales.
- To enable managers decide better and accurate pricing strategies.
- To understand the association rules between items.
- To implement an association rules algorithm (FP-Growth Algorithm) using web-based scripting language.
- To provide application’s result with good visualization and easy to understand form.
- To keep good relationship between Duta Wacana Christian University Student and Amigo Group.
- To complete and finish writer’s Final Year Project II.

1.8 Project Scope

The agreeable scope of the project involves developing a web-based data mining system that covers a total of 16 use cases. Those use cases serve the administrator and the user (manager) of Amigo Groups’ management staff. The total time available for writer to finish this project is about one year. The writer’s supervisor advised to use three years data from 2006 until 2008 transactions data from one of their shop, in this case, Bimbo was chosen. The data itself needs to be cleaned with writer’s own method because the data still classified as dirty data. To be more attractive and interactive, a simple dashboard will be built for the manager’s operation with prerequisite of a table and a chart.

1.9 Software and Hardware Requirements

Hardware requirement:

- Laptop (with at least Dual Core processors, 1 GB of RAM, and 40GB of Hard drive space for the convenience of using the application)
- Back up drive

Software requirement:

- XamPP 1.7.3 (including PHP 5.3.1, MySQL 5.1.41 and Apache 2.2.14)
- Star UML
- MySQL Workbench 5.1
- Microsoft Visio 2003
- Microsoft Project 2007
- Webyog SQLyog Ultimate 8.5
- MockUp Screens 4.24 Trial version
- Editor (Notepad ++ or Adobe Dreamweaver)
- Adobe Photoshop CS5
- Corel Draw X5

1.10 Development Methodology

The system development methodology implemented for this project is prototyping. Prototyping is the process of building a model of a system. In terms of an information system, prototypes are employed to help system designers build an information system that is intuitive and easy to manipulate for end users [Ake99]. In addition, prototyping is suitable when user's feedback is easy to obtained. In this project, the end-user feedback is easy to obtain because the end-user already has good relation with developer's working environment.

Prototyping is chosen with a reason that the user requirements can change dynamically. With prototyping, the cycle will do iteratively, so the requirement changes will not be a problem because it will be reviewed at every cycle [Com07]. This methodology also reduce work load for the developer and enable faster feedback from end-users. The approach will be used for this project is called 'incremental approach'. For both developer and the client, incremental prototype has some advantages. Like stated in [Com07], one of the advantages from incremental approach is that the client and/or end-users have the opportunity to test the developed components and their functionality. They also have opportunities to provide feedback while other components are still in development, and can thus influence the outcome of further development [Com07].

1.12 Project Plan

All of the task planned are already identified in the Appendix C of the writer's Final Year Project I report. There are some changes occur at the actual plan in this Final Year Project II implementation stage. The changes and updates of the Gant Chart can be seen on the Appendix B in the end of this project report. Also, please refer to Appendix D for the Work Items List.

1.13 Vision

Please refer to Appendix C for the updated Vision document.

Chapter 5: Implementation and Testing

5.1 Introduction

This chapter will explain about the implementation of the use cases during the implementation stage of this project. Implementation methodology, techniques used to do programming (data cleaning and FP-Growth implementation) and testing phase to achieve project completion are the main focuses described in this chapter.

At the end of this stage, the final release which is completed after prototype V, will cover all of the use cases which were built based on the user requirement. Two use cases have been implemented in Final Year Project I. The rest of 14 use cases will be implemented regularly in every deployment of the prototype. The developed use cases are tested by our supervisor, client, and also friends and lecturers from Duta Wacana Christian University to ensure that the system runs without errors.

5.2 Implementation Methodology

The implementation strategy used is prototyping. The writer uses incremental type of prototyping which has advantage of common feedback from the client / end-user. While producing feedback, the writer may develop the other components of the system.

The programming language used is a scripting language, PHP. Edited CSS from other designer is used to create a new layout and JQuery is used to make the interface become more attractive. Apache web server is running on the local computer as the web server for the web application to be developed. MySQL is installed as the RDBMS of the application. In addition, the installation of a software named ‘SQLyog’ is suggested because the software contains acceleration engine for MySQL. By installing it, the query process will be faster than the bundled MySQL installation in XAMPP software.

Explanations about the implementations, screenshots, and also problems faced during the implementation stage of prototype II until the final release of the prototype can be read in the following section.

5.2.1 Project preparation

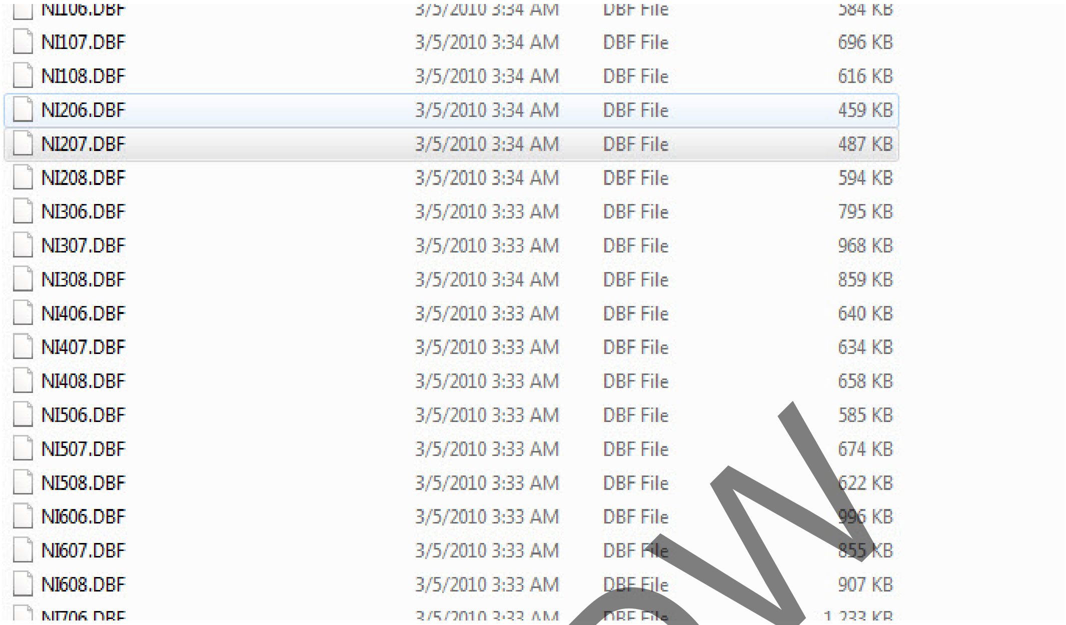
To start the implementation of the project, first thing to do is to make sure that the project planning has been completed and clearly understood. This steps can be done through reviewing the project plan which already made in Final Year Project I. By reviewing the previous studies, the writer can acknowledge some mistakes and consider changes that should be made during Final Year Project II implementation. The next step is doing more research for the literature, algorithm supporting the project and then decide the algorithm that should be used based on the knowledge that has already learned. Considering better time process and the effectiveness of the algorithm (rather than Apriori algorithm), FP-Growth algorithm will be the chosen algorithm for this project. Then, writer must search the necessary changes and updates for use cases, diagrams, and also think about new database design (if necessary). Having completed those steps, writer should install the software needed to develop this application.

5.2.2 Prototype II

First thing to do in this prototype is to create database and its tables in MySQL. The writer uses SQLyog software to facilitate easier database actions. This prototype contains two big parts, first is the data preparation and second is the implementation of data cleaning techniques.

5.2.2.1 Data Preparation

While waiting ‘lini’ table delivered from Amigo’s IT staff, the writer prepares the 2006 – 2008 transactions table and detail transactions table which already sent during the BIT304. Both of them have .dbf extension and each of them consist of 36 files (1 file means 1 month). Total of 72 .dbf files must be combined and formatted into two files called nidump.csv and ntdump.csv. Some of these 72 files can be seen on Figure 5.1.



NI106.DBF	3/5/2010 3:34 AM	DBF File	584 KB
NI107.DBF	3/5/2010 3:34 AM	DBF File	696 KB
NI108.DBF	3/5/2010 3:34 AM	DBF File	616 KB
NI206.DBF	3/5/2010 3:34 AM	DBF File	459 KB
NI207.DBF	3/5/2010 3:34 AM	DBF File	487 KB
NI208.DBF	3/5/2010 3:34 AM	DBF File	594 KB
NI306.DBF	3/5/2010 3:33 AM	DBF File	795 KB
NI307.DBF	3/5/2010 3:33 AM	DBF File	968 KB
NI308.DBF	3/5/2010 3:34 AM	DBF File	859 KB
NI406.DBF	3/5/2010 3:33 AM	DBF File	640 KB
NI407.DBF	3/5/2010 3:33 AM	DBF File	634 KB
NI408.DBF	3/5/2010 3:33 AM	DBF File	658 KB
NI506.DBF	3/5/2010 3:33 AM	DBF File	585 KB
NI507.DBF	3/5/2010 3:33 AM	DBF File	674 KB
NI508.DBF	3/5/2010 3:33 AM	DBF File	622 KB
NI606.DBF	3/5/2010 3:33 AM	DBF File	996 KB
NI607.DBF	3/5/2010 3:33 AM	DBF File	855 KB
NI608.DBF	3/5/2010 3:33 AM	DBF File	907 KB
NI706.DBF	3/5/2010 3:33 AM	DBF File	1,233 KB

Figure 5.1 Total of 72 .dbf files must be combined into two CSV files.

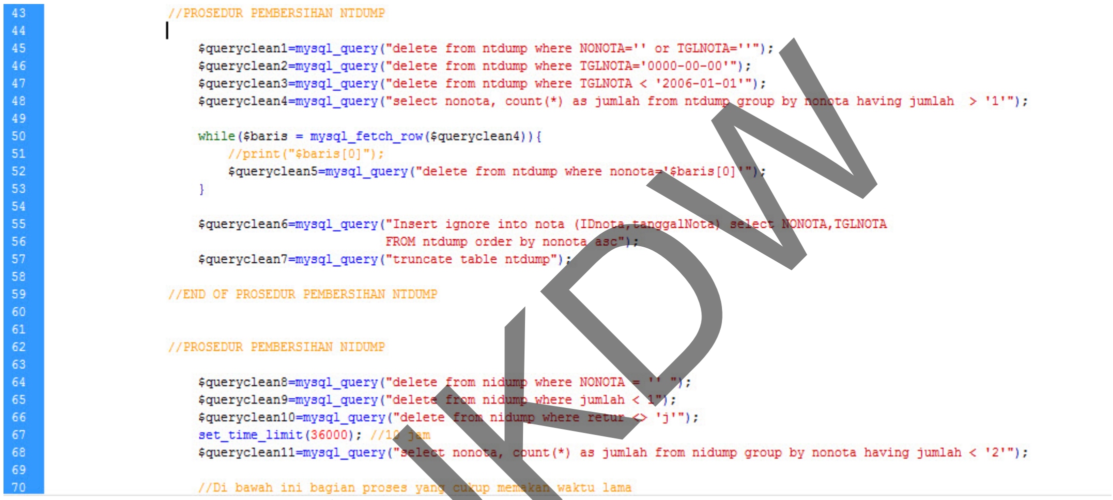
‘Ntdump.csv’ contains the dirty data of transactions combined from 36 files which have name prefixed with ‘NTxxx’. ‘Nidump.csv’ contains the dirty data of detailed transactions combined from 36 files which have name prefixed with ‘NIxxx’.

To combine 72 files into ‘ntdump.csv’ and ‘nidump.csv’, the writer uses Microsoft Access and Microsoft Excel. The file saved into CSV format and then inserted into ‘nidump’ table and ‘ntdump’ table in the database. The MySQL database name for this project is ‘amigomining’. After several days, the ‘lini’ table which contains items ID and items name data is received and then converted into ‘lini’ table in MySQL database. This mean all of the data needed to do this project has been inserted into the database and ready to be cleaned.

5.2.2.2 Data cleaning

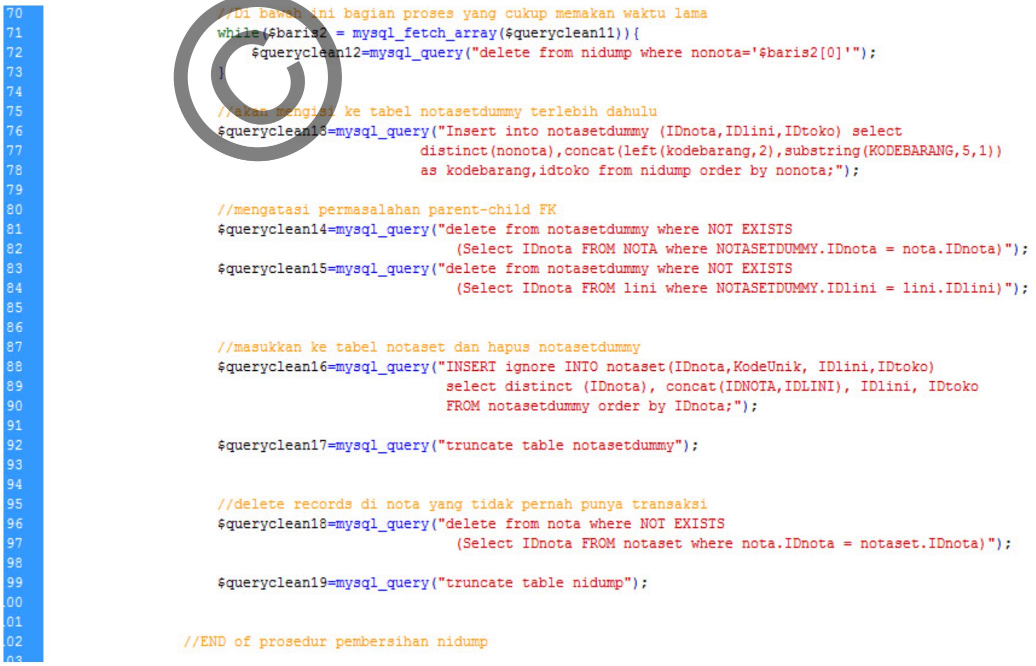
After having consultation with supervisor, the writer define the cleaning rules will be used to clean the prepared dirty data in ‘nidump’ table and ‘ntdump’ table. MySQL queries through PHP pages seems to be the best techniques to clean the data. The explanation about cleaning rules will be explain soon in this section.

To clean data, a user must log into the system with Administrator privileges and click the ‘Data cleaning’ page in the link menu. Clicking and confirm the action will bring the administrator into ‘cleaningresult.php’ and he will informed that the data cleaning process has been done. Actually, the data cleaning process is done through this ‘cleaningresult.php’ page. Figure 5.2 and Figure 5.3 show the implemented code in ‘cleaningresult.php’.



```
//PROSEDUR PEMBERSIHAN NTDUMP

    $queryclean1=mysql_query("delete from ntdump where NONOTA='' or TGLNOTA=''");
    $queryclean2=mysql_query("delete from ntdump where TGLNOTA='0000-00-00'");
    $queryclean3=mysql_query("delete from ntdump where TGLNOTA < '2006-01-01'");
    $queryclean4=mysql_query("select nonota, count(*) as jumlah from ntdump group by nonota having jumlah  > '1'");

    while($baris = mysql_fetch_row($queryclean4)){
        //print("$baris[0]");
        $queryclean5=mysql_query("delete from ntdump where nonota='$baris[0]'");
    }

    $queryclean6=mysql_query("Insert ignore into nota (IDnota,tanggalNota) select NONOTA,TGLNOTA
                             FROM ntdump order by nonota asc");
    $queryclean7=mysql_query("truncate table ntdump");

//END OF PROSEDUR PEMBERSIHAN NTDUMP


//PROSEDUR PEMBERSIHAN NIDUMP

    $queryclean8=mysql_query("delete from nidump where NONOTA = '' ");
    $queryclean9=mysql_query("delete from nidump where jumlah < 1");
    $queryclean10=mysql_query("delete from nidump where retur <> 'j'");
    set_time_limit(36000); //10 jam
    $queryclean11=mysql_query("select nonota, count(*) as jumlah from nidump group by nonota having jumlah < '2'");

    //Di bawah ini bagian proses yang cukup memakan waktu lama
```

Figure 5.2 Ntdump cleaning query and part of nidump cleaning query



```
    //Di bawah ini bagian proses yang cukup memakan waktu lama
    while($baris2 = mysql_fetch_array($queryclean11)){
        $queryclean12=mysql_query("delete from nidump where nonota='$baris2[0]'");
    }

    //akan mengisi ke tabel notasetdummy terlebih dahulu
    $queryclean13=mysql_query("Insert into notasetdummy (IDnota,IDlini,IDtoko) select
                            distinct(nonota),concat(left(kodebarang,2),substring(KODEBARANG,5,1))
                            as kodebarang,idtoko from nidump order by nonota;");

    //mengatasi permasalahan parent-child FK
    $queryclean14=mysql_query("delete from notasetdummy where NOT EXISTS
                               (Select IDnota FROM NOTA where NOTASETDUMMY.IDnota = nota.IDnota)");
    $queryclean15=mysql_query("delete from notasetdummy where NOT EXISTS
                               (Select IDnota FROM lini where NOTASETDUMMY.IDlini = lini.IDlini)");


    //masukkan ke tabel notaset dan hapus notasetdummy
    $queryclean16=mysql_query("INSERT ignore INTO notaset(IDnota,KodeUnik, IDlini,IDtoko)
                              select distinct (IDnota), concat(IDNOTA,IDLINI), IDlini, IDtoko
                              FROM notasetdummy order by IDnota;");

    $queryclean17=mysql_query("truncate table notasetdummy");


    //delete records di nota yang tidak pernah punya transaksi
    $queryclean18=mysql_query("delete from nota where NOT EXISTS
                               (Select IDnota FROM notaset where nota.IDnota = notaset.IDnota)");

    $queryclean19=mysql_query("truncate table nidump");


  //END of prosedur pembersihan nidump
```

Figure 5.3 Nidump cleaning query

First, let's take a look on the cleaning method for 'ntdump' table.

```
$queryclean1=mysql_query("delete from ntdump where NONOTA='' or
TGLNOTA=''");

$queryclean2=mysql_query("delete from ntdump where TGLNOTA='0000-00-
00'");

$queryclean3=mysql_query("delete from ntdump where TGLNOTA < '2006-
01-01'");

$queryclean4=mysql_query("select nonota, count(*) as jumlah from
ntdump group by nonota having jumlah  > '1'");

while($baris = mysql_fetch_row($queryclean4))
{
      $queryclean5=mysql_query("delete from ntdump where
nonota='$baris[0]'");
}

$queryclean6=mysql_query("Insert ignore into nota
                        (IDnota,tanggalNota)
                        select NONOTA,TGLNOTA  FROM ntdump order by
                        nonota asc");

$queryclean7=mysql_query("truncate table ntdump");
```

The code snippet above reflects queries to clean the 'ntdump' table:

- $queryclean1 → delete data in 'ntdump' where the TID (Transaction ID) is blank and delete data in 'ntdump' where the transaction date is blank.
- $queryclean2 → delete data in 'ntdump' where transaction date is '0000-00-00'
- $queryclean3 → delete data in 'ntdump' where transaction date occurs before '2006-01-01'

- $queryclean4 → select TID and its count (amount) group by the TID in ‘ntdump’ where count is more than 1.
- Based on the select on $queryclean4, the data will be fetched and cleaned using $queryclean5 → delete data in ‘ntdump’ where TID = TID fetched by $queryclean4. This step’s purpose is to clean multiple TID.
- $queryclean6 → insert all the ‘ntdump’ data in column TID and Transaction date (the real columns name are NONOTA and TGLNOTA) into ‘nota’ table.
- $queryclean7 → truncate data in table ‘ntdump’.

Now the system has clean data in ‘nota’ table. Next, let’s take a look on the cleaning method for ‘nidump’ table.

```
$queryclean8=mysql_query("delete from nidump where NONOTA = '' ");

$queryclean9=mysql_query("delete from nidump where jumlah < 1");

$queryclean10=mysql_query("delete from nidump where retur <> 'j'");

set_time_limit(36000); //10 jam
$queryclean11=mysql_query("select nonota, count(*) as jumlah from
nidump group by nonota having jumlah < '2'");

//Di bawah ini bagian proses yang cukup memakan waktu lama
while($baris2 = mysql_fetch_array($queryclean11))
{
      $queryclean12=mysql_query("delete from nidump where
      nonota='$baris2[0]'");
}

//akan mengisi ke tabel notasetdummy terlebih dahulu
$queryclean13=mysql_query("Insert into notasetdummy
            (IDnota,IDlini,IDtoko) select
            distinct(nonota),concat(left(kodebarang,2),substring(KOD
            EBARANG,5,1) as kodebarang,idtoko from nidump order by
            nonota;");
```

```
//mengatasi permasalahan parent-child FK
$queryclean14=mysql_query("delete from notasetdummy where NOT EXISTS
            (Select IDnota FROM NOTA where NOTASETDUMMY.IDnota =
            nota.IDnota)");

$queryclean15=mysql_query("delete from notasetdummy where NOT EXISTS
            (Select IDnota FROM lini where NOTASETDUMMY.IDlini =
            lini.IDlini)");


//masukkan ke tabel notaset dan hapus notasetdummy
$queryclean16=mysql_query("INSERT ignore INTO
      notaset(IDnota,KodeUnik, IDlini,IDtoko)
      select distinct (IDnota),concat(IDNOTA,IDLINI),IDlini, IDtoko
      FROM notasetdummy order by IDnota;");


$queryclean17=mysql_query("truncate table notasetdummy");

//delete records di nota yang tidak pernah punya transaksi
$queryclean18=mysql_query("delete from nota where NOT EXISTS
            (Select IDnota FROM notaset where nota.IDnota =
            notaset.IDnota)");

$queryclean19=mysql_query("truncate table nidump");
```

The code snippet above reflects queries to clean the ‘nidump’ table:

- $queryclean8 → delete data in ’nidump’ where the TID (Transaction ID) is blank.
- $queryclean9 → delete data in ‘nidump’ where value in column jumlah less than one.
- $queryclean10 → delete data in ‘nidump’ where value in column retur is not ‘j’.
- $queryclean11 → select TID and its count (amount) group by the TID in ‘nidump’ where count is less than 2.

- Based on the select on $queryclean11, the data will be fetched and cleaned using $queryclean12 → delete data in 'ntdump' where TID = TID fetched by $queryclean11. This step's purpose is to clean TID which only has one item because it cannot be used to generate association rule. This process will take long enough time because there are so many records to be checked and cleaned during this processes. In the real implementation time, more than 3 hours are needed to clean three years data (about 413.813 records).
- $queryclean13 → insert all the 'nidump' data in column distinct (TID), concat (left(TID,2),substring(TID,5,1) and shop ID (the real columns name are nonota, kodebarang, and idtoko) into 'notasetdummy' table. The concat part takes value from the first two alphabets in 'kodebarang' column and concatenate it with fifth letter from 'kodebarang' column. The format of the original kodebarang (TID) from Amigo is not the same with the TID format that is used in this system. For the example, ICMFE9304125 means IC as lini product, MF is the supplier, E as depth, 9 as brand, 304 is the real code, 12 is month, and 5 means year of the item.
- $queryclean14 → delete the data in 'notasetdummy' where the TID is not exist in 'nota' table's TID.
- $queryclean15 → delete the data in 'notasetdummy' where the lini ID is not exist in 'lini' table's lini ID.
- $queryclean16 → insert all the 'notasetdummy' data in column distinct (TID), concat (TID, Lini ID), lini ID and shop ID (the real columns name are IDNOTA, concat (IDNOTA, IDLINI), IDLINI, and IDtoko) into 'notaset' table.
- $queryclean17 → truncate data in table 'notasetdummy'.
- After having a cleaned data in 'notaset' table, now the system must check once again in table 'nota' to delete TID in 'nota' table where not exists in 'notaset' table's TID. It is done through $queryclean18.
- $queryclean19 → truncate data in table 'nidump'.

After $queryclean19 has been executed, the system has complete the cleaning process. Now, it has clean data in 'nota' and 'notaset' table. These two tables will be processed during the implementation of FP-Growth algorithm later.

This is the end of prototype II. The writer backups this prototype in the local storage of his laptop and keeps it available on the other external storage devices too. The last part of this prototype will be testing and evaluation which will be described in section 5.4. There are no significant errors occur during this testing.

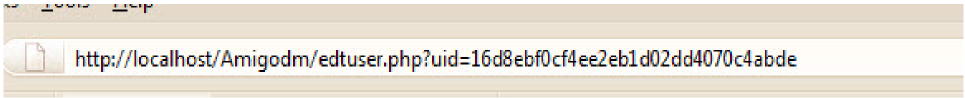
5.2.3 Prototype III

First thing to do in this prototype is to fix previous bugs realized in the testing phase of prototype II. After bugs are fixed, the writer searches for visualization method to display the data in the 'nota' and 'notaset' table. A JQuery plug-in called datatables can be the solution. A modified datatables combined with AJAX implementation is a better visualization method rather than displaying HTML table. It has advantage of some features like search box, sort arrow, pagination, and also total records information. The interface of this datatables can be seen on Figure 4.14.

After decision has reached in visualization method, the writer then continue to work on the administrator's page on the system. First, edit the HTML text and other components on the administrator's homepage. There are no difficulty in this step because it has to do only with static HTML text.

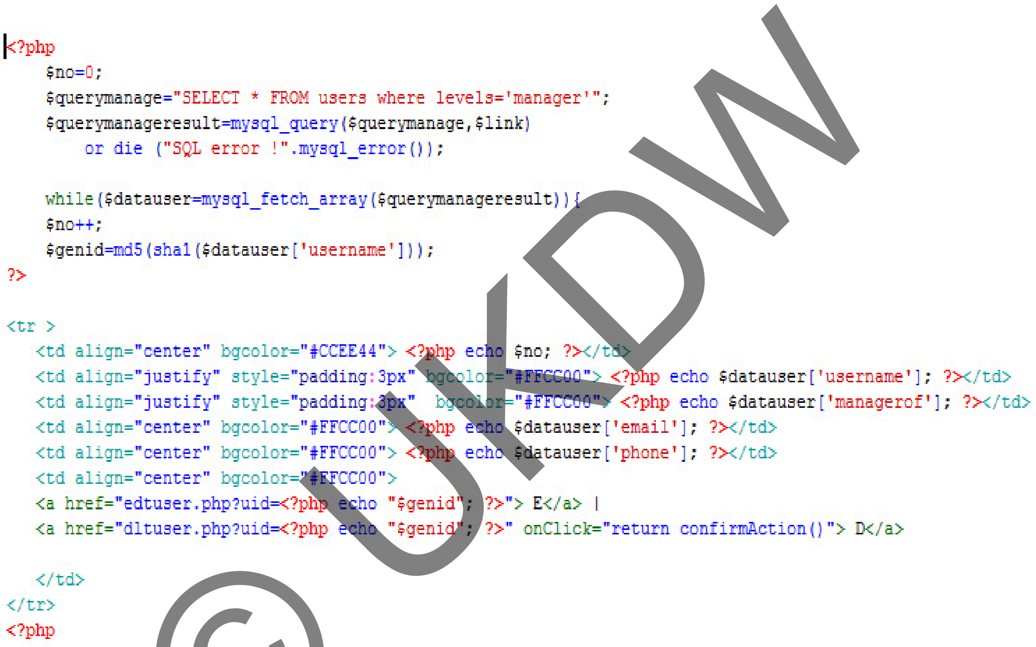
Second, create the 'manage user' page and implement code in it. Figure 5.5 shows the implementation code of 'mnguser.php'. Figure 5.5 shows that manager's account is displayed from the query result of `Select * from users where levels='manager'`. This query being fetched and displayed in a HTML table, so that the administrator can select action whether he wants to edit user (edtuser.php) or delete user (dltuser.php). The variable $genid is an id that hashed using SHA1 and then MD5-hash to send unique identifier (uid) to 'edtuser.php' or 'dltuser.php' page.

Figure 5.4 below is the example of the URL generated if we click username ‘bobby’ from the manage user page. ‘bobby’ is hashed with SHA1 and then hashed again with MD5 until the string becomes *16d8ebf0cf4ee2eb1d02dd4070c4abde*. This value sent in the URL to avoid URL injection and manipulation from the naughty user.



http://localhost/Amigodm/edtuser.php?uid=16d8ebf0cf4ee2eb1d02dd4070c4abde

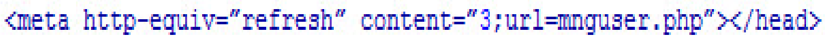
Figure 5.4 uid generated in the URL



```
<?php
    $no=0;
    $querymanage="SELECT * FROM users where levels='manager'";
    $querymanageresult=mysql_query($querymanage,$link)
        or die ("SQL error !".mysql_error());

    while($datauser=mysql_fetch_array($querymanageresult)){
    $no++;
    $genid=md5(sha1($datauser['username']));
?>

<tr >
   <td align="center" bgcolor="#CCEE44"> <?php echo $no; ?></td>
   <td align="justify" style="padding:3px" bgcolor="#FFCC00"> <?php echo $datauser['username']; ?></td>
   <td align="justify" style="padding:3px"  bgcolor="#FFCC00"> <?php echo $datauser['managerof']; ?></td>
   <td align="center" bgcolor="#FFCC00"> <?php echo $datauser['email']; ?></td>
   <td align="center" bgcolor="#FFCC00"> <?php echo $datauser['phone']; ?></td>
   <td align="center" bgcolor="#FFCC00">
   <a href="edtuser.php?uid=<?php echo "$genid"; ?>"> E</a> |
   <a href="dltuser.php?uid=<?php echo "$genid"; ?>" onClick="return confirmAction()"> D</a>

   </td>
</tr>
<?php
```

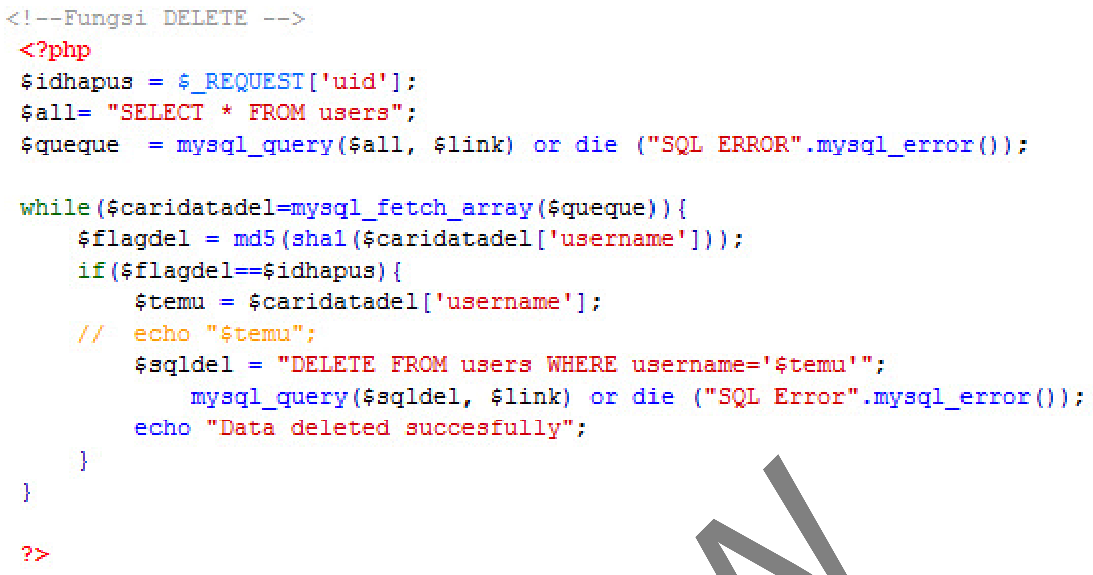
Figure 5.5 Manage user implementation

A confirmation box will appear if we click the page to ‘dltuser.php’ because in ‘dltuser.php’, the page will refresh automatically and redirected to display the manager’s account again. Figure 5.6 shows the HTML header used to do this function.



```
<meta http-equiv="refresh" content="3;url=mnguser.php"></head>
```

Figure 5.6 HTML header to refresh the page and redirect to mnguser.php



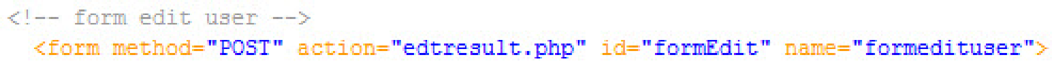
```
<!--Fungsi DELETE -->
<?php
$idhapus = $_REQUEST['uid'];
$all= "SELECT * FROM users";
$queque  = mysql_query($all, $link) or die ("SQL ERROR".mysql_error());

while($caridatadel=mysql_fetch_array($queque)){
    $flagdel = md5(sha1($caridatadel['username']));
    if($flagdel==$idhapus){
        $temu = $caridatadel['username'];
    //  echo "$temu";
        $sqldel = "DELETE FROM users WHERE username='$temu'";
            mysql_query($sqldel, $link) or die ("SQL Error".mysql_error());
        echo "Data deleted succesfully";
    }
}

?>
```

Figure 5.7 Code implementation of dltuser.php

This is the explanation of code above :

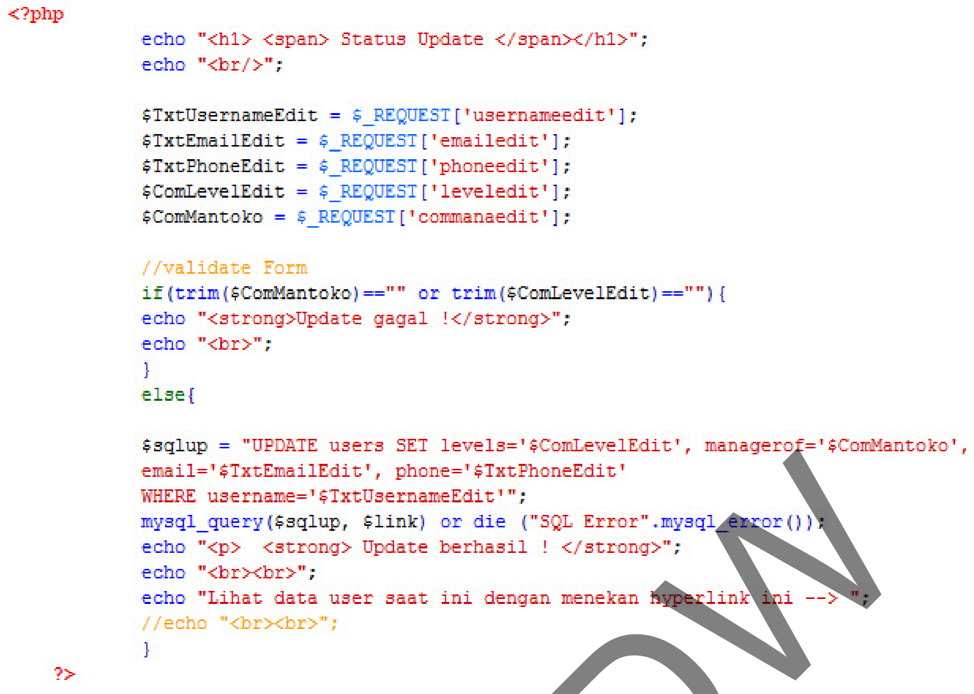
- $idhapus → receive the uid.
- $all → select all data in table users.
- $queque → check $all status query.
- $caridatadel → fetch all data from $queque query with array.
- While data is being fetched, do this every loop:
    - $flagdel → compare uid with the username.
    - If found (same) then set the username in the $temu variable.
    - $sqldel → delete from table users where the username is $temu value.
- Data deleted successfully ☺

The same thing happen when ‘edtuser.php’ is opened. The uid will be received and a form containing the specific data of the related username will displayed. The form itself can be seen on the previous section in Figure 4.9. This form will take action into ‘edtresult.php’ to return status whether the edit action was success or not.



```
<!-- form edit user -->
  <form method="POST" action="edtresult.php" id="formEdit" name="formedituser">
```

Figure 5.8 Form action in edtuser.php



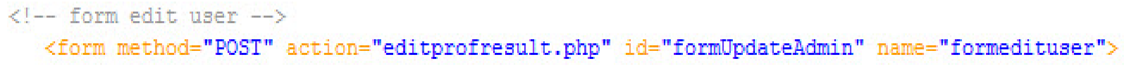
```
<?php
            echo "<h1> <span> Status Update </span></h1>";
            echo "<br/>";

            $TxtUsernameEdit = $_REQUEST['usernameedit'];
            $TxtEmailEdit = $_REQUEST['emailedit'];
            $TxtPhoneEdit = $_REQUEST['phoneedit'];
            $ComLevelEdit = $_REQUEST['leveledit'];
            $ComMantoko = $_REQUEST['commanaedit'];

            //validate Form
            if(trim($ComMantoko)=="" or trim($ComLevelEdit)==""){
            echo "<strong>Update gagal !</strong>";
            echo "<br>";
            }
            else{

            $sqlup = "UPDATE users SET levels='$ComLevelEdit', managerof='$ComMantoko',
            email='$TxtEmailEdit', phone='$TxtPhoneEdit'
            WHERE username='$TxtUsernameEdit'";
            mysql_query($sqlup, $link) or die ("SQL Error".mysql_error());
            echo "<p>  <strong> Update berhasil ! </strong>";
            echo "<br><br>";
            echo "Lihat data user saat ini dengan menekan hyperlink ini --> ";
            //echo "<br><br>";
            }
    ?>
```

Figure 5.9 Code implementation of edtresult.php

This is the explanation of code above :

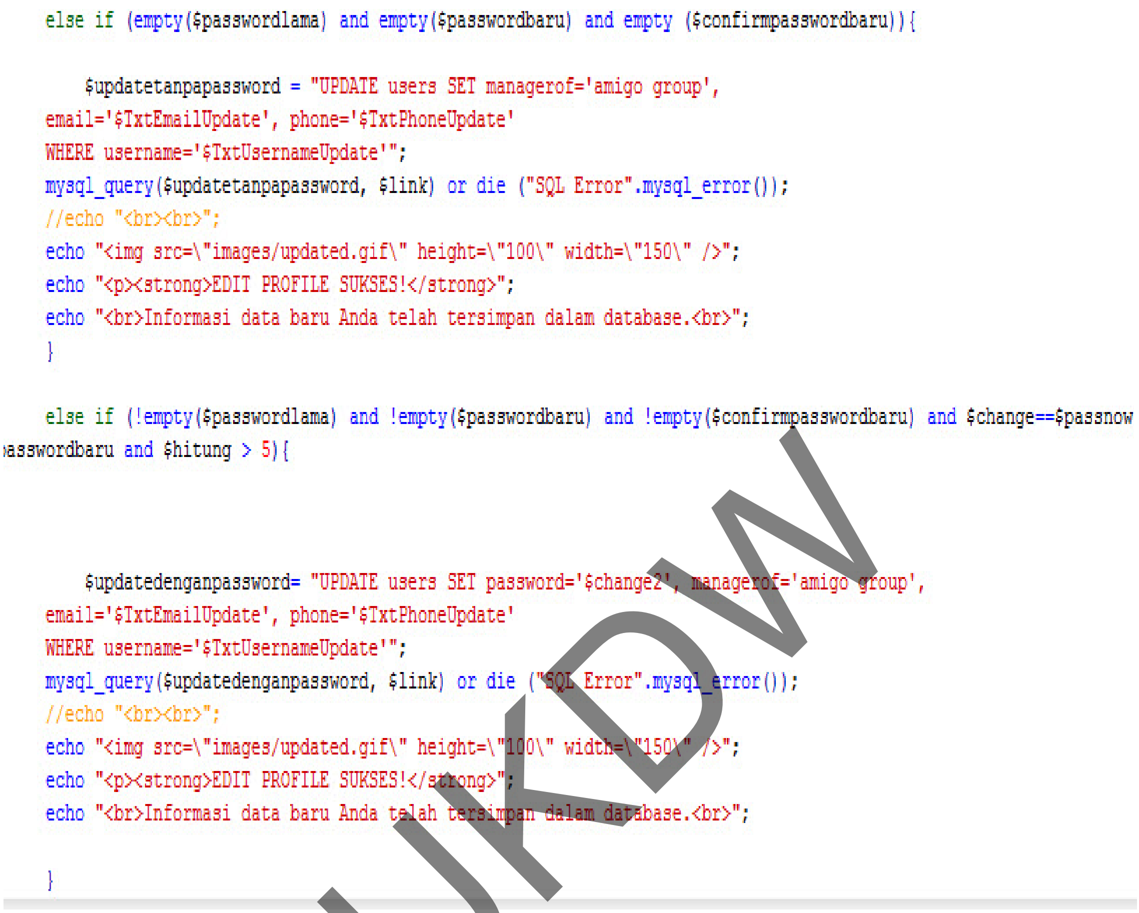
- Form values were received by the variables ($TxtUsernameEdit, $TxtEmailEdit, $TxtPhoneEdit, $ComLevelEdit, $ComMantoko)
- If ‘level’ and ‘manager of’ are not blank, then it will do the $sqlup query.
- $sqlup → update data in table users set levels, managerof, email, and phone into values that were received by the proper variables where the username is the username value in $TxtUsernameEdit.
- Data updated successfully

Third thing to do is to create edit profile page in the administrator’s account. This page can be seen on the previous section in Figure 4.15. This form will take action into ‘edtprofresult.php’ to return status whether the edit profile action was success or not.



```
<!-- form edit user -->
   <form method="POST" action="editprofresult.php" id="formUpdateAdmin" name="formedituser">
```

Figure 5.10 Form action in editprof.php

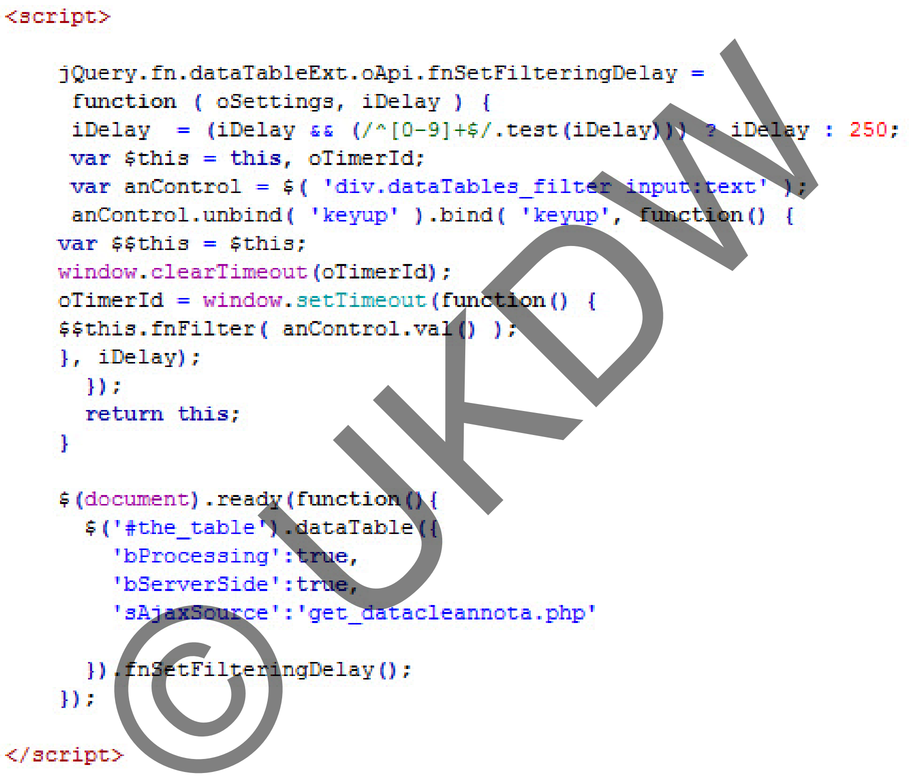


```
    else if (empty($passwordlama) and empty($passwordbaru) and empty ($confirmpasswordbaru)){

        $updatetanpapassword = "UPDATE users SET managerof='amigo group',
    email='$TxtEmailUpdate', phone='$TxtPhoneUpdate'
    WHERE username='$TxtUsernameUpdate'";
    mysql_query($updatetanpapassword, $link) or die ("SQL Error".mysql_error());
    //echo "<br><br>";
    echo "<img src=\"images/updated.gif\" height=\"100\" width=\"150\" />";
    echo "<p><strong>EDIT PROFILE SUKSES!</strong>";
    echo "<br>Informasi data baru Anda telah tersimpan dalam database.<br>";
    }

    else if (!empty($passwordlama) and !empty($passwordbaru) and !empty($confirmpasswordbaru) and $change==$passnow
asswordbaru and $hitung > 5){



        $updatedenganpassword= "UPDATE users SET password='$change2', managerof='amigo group',
    email='$TxtEmailUpdate', phone='$TxtPhoneUpdate'
    WHERE username='$TxtUsernameUpdate'";
    mysql_query($updatedenganpassword, $link) or die ("SQL Error".mysql_error());
    //echo "<br><br>";
    echo "<img src=\"images/updated.gif\" height=\"100\" width=\"150\" />";
    echo "<p><strong>EDIT PROFILE SUKSES!</strong>";
    echo "<br>Informasi data baru Anda telah tersimpan dalam database.<br>";

    }
```

Figure 5.11 Code implementation part of edtprofresult.php

Figure 5.11 is part of code implementation in edit profile result page. If old password, new password and confirm new password were successfully matched, the system will do $updatedenganpassword query to update the data in table users. But if the passwords are blank or empty, the system will do $updatetanpapassword query to update the data in table users.

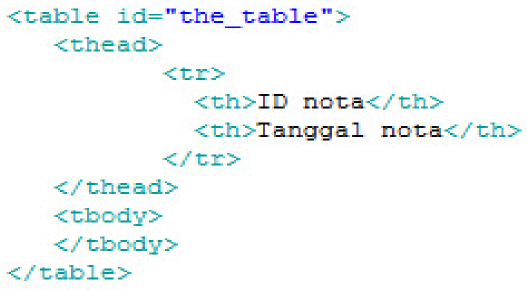
The last page to complete the administrator's account is 'Table data' page which will display hyperlink into pages that containing cleaned data ('tabelnota.php' and 'tabelnotaset.php'). These two pages are basically similar to each other. The difference is just the query used to display the table. The 'tabelnota.php' page can be seen on the previous section in Figure 4.14. This page is using datatables JQuery to display the data. The original datatables mechanism is to send whole data from the query executed in JSON format. This is not a good idea when it comes to data mining because usually it deals with large data. The solution to this problem is to modify the functions in the source data into AJAX format. The idea is to send only 10 data in the beginning of the query, not the whole data. With help from [Ver09] the page is successfully implemented. The ‘tabelnota.php’ is the page to display the datatables itself, but it also needs the source data page. In this case, ‘get_datacleannota.php’ is the source page of ‘tabelnota.php’. It has to be declared on the head section of ‘tabelnota.php’ HTML.



```
<script>

    jQuery.fn.dataTableExt.oApi.fnSetFilteringDelay =
     function ( oSettings, iDelay ) {
     iDelay  = (iDelay && (/^[0-9]+$/.test(iDelay))) ? iDelay : 250;
     var $this = this, oTimerId;
     var anControl = $( 'div.dataTables_filter input:text' );
     anControl.unbind( 'keyup' ).bind( 'keyup', function() {
    var $$this = $this;
    window.clearTimeout(oTimerId);
    oTimerId = window.setTimeout(function() {
    $$this.fnFilter( anControl.val() );
    }, iDelay);
      });
      return this;
    }

    $(document).ready(function(){
      $('#the_table').dataTable({
        'bProcessing':true,
        'bServerSide':true,
        'sAjaxSource':'get_datacleannota.php'

      }).fnSetFilteringDelay();
    });

</script>
```

Figure 5.12 Datatables declaration on targeted page

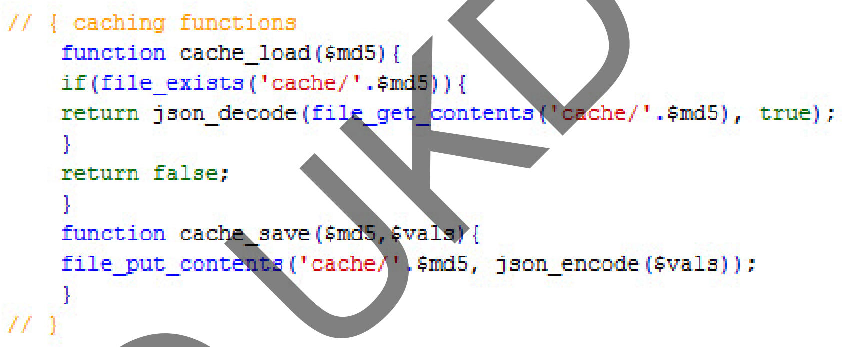
To display datatables, the structure of the HTML table must use <thead> and <tbody> tags like shown in Figure 5.13. <thead> is used to declare <th> as the column header or column title. <tbody> will generated from the source data page, in this case: ‘get_datacleannota.php’.



```
<table id="the_table">
   <thead>
          <tr>
            <th>ID nota</th>
            <th>Tanggal nota</th>
          </tr>
   </thead>
   <tbody>
   </tbody>
</table>
```

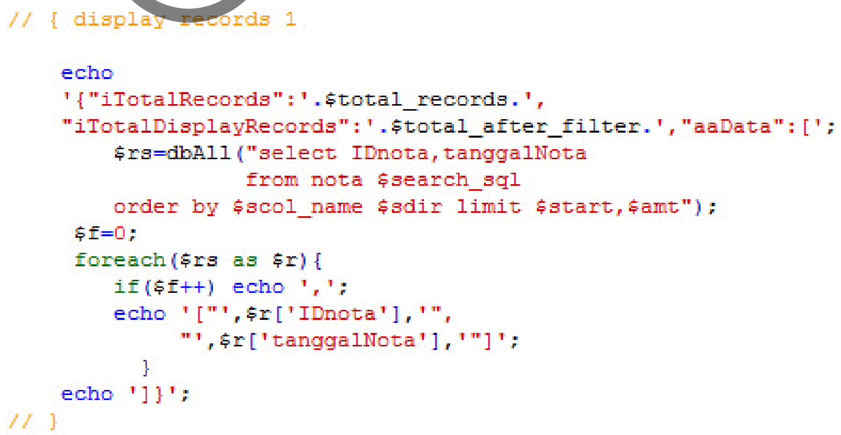
Figure 5.13 Datatables table structure

The important part of the modification of the source page is the cache function. With cache function, the queries are stored in the folder ‘cache’ and it will generate a cache files every queries being executed. This makes the same queries being cached and makes the result generated faster. In addition, the problem of using this cache will be the main problem at the end of this prototype.



```
// { caching functions
    function cache_load($md5){
    if(file_exists('cache/'.$md5)){
    return json_decode(file_get_contents('cache/'.$md5), true);
    }
    return false;
    }
    function cache_save($md5,$vals){
    file_put_contents('cache/'.$md5, json_encode($vals));
    }
// }
```

Figure 5.14 Datatables cache function



```
// { display records 1

    echo
    '{"iTotalRecords":'.$total_records.',
    "iTotalDisplayRecords":'.$total_after_filter.',"aaData":[';
        $rs=dbAll("select IDnota,tanggalNota
                     from nota $search_sql
        order by $scol_name $sdir limit $start,$amt");
     $f=0;
     foreach($rs as $r){
        if($f++) echo ',';
        echo '["',$r['IDnota'],'",
             "',$r['tanggalNota'],'"]';
          }
    echo ']}';
// }
```

Figure 5.15 Datatables display <tbody> content

After all, the <tbody> needs to be filled with the data from select query. The main query is to select and display all data in table 'nota'. $rs in Figure 5.15 reflects the main query of this datatables. After the query has been executed, the system will do the loop (foreach). For every data as an array, it will display column 'IDnota' and 'tanggalNota' from table 'nota'. Now, the datatables in 'tabelnota.php' page is ready to display live data from table 'nota'.

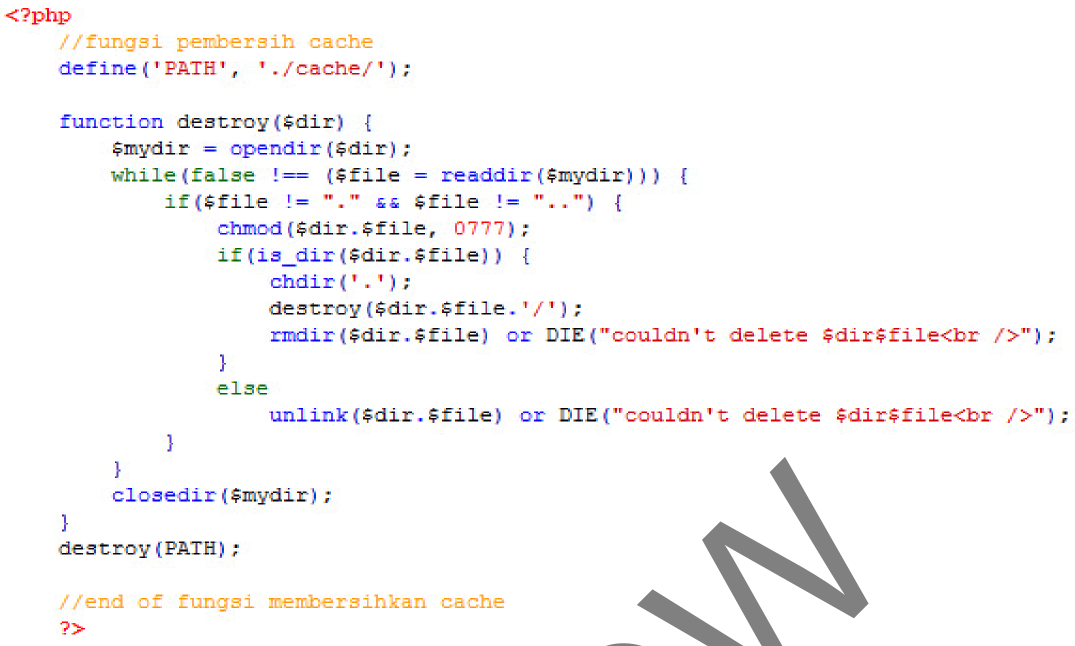
This is the end of prototype III. The writer backups this prototype in the local storage of his laptop and keeps it available on the other external storage devices too. The last part of this prototype will be testing and evaluation which will be described in section 5.4. There is a problem found on this testing phase, the cache files in the 'cache' folder seems to cause the datatables display keeps static like a previous cache's view even the data in table 'nota' has changed. After some problem solving analysis, this happened because the cache files in the cache folder must be un-cached (delete) every time the page reload. This problem is noted and will be solved in the beginning of prototype IV.

5.2.4 Prototype IV

This prototype begins with fixing bug about cache files from the previous prototype. The solution for this bug is to clean the directory 'cache' before the pages containing datatables are opened. Figure 5.16 shows the code implemented to delete all the files existed in the 'cache' folder.

- First, the path is defined with the code: `define('PATH','./cache');` which means the target location for this function is located under its index path on the 'cache' folder.
- Create function destroy and give permission (set chmod into 777) that will delete the files in the directory.
- After function has been executed, destroy the path definition.

This solution works properly for the system and there is no more problems about caching functions until now.



```
<?php
    //fungsi pembersih cache
    define('PATH', './cache/');

    function destroy($dir) {
        $mydir = opendir($dir);
        while(false !== ($file = readdir($mydir))) {
            if($file != "." && $file != "..") {
                chmod($dir.$file, 0777);
                if(is_dir($dir.$file)) {
                    chdir('.');
                    destroy($dir.$file.'/');
                    rmdir($dir.$file) or DIE("couldn't delete $dir$file<br />");
                }
                else
                    unlink($dir.$file) or DIE("couldn't delete $dir$file<br />");
            }
        }
        closedir($mydir);
    }
    destroy(PATH);

    //end of fungsi membersihkan cache
    ?>
```

Figure 5.16 Code implementation to solve cache problems

After bug has been fixed, the next step is to implement the manager’s homepage and functionality. There are two big parts of this prototype and that makes this prototype becomes the longest period prototype to work on this project. The two big parts are the implementation of most manager’s use cases and the implementation of FP-Growth algorithm, continued with mining the association rules.

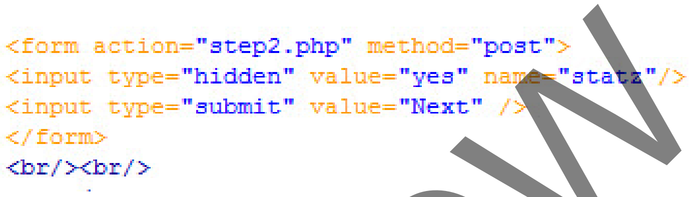
5.2.4.1 Implementation of manager use cases

At this stage, the manager’s homepage, edit profile, and the form designs for the setup wizard are created. The manager’s index page contains some information that the manager should know before operating the system. The edit profile page is similar with the one that already discussed on the previous administrator’s edit profile section. There are no difference in editing the manager’s edit profile because it is processed on the same page with administrator’s edit profile page (editprofresult.php). After completed the homepage and the edit profile page, now it is the time to implement code on the setup wizard interface. The interface of the wizard has already displayed in the Figure 4.19 – 4.25 in chapter 4. For the information, the ‘Dashboard’ page and the ‘Rules Table’ page will be implemented in the next prototype.

5.2.4.2 FP-Growth implementation and mining association rules

To start explain about the implementation of FP-Growth algorithm, let's take a look on each of the wizard steps.

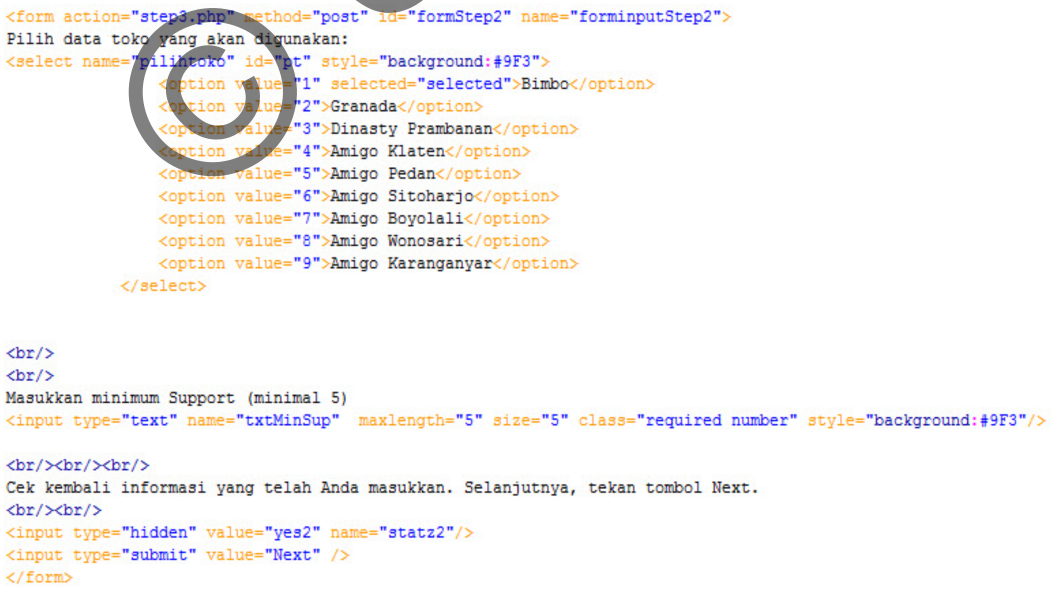
- step1.php → the wizard starts with confirming will of the manager to do the wizard. Once done, the wizard should be completed for the proper result of this system.
- Every page move of the wizard will send hidden input to be checked in the next page. This is done to avoid URL manipulation and prevent the wizard's failure.



```
<form action="step2.php" method="post">
<input type="hidden" value="yes" name="statz"/>
<input type="submit" value="Next" />
</form>
<br/><br/>
```

Figure 5.17 Example of hidden input and its value

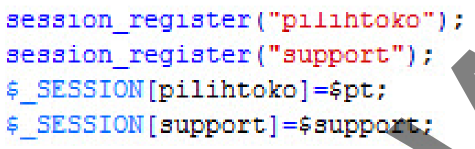
- step2.php → In this page, the system asks inputs for which shop's data and how much is the minimum support will be used for processing the next steps. Shop data sent as 'pilihtoko' name in drop down menu and minimum support sent as 'txtMinSup' name in a textbox.



```
<form action="step3.php" method="post" id="formStep2" name="forminputStep2">
Pilih data toko yang akan digunakan:
<select name="pilihtoko" id="pt" style="background:#9F3">
                <option value="1" selected="selected">Bimbo</option>
                <option value="2">Granada</option>
                <option value="3">Dinasty Prambanan</option>
                <option value="4">Amigo Klaten</option>
                <option value="5">Amigo Pedan</option>
                <option value="6">Amigo Sitoharjo</option>
                <option value="7">Amigo Boyolali</option>
                <option value="8">Amigo Wonosari</option>
                <option value="9">Amigo Karanganyar</option>
            </select>


<br/>
<br/>
Masukkan minimum Support (minimal 5)
<input type="text" name="txtMinSup"  maxlength="5" size="5" class="required number" style="background:#9F3"/>

<br/><br/><br/>
Cek kembali informasi yang telah Anda masukkan. Selanjutnya, tekan tombol Next.
<br/><br/>
<input type="hidden" value="yes2" name="statz2"/>
<input type="submit" value="Next" />
</form>
```

Figure 5.18 HTML code for step2.php

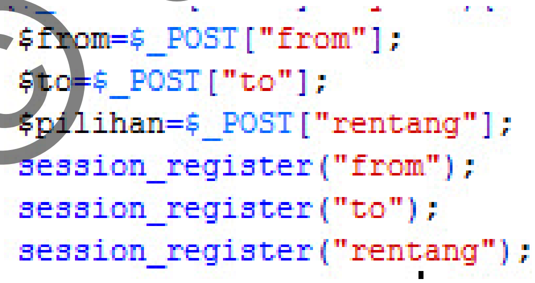
- step3.php → values sent from 'step2.php' form accepted in this page and then stored as sessions. There is also a form that must be filled in this page. The manager should choose whether he/she wants to choose data in fixed period, custom date, or all data in the database. The options provided in existing system is the Idul Fitri Day (Ramadhan Feast), Christmas holiday period, all data or custom data that can be inputted by checking a checkbox. Figure 4.21 in the previous chapter shown the formatted HTML for this page.



```
session_register("pilihtoko");
session_register("support");
$_SESSION[pilihtoko]=$pt;
$_SESSION[support]=$support;
```

Figure 5.19 Shop chosen and minimum support registered as sessions

- step4.php → The value chosen in 'step3.php' accepted in this page and then stored as sessions. Value of $pilihan is received if the manager chooses one of the radio button provided on the form. Meanwhile, values of $from and $to are received if the manager inputted the custom date range through the textbox in the 'step3.php' form.

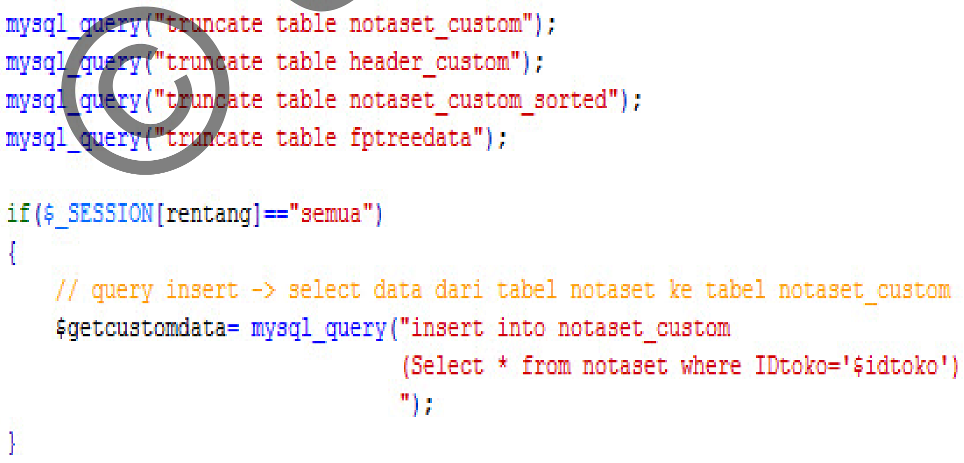


```
$from=$_POST["from"];
$to=$_POST["to"];
$pilihan=$_POST["rentang"];
session_register("from");
session_register("to");
session_register("rentang");
```

Figure 5.20 $from, $to and $pilihan registered as sessions

There is no form in this page, just a notification informed the manager that the system is ready to create new FP-Tree. The FP-Growth algorithm uses FP-Tree as a compact data structure. This tree will be used to support frequent-itemset generation of the algorithm. When 'Create' button is pressed, the system will open 'step5.php'.

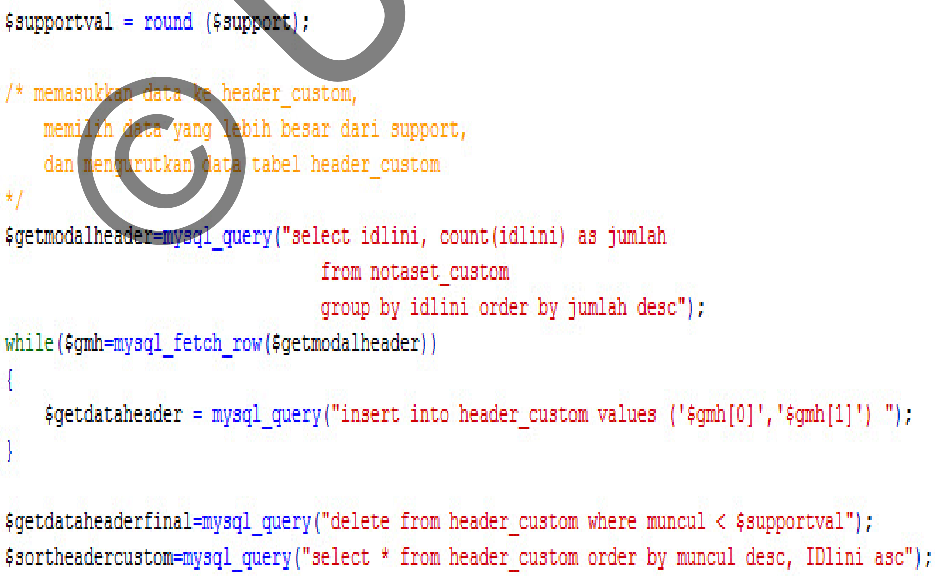
- step5.php → This is the page where FP-Tree is created. The system will create the tree based on the value of the sessions that are already registered on the previous steps. Explanation and screenshots about the creation of FP-Tree will be explained step-by-step below.
    - First of all, the system will check whether the data will be used is using custom date ($from and $to) or fixed date ($rentang). Let's assume that the manager inputs a fixed date, which is 'all data' from the radio button's options. This means the system will use data from 01-01-2006 until 31-12-2008. Both custom and fixed date are using the same algorithm, the difference is just the data being used by the system is based on each of the time period.
    - To begin the creation of FP-Tree, all of the data on the tables related with FP-Tree must be cleaned first. Figure 5.21 shows the process of truncated those tables. Then, because all data was chosen as the data source, the $getcustomdata query will be executed. This query inserting the all the data chosen in the 'notaset' table into 'notaset_custom' table. This is done to keep the data in 'notaset' table clean and original.



```
mysql_query("truncate table notaset_custom");
mysql_query("truncate table header_custom");
mysql_query("truncate table notaset_custom_sorted");
mysql_query("truncate table fptreedata");

if($_SESSION[rentang]=="semua")
{
    // query insert -> select data dari tabel notaset ke tabel notaset_custom
    $getcustomdata= mysql_query("insert into notaset_custom
                                (Select * from notaset where IDtoko='$idtoko')
                                ");
}
```

Figure 5.21 Process of creating FP-Tree

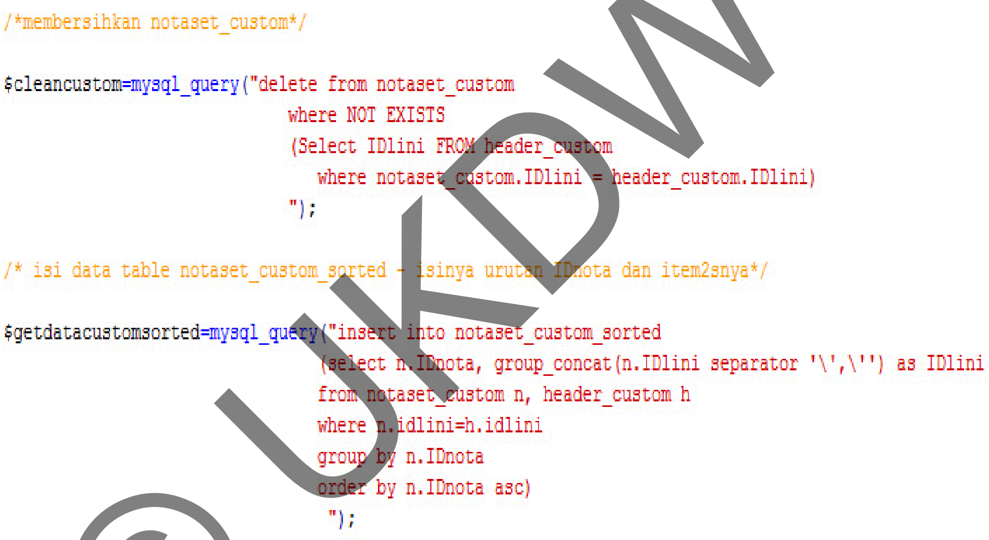
- The next step is to count and round the minimum support. For example, minimum support is 6.42 will be rounded to 6. This is done through a PHP function; round().
- $getmodalheader → select IDlini and its count grouped by IDlini and ordered descending by the count itself. This is the process that analogically the same with the creation of header table in FP-Growth algorithm.
- Because the data has successfully selected by the $getmodalheader query, the system has to fetched every data from that query and insert it into ‘header_custom’ table. ‘header_custom’ table is the header table of the items on this project. It contains the items and its support from the data range that were selected.
- $getdataheaderfinal → clean the data in ‘header_custom’ table which has the frequency of occurrence less than minimum support.
- $sortheadercustom → do descending sort to all the data in ‘header_custom’ table by its frequency of occurrence. (Figure 5.22 shows the code implemented until this step)



```
$supportval = round ($support);

/* memasukkan data ke header_custom,
   memilih data yang lebih besar dari support,
   dan mengurutkan data tabel header_custom
*/
$getmodalheader=mysql_query("select idlini, count(idlini) as jumlah
                             from notaset_custom
                             group by idlini order by jumlah desc");
while($gmh=mysql_fetch_row($getmodalheader))
{
   $getdataheader = mysql_query("insert into header_custom values ('$gmh[0]','$gmh[1]') ");
}

$getdataheaderfinal=mysql_query("delete from header_custom where muncul < $supportval");
$sortheadercustom=mysql_query("select * from header_custom order by muncul desc, IDlini asc");
```

Figure 5.22 Process of creating FP-Tree (2)

- The next step is executing $cleancustom → to delete 'notsaet_custom' data by checking its 'Idlini' where not exists in 'header_custom' table. This step is very important for the creation of FP-Tree.
- $getdatacustomsorted → insert the data to 'notaset_custom_sorted' based on the data from 'notaset_custom' table and 'header_custom' table. At the end of this query, the system will have a table that similar with compact TID table on the FP-Growth algorithm. It contains the IDnota (TID) and the items (Items) bought in that TID. (Figure 5.23 shows the code implemented until this step)



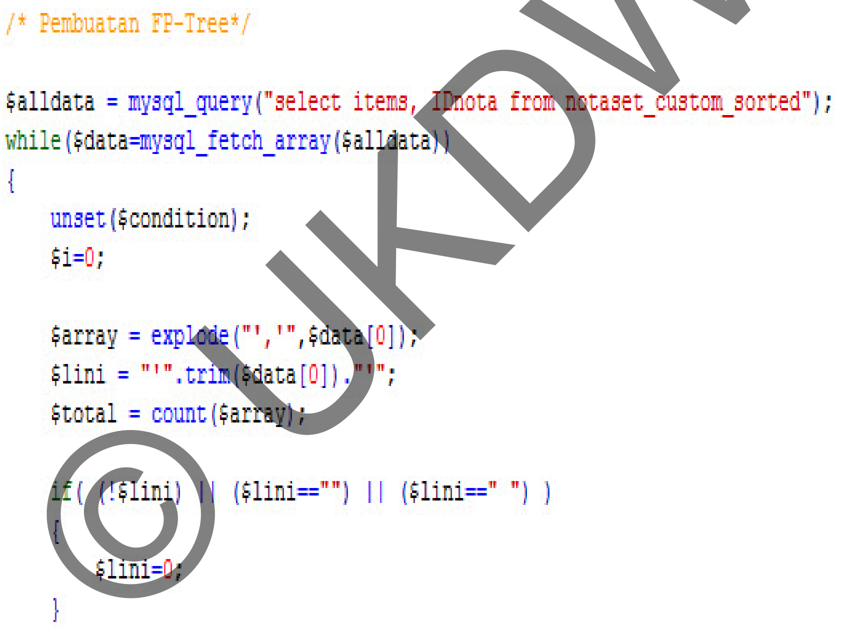
```
/*membersihkan notaset_custom*/

$cleancustom=mysql_query("delete from notaset_custom
                             where NOT EXISTS
                             (Select IDlini FROM header_custom
                                where notaset_custom.IDlini = header_custom.IDlini)
                             ");

/* isi data table notaset_custom_sorted - isinya urutan IDnota dan item2snya*/

$getdatacustomsorted=mysql_query("insert into notaset_custom_sorted
                                (select n.IDnota, group_concat(n.IDlini separator '\',\'') as IDlini
                                from notaset_custom n, header_custom h
                                where n.idlini=h.idlini
                                group by n.IDnota
                                order by n.IDnota asc)
                                 ");
```

Figure 5.23 Process of creating FP-Tree (3)

- Do $alldata → select items, IDnota from 'notaset_custom_sorted' table.
- The next process is fetch $alldata into arrays. 'Items' is $data[0] and 'IDnota' is $data[1].
- Then, the FP-Tree is built using the code inside the looping of fetched data. The code of this looping is:

```
while($data=mysql_fetch_array($alldata)
{ …the code.. }
```

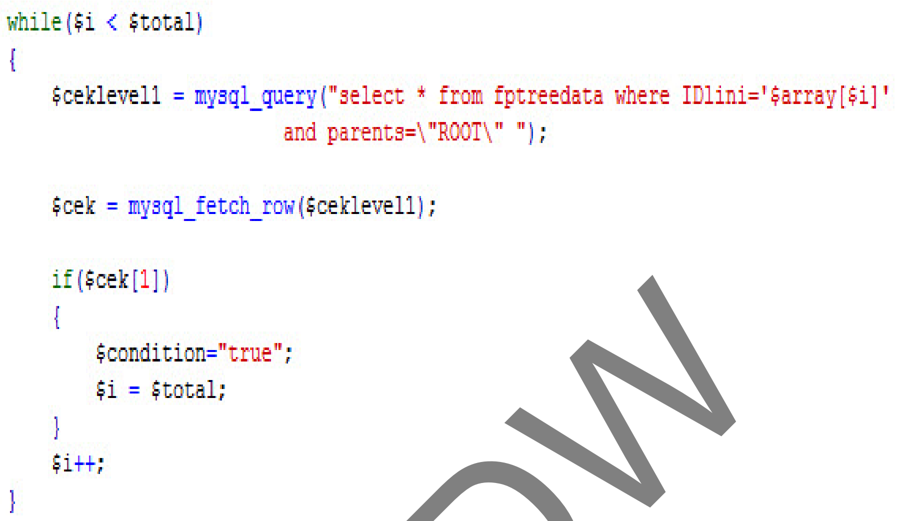
- The first line of the looping is unset $condition. $condition is a variable that will be used for flag (true or false) condition in this project. Then, set $i to 0.
- $array → explodes items in column items with separator **‘,’**
- $lini → trim the items and format it into specified format like: `‘<itemsID>’.` Items ID here is the same with groups of IDlini
- $total → count how much IDlini are available in the $array.
- If $lini is empty or blank then set $lini to 0. It means that there is no items in that IDnota (TID) . (Figure 5.24 shows the code implemented until this step)



```
/* Pembuatan FP-Tree*/

$alldata = mysql_query("select items, IDnota from notaset_custom_sorted");
while($data=mysql_fetch_array($alldata))
{
    unset($condition);
    $i=0;

    $array = explode("','",$data[0]);
    $lini = "'".trim($data[0])."'";
    $total = count($array);

    if( (!$lini) || ($lini=="") || ($lini==" ") )
    {
        $lini=0;
    }
```

Figure 5.24 Process of creating FP-Tree (4)

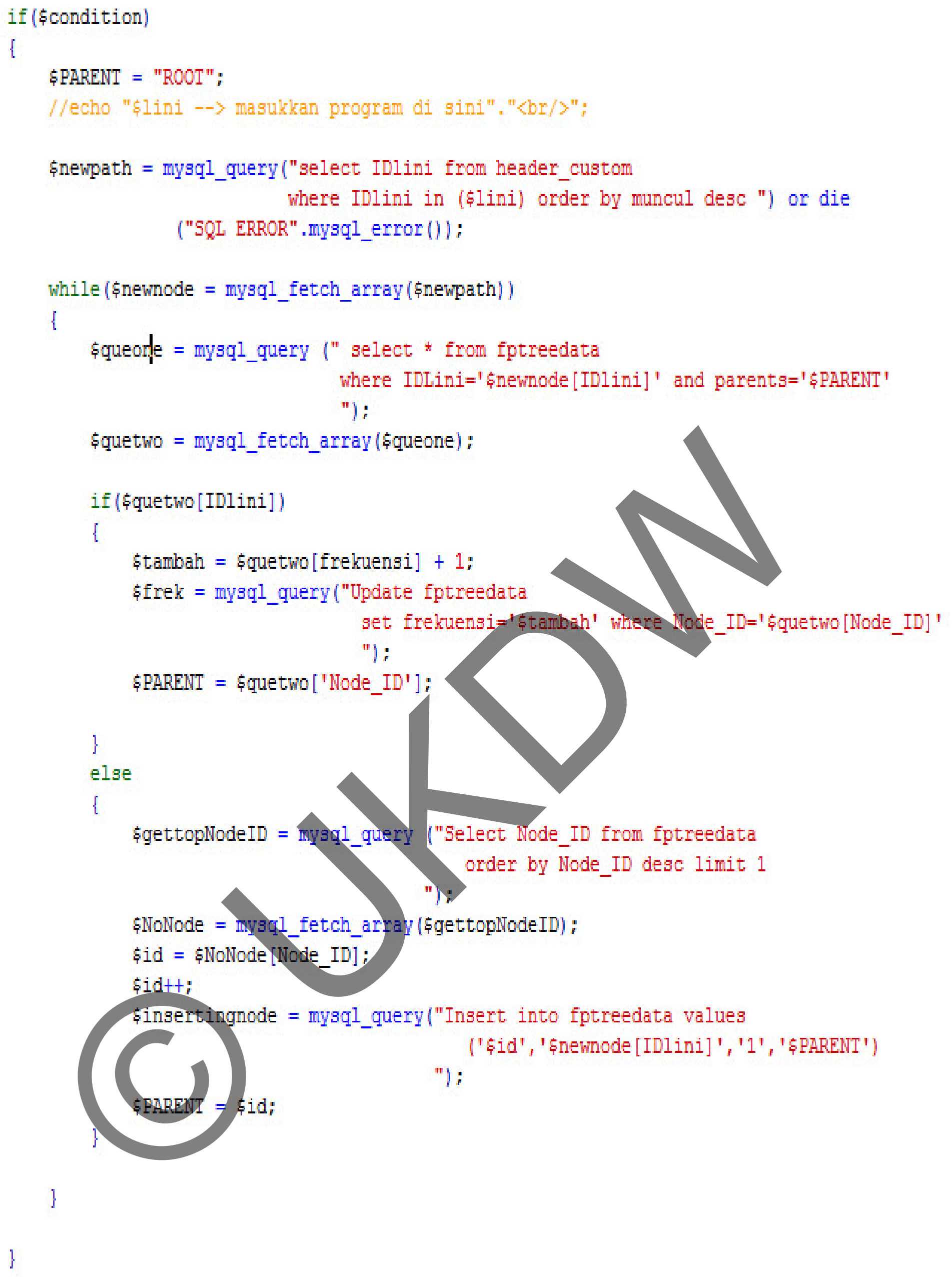
- The code in Figure 5.25 explain a loop (while) to check whether in every items in $array is a direct child from ROOT or not (level 1 node). If one of the items in $array is already on the ‘fptreedata’ table and it is the direct child of the ROOT, then $condition will be set into true. Else, $condition remains false. It can also said that this loop will be influence starting from second records of $data (remember that when the data from the first records being fetched, the ‘fptreedata’ table is still empty).



```
while($i < $total)
{
    $ceklevel1 = mysql_query("select * from fptreedata where IDlini='$array[$i]'
                        and parents=\"ROOT\" ");

    $cek = mysql_fetch_row($ceklevel1);

    if($cek[1])
    {
        $condition="true";
        $i = $total;
    }
    $i++;
}
```

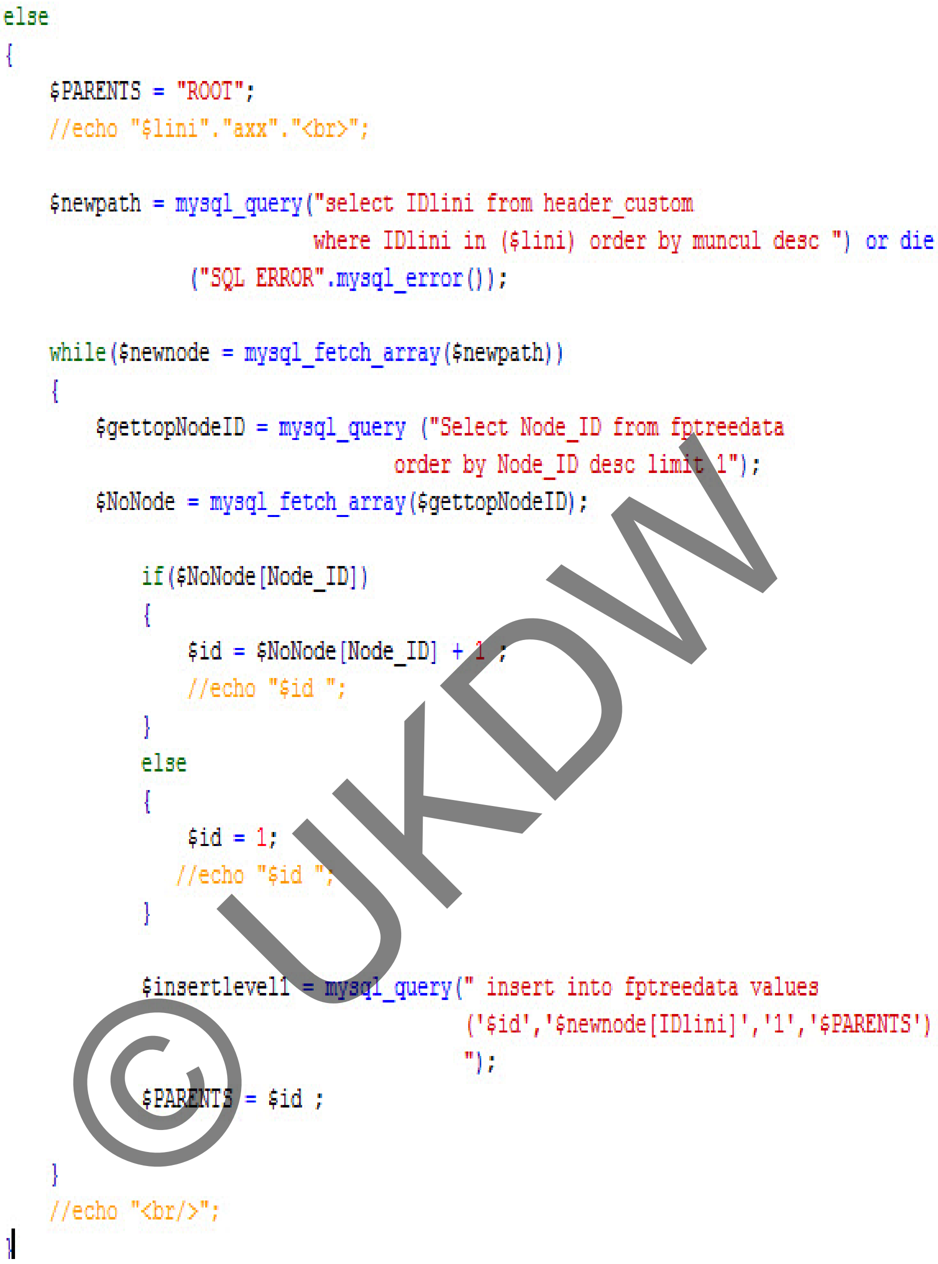
Figure 5.25 Process of creating FP-Tree (5)

- There is two branches await to process the data. First is the condition where $condition is true and the other is the condition where $condition is false (unset).
- Figure 5.26 explains the first branch. This condition will be run if the IDnota (TID) is not the first record fetched from $data and items on that IDnota (TID) has already become level 1 node (direct child of ROOT).
- Figure 5.27 explains the second branch. This condition will be run if the IDnota (TID) is the first record fetched from $data.



```
if($condition)
{
    $PARENT = "ROOT";
    //echo "$lini --> masukkan program di sini"."<br/>";

    $newpath = mysql_query("select IDlini from header_custom
                           where IDlini in ($lini) order by muncul desc ") or die
                ("SQL ERROR".mysql_error());

    while($newnode = mysql_fetch_array($newpath))
    {
        $queone = mysql_query (" select * from fptreedata
                                where IDLini='$newnode[IDlini]' and parents='$PARENT'
                                ");
        $quetwo = mysql_fetch_array($queone);

        if($quetwo[IDlini])
        {
            $tambah = $quetwo[frekuensi] + 1;
            $frek = mysql_query("Update fptreedata
                                  set frekuensi='$tambah' where Node_ID='$quetwo[Node_ID]'
                                  ");
            $PARENT = $quetwo['Node_ID'];

        }
        else
        {
            $gettopNodeID = mysql_query ("Select Node_ID from fptreedata
                                            order by Node_ID desc limit 1
                                        ");
            $NoNode = mysql_fetch_array($gettopNodeID);
            $id = $NoNode[Node_ID];
            $id++;
            $insertingnode = mysql_query("Insert into fptreedata values
                                            ('$id','$newnode[IDlini]','1','$PARENT')
                                         ");
            $PARENT = $id;
        }

    }

}
```

Figure 5.26 Process of creating FP-Tree (6)



```
else
{
    $PARENTS = "ROOT";
    //echo "$lini"."axx"."<br>";

    $newpath = mysql_query("select IDlini from header_custom
                           where IDlini in ($lini) order by muncul desc ") or die
                ("SQL ERROR".mysql_error());

    while($newnode = mysql_fetch_array($newpath))
    {
        $gettopNodeID = mysql_query ("Select Node_ID from fptreedata
                                  order by Node_ID desc limit 1");
        $NoNode = mysql_fetch_array($gettopNodeID);

            if($NoNode[Node_ID])
            {
                $id = $NoNode[Node_ID] + 1 ;
                //echo "$id ";
            }
            else
            {
                $id = 1;
               //echo "$id ";
            }

            $insertlevel1 = mysql_query(" insert into fptreedata values
                                        ('$id','$newnode[IDlini]','1','$PARENTS')
                                        ");

            $PARENTS = $id ;

    }
    //echo "<br/>";
}
```

Figure 5.27 Process of creating FP-Tree (7)

- For example, assume that ‘items’ column fetched in $data is:
    - Records 1: ‘A’,’B’,’C’
    - Records 2: ‘A’,’C’,’D’
    - Records 3: ‘B’,’A’,’E’

- Please remember that the first records (records 1) from the $data will run code on Figure 5.27. In this condition:
    - First, set $PARENTS as “ROOT”
    - Execute $newpath query → in this case: select IDlini from header_custom where IDlini in (‘A’,’B’,’C’) order by muncul desc; The result will be fetched in $newnode. It means that each of ‘A’ – ‘B’ – ‘C’ will go through looping (while) in the next line.
    - In every block statement of while, do:
        - Execute $gettopNodeID query to select the highest Node_ID from ‘fptreedata’ table limit by one.
        - Enter that Node_ID value into $NoNode.
        - If the query return a value, it means that the ‘fptreedata’ table already has data in it, then set $id into $NoNode + 1.
        - Else, set $id = 1.
        - Execute $insertlevel query → insert data into ‘fptreedata’ table with values ($id, $newnode[IDlini], 1, $PARENTS).
        - Set $PARENTS = $id.
    - At the end of this stage, data on the ‘fptreedata’ will looks like Table 5.1

Table 5.1 data on ‘fptreedata’ table after first records

Node_ID	**IDlini**	**frekuensi**	**parents**
1	A	1	ROOT
2	B	1	1
3	C	1	2

- Now, the second records which is ‘A’,’C’,’D’ will be checked with code implemented in Figure 5.25. If one of the items on ‘A’,’C’,’D’ is a level 1 node (direct child of ROOT), then the $condition will set into “true”. It means that the system will match with the first branch and it will run the code inside if($condition) block statement in Figure 5.26. In this condition:

- First, set $PARENT as “ROOT”
- Execute $newpath query → in this case: select IDlini from header_custom where IDlini in (‘A’,’C’,’D’) order by muncul desc; The result will be fetched in $newnode. It means that each of ‘A’ – ‘C’ – ‘D’ will go through looping (while) in the next line.
- In every block statement of while, do:
    - Execute $queone query to select all column from ‘fptreedata’ table where IDlini = $newnode[IDlini] and parents = ‘$PARENT’
    - Fetched array $queone values into $quetwo
    - If $quetwo query return a value, it means that the first IDlini in the ‘A’-‘C’-‘D’, which is A, is already a level 1 node. This means the system doesn’t have to insert new data to the ‘fptreedata table’ but updated its frequency.
        - $tambah = $quetwo[frekuensi] + 1;
        - Do $frek query → update the level 1 ‘A’ node frequency into $tambah.
        - Set $PARENT into $quetwo['Node_ID'];
    - The next $quetwo queries for IDlini ‘C’ and ‘D’ will enter the else condition. In this block statement, do:
        - Execute $gettopNodeID query to select the highest Node_ID from ‘fptreedata’ table limit by one.
        - Enter that Node_ID value into $NoNode.
        - Set $id = $NoNode[Node_ID];
        - Do $id++ to create new Node_ID for the new data

- Execute $insertingnode query → insert data into 'fptreedata' table with values ($id, $newnode[IDlini], 1, $PARENT).
- Set $PARENT = $id.

- At the end of this stage, data on the 'fptreedata' will looks like Table 5.2

Table 5.2 data on 'fptreedata' table after second records

Node_ID	IDlini	frekuensi	parents
1	A	2	ROOT
2	B	1	1
3	C	1	2
4	C	1	1
5	D	1	4

- The third records which is 'B','A','E' will be checked with code implemented in Figure 5.25. Because 'A' has been already marked as a level 1 node, then this record will also enter if($condition) block statement just like the second records. Then, because the first IDlini ('B') is not a level 1 node, the system will enter else condition three times to insert new data into 'fptreedata' table. At the end of this stage, data on the 'fptreedata' will looks like Table 5.3

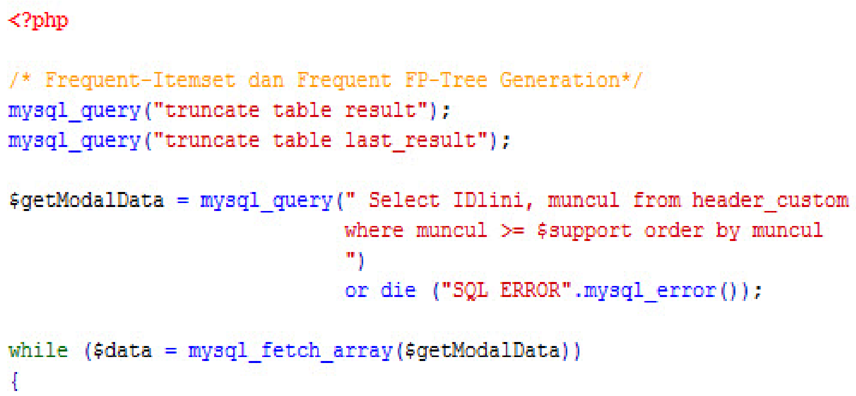
Table 5.3 data on 'fptreedata' table after third records

Node_ID	IDlini	frekuensi	parents
1	A	2	ROOT
2	B	1	1
3	C	1	2
4	C	1	1
5	D	1	4
6	B	1	ROOT
7	A	1	6

8	E	1	7

- When all of the processes above have been completed, the system will have a complete FP-Tree structure in ‘fptreedata’ table and the page will display information that the FP-Tree has been successfully created. The next step is the frequent-itemset generation. This process will take a various process time just like the creation of FP-Tree. To process the frequent-itemset generation and get the final result of this project, press ‘Next’ button on ‘step5.php’.

- step6.php (final wizard page) → when the page opened, the system will run script to generate frequent-itemset and produce the association mining result of the system. The full script can be seen on file ‘step6.php’ in ‘Amigodm’ folder. The explanation of important parts of the script will be explained sequentially.
    - First of all, truncate ‘result’ table and ‘last_result’ table to clean the existing data inside both tables.
    - Execute $getModalData query → select IDlini and its frequency from ‘header_custom’ table where the frequency is greater or equal than the minimum support, sorted by the frequency.
    - Fetch $getModalData query result into $data and do looping (while) for each $data (Figure 5.28 shows the code implemented until this step). This while ($data loop) consists of three big parts. First is the conditional-pattern base generation. Second is the conditional FP-Tree generation and the last part is inserting the data into ‘result’ table. The next point will explained about these three big parts.



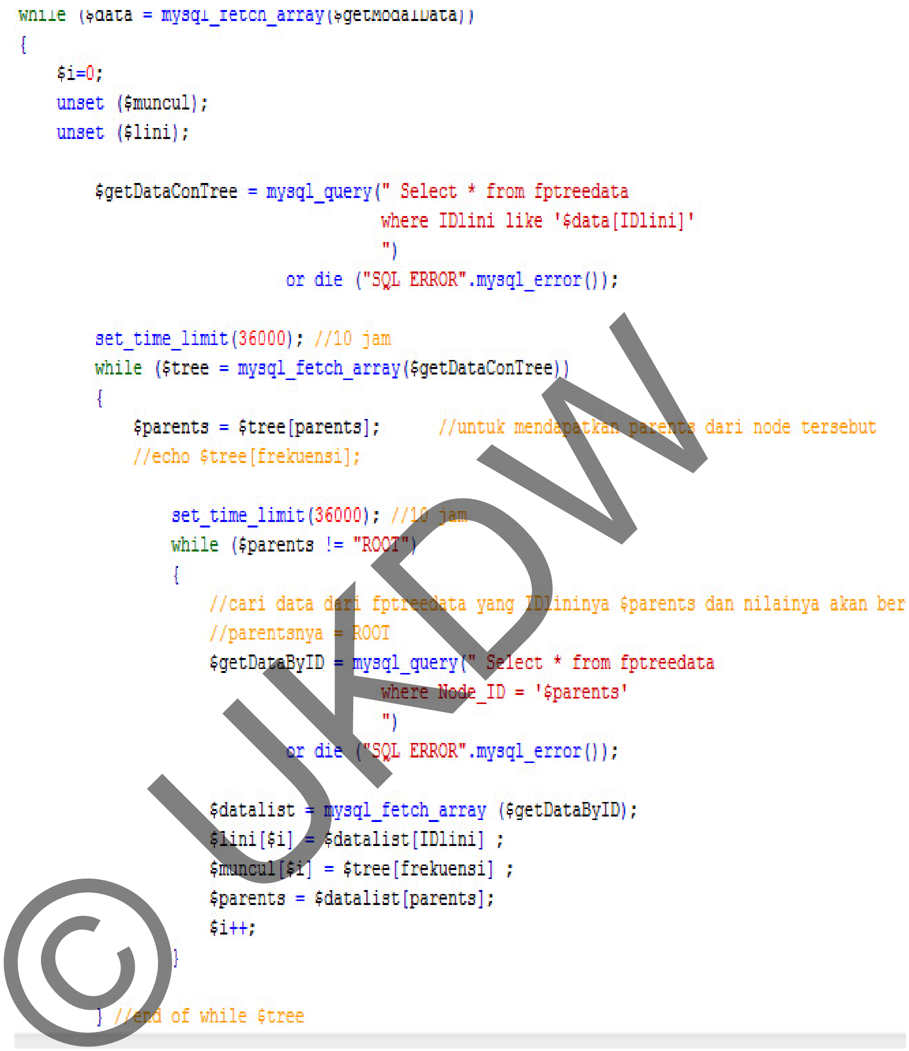
```
<?php

/* Frequent-Itemset dan Frequent FP-Tree Generation*/
mysql_query("truncate table result");
mysql_query("truncate table last_result");

$getModalData = mysql_query(" Select IDlini, muncul from header_custom
                            where muncul >= $support order by muncul
                            ")
                            or die ("SQL ERROR".mysql_error());

while ($data = mysql_fetch_array($getModalData))
{
```

Figure 5.28 Process of frequent-itemset generation (1)

- In each loop of $data, do:
    - Set $i = 0, unset $muncul value and unset $lini value.
    - Do query $getDatConTree → select all data from ‘fptreedata’ table where IDlini like $data[IDlini].
    - Fetch the $getDataConTree into $tree and do looping (while) for each $tree. In each of the looping, do:
        - Set $parents = $tree[parents]
        - Check $parents value. If the value is not “ROOT”, do:
            - $getDataByID query → select all data from ‘fptreedata’ table where Node_ID = $parents.
            - Fetch the result into $datalist.
            - Set $lini[$i] = $datalist[IDlini] and $parents = $datalist[$parents].
            - Set $muncul[$i] = $tree[frekuensi];
            - Set $i++
    - Until this stage, the system have done the first part which is conditional pattern-base generation. The next part is to check whether there is IDlini that must be combined into one or not. This part is also considered as conditional FP-Tree generation in FP-Growth Algorithm.

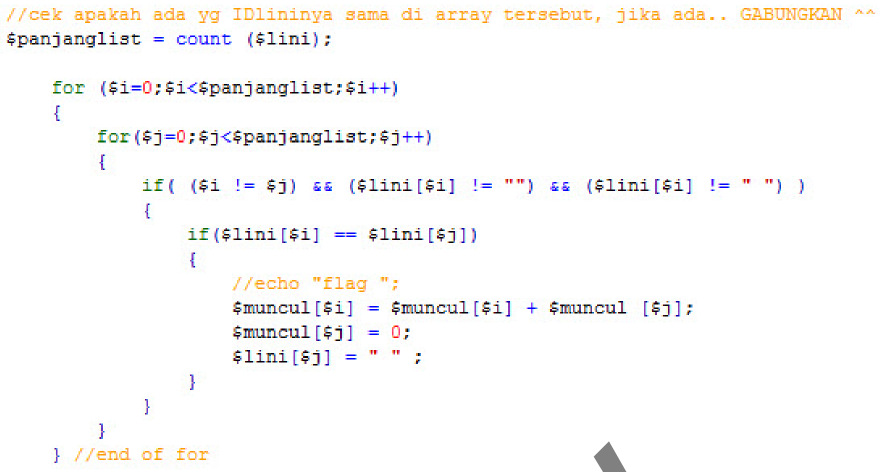
- To do this, count the size of $lini array and set the value into $panjanglist variable.
- Do multiple looping conditions like stated in Figure 5.30. In inner looping, do:
  - Check the condition:
    ```
    if( ($i != $j) && ($lini[$i] != "")
    && ($lini[$i] != " ") )
    ```
  - If the condition is match, then check it again with:
    ```
    if($lini[$i] == $lini[$j])
    ```
  - If the previous if condition is matched, then do the following steps:
    - Set $muncul[$i] = $muncul[$i] + $muncul [$j];
    - Set muncul[$j] = 0 and set $lini[$j] = " " to avoid repetition of added value.
- Now the system have done the second part and already has conditional FP-Tree data.
- The last part is to insert the values of the variables into 'result table (Figure 5.31).
- In this part, do looping from $i=0 to $i< size of $panjanglist with increasing one value of $i. In every loop, check if $muncul[$i] >= $support then do the following steps:
  - $persentase = $mncul[$i] / $data[muncull] * 100 (this statement is to find percentage of the confidence)
  - $persentase = substr ($persentase,0,7) (to get first seven characters from the confidence's calculation result)
  - Execute a mysql query → insert data into 'result' table with values ($data[IDlini], $lini[$i], $muncul[$i], $persentase).
- This is the end of $data looping (while condition). In addition, the script on 'step6.php' is not over yet. The system still have to process the 'result' table into 'result_translated' table. The

data in ‘result’ table is still the IDlini of the items. The system should translate it into items name.



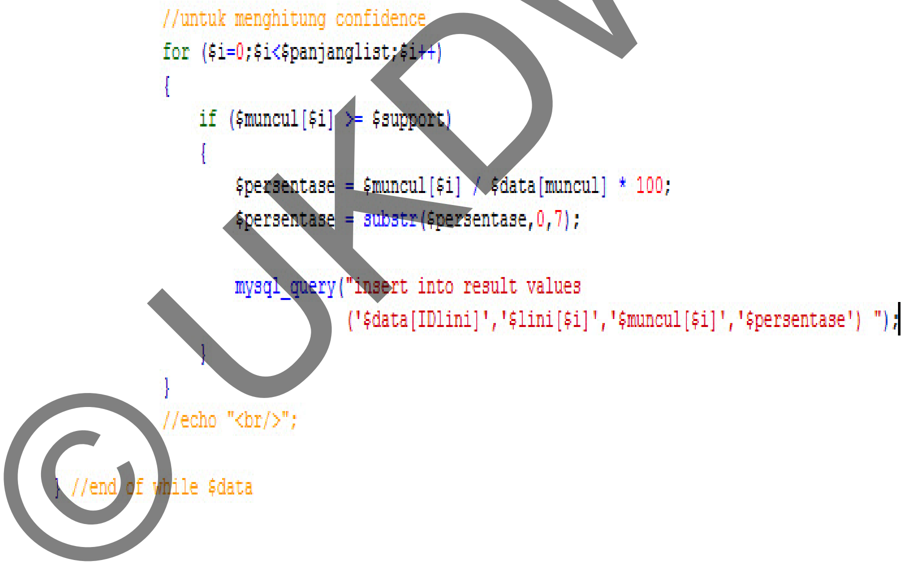
```
while ($data = mysql_fetch_array($getModalData))
{
    $i=0;
    unset ($muncul);
    unset ($lini);

        $getDataConTree = mysql_query(" Select * from fptreedata
                                      where IDlini like '$data[IDlini]'
                                      ")
                            or die ("SQL ERROR".mysql_error());

        set_time_limit(36000); //10 jam
        while ($tree = mysql_fetch_array($getDataConTree))
        {
            $parents = $tree[parents];      //untuk mendapatkan parents dari node tersebut
            //echo $tree[frekuensi];

                set_time_limit(36000); //10 jam
                while ($parents != "ROOT")
                {
                    //cari data dari fptreedata yang IDlininya $parents dan nilainya akan ber
                    //parentsnya = ROOT
                    $getDataByID = mysql_query(" Select * from fptreedata
                                      where Node_ID = '$parents'
                                      ")
                            or die ("SQL ERROR".mysql_error());

                    $datalist = mysql_fetch_array ($getDataByID);
                    $lini[$i] = $datalist[IDlini] ;
                    $muncul[$i] = $tree[frekuensi] ;
                    $parents = $datalist[parents];
                    $i++;
                }

        } //end of while $tree
```

Figure 5.29 Process of frequent-itemset generation (2)



```
//cek apakah ada yg IDlininya sama di array tersebut, jika ada.. GABUNGKAN ^^
$panjanglist = count ($lini);

    for ($i=0;$i<$panjanglist;$i++)
    {
        for($j=0;$j<$panjanglist;$j++)
        {
            if( ($i != $j) && ($lini[$i] != "") && ($lini[$i] != " ") )
            {
                if($lini[$i] == $lini[$j])
                {
                    //echo "flag ";
                    $muncul[$i] = $muncul[$i] + $muncul [$j];
                    $muncul[$j] = 0;
                    $lini[$j] = " " ;
                }
            }
        }
    } //end of for
```

Figure 5.30 Process of frequent-itemset generation (3)

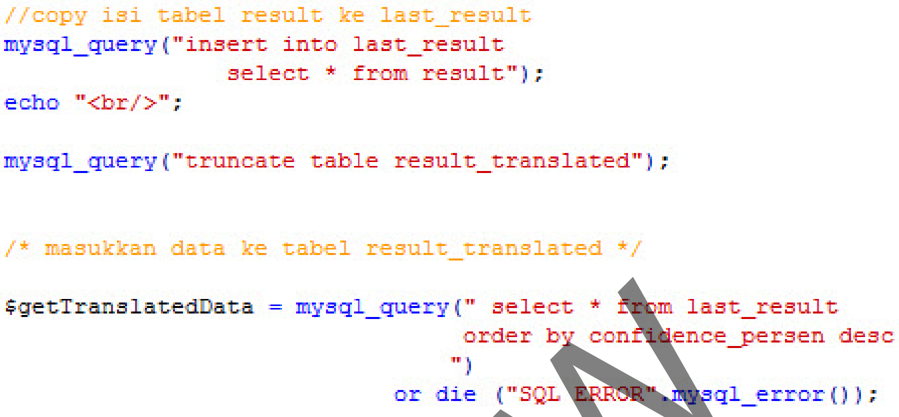


```
    //untuk menghitung confidence
    for ($i=0;$i<$panjanglist;$i++)
    {
        if ($muncul[$i] >= $support)
        {
            $persentase = $muncul[$i] / $data[muncul] * 100;
            $persentase = substr($persentase,0,7);

            mysql_query("insert into result values
                        ('$data[IDlini]','$lini[$i]','$muncul[$i]','$persentase') ");
        }
    }
    //echo "<br/>";

} //end of while $data
```

Figure 5.31 Process of frequent-itemset generation (4)

- After ‘result’ table was filled with data, now the system must do a data translation from ‘IDlini_A’ and ‘IDlini_B’ column in ‘result’ table into ‘Items’ and ‘to_Items’ column in ‘result_translated’ table.
- Execute a query to duplicate data in ‘result’ table to ‘last_result’ table → `insert into last_result select * from result`.
- Truncate table ‘result_translated’ to make sure that table is clean from previous wizard’s result → `mysql_query (truncate table result_translated”);`

- Execute $getTranslatedData query to select all data in ‘last_result’ table and sort it descending by the confidence percentage. (Figure 5.32 shows the code implemented until this step).



```
//copy isi tabel result ke last_result
mysql_query("insert into last_result
                select * from result");
echo "<br/>";

mysql_query("truncate table result_translated");


/* masukkan data ke tabel result_translated */

$getTranslatedData = mysql_query(" select * from last_result
                                    order by confidence_persen desc
                                   ")
                               or die ("SQL ERROR".mysql_error());
```

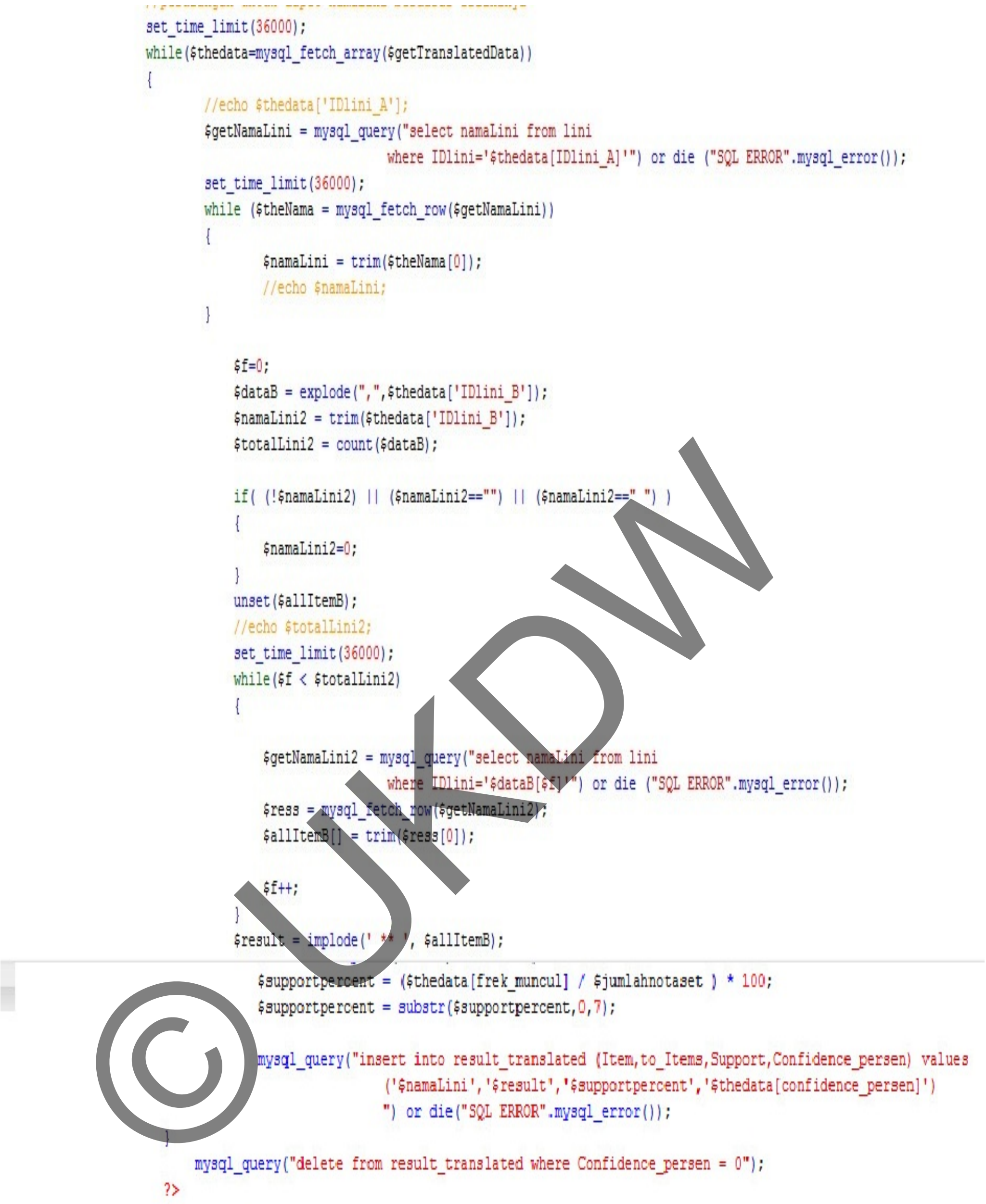
Figure 5.32 Process of data translation (1)

- Get the value of total items used in the dataset with query “SELECT SUM(muncul) AS jumlah FROM header_custom” and input the value into $jumlahnotaset.
- Do looping while $getTranslatedData query results are being fetched_array in $thedata. Inside every loop, do:
    - Execute $getNamaLini query to select namaLini from ‘lini’ table where IDlini is $thedata[IDlini_A].
    - Do looping while $getNamaLini query results are being fetched_array in $thedata. Inside every looping trim the namaLini (items name).
    - Set $f = 0.
    - $dataB → explode items in $thedata[‘IDlini_B’] with ‘,’ as separator.
    - $namaLini2 → trim $thedata[‘IDlini_B’].
    - $totalLini2 → count items in $dataB.
    - If $namaLini2 value is blank or empty, then set $namaLini2 = 0.
    - Clear value in $allItemB variable with unset() function.

- Do looping with condition: `while($f <$totalLini2){ … }.` Inside every loop do:
    - $getNamaLini2 query → select namaLini from 'lini' table where IDlini = $dataB[$f].
    - Fetch_row the result and insert into $ress.
    - Set $allItemB[] array with `trim($ress[0])` function.
    - Set $f increment → `$f++;`
- Now, implode each item in $allitemB with this sign → ** as separator.
- Set $supportpercent with the support percentage value. This value can be gained using this formula: `($thedata[frek_muncul] / $jumlahnotaset) * 100;`
- Execute MySQL query in the textbox below to insert the translated data into 'result_translated' table.

```
mysql_query("insert into result_translated
(Item,to_Items,Support,Confidence_persen)
values
('$namaLini','$result','$supportpercent','$hedata[c
onfidence_persen]')
") or die("SQL ERROR".mysql_error());
```

- Last, delete data in 'result_translated' table which has 'Confidence_persen' value = 0 because it doesn't reflect an association rule.



```
set_time_limit(36000);
while($thedata=mysql_fetch_array($getTranslatedData))
{
        //echo $thedata['IDlini_A'];
        $getNamaLini = mysql_query("select namaLini from lini
                                    where IDlini='$thedata[IDlini_A]'") or die ("SQL ERROR".mysql_error());
        set_time_limit(36000);
        while ($theNama = mysql_fetch_row($getNamaLini))
        {
                $namaLini = trim($theNama[0]);
                //echo $namaLini;
        }

            $f=0;
            $dataB = explode(",",$thedata['IDlini_B']);
            $namaLini2 = trim($thedata['IDlini_B']);
            $totalLini2 = count($dataB);

            if( (!$namaLini2) || ($namaLini2=="") || ($namaLini2==" ") )
            {
                $namaLini2=0;
            }
            unset($allItemB);
            //echo $totalLini2;
            set_time_limit(36000);
            while($f < $totalLini2)
            {

                $getNamaLini2 = mysql_query("select namaLini from lini
                                  where IDlini='$dataB[$f]'") or die ("SQL ERROR".mysql_error());
                $ress = mysql_fetch_row($getNamaLini2);
                $allItemB[] = trim($ress[0]);

                $f++;
            }
            $result = implode(' ** ', $allItemB);
                $supportpercent = ($thedata[frek_muncul] / $jumlahnotaset ) * 100;
                $supportpercent = substr($supportpercent,0,7);

                mysql_query("insert into result_translated (Item,to_Items,Support,Confidence_persen) values
                                 ('$namaLini','$result','$supportpercent','$thedata[confidence_persen]')
                                 ") or die("SQL ERROR".mysql_error());
}
    mysql_query("delete from result_translated where Confidence_persen = 0");
?>
```

Figure 5.33 Process of data translation (2)

- o The processes have been completed.

- At the end of step6.php, the manager can press ‘FINISH’ button to move into ‘Rules Table’ page to see the generated association rules resulted by the wizard.

This is the end of prototype IV.. The writer backups this prototype in the local storage of his laptop and keeps it available on the other external storage devices too. The last part of this prototype will be testing and evaluation which will be described in section 5.4. There are no significant errors occur after the code has been implemented. The writer faced same problems during the implementation of the FP-Tree algorithm because this system has to go through the database every time the system deals with ‘data process’. It makes the laptop being used become overload when the query processed is wrong or endless. The other problem is when the writer tried to do items combination on ‘to_Items’ column in order to generate more association rule result. The process of the query took very long time (and never finished during the experiment) and the results produced were not good. A lot of rules contain 0 % of confidence percent. In order to make the system more effective and faster, the writer decide to keeps the existing system and not to use the combination function.

5.2.5 Prototype V

After removing the combination function from the previous prototype, the system needs to display the association rules result and dashboard to see the status of the latest wizard’s result done by the manager. The interface of ‘Rules Table’ and ‘Dashboard’ pages have been shown in Figure 4.26 and 4.27 in Chapter 4. To create this page, the writer designed some rules that will be applied for the ‘Rules Table’ page and what information will be displayed for the ‘Dashboard’ page. Some rules for ‘Rules Table’ page are:

- This page will use datatables JQuery plug-in to display the data from ‘result_translated’ table.
- The features available is to sort, search items, minimum confidence filter, and convert the data into CSV format.
- To do the minimum confidence filter, use the textbox (search box) in right-top of datatables. If the characters inputted in the textbox is less than three, the datatables will automatically display the data which has minimum confidence higher or equal than the input. For example, if the manager input

50 in the textbox, datatables will display data which has confidence (%) >= 50.

- To do search items, use the textbox (Search box) in right-top of datatables. Input must be more than two (three or more) to search the item. For example,, if the manager input ‘CLN’ in the textbox, datatables will display data which contain ‘CLN’ in the ‘Item’ column.

Some rules for ‘Dashboard’ page are:

- This page will have summary of the existing rules, a visual representation of the rules, and top ten rules sorted by the highest confidence (%).
- This page will use JPgraph (http://jpgraph.ne) for the visualization (bar charts).
- Bar charts generated from top ten association rules result sorted by the highest confidence (%).

After designed the rules and the page, the writer implements the PHP code into these two pages. The implementation was not hard and everything is clear. There is no problem faced during implementation of this prototype. After successes with the implementation, now the system has completed all the requirements. For the information, the items name on the ‘Item’ column and ‘Related Items’ are based on the real name that Amigo’s staff understand. The writer did not enter the items name, it is the original name that should be understood by Amigo’s staff.

This is the end of prototype V. Before the writer backups this prototype in the local storage and other external storage, the writer prepares sample database for the presentation and finalized database that will be included in the DVD. Last but not least, the testing and evaluation stage is always conducted in the end of each prototype to test the system’s operation.

5.2.6 Final release

This is a stage to fix all the bugs found during the testing phase in every previous prototype. The writer makes sure that the bugs found are covered and fixed before the final release of the prototype is launched. After everything is done, the writer finalized the website of this Amigo Data Mining System and backup both the finalized web folder and the finalized database in internal/external media storage. This folders and files will be included on the DVD.

5.3 Implementation

5.3.1 Work Items List

There are some changes occur at the actual plan in this Final Year Project II implementation stage. Please refer to Appendix D for the Work Items List.

5.3.2 Iteration Plan

Please refer to Appendix E for the Iteration plan.

5.4 Testing

List of the testing done:

- User login testing
- User edit profile testing
- Add user testing
- Edit and delete user testing
- Data Cleaning testing
- Display cleaned data testing
- Setup Wizard testing
- Display association rules testing
- Convert to CSV testing
- Display dashboard testing

All of the information about the testing phase can be seen on the Test Case in Appendix F and Test Script in Appendix G.

5.5 Testing Phase Conclusion

All of the use cases (16 use cases) are successfully implemented in the end of prototype V. After much testing and debugging are performed, the prototype finalized in the final release of the project. All of the bugs were fixed and the system is ready to be used in the real system.

Chapter 6: Project Review

6.1 Introduction

The intended project scope to be completed has been met. The prototype covering 16 use cases has been developed. The final version of this web-based association rules project is successfully completed. The following sections explore on the evaluation of the aims and objectives, work items, successes, shortcomings, discussions, learning outcomes, project management, and cost-effectiveness.

6.2 Aims and Objectives

The final version of the website has been successfully completed the aims, objectives, and requirements of the project. The aim of this project is to create a web-based system to help Amigo Group's manager come to the decision about sales by using association rule's application. The main objective of this project is to implement association rules mining to a new Amigo Group's data mining system. All 16 use cases agreed were reviewed by the supervisor and the client. The use cases reflect the aims and objectives of the project, as well as the system requirements.

There is no modification made during the implementation time from BIT304 to BIT305 project. People from Amigo Group keep their commitment to build a web-based system to help their managers do the analysis works. There is a good relation made between the writer as the programmer, the supervisor, the Amigo group's people during the implementation of this project.

It makes a condusive work environment and it helps succesfull implementation of the project. Although there are no changes and updates for aims and objectives, but there are changes and updates for the requirements. This is happened because several requirements are not necessary to do, meanwhile there are new requirements proposed by the client. For example, the 'append data' use case has been changed into 'data cleaning' use case because the IT consultant of Amigo Group asked the writer to make a data cleaning process in the system. For Amigo Group's staff, a data cleaning process is considered as more important feature than appending data into database. As the consequences, the system will use a manual data entry for the future use. The client is understand about the project being faced by the writer is a Final Year Project II project with limited time to do code implementation and a detailed project report. This understanding makes both sides to communicate well and resulting in the 16 agreed use cases.

Finally, in the end of this project, both the writer and the client can meet the aims and objectives of the project. Both sides are satisfied.

6.3 Work Items

The project is able to meet its aim, objectives and requirements before the end period of Final Year Project II. The proposed schedule on the Gant Chart included in Appendix B is used to help the writer arranges the time to finish all of the work item list. It also increase spirit of the writer every time a work item is finished before the estimated completion time. At the end of this project, the actual completion time is faster than the estimated completion time. The writer is successfully finished all the work about ten days before the end period of Final Year Project II project. Figure 6.1 below shows a 'project burndown' chart between estimated completion time and actual hours for each of work item.

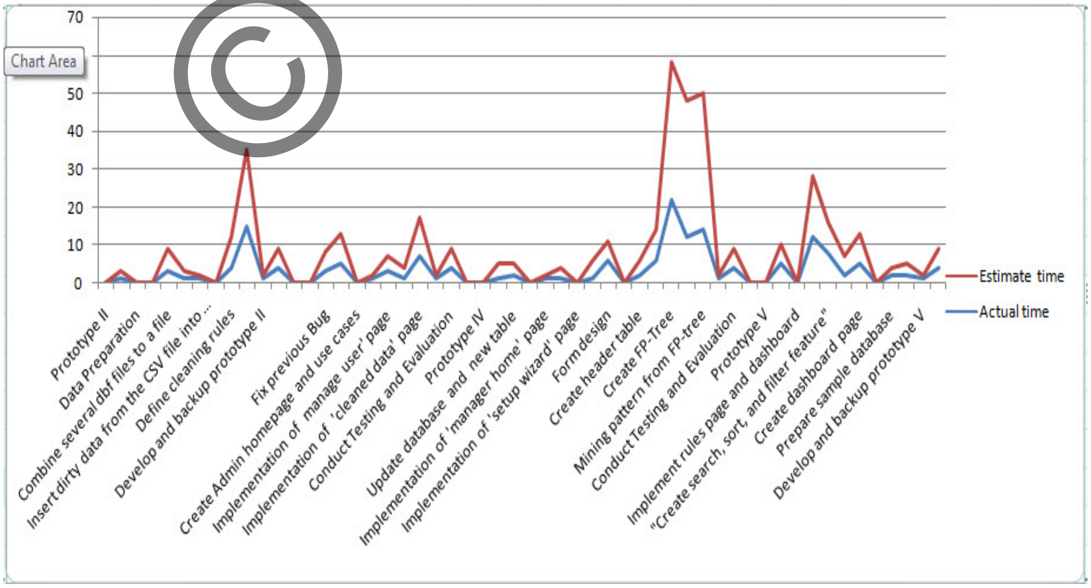


Figure 6.1 Project burndown charts between estimate hours and actual hours.

In Figure 6.1, X-asis is the name of the work item included in the work item list, meanwhile Y-axis is number of hours. There are big differences between the actual implementation time and estimation time for the main implementation processes in this project. Defining cleaning rules, implementation of 'data cleaning' use case, and the implementation of FP-Growth algorithm take faster completion time than the estimated time. This is happened because the writer has predicted what processes will take long period to complete and then the writer set those processes for a longer estimated time that it should be. By doing this, the writer can avoid 'lack of time' problem in the real implementation time of those work items, so that the writer can finish this project on track and on time.

6.4 Successes

For the writer, this project is considered as a succesfull project. The reason is because this is the first time the writer works in a data mining project for a real company. Besides, there are so many aspects that makes this project become an enjoyable project to be done. What went right on the project ? Here is a list that contain some success aspects:

- The problem is clearly defined in the beginning of the project and that makes the scopes and the requirements become easy to define.
- Support from Amigo Group is considered as a success aspect, this has been proven with their readiness to send the raw transactional database. The raw data sent was the sales transactions data from IT department.
- Communication between client and the writer is going well and both sides can understand the needs of each other.
- There is no payment or transaction cost spent during implementation of this project. The cost spent is just for the electricity. The time and power to do this project are not considered as a cost because this is considered as an effort that should be done by the writer in order to complete the project and his dual degree program.

- The time management for prototyping is going well and the actual process is faster than the estimated process, the only over-scheduled task is the 'visual aid assignment' completion time.
- The actual implementation of data cleaning and FP-Growth algorithm take shorter time than the estimated time. This is good because there is a problem to find appropriate literatures and sources in early stage of this project. The problem can be covered by having consultation with the supervisor and learn the algorithm through the online forum together with the community.
- The responds and guidances from the supervisors are also considered as a success factor that makes this project success.
- In the end, all the requirements are completed and all the use cases are made succesfully without any complaints (until this stage) from both sides (writer and client). This is the main success of the project.

6.5 Shortcomings

While there are so many success aspects in this project, there are also some aspects that went badly during the whole process of this project. Here is a list that contain some bad aspects (what went wrong):

- At the very early phase of this project, the writer is finding difficulty to get the sample data from Amigo Group until the writer made decision to go directly to the Amigo's branches. After this visit, there is no more diffculty to get the database because the writer knows where to go if he needs another data.
- Working alone in a new project is always hard, especially when there are just little sources and people around the work environment that understand the problem faced by the writer.
- The data provided from Amigo Group's is raw data and needs to be prepared better. The data they have given to the writer is not the data from the existing information system but it comes from the previous

information system, this makes the data structures are very bad and contain lots of multiple data inside the tables.

- Defining the cleaning rules and starting the code implementation of FP-Growth algorithm is really hard at the beginning. The writer can not find any sources related with a web-based association rule project using FP-Growth algorithm. The original FP-Growth algorithm uses OOP programming and datasets as their data sources, the understanding to make those codes work in a web-based + RDBMS system takes much enough time.
- Three until four hours of data cleaning process must be done several times in the implementation phase of data cleaning method. There are some mistakes occur in each attempt of data cleaning process. For example, the first attempt of data cleaning process didn’t include the ‘IDtoko’ column in the ‘nidump’ table, so the writer should repeat the cleaning process again. There are some test to ensure that the cleaning process run properly. In the writer opinion, this step is considered as an exhausting testing step because the writer should wait until the whole cleaning process has been completed before another modules can be implemented.
- The difficulty to implement efficient process time to make items combination as the rules generated.
- The writer needs more inputs from other developer or programmer which has experiences in data mining project.
- Another factor that makes the implementation distracted is a natural disaster, the Mount Merapi’s eruptions which happened while the writer is trying to finished all the code implementation of the use cases has ‘forced’ the writer to do a better time management. In the morning to evening, the writer works as a volunteer, meanwhile at the night, the writer is doing the planned work item that should be done.

6.6 Discussion

This project has been a grreat experience for the writer to work on a web-based data mining system. The association rule mining is one of the basic knowledge in data mining area. There are many other discilpines in data mining like classification, clustering and prediction analysis. The writer will take the good aspects of this BIT305 project to works on another project and improves the bad ascpects in order to make them better. The scope, time, and project integration management are already good but still must be improved to be better. The cost management will be a real challenge in a software development and must be declared clearly before the implementation of the system.

For another association rules mining in another project, the writer might consider to use desktop application with also considering the user requirements. The reason is because desktop application is faster in case of utilizing the main memory of the CPU. It doesn’t have to do request – receive operations through the web server which can be located on other computer. Besides, a desktop application is usually easier to carry (handy) than a web-based application. Another concern is the database used, using MySQL is a good idea for a web-based application, but datasets in CSV format or desktop database might be more appropriate for large data.

The human resources aspect is a very important aspect. By adding numbers of programmers to work in a data mining project, the writer thinks that the implementation will be easier, the problem solving will be faster, and the result will be better. The programmers can do brainstorming, system testing, and fixing the bug continously.

Utilizing open source software is also a good idea for creating a data mining system. Pentaho, Rapid Miner and Weka are examples of free softwares that were made by the developers all around the world. By using those open source softwares, the programmers can save much time and concentrate on the implementation of formatting the company data into datasets that are understood by each software.

6.7 Learning Outcomes

The writer have learnt so many things in this project. For the writer, the most important learning outcome is the learning processes during the whole project with helps from supervisors, new clients and also friends. By working with them, the author can improve knowledges, learn new point of views, get new friends and some self-improvements like improvement of patience, communication skills, listening skills and disciplines.

Another learning outcome from this project is the importance of data cleaning techniques in a data mining project. The data cleaning process must be done before the data can be used for a specific algorithm because processing the wrong data can result in a wrong result too. The time needs to clean the data itself is various, depending on the size of the data, the cleaning techniques used, and also the ability of the computer which run the data cleaning process.

In the literature review stage, the writer learns about the importance of selectng a proper algorithm to the system. This project is using FP-Grwoth algorithm with reason that the most commonly used algorithm, which is Apriori, will take longer time in processing larger data than FP-Growth algorithm. The selection of the algorithm can be done by doing contionus search about the algorithms and literature review about its implementation.

Learning people's perspective and skills are also considered as one of the learning outcomes. In this project, the writer learn how to deal with people from different background which has different skills and different understanding about the importance of this project. By understanding what they want and communicate what the writer's purpose nicely, the communication line is going well. The negotiation about boundaries of the system, purpose of the system, requirements gathering stage, and the transactional data sharing are agreed and achieved in a good relation between the stakeholders.

6.8 Project Management

Like stated in Schwalbe and College (2007), a project manager framework is consist of stakeholder needs and expectations + 9 knowledge areas + tools and techniques in order to achieve a project success. The project management tools and techniques help the project execution of the project become more planned and organized to do. Actually, there are some areas from the 9 knowledge areas of Project Management which are not clearly exist in this project, namely project cost management and procurement management. The reason is because this project is for academic purpose. This project is considered as a duty that must be finished in order to complete the writer's bachelor degree.

The other knowledge areas are highly affected by the project management tools and techniques. The creation of WBS and requirement documentation help the writer to identify the scope of the project. Gantt charts are used to define the time estimation for the implementation of the project. Testing is done every end of the methodologies used, which is prototype. The ability to communicate with others and self-motivating are techniques used in communication and human resource management.

All of these tools and techniques are really help the execution of the project. The impact of using them is that the writer can give more focus to the detailed work item, what should be done, when should it be finished, who will affected by the action, how to test the credibility of the code, and what risk should be consider after doing the specific work item. In the end, a more detailed and organized work item will be produce on a better result too.

6.9 Cost-Effectiveness

In some aspect, this project is considered as a quite economic project for the client. Half of the softwares and hardwares used in this project have been available in the writer's laptop before this project is started. Mostly, the softwares being used are open source softwares, which can be downloaded free from Internet. Time spent for efforts has reached 169 hours during the last four months of this project. Indirect costs like internet connection, phone connection, and electricity spent during the implementation are handled by the writer. In short, the writer's cost is higher than the client.

This project is valuable for both the client and the writer, who works as the programmer. For the client, this project is a new 'analysis helper' system that can be used for marketing decision in a real business. For the writer, this project is considered as the last duty to finish his studies. In short, both sides are having their own purposes and advantages from the system.

Chapter 7: Conclusion

7.1 Introduction

The project has been done completely. The web-based application finalized after fifth development of prototypes. The finalized application covers 16 use cases has been achieved and ready to be used in a real implementation of the system. The following sections explore on the further improvement, future work and the summary of the project.

7.2 Further Improvement

During this project, there are some issues that were faced by the writer. These issues are already mentioned in the project review in chapter 5 and chapter 6. After all, there is still a module that might be implemented for the further improvement of the project. The module is the ‘append data’ use case which has been changed in Final Year Project II into ‘data cleaning’ use case. The ‘append data’ use case can be done in two methods. First method is using form and input them directly to the cleaned table (‘nota’ and ‘notaset’ table) and the second method will be to convert CSV file or DBF file containing data directly into MySQL table.

Other improvement that can be made is improvement in the data availability. The data available in this system is only from the years of 2006 until 2008. There should be data from years 2009 and 2010 too because these will keep the system up-to-date. Data from other shops can be inserted into database too (not only ‘Bimbo’) and it will make the system’s availability become larger.

7.3 Future Work

This system is limited with producing only a pair of item in the association rules generated. For the future system, the programmer can improve this module into more combination items to produce more association rules result. Of course, the programmer should consider about the process time of execution and the validity of the result generated by these combinations. Producing lots of results that has confidence less than 10-15% in a very long period are not considered as improvement for the system. Faster process execution and more valid combination result is considered as an improvement.

With the consideration of the system's name, which is Amigo Data Mining System, the project can be expanded with more data mining algorithms and methodologies beside association rule's discovery. This is considered as an advantage of having data mining system for the client's future system.

7.4 Summary

The proposed web-based application for this project – The Amigo Data Mining System is to be used by two groups of users – administrators and managers. The system consists of 16 use cases. All of the use cases have been implemented in the end of this project. The result of this system reflects pair of association rules from Amigo Group sales transactions data over three years. Pattern of items that frequently purchased together can be analyzed to explore on marketing strategies and items mixing. For example, the results of Idul Fitri 2006 until 2008 data show a changing trend of customer wear. In recent years, people tend to buy more casual clothes instead of Moslem wear. Based on these result, Amigo Group can prepare the next Idul Fitri Feast Day and create a catalog design to attract customers buy items that are usually purchased together. In other way, the catalog design can also be utilized to mix items or rules that have medium confidence percentages in order to increase sales for those items.

This is the first time for the author to build a data mining application which will be used for real enterprise use. To work with the requirements from the client is also a challenge. Lessons are learned by doing this project. Some mistakes are made along the way, but new experiences are also reached. This project has been a very good experience for the writer's to improve technical skill and communication skill. More importantly, the aims and objectives of the system are now completed. Thanks to all people who made this project successfully finished.

Bibliography

[Ass10] ‘Association Rules (Market Basket Analysis)’, viewed 17 August 2010,

<http://www.google.co.id/url?sa=t&source=web&cd=1&ved=0CBcQFjAA&url=http%3A%2F%2Fwww2.hawaii.edu%2F~milica%2Fics491_spring04%2FAssociationRules.doc&rct=j&q=association%20rules%20%28Market%20Basket%20Analysis%29%20filetype%3Adoc&ei=Yf7STN6HHsP4cdr13bQO&usg=AFQjCNGzcY6pDEMcXRVOuMF_uXk_DNcIBw&sig2=nAihySY0Mpqdq-_xEvMrJQ&cad=rja>

[Bar10] BarelyFitz Designs, ‘Learn CSS Positioning in ten steps’, viewed 9 September 2010,

<http://www.barelyfitz.com/screencast/html-training/css/positioning/>

[Bor10] Borgelt, C 2010, ‘An Implementation of FP-Growth Algorithm’, viewed 19 August 2010,

<http://www.google.co.id/url?sa=t&source=web&cd=2&ved=0CCcQFjAB&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.162.1209%26rep%3Drep1%26type%3Dpdf&rct=j&q=An%20Implementation%20of%20the%20FP-growth%20Algorithm&ei=Pd3gTLCMK4PJcae3rZcM&usg=AFQjCNGweS3MfgrAsoN9sCCVh_n3P0X-pg&sig2=_381b7TIHzjAJf3TEisx2w&cad=rja>

[Bpk08] Bala, PK 2008 ‘A technique for mining generalized quantitative association rules for retail inventory management’, viewed 6 May 2010,

<http://findarticles.com/p/articles/mi_6766/is_2_8/ai_n31125903/pg_4/?tag=content;col1>

[Dep10] Department of Computer Science and Engineering. The Chinese University of Hong Kong, ‘Association Rules/Frequent Itemsets’, viewed 18 August 2010,

<http://appsrv.cse.cuhk.edu.hk/~kdd/program.html>

[Fre10] 'Frequent Pattern Mining Implementations', viewed 17 August 2010, <http://adrem.ua.ac.be/~goethals/software/>

[Ind10] 'Industry Applications of Data Mining', viewed 5 May 2010, <http://www.pearsonhighered.com/samplechapter/0130862711.pdf>

[Jel05] Jelsoft Enterprises Ltd 2005, 'Delete ALL files in directory' in forum 'Codewalkers', viewed 22 November 2010,

<http://forums.codewalkers.com/php-coding-7/delete-all-files-in-directory-714057.html>

[JJ10] Jacob, H & Jonas, H, 'Triminds – Enable and Disable', viewed 21 September 2010,

<http://www.htmlgoodies.com/tutorials/forms/article.php/3479101/Triminds---Enable-and-Disable.htm>

[Kas10] Kaskus 2010, 'Rumahnya Para Data Mining Enthusiast', viewed 11 September 2010,

<http://www.kaskus.us/showthread.php?t=4057452&page=8>

[Lar09] Lars 2009, '75+ Tools for Visualizing your Data, CSS, Flash, jQuery', viewed 9 September 2010,

<http://www.tripwiremagazine.com/2009/12/70-tools-for-visualizing-your-data-css-flash-jquery-php.html>

[MM05] Margahny, MH & Mitwaly, AA D 2005, 'Fast Algorithm for Mining Association Rules', paper presented to the AIML 05 Conference, Cairo, Egypt, 19-21 December.

[Ore10] O'Reilly & Associates, 'Finding all element combinations of an array' in O'Reilly, 'PHP Cookbook', viewed 1 November 2010,

<http://docstore.mik.ua/orelly/webprog/pcook/ch04_25.htm>

[Rec00] 'Recharge Your Retail marketing Strategy with Market Basket Analysis', viewed 4 May 2010, <http://www.free-press-release.com/news/200906/1244023265.html>

[RIP10] Redlon, M, Intelligence, D & Prairie, MN 2010, 'A SAS® Market Basket Analysis Macro: The "Poor Man's Recommendation Engine', viewed 5 May 2010, <http://www2.sas.com/proceedings/sugi28/223-28.pdf>

[San03] Santoso, LW 2003, 'Pembuatan Perangkat Lunak Data Mining untuk Penggalian Kaidah Asosiasi Menggunakan Metode Apriori', Jurnal Informatika, vol. 4, no. 2, pp. 49-56

[Sql07] 'SQL injection' 2007, viewed 17 September 2010, <http://miqrosoft.wordpress.com/sql-injection/>

[Sta10] StatSoft Electronic Statistics Textbook, 'Association Rules', viewed 18 August 2010, <http://www.statsoft.com/textbook/association-rules/?button=1>

[Swe10] Swedberg, K, 'Slicker Show and Hide', viewed 9 September 2010, <http://www.learningjquery.com/2006/09/slicker-show-and-hide>

[Tiz10] Tizag.com, 'mysql aggregate functions - sum()', viewed 17 September 2010, <http://www.tizag.com/mysqlTutorial/mysqlsum.php>

[Ver08] Verhein, F 2008, 'Frequent Pattern Growth (FP-Growth) Algorithm An Introduction', viewed 18 August 2010,

<http://www.florian.verhein.com/teaching/2008-01-09/fp-growth-presentation_v1%20%28handout%29.pdf>

[Vie09] Vielmetter, D 2009, 'Convert an HTML table to CSV using PHP', viewed 2 November 2010,

<http://davidvielmetter.com/?p=1357>

References

[Ake99] Akers, G 1999, 'What is Prototyping', viewed 3 May 2010, <http://www.umsl.edu/~sauterv/analysis/prototyping/proto.html>

[AMS+96] Agrawal, R, Mannila, H, Srikant, R, Toivonen, H & Verkarno ,AI 1996, 'Fast discovery of Association Rules', viewed 5 May 2010, <http://sci2s.ugr.es/keel/pdf/algorithm/capitulo/agrawal96.pdf>

[Bao07] Bao, X 2007, 'Mining Transaction / Order Data Using SAS® Enterprise Miner™ Association Node', viewed 7 May 2010, <http://www2.sas.com/proceedings/forum2007/132-2007.pdf>

[BL05] Budhi, GS & Lim, R 2005, 'Penggunaan Metode Fuzzy E-Covering untuk Analisa Market Basket pada Supermarket', Jurnal Informatika, vol. 6, no. 1, pp. 51-58

[Blm06] Bashin, L M 2006, 'Data Mining: A Competitive Tool in the Banking and Retail Industries', viewed 7 May 2010, <http://www.icai.org/resource_file/9935588-594.pdf>

[BSS06] Budhi, GS, Santoso, LW & Susanto, E 2006, 'Metode Market Basket Analysis menggunakan Algoritma Pincer Search untuk Sistem Pembantu Pengambilan Keputusan', 1st Seminar on Apllication and Research in Industrial Technology, SMART, Yogyakarta

[Com07] Computing, HN 2007, 'Incremental Prototyping', viewed 3 May 2010,

<http://www.sqa.org.uk/e-learning/IMAuthoring01CD/page_09.htm>

[Erw09] Erwin 2009, ‘Analisis Market Basket Dengan Algoritma Apriori dan FP-Growth’, Jurnal Generic, viewed 19 August 2010, <http://uppm.ilkom.unsri.ac.id/userfiles/JurnalVol_4_No_2_Juli_2009/6-Erwin.pdf>

[Gui06] Gutierrez, N 2006, ‘Demystifying Market Basket Analysis’ , viewed 18 August 2010,

<http://www.information-management.com/specialreports/20061031/1067598-1.html>

[Hen04] Henderson, S 2004, ‘It’s all too easy to forget’, viewed 22 August 2010, <http://www.independent.co.uk/life-style/health-and-families/health-news/its-all-too-easy-to-forget-568072.html>

[HK01] Han J & Kamber M 2001, ’Data Mining – concepts and techniques’, Morgan Kaufmann, San Fransisco.

[NXC09] Ngai, EWT, Xiu L & Chau, DCK 2009, ‘Application of data mining techniques in customer relationship management: A literature review and classification’, Elsevier, Expert Systems with Applications, vol. 36

[SC07] Schwalbe K & College A 2007, ’Information Technology Project Management Fifth Edition’, Course Technology, Boston

[Shi06] Shishirs 2006, ‘Data Mining for the Retail industry’, viewed 7 May 2010,

<http://blogs.msdn.com/b/shishirs/archive/2006/09/04/740491.aspx>

[Ver09] Verens, K 2009, ‘Data Tables and DataTables Plugin in jQuery 1.3 with PHP’, viewed 9 September 2010,

<http://www.packtpub.com/article/data-tables-datatables-plugin-jquery-1.3-with-php>

[Vya07] Vyas, R 2007, ‘Comparative Study of Apriori and FP-Growth Method’, viewed 20 August 2010,

<http://www.isi.it/progetti/giacs-school/files/Ranjana_Vyas_omsai_rv2.pdf>

[Xlm10] XLMiner 2010, ‘Association Rules’, viewed 5 May 2010,

<http://www.resample.com/xlminer/help/Assocrules/associationrules_intro.htm>

Bibliography

[Ass10] ‘Association Rules (Market Basket Analysis)’, viewed 17 August 2010,

<http://www.google.co.id/url?sa=t&source=web&cd=1&ved=0CBcQFjAA&url=http%3A%2F%2Fwww2.hawaii.edu%2F~milica%2Fics491_spring04%2FAssociationRules.doc&rct=j&q=association%20rules%20%28Market%20Basket%20Analysis%29%20filetype%3Adoc&ei=Yf7STN6HHsP4cdr13bQO&usg=AFQjCNGzcY6pDEMcXRVOuMF_uXk_DNcIBw&sig2=nAihySY0Mpqdq-_xEvMrJQ&cad=rja>

[Bar10] BarelyFitz Designs, ‘Learn CSS Positioning in ten steps’, viewed 9 September 2010,

<http://www.barelyfitz.com/screencast/html-training/css/positioning/>

[Bor10] Borgelt, C 2010, ‘An Implementation of FP-Growth Algorithm’, viewed 19 August 2010,

<http://www.google.co.id/url?sa=t&source=web&cd=2&ved=0CCcQFjAB&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.162.1209%26rep%3Drep1%26type%3Dpdf&rct=j&q=An%20Implementation%20of%20the%20FP-growth%20Algorithm&ei=Pd3gTLCMK4PJcae3rZcM&usg=AFQjCNGweS3MfgrAsoN9sCCVh_n3P0X-pg&sig2=_381b7TIHzjAJf3TEisx2w&cad=rja>

[Bpk08] Bala, PK 2008 ‘A technique for mining generalized quantitative association rules for retail inventory management’, viewed 6 May 2010,

<http://findarticles.com/p/articles/mi_6766/is_2_8/ai_n31125903/pg_4/?tag=content;col1>

[Dep10] Department of Computer Science and Engineering. The Chinese University of Hong Kong, ‘Association Rules/Frequent Itemsets’, viewed 18 August 2010,

<http://appsrv.cse.cuhk.edu.hk/~kdd/program.html>

[Fre10] ‘Frequent Pattern Mining Implementations’, viewed 17 August 2010, <http://adrem.ua.ac.be/~goethals/software/>

[Ind10] ‘Industry Applications of Data Mining’, viewed 5 May 2010, <http://www.pearsonhighered.com/samplechapter/0130862711.pdf>

[Jel05] Jelsoft Enterprises Ltd 2005, ‘Delete ALL files in directory’ in forum ‘Codewalkers’, viewed 22 November 2010,

<http://forums.codewalkers.com/php-coding-7/delete-all-files-in-directory-714057.html>

[JJ10] Jacob, H & Jonas, H, ‘Triminds – Enable and Disable’, viewed 21 September 2010,

<http://www.htmlgoodies.com/tutorials/forms/article.php/3479101/Triminds---Enable-and-Disable.htm>

[Kas10] Kaskus 2010, ‘Rumahnya Para Data Mining Enthusiast’, viewed 11 September 2010,

<http://www.kaskus.us/showthread.php?t=4057452&page=8>

[Lar09] Lars 2009, ‘75+ Tools for Visualizing your Data, CSS, Flash, jQuery’, viewed 9 September 2010,

<http://www.tripwiremagazine.com/2009/12/70-tools-for-visualizing-your-data-css-flash-jquery-php.html>

[MM05] Margahny, MH & Mitwaly, AA D 2005, ‘Fast Algorithm for Mining Association Rules’, paper presented to the AIML 05 Conference, Cairo, Egypt, 19-21 December.

[Ore10] O'Reilly & Associates, ‘Finding all element combinations of an array’ in O’Reilly, ‘PHP Cookbook’, viewed 1 November 2010,

<http://docstore.mik.ua/orelly/webprog/pcook/ch04_25.htm>

[Rec00] ‘Recharge Your Retail marketing Strategy with Market Basket Analysis’, viewed 4 May 2010,
<http://www.free-press-release.com/news/200906/1244023265.html>

[RIP10] Redlon, M, Intelligence, D & Prairie, MN 2010, ‘A SAS® Market Basket Analysis Macro: The “Poor Man’s Recommendation Engine’, viewed 5 May 2010,
<http://www2.sas.com/proceedings/sugi28/223-28.pdf>

[San03] Santoso, LW 2003, ‘Pembuatan Perangkat Lunak Data Mining untuk Penggalian Kaidah Asosiasi Menggunakan Metode Apriori’, Jurnal Informatika, vol. 4, no. 2, pp. 49-56

[Sql07] ‘SQL injection’ 2007, viewed 17 September 2010,
<http://miqrosoft.wordpress.com/sql-injection/>

[Sta10] StatSoft Electronic Statistics Textbook, ‘Association Rules’, viewed 18 August 2010,
<http://www.statsoft.com/textbook/association-rules/?button=1>

[Swe10] Swedberg, K, ‘Slicker Show and Hide’, viewed 9 September 2010,
<http://www.learningjquery.com/2006/09/slicker-show-and-hide>

[Tiz10] Tizag.com, ‘mysql aggregate functions - sum()’, viewed 17 September 2010,
<http://www.tizag.com/mysqlTutorial/mysqlsum.php>

[Ver08] Verhein, F 2008, ‘Frequent Pattern Growth (FP-Growth) Algorithm An Introduction’, viewed 18 August 2010,

<http://www.florian.verhein.com/teaching/2008-01-09/fp-growth-presentation_v1%20%28handout%29.pdf>

[Vie09] Vielmetter, D 2009, 'Convert an HTML table to CSV using PHP', viewed 2 November 2010,

<http://davidvielmetter.com/?p=1357>

References

[Ake99] Akers, G 1999, ‘What is Prototyping’, viewed 3 May 2010, <http://www.umsl.edu/~sauterv/analysis/prototyping/proto.html>

[AMS+96] Agrawal, R, Mannila, H, Srikant, R, Toivonen, H & Verkarno ,AI 1996, ‘Fast discovery of Association Rules’, viewed 5 May 2010, <http://sci2s.ugr.es/keel/pdf/algorithm/capitulo/agrawal96.pdf>

[Bao07] Bao, X 2007, ‘Mining Transaction / Order Data Using SAS® Enterprise Miner™ Association Node’, viewed 7 May 2010, <http://www2.sas.com/proceedings/forum2007/132-2007.pdf>

[BL05] Budhi, GS & Lim, R 2005, ‘Penggunaan Metode Fuzzy E-Covering untuk Analisa Market Basket pada Supermarket’, Jurnal Informatika, vol. 6, no. 1, pp. 51-58

[Blm06] Bashin, L M 2006, ‘Data Mining: A Competitive Tool in the Banking and Retail Industries’, viewed 7 May 2010, <http://www.icai.org/resource_file/9935588-594.pdf>

[BSS06] Budhi, GS, Santoso, LW & Susanto, E 2006, ‘Metode Market Basket Analysis menggunakan Algoritma Pincer Search untuk Sistem Pembantu Pengambilan Keputusan’, 1st Seminar on Apllication and Research in Industrial Technology, SMART, Yogyakarta

[Com07] Computing, HN 2007, ‘Incremental Prototyping’, viewed 3 May 2010,

<http://www.sqa.org.uk/e-learning/IMAuthoring01CD/page_09.htm>

[Erw09] Erwin 2009, ‘Analisis Market Basket Dengan Algoritma Apriori dan FP-Growth’, Jurnal Generic, viewed 19 August 2010, <http://uppm.ilkom.unsri.ac.id/userfiles/JurnalVol_4_No_2_Juli_2009/6-Erwin.pdf>

[Gui06] Gutierrez, N 2006, ‘Demystifying Market Basket Analysis’ , viewed 18 August 2010,

<http://www.information-management.com/specialreports/20061031/1067598-1.html>

[Hen04] Henderson, S 2004, ‘It’s all too easy to forget’, viewed 22 August 2010, <http://www.independent.co.uk/life-style/health-and-families/health-news/its-all-too-easy-to-forget-568072.html>

[HK01] Han J & Kamber M 2001, ’Data Mining – concepts and techniques’, Morgan Kaufmann, San Fransisco.

[NXC09] Ngai, EWT, Xiu L & Chau, DCK 2009, ‘Application of data mining techniques in customer relationship management: A literature review and classification’, Elsevier, Expert Systems with Applications, vol. 36

[SC07] Schwalbe K & College A 2007, ’Information Technology Project Management Fifth Edition’, Course Technology, Boston

[Shi06] Shishirs 2006, ‘Data Mining for the Retail industry’, viewed 7 May 2010,

<http://blogs.msdn.com/b/shishirs/archive/2006/09/04/740491.aspx>

[Ver09] Verens, K 2009, ‘Data Tables and DataTables Plugin in jQuery 1.3 with PHP’, viewed 9 September 2010,

<http://www.packtpub.com/article/data-tables-datatables-plugin-jquery-1.3-with-php>

[Vya07] Vyas, R 2007, ‘Comparative Study of Apriori and FP-Growth Method’, viewed 20 August 2010,

<http://www.isi.it/progetti/giacs-school/files/Ranjana_Vyas_omsai_rv2.pdf>

[Xlm10] XLMiner 2010, ‘Association Rules’, viewed 5 May 2010,

<http://www.resample.com/xlminer/help/Assocrules/associationrules_intro.htm>