Software Engineering to Develop Patterns for Accumulated Values of Accounting Transactions

Sigit Widadi[1], Parwoto[2]

[1,2]Electromedical Engineering Program, Vocational Program, Universitas Muhammadiyah Yogyakarta
Applied Accounting Program, Vocational Program, Universitas Muhammadiyah Yogyakarta
Integrated Campus of UMY, Lingkar Selatan Street, Kasihan, Bantul Yogyakarta 55183
E-mail: swdskom@gmail.com, parwoto@umy.ac.id

Abstract – *Software for accounting information systems has its own peculiarities, because the transaction data is recorded in pairs over a period of time, must be traceable and can be seen again in the future at the time period desired by the user according to the accountability of financial information systems. This research presents a prototype development technique about inter-account configuration in the accounting structure and aims to provide technical assistance on how to develop a pattern of financial transaction information presentation on a double entry record system model using financial applications. Relational data set theory and concepts are used as a scientific explanation framework on how the correlation between accounts can be constructed mathematically so that the concepts can be understood by programmers when they develop software. The result of this study has been tested by applying to the financial applications in one of the microfinance institutions and it can work properly. The prototype development techniques of this research can be applied to the software development in the business field, such as services, manufacturing and trade by implementing in the form of the software configuration system tools. Furthermore, the financial application developers are expected to utilize this prototype development technique to produce more dynamic software as a financial data processor according to the accounting rules in the business organizations.*

***Keywords**: configuration, transaction mapping, double entry system, accounting information system, accumulation*

I. Introduction

In general, financial transactions in accounting are described in the account and software developers do not commonly master the problem of which the solution method is derived from the concept of accounting, so the needs desired by accounting practitioners still cannot be fulfilled by the software developers. The understanding of the concept of applied sciences across the field affects the efficiency of making program code patterns to solve the problem [7]. The main problem in developing a financial application with accounting standards is the recurring account readings and the accumulation of transaction value to the account at the above level [3]. In everyday implementation, the accumulation problem is formulated through a spreadsheet software, such as MS Excel. The tendency which occurs in that software is to change the program code if there is a change in the treatment of the formula used to solve the problem [1].

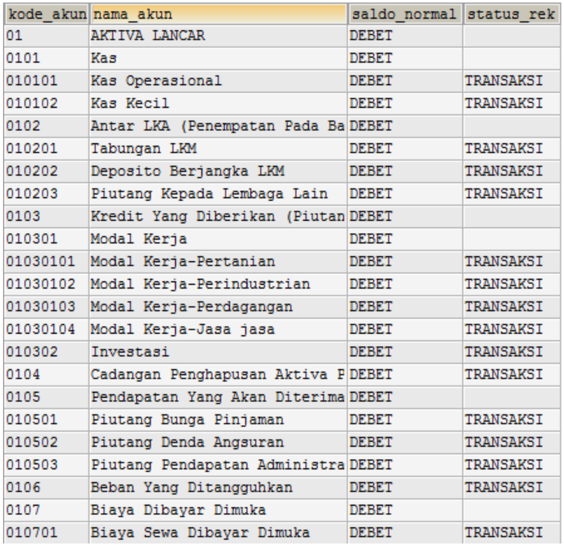
In an information system software built of programming languages and utilizing database systems, recoding is not easy to do and requires a much more complex step [7]. Changes in transaction formulas are inevitable in the treatment of transactional account records since they involve the dynamics of the financial system and the transaction service competition by financial

Manuscript received July 2017, revised September 2017

institutions, therefore the most feasible alternative is to create software engineering techniques capable of addressing the dynamics of applied application needs [9].

This research involves some basic terminologies in accountancy, such as:

1. An account i.e. numbers or estimates to record value changes
2. A transaction account i.e. an account directly related to the status record of value changes
3. An accumulated account i.e. an account that accommodate value changes of one or more accounts below it, also commonly called the header account [4].



kode_akun	nama_akun	saldo_normal	status_rek
01	AKTIVA LANCAR	DEBET	
0101	Kas	DEBET	
010101	Kas Operasional	DEBET	TRANSAKSI
010102	Kas Kecil	DEBET	TRANSAKSI
0102	Antar LKA (Penempatan Pada Ba	DEBET	
010201	Tabungan LKM	DEBET	TRANSAKSI
010202	Deposito Berjangka LKM	DEBET	TRANSAKSI
010203	Piutang Kepada Lembaga Lain	DEBET	TRANSAKSI
0103	Kredit Yang Diberikan (Piutan	DEBET	
010301	Modal Kerja	DEBET	
01030101	Modal Kerja-Pertanian	DEBET	TRANSAKSI
01030102	Modal Kerja-Perindustrian	DEBET	TRANSAKSI
01030103	Modal Kerja-Perdagangan	DEBET	TRANSAKSI
01030104	Modal Kerja-Jasa jasa	DEBET	TRANSAKSI
010302	Investasi	DEBET	TRANSAKSI
0104	Cadangan Penghapusan Aktiva P	DEBET	TRANSAKSI
0105	Pendapatan Yang Akan Diterima	DEBET	
010501	Piutang Bunga Pinjaman	DEBET	TRANSAKSI
010502	Piutang Denda Angsuran	DEBET	TRANSAKSI
010503	Piutang Pendapatan Administra	DEBET	TRANSAKSI
0106	Beban Yang Ditangguhkan	DEBET	TRANSAKSI
0107	Biaya Dibayar Dimuka	DEBET	
010701	Biaya Sewa Dibayar Dimuka	DEBET	TRANSAKSI

Fig 1. Example of an account structure table in a microfinance institution

Based on Figure 1, an account with code 010101 is an account to record the changes directly to *Kas Operasional* transaction value, where the changes will accumulate on the above header account, i.e. 0101 *(Kas)* and 01 *(Activa Lancar)*. Value changes or transactions that may occur are cash addition *(Kredit)* or cash reduction *(Debet)*. The next picture is an account with code 01030104 (*Modal Kerja Jasa-jasa*) which is a transaction account. The value change in that account will be accumulated to the above header account, that is 010301(*Modal Kerja*) and the value change of *Modal Kerja* account will be accumulated to account 0103 i.e. *Piutang Usaha* and the accumulated value changes ends on account 01 i.e. *Activa Lancar* [6].

This research tries to present a configuration technique or transaction mapping in order for programmers to avoid the customization of program codes when dealing with changing pattern of transaction accumulation caused by changes of accounting structure. If a business organization has a complex accounting structure, then customization is a job that requires high precision because a transaction is not only related to one account only, but at least two accounts for debit and credit paired record model. The techniques discussed in this study can be used by a financial application developer in the form of a transaction pattern mapping configuration feature to allow application users to create transactional relationships from a detail account to a header account in runtime mode.

The hypothesis of how to create an account mapping pattern in the example above is that the transaction account can be arranged in a relational database structure model between the detail account and the header account. Furthermore, the correctness of the hypothesis will be proven in this research steps. The proof of mapping the accumulation patterns of the transaction values from the detail account to the master account is expected to generate an output in the form of a prototype that contains database programming techniques to build financial information systems according to the accounting rules.

II. Theoretical Review

2.1 Basic Theories Of Accounting

In this study, a brief description of the basic theory of accounting is taken from a technological perspective on the basis of the positive theory of accounting itself. The technological perspective here is accounting as software technology that must be developed to achieve certain objectives by information engineering with the final result of techniques, methods, or principles in order to achieve these goals. The basic positive theories of accounting are used in this study because it only observes empirical facts to accounting phenomena as they are to support the accounting syntactic theory. The definition of syntactic theory is about how the technique of presenting effective and efficient financial statements [15]. The basics of accounting reasoning, financial reporting engineering, conceptual framework and accounting structure model are not specifically discussed in this study.

According to accounting experts, the basic concept used in the double entry model is the basic equation of accounting. The system has existed since the 15th century. The inventor of the double entry system is still a debate to date. Some claim that the inventor is Lucas Pacioli and some others claim that the inventor is Benedetto Cortrugly. The double entry system with recording of debit-credit transactions has been used worldwide to date. Debits are used as the signs of additions to the account, while credits are the signs of deductions on the account. With respect to Cash accounts, Debits are used to record reductions in Cash balances, while credits are used to record the addition of Cash balances. In the table of transaction records, the position of debit and credit are on the opposite side. Debit is on the left and credit is on the right. Every debit transaction must be followed by credit transactions that are opposed on the different side, so it must be understood about the transactional recording system to be done, whether on the debit side or on the credit side. To understand it, it is necessary to know the properties of the account group. If assets and expenses increase then the recording is on the debit side, if reduced it is recorded on the credit side. If incomes, capitals and debts increase then it is recorded on the credit side, if reduced then it is recorded on the debit side. If this logic is well understood it will make it easier in determining debit and credit on the development of the prototype of accounting transaction pattern.

2.2 The Basic Theories Of An Accounting Information Systems (AIS)

According to [2], accounting is defined as the process of recording, measuring and delivering economic information to be used as a basis for decision-making or wisdom. Meanwhile, according to [8], accounting is a process of collecting, recording, analyzing, summarizing, classifying and reporting financial transactions from an economic entity to provide financial information for users of the report useful for decision making. From these two basic theories, two important terms can be derived as the foundation of this research, namely:

1. Financial transaction

   Financial transaction is an economic activity that can be measured with a certain unit of money that can change the financial position of an organization or business entity.

2. Financial information

   Financial Information contains an explanation of the financial status of an organization. Based on the financial information it can be concluded about the liquidity of an organization. Liquidity is related to a variety of things including the availability and ability of the organization to meet its financial obligations.

The two basic terms above are very close to this research object because both terms have transaction and information keywords.

According to [7] *"An Accounting Information System (AIS) is a collection of activities of the organization responsible for providing financial information and the information obtained from data transactions for internal reporting purposes to managers to be used in current control and planning and future operations and also an external reporting to the shareholders, government, and other outside parties"*.

There are several other definitions of AIS proposed by some experts, but all of the definitions refer to the conclusion that AIS is processing transaction data into information. From the information system model below, it can be seen that the position of the object under study is the element of the accounting information system (AIS), where the information system is one of the input data subsystems sourced from internal transactions and external environment [10].

From Figure 2 it can be see that the accounting information system provides information supply to the database system to be used by the forecasting system. The definition of forecasting in general is prediction or forecast based on historical data through mathematical calculations. The prediction accuracy is influenced by many factors. The results of this study do not have a close relationship with the financial forecasting system.

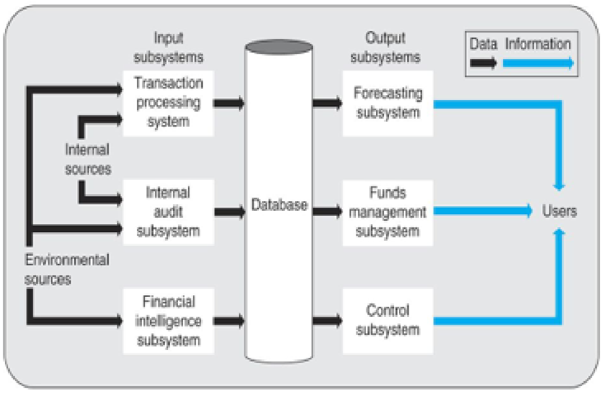


Fig 2. Financial Information System Model
Source: [10]

The characteristics of an AIS that distinguishes it from other subsystems:

1. AIS delivers the required tasks
2. Adheres to a relatively standard procedure
3. Handles detailed data
4. Has a historical focus

5. Provides minimal solving information

In accordance with the characteristics of points 3 and 4 above, the detailed data are the transaction data which are the value changes of an account. Meanwhile, the historical focus is how transaction data flow over time.

This research belongs to a design type, which is making prototype tool configuration to facilitate the adaptation of financial applications in facing the changes and the development of patterns and changes in variants of financial transaction accounts. This can be seen in the systems development life cycle of accounting information presented in Figure 3.

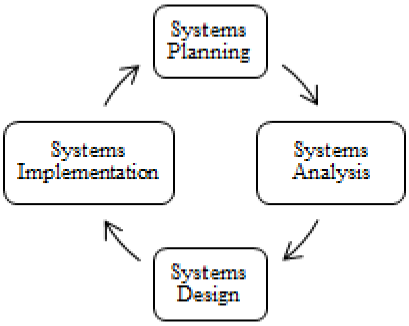


Fig. 3. Systems Development Life Cycle of Accounting Information. Source: [14]

In order not to fail in the application development process, the developer must study the scope of the object to be developed, choose the right tools during development and know the limits of possible development. To keep the resulting design perfectly valid, the designer generally evaluates and reviews the design periodically over a specified time interval. The objective is to determine whether the design results are reliable to the user and whether the user is satisfied with the resulting design.

2.3. Reconfiguration of the Systems

According to the experts, configuration means to form an arrangement or make settings on an object so that the object can work properly in a certain condition. Configuration has meaning according to the field under review [12]. This research aims to add the references of engineering techniques to software developers in building a financial transaction configuration tool, so that it requires understanding of the arrangement of the variables involved.

The technology makes it possible to reconfigure business practices even though those business practices are beyond a strategic plan to repack business strategies to fit a growing business pattern [12].

2.4. Relational Database

A relational database consists of a collection of tables, each of which is given a unique name [1]. A row in a table represents a relationship of several values. The relational database model is still the most popular data model to date because of its simplest form compared to other models [5]. This simple form makes the work of a software developer easier, i.e. when performing various data operations (select, insert, update, delete).

In the case of the accounting system, the account code is a unique naming of the values in the database table [4]. Referring to the set theory in basic mathematics, the relationship model between set A and set B can be denoted as follows [11]:

$R = (A, B, P\ (x, y))$

$A = \{2,3,4\}$

$B = \{3,4,5,6\}$

So the result is obtained as follows:

$R = \{(2,4), (2,6), (3,3), (3,6), (4,4)\}$

In the analogy of the accounting database model, the following diagram illustrates the relationship between the detail account and header account:

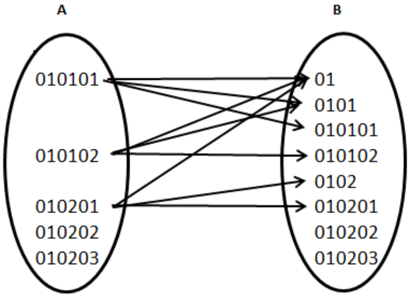


Fig. 4. An accumulated account mapping scheme

In the diagram above, it shows that the mapping function from domain A to range B produces the accumulated account code B from transaction account A. Domain is the origin area of the account, which is the detail account. Range is the result area, which is the header account. In accordance with the account code structure in accounting, value changes in the detail account will produce results on the header account.

III. SYSTEM PLANNING

The research methods are conducted in several stages (see Fig. 1), i.e. literature study, sampling data collection, data analysis, modelling of data analysis result, developing the prototype, testing the prototype, and conclusion.

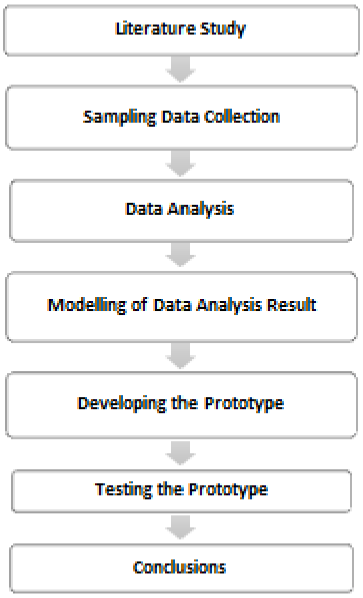


Fig. 5. Research Stages

These steps can be briefly described as follows:
1. Literature Study
This study is an interdisciplinary of science so it is necessary to review the various theories of each science, including theories about basic accounting, empirical accounting studies, engineering financial statement presentation, basic mathematics, relational database systems, programming languages and information management. Studies that studied previous research intend to find the theorems and techniques used by previous researchers in solving accounting problems using the information technology approach.
2. Sampling Data Collection
This research is a software development using a quantitative method, hence it requires samples of financial transaction data from an organization. The capacity of the data taken at least transaction data during one period of financial statement to be used as research object to get prediction reference for the next period. The selected organization is a microfinance institution by considering the amount of data that is not too large but has a fairly complete transaction pattern and managed using good data management.
3. Data Analysis
The sample data obtained from the financial business organization is a relational database system, so it is necessary to study the correlation between the tables and the structure of the database. Then the pattern of financial statement presentation sourced from the relational database is learnt to get description of the pattern between transaction and transaction accumulation pattern.
4. Creating the Design of Transaction and Accumulation Pattern
The data abstraction obtained from the step of analysis in the previous step is used to describe the model of transaction pattern and accumulation pattern and juxtaposed with the theorem about the engineering of financial statement presentation.
5. Developing the Prototype
If the model described is in accordance with the engineering techniques of financial statement presentation, then the development of the prototype is done using database applications and programming languages.
6. Testing the Prototype
The prototype produced is a set of configuration tools to map out the accumulated pattern of accounting transactions. Tests conducted on three groups of accounts, namely balance sheet, income statement, and cash flow statement. The test result is the transaction value taken from the sample data and not the simulative transactions to avoid creating interface to enter transaction to the database system.
7. Conclusions
This step describes the findings obtained during the development and testing of the prototype, including conclusions about engineering techniques, test results and the advantages and disadvantages of the prototype.

IV. RESULTS AND ANALYSIS

From the result of data analysis of transaction sample which is depicted using the set relation model, the result is obtained as shown in Figure 6.

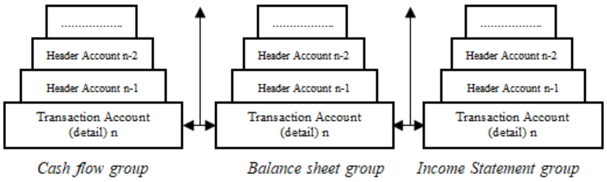


Fig. 6. Model of transaction account mapping of cash flow, balance sheet, and income statement

From Figure 6, changes to the account values in the balance sheet account group will result in changes in the values of the other account groups, i.e. the cash flow account group and the income statement account groups. Any changes to the detail account value in each group will result in a change in the value of the header account.

The table relation diagram of the mapping model in Figure 6 can be seen in Figure 7. The *accumulation_account* table is a duplicate of the *chart_of_account* list. Both tables are related using the mapping table.

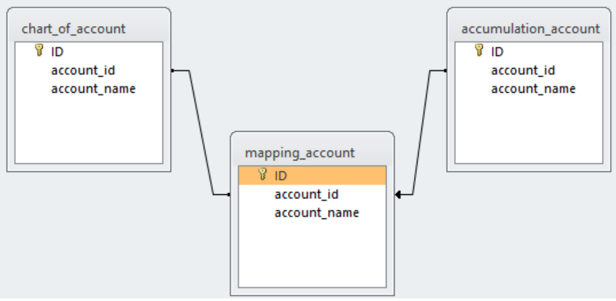
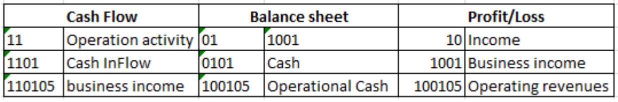


Fig. 7. Diagram relation of account code list table to accumulation account table

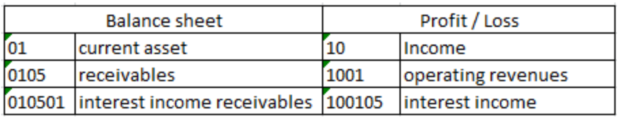
The description in the form of table design is presented in Figure 8, Figure 9, Figure 10 and Figure 11. Each image represents only one account as a sample of the table design in the relational database to be used by the prototype.



Cash Flow		Balance sheet		Profit/Loss	
11	Operation activity	01	1001	10	Income
1101	Cash InFlow	0101	Cash	1001	Business income
110105	business income	100105	Operational Cash	100105	Operating revenues

Fig. 8. Design of cash base mapping table of Interest Revenue Account to Operational Cash and Revenue of Income Interest

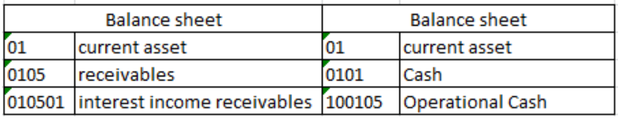
Figure 8 describes an Interest Income transaction borrowed from a customer (debtor) paid at the time of the event. It means that in that occasion there are additional balances on Cash.



Balance sheet		Profit / Loss	
01	current asset	10	Income
0105	receivables	1001	operating revenues
010501	interest income receivables	100105	interest income

Fig. 9. Design of accrual basis mapping table of Interest Income accounts to Interest Receivable accounts

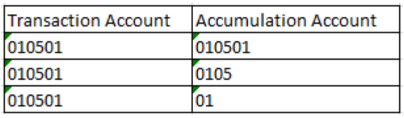
Figure 9 describes an addition to the interest receivable balance and does not involve Cash, because at the time of maturity there is no payment of interest on the loan from the customer that should be received by the bank or financial institution at the maturity date.



Balance sheet		Balance sheet	
01	current asset	01	current asset
0105	receivables	0101	Cash
010501	interest income receivables	100105	Operational Cash

Fig. 10. Design of cash base mapping table of Interest Receivable Accounts to Cash accounts

Figure 10 describes the transaction receipt of Interest Receivable by the bank, or payment of debt by the customer to the bank resulting in the addition of Cash balance. In this case Interest Income is not involved because it has been involved in the event of the addition of interest receivable balance.

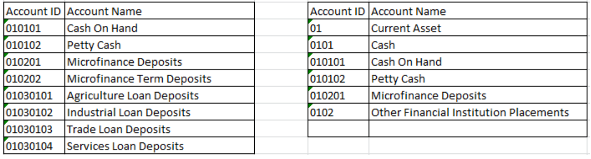


Transaction Account	Accumulation Account
010501	010501
010501	0105
010501	01

Fig. 11. Example of cash basis mapping table of Accounts Receivable Interest against Cash account

Figure 11 describes a sample transaction on account 010501. The transaction value will be accumulated on the header account above it, i.e. Accounts Receivable (0105) and Current Assets account (01).

The implementation of the prototype in the form of interface and syntax of Delphi 7 program can be seen in Figure 12, Figure 13 and Figure 14.

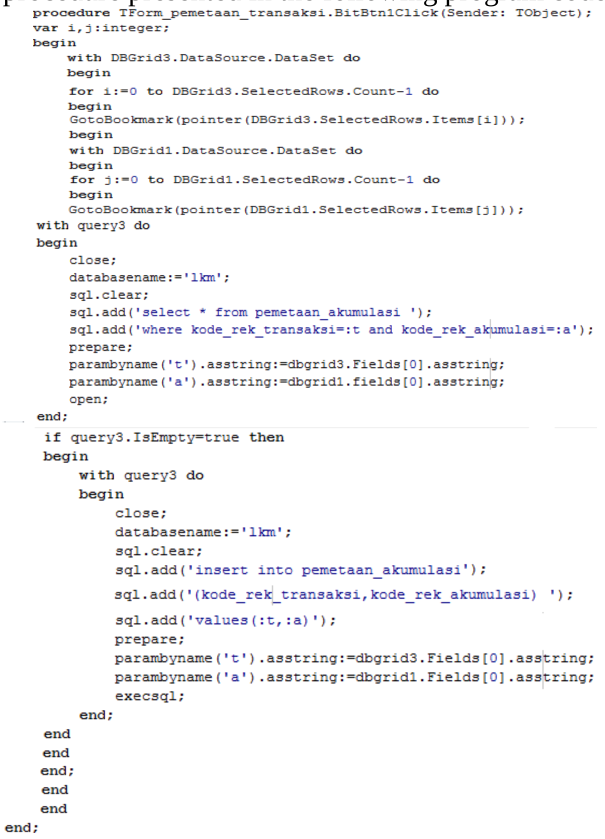


Account ID	Account Name
010101	Cash On Hand
010102	Petty Cash
010201	Microfinance Deposits
010202	Microfinance Term Deposits
01030101	Agriculture Loan Deposits
01030102	Industrial Loan Deposits
01030103	Trade Loan Deposits
01030104	Services Loan Deposits

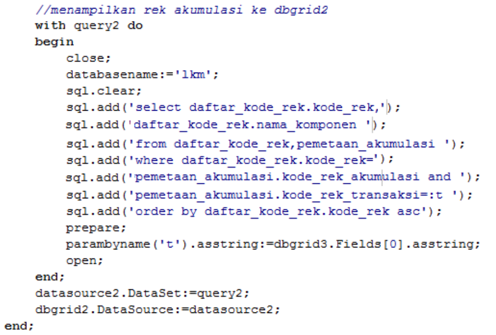
Account ID	Account Name
01	Current Asset
0101	Cash
010101	Cash On Hand
010102	Petty Cash
010201	Microfinance Deposits
0102	Other Financial Institution Placements

Fig. 12. Design of the prototype interface for the account pair

Execution of account mapping is done by the procedure presented in the following program code:



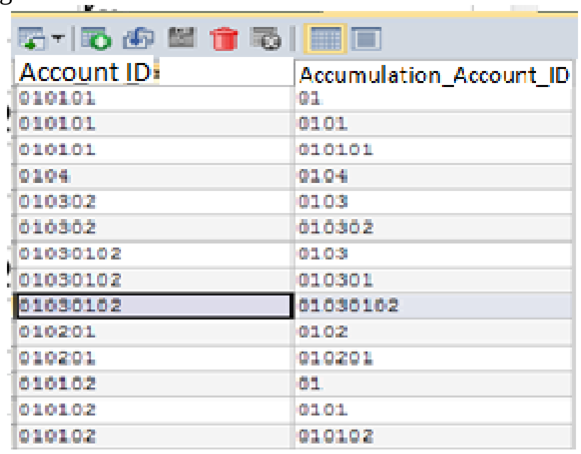
```
procedure TForm_pemetaan_transaksi.BitBtn1Click(Sender: TObject);
var i,j:integer;
begin
    with DBGrid3.DataSource.DataSet do
    begin
    for i:=0 to DBGrid3.SelectedRows.Count-1 do
    begin
    GotoBookmark(pointer(DBGrid3.SelectedRows.Items[i]));
    begin
    with DBGrid1.DataSource.DataSet do
    begin
    for j:=0 to DBGrid1.SelectedRows.Count-1 do
    begin
    GotoBookmark(pointer(DBGrid1.SelectedRows.Items[j]));
 with query3 do
 begin
    close;
    databasename:='lkm';
    sql.clear;
    sql.add('select * from pemetaan_akumulasi ');
    sql.add('where kode_rek_transaksi=:t and kode_rek_akumulasi=:a');
    prepare;
    parambyname('t').asstring:=dbgrid3.Fields[0].asstring;
    parambyname('a').asstring:=dbgrid1.fields[0].asstring;
    open;
 end;
  if query3.IsEmpty=true then
  begin
     with query3 do
     begin
        close;
        databasename:='lkm';
        sql.clear;
        sql.add('insert into pemetaan_akumulasi');
        sql.add('(kode_rek_transaksi,kode_rek_akumulasi) ');
        sql.add('values(:t,:a)');
        prepare;
        parambyname('t').asstring:=dbgrid3.Fields[0].asstring;
        parambyname('a').asstring:=dbgrid1.Fields[0].asstring;
        execsql;
     end;
  end
  end
  end;
  end
  end
end;
```



```
    //menampilkan rek akumulasi ke dbgrid2
    with query2 do
    begin
        close;
        databasename:='lkm';
        sql.clear;
        sql.add('select daftar_kode_rek.kode_rek,');
        sql.add('daftar_kode_rek.nama_komponen ');
        sql.add('from daftar_kode_rek,pemetaan_akumulasi ');
        sql.add('where daftar_kode_rek.kode_rek=');
        sql.add('pemetaan_akumulasi.kode_rek_akumulasi and ');
        sql.add('pemetaan_akumulasi.kode_rek_transaksi=:t ');
        sql.add('order by daftar_kode_rek.kode_rek asc');
        prepare;
        parambyname('t').asstring:=dbgrid3.Fields[0].asstring;
        open;
    end;
    datasource2.DataSet:=query2;
    dbgrid2.DataSource:=datasource2;
end;
```

Fig. 13. Codes of the mapping program of the transaction value accumulation

To create a transaction pattern mapping can use the same programming techniques as in the program code. The prototype test results in a relation table as shown in Figure 14.



Account ID	Accumulation_Account_ID
010101	01
010101	0101
010101	010101
0104	0104
010302	0103
010302	010302
01030102	0103
01030102	010301
01030102	01030102
010201	0102
010201	010201
010102	01
010102	0101
010102	010102

Fig. 14. Test results of making transaction accumulation patterns

Some of the images presented in the discussion section above are the results of the software prototype to create patterns of accumulation on accounting transactions. Example of testing the device configuration of the patterns of accumulated transaction values for the income statement format in the MFI Application at the Micro Finance Institution in Brosot Village, Galur Sub-District, Kulon Progo Regency, Yogyakarta.

V. Conclusion

Based on the research that has been done, some conclusions can be drawn at below.

1. Mapping the accumulated values of accounting transactions is able to give additional values to financial applications because it can reduce the customization process of program codes.
2. Transaction mapping using the techniques presented in this study can be successful if the user of the configuration tool used to create the transaction pattern understands the philosophy of accounting structure.
3. The prototype produced by this research is intended to present a visualization of the technique of developing the accounting transaction pattern so that in its implementation it needs to be reengineered in the database design and the software interface design that have been developed by the programmer.

References

[1] Abdul Kadir.2003, *Dasar Perancangan Dan Implementasi Database Relasional. Yogyakarta: Andi offset*

[2] Charles, Horngren. 2003. *Akuntansi Biaya Penekanan Manajerial Jilid satu*. Jakarta: PT. Indeks Kelompok Gramedia

[3] Dwi Prastowo dan Rifka Juliaty. 2005. *Edisi Revisi Analisis Laporan Keuangan* . Yogyakarta : Unit Penerbit Dan Percetakan AMP YKPN

[4] Harahap, Sofyan Sahri. 2010. *Teori Akuntansi. Edisi Revisi 2010* . Jakarta: Rajawali Press

[5] Howe, D.R. 1991. *Data analysis for data base design. Chapman and Hall*, Inc. New York

[6] JIBI. (2013, 11 27). *bisnis jateng*. Retrieved 09 10, 2017, from semarang.bisnis.com: http://semarang.bisnis.com/read/20131127/2/68767/lembaga-keuangan-mikro-masuk-radar-otoritas-jasa-keuangan

[7] Jogiyanto, HM. 2007. *Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis.* Yogyakarta: Andi Offset.

[8] Keputusan Menteri Keuangan RI (NO. 476 KMK. 01 1991 *Tentang Pedoman Kapitalisasi Barang Milik/Kekayaan Negara Dalam Sistem Akuntansi Pemerintah*, Jakarta: Sekretariat Kabinet Nomor 143

[9] Leman, 1997 *Metodologi Pengembangan Sistem Informasi*, PT. Elex Media Komputindo, Jakarta

[10] McLeod, Raymond Jr. 2002. *Sistem Informasi Manajemen Jilid 2*. PT Prenhallindo : Jakarta

[11] Mursita, Danang. 2002. *Matematika Dasar*. STTTELKOM: Bandung.

[12] Noel Carroll,Eoin Whelan,Ita Richardson. 2015. “*Understanding the Value of Business Process Configuration*”,University Of Limerick. Ireland.

[13] Sugiyono, 2013. *Metode Penelitian Kuantitatif*. Bandung:Alfabeta

[14] Sutabari, Tata, 2004, *Sistem Informasi Akuntansi*, Yogyakarta : Andi.

[15] Suwardjono, 2005, *Teori Akuntansi : Perekayasaan Pelaporan Keuangan Edisi ke-3, Badan Penerbitan Fakultas Ekonomi UGM, Yogyakarta*

AUTHORS’ INFORMATION

Sigit Widadi, received Bachelor of Computer Science degree from Department of Mathematic And Natural Sciences Gadjah Mada University in 2001, Student in Department of Magister Informatics Technology, Universitas Islam Indonesia, Yogyakarta, Indonesia. He is a Lecturer in Department of Medical Electronics Technology, Vocational Program, Universitas Muhammadiyah Yogyakarta, Indonesia. His research interests are in mobile application, enterprise architecture and medical information systems.