MODELING RISK CLUSTER BASED ON SENTIMENT ANALYSIS IN BAHASA INDONESIA FOR SME BUSINESS RISK ANALYSIS DOCUMENTS

IRFAN WAHYUDIN

GRADUATE SCHOOL
BOGOR AGRICULTURAL UNIVERSITY
BOGOR
2015

DECLARATION OF ORIGINALITY AND COPYRIGHT TRANSFER*

I hereby declare that the thesis entitled Modeling Risk Cluster based on Sentiment Analysis in Bahasa Indonesia for SME Business Risk Analysis Documents is my own work and to the best of my knowledge it contains no material previously published in any university. All of incorporated originated from other published as well as unpublished papers are stated clearly in the text as well as in the references.

Hereby, I state that the copyright to this paper is transferred to Bogor Agriculture University.

Bogor, June 2015

Irfan Wahyudin
Student Id G651130734

SUMMARY

IRFAN WAHYUDIN. Modeling Risk Cluster Based on Sentiment Analysis in Bahasa Indonesia for SME Business Financing Risk Analysis Documents. Supervised by TAUFIK DJATNA and WISNU ANANTA KUSUMA.

Currently, there are two risk analysis models that commonly used for business financing in banking industry, namely, the quantitative model and the qualitative model. The quantitative model are mostly implemented as an credit scoring system that consists of several accounting formulation that calculates the financial statement and business performance to determine the feasibility of bank customers in accepting loan. The second model, namely qualitative model, emphasizes the risk analysis, opinion, and mitigation from risk analyst to support the decision makers in accepting loan proposal.

From the observation through the Standard Operating Procedure (SOP) the quantitative model has some drawbacks in measuring the acceptance criteria, since the data are originated from the customer itself and vulnerable to have a manipulation, especially when the financial statement has no any inspection from external auditor. The second drawback is that the quantitative model tend to be subjective since the credit scoring calculation are performed by the marketing staff that stand sides to the customer. Hence, the qualitative model are deployed to overcome these drawbacks, where the analysis is objectively proceed by risk analysts. However, the implementation of qualitative model are not a hundred percent perfect, since the qualitative model neither has decision criteria nor risk measurement. Another issue is that the risk analysis documents that consist of risk opinion and mitigation from previous analysis, are not well managed. Actually, these documents are useful for the risk analyst to evaluate and relearn from the previous analysis.

In this research, the opinion or sentiment analysis against the risk analysis documents is conducted by modeling the risk cluster to help the risk analyst in refine the analysis. There are three tasks that have been conducted, those are clustering the risk analysis documents based on the term presence. Secondly is quantify the risk level within each cluster by measuring the term importance and sentiment score using TF-IDF and SentiWordNet 3.0 respectively. The task is eventually finished by evaluating the cluster quality using Silhouette function and examining the most frequent terms by its importance and sentiment. We also develop a prototype that enables risk analysts to retrieve the risk analysis documents by entering query terms and presents the level of risk from each document. The results has been shown that sentiment mining technique is effective and could be utilized to model risk cluster. This could be seen in how relevant is the cluster model with the 5Cs Credit criteria that commonly used in banking industry. In discussion, there are also some suggestions for the management on what criteria they should resharpen in conducting the qualitative model. By giving the risk score in each cluster, it is expected that the model also could be used as a benchmark to qualify the submitted loan proposal and assist the decision maker to have a better decision making.

Keywords: centroid optimation, K-Means clustering, opinion mining, risk analysis, SME business, sentiment analysis.

RINGKASAN

IRFAN WAHYUDIN. Pemodelan Klaster Risiko Berdasarkan Analisis Sentimen dalam Bahasa Indonesia pada Dokumen Analisa Risiko Pembiayaan Bisnis UMKM. Dibimbing oleh TAUFIK DJATNA dan WISNU ANANTA KUSUMA.

Saat ini, terdapat dua model analisa risiko yang umum digunakan untuk pembiayaan bisnis pada industri perbankan, yaitu, model kuantitatif dan model kualitatif. Model kuantitatif paling banyak diimplementasikan dalam bentuk sistem skoring kredit yang terdiri atas beberapa formula akuntansi yang menghitung laporan keuangan dan performa bisnis dari nasabah untuk menentukan feasibilitas dalam menerima fasilitas pinjaman. Adapun model kualitatif, menitikberatkan pada analisa risiko, opini, dan mitigasi dari analis risiko untuk mendukung pengambil keputusan dalam menyetujui proposal pinjaman.

Dari observasi terhadap Standar Operasional dan Prosedur (SOP) model kuantitatif memiliki beberapa kekurangan dalam menghitung kriteria persetujuan, dikarenakan data berasal dari nasabah itu sendiri dan rentan manipulasi, terutama untuk laporan keuangan yang tidak diaudit oleh audit eksternal. Kekurangan kedua adalah, model kuantitatif cenderung subjektif, dikarenakan perhitungan skoring kredit dilakukan oleh staf pemasaran yang sedikit banyak mempunyai keberpihakan kepada nasabah. Untuk itu, model kualitatif diimplementasikan untuk mengatasi kekurangan-kekurangan ini, di mana analisa dilakukan secara objektif oleh analis risiko. Bagaimanapun, implementasi dari model kualitatif ini, masih belum berjalan sempurna, dikarenakan model ini belum mempunyai kriteria pengambilan keputusan ataupun perhitungan tingkatan risiko. Masalah lain adalah dokumen analisa risiko yang terdiri atas opini dan mitigasi risiko dari pengajuan sebelumnya masih belum dikelola dengan baik. Dokumen-dokumen ini dapat berguna bagi analis risiko untuk mengevaluasi dan mempelajari analisa sebelumnya.

Pada penelitian ini, analisa sentimen terhadap dokumen analisa risiko dilakukan dengan memodelkan klaster risiko untuk membantu analis risiko dalam melakukan analisa. Terdapat tiga pekerjaan yang dilakukan, yang pertama adalah mengklaster dokumen analisa risiko berdasarkan kemunculan kata. Kedua adalah mengkuantifikasi tingkatan risiko di dalam tiap klaster dengan mengukur tingkat kepentingan kata dan skor sentimen menggunakan TF-IDF dan SentiWordNet 3.0. Selanjutnya diakhiri dengan evaluasi kualitas klaster menggunakan fungsi Silhouette dan mengukur tingkat kepentingan dan sentimen dari kata yang sering muncul. Sebuah prototipe aplikasi juga dibangun untuk mengunduh dokumen berdasarkan kata pencarian dan menampilkan dokumen terkait dengan level risikonya. Hasil dari penelitian menunjukkan teknik *sentiment mining* efektif dan dapat digunakan untuk memodelkan klaster risiko. Hal ini dapat dilihat dari relevansi model klaster dengan kriteria 5C Kredit yang umum digunakan pada industri perbankan. Pada bagian diskusi, terdapat beberapa saran untuk manajemen terkait kriteria analisa risiko yang harus dipertajam dalam implementasi model kualitatif. Skor risiko pada tiap klaster dapat digunakan sebagai tolak ukur untuk mengkualifikasi proposal dan membantu dalam mengambil keputusan.

Kata Kunci: analisa risiko, bisnis UMKM, K-Means clustering, optimasi centroid, opinion mining, sentiment analysis.

MODELING RISK CLUSTER BASED ON SENTIMENT ANALYSIS IN BAHASA INDONESIA FOR SME BUSINESS RISK ANALYSIS DOCUMENTS

Irfan Wahyudin

Thesis
As partial fulfillment of the requirements for the
Degree of Magister of Computer Science
in Computer Science Study Program

GRADUATE SCHOOL
BOGOR AGRICULTURAL UNIVERSITY
BOGOR
2015

Non-committee examiner : Irman Hermadi, S Kom, MS, PhD

Title	: Modeling Risk Cluster based on Sentiment Analysis in Bahasa Indonesia for SME Business Risk Analysis Documents
Name	: Irfan Wahyudin
Student Id	: G651130734

Approved by

Supervision Commissioner

Dr Eng Taufik Djatna, STp MSi
Chairman

Dr Eng Wisnu A Kusuma, ST, MT
Member

Acknowledged by

Head of Dept. Computer Science Graduate Program

Dr Eng Wisnu A Kusuma ST, MT

Dean of Graduate School

Dr Ir Dahrul Syah, MSc Agr

Examination date:

Passed date:

PREFACE

Alhamdulillah, the author praise to Allah Subhanahu Wa Ta'ala, The Almighty Lord, without His blessing and favour, this research would not be completed.

I would like to thank Dr Eng Taufik Djatna S Tp, MSi and Dr Eng Wisnu Ananta Kusuma ST MT, who have given insight and suggestion for the completion of this thesis book. This research was conducted in one of national private bank in Indonesia during December 2014 until May 2015, entitled Modeling Risk Cluster Based on Sentiment Analysis in Bahasa Indonesia for Small Medium Enterprise Business Financing Risk Analysis Documents. I also would like to appreciate the Risk Management Division from the bank where this research was taken place, for the knowledge sharing and for allowing the author using the risk analysis documents as a research sample. Also, thank you for all of the love, support and prayer of my father, my mother, my wife, my child, and all of family's member.

Hopefully, this research will brings many of benefits and has a contribution to science.

Bogor, June 2015

Irfan Wahyudin

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDIXES

1 INTRODUCTION

Background

Risk is defined as an uncertain event or set of events that would affect the objective, goal, and achievement of an organization (Newey 2014). Since it is inevitable, organizations must be aware and prepare to the risk that would occur by establishing a risk management, a group or division that consists of risk analysts. The main objective of the risk analysts is to measure the combination of the probability of a perceived threat or opportunity occurring and the magnitude of its impact on the organization's objectives. Thus, risk management is not only strongly related to threat or negativity, but it also related to opportunity or positivity as well. Thus, a good risk management would determine the success of an organization, not only in term of risk or threat avoidance, but also in term of obtaining success such as market opportunity and profit gaining.

In banking and financing industry, there are two models that have been widely used in implementing risk management namely quantitative model and qualitative model (Soares *et al.* 2011). These two models are commonly based on 5Cs credit criteria: Character, Capacity, Capital, Condition, and Collateral. In summary, these criteria tell a story about the debtor (Thomas 2000). For instance, the Character criteria reveals how good the track record and reputation of the management or the stakeholder of the business. From Capacity, the bank can ensured that the business is well managed by the management since it has a good production capacity. The Capital criteria shows that there is a reliable investment source to support the business running. Condition describes about the related business condition such as the market trend, the economic and atmosphere of related business support in the country where the business is about to run. In other word, Collateral also means as a guarantee that the debtor would repay the loan where they hand over the asset they have as the loan status is default.

These criteria are decisive for the top management in making a decision for the submitted financing proposal. Risk assessment for SME business in national bank in Indonesia is commonly dominated by the implementation of credit scoring system (quantitative model). The credit scoring model is basically consist of accounting formulation with time-cost dimension (Newey 2014) to measure the quality of the business. As an input, the information come from the debtor's financial report.

From the observation through the loan assesment's Standard Operating Procedure (SOP) in the bank where this research is conducted, it is found that the there are some leakages in measuring the acceptance criteria. In fact, the leakages are dominantly found in the credit scoring system, and customer's financial quality analysis which are part of what we called as quantitative model. In quantitative model, the objectivity is questioned, since it is performed by the marketing staff that stand sides to the customer. Moreover, the data are originated from the customer itself, vulnerable to have a manipulation, especially when the financial statement has no any inspection from the external auditor. Hence, the qualitative model are deployed to overcome these drawbacks, where the analysis is objectively proceed by some risk analysts. However, the implementation of qualitative model is not wholly reliable, which was indicated in the bank's Non Performing Loan ratio.

Speaking of information technology implementation, the credit scoring has been commonly integrated with the Loan Originating System (LOS). The LOS is basically a workflow system that records the administration process, Service Level Agreement (SLA), notice from bank analysts that yields a numeric score used by the decision makers in making a decision whether the proposal is going to be approved or declined. To complement this model, some banks also implement the dimensionless qualitative model where the information came from the risk analysts by commenting the proposal and financial information.

Currently, the government of Indonesia requires all national bank in Indonesia to support the SME business by providing working capital loan. According to the central bank of Indonesia (BI 2012), a business is classified as a small business when it has a net worth ranging from IDR 50,000,000.00 up to IDR 500,000,000.00, and also it has an annual worth ranging from IDR 300,000,000.00 up to IDR 2,500,000,000.00. A business is classified as the medium one, when it has a net worth ranging from 500,000,000.00 up to IDR 10,000,000,000.00, and also it has an annual worth ranging from IDR 2,500,000,000.00 up to IDR 50,000,000,000.00. As of June 2015, there are about 11,1 million SME businesses that obtained financing from national banks in Indonesia[1]. In the other hand, the central bank also insists the national banks to have risk management before granting a loan to minimize the risk that might occur such as loan default, as regulated in the central bank regulation (BI 2003). To measure the performance of the bank on channeling loan, one of main indicator that central bank use Non Performing Loan (NPL) ratio that would be issued by the bank through their annual report. Mostly, every bank would set up a target in how big the NPL ratio they should achieve annually.

As a case study, this research is taken place in a national private bank in Indonesia where the SME financing is one of their main business where their NPL ratio has not achieved the target, which is lower than 2%, for three consecutive years (2012: 2.66%; 2013: 2.26%; 2014: 2.78%). This problem has motivated the management through the New Year speech in 2015 to ask the risk analysts to have a refinement on analyzing the financing proposal. The first task to be done is by retrieving some information regarding the implementation of financing Standard Operating Procedure (SOP) as summarized in Figure 1.

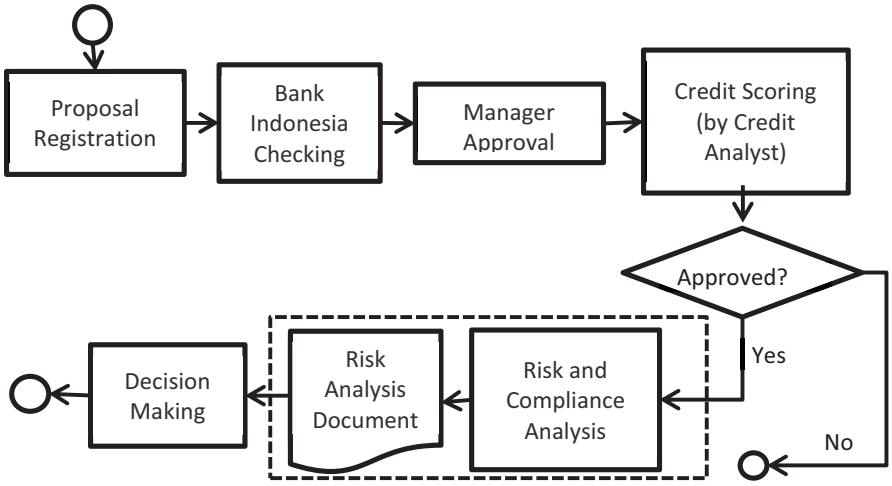


Figure 1 Brief workflows of SME financing Standard Operating Procedure

[1]Bank Indonesia, "*Laporan Perkembangan Kredit UMKM Triwulan 1 2015*", Juni 2015, http://www.bi.go.id/id/umkm/kredit/laporan/Documents/Laporan%20Perkembangan%20Kredit%20UMKM%20Triwulan%20I-2015.pdf

What became our concern is there is a drawback in implementing the two model of risk analysis. Eventually, the output from both models combined, was used by the decision makers in making a decision. The quantitative model already has an obvious decision criteria that the proposal to be approved or rejected. Yet, the qualitative model does not has this criteria. By reading the narrative opinion from the risk analysts the decision makers must read carefully on what the analysts have written down in the documents. All the decisions are only made based on their experiences and knowledge in banking particularly in business financing. Here is where the gap between both model and become the main motivation for this research to be conducted.

Problem Statements

From our observation, the bank where this research has taken place is already successfully used the quantitative model that implemented in their LOS. Problems were founded in the qualitative model where the implementation is conducted by delivering the risk opinion and risk mitigation through a risk analysis document. In this research, we formulate three problem statements to be solved in this research conduction: (1) What is the appropriate model for these SME business risk analysis documents, is it by clustering or by classifying the documents? (2) How to quantify the risk in the documents? (3) How to evaluate the model that have been built?

Objectives

There are three objectives of this research to address the problem statements above: (1) To perform clustering task to group the risk analysis documents, since there is no labeled documents yet, (2) To measure the risk level in each cluster using term-importance and sentiment weighting, and (3) To evaluate clustering task and sentiment measurement to reveal the implication with the criteria in assessing the loan risk.

Benefits

The benefit of this research is to enhance the Standard Operating Procedure in assesing written loan proposal. By providing the information regarding the risk group and its risk level, the expectation is that the conventional bank and particularly the risk management division would be able to compare the newly submitted loan proposal against the available risk model available in the risk cluster. Thus, prior to approving the loan proposal, the decision makers in the bank are expected to be able to decide that an SME business is considered as a high risk business or not. Another expectation is that the risk cluster and measurement can be used in helping the bank to evaluate and refine their current risk analysis implementation.

Boundaries

This research has several following boundaries: (1) According to the central bank regulation, risk opinion and mitigation from a risk analyst is required to all of SME financing proposal with the filing value above 5 billion Rupiahs. Therefore, all of documents that observed in this research were intended for the SME financing

with the filing value above 5 billion Rupiahs (2) There are 519 risk analysis documents used in this research that have been written by the risk analysts from 2013 to January 2014. Any risk opinion and mitigation beyond that period are not covered yet in this research (3) Although the cluster result may describe the level of the risk. However, for further implementation in Loan Originating System to support the decision makers, the risk management must define at what level is the approval criteria for decision makers, since this research did not cover any deeper analysis to define the approval criteria.

2 LITERATURE REVIEW

Risk Clustering

Clustering risks in financing are majorly dominated by the usage of numerical data which are mostly utilized to reflect the financial performance such as to forecast bankruptcy of a company, and detect default status of a financing (Kou *et* al 2014). In this research, since our effort is to utilize the risk opinion documents, We do not use those common approach that using the numerical data either to cluster or to classifiy the data. Instead, we use semantic approach to analyze the narrative opinion from risk analysis documents by conducting sentiment analysis.

In addition, they emphasize to evaluate six clustering algorithm that implemented for risk analysis in financing. And all of them are based on the structured data that mostly came from the financial statement of the bank customers. Their research also motivated from the previous works in risk analysis that implemented either supervised and unsupervised learning that also utilized the financial data to define risk in financing. Thus, by observing the unstructured data that came from the risk analysts opinion will be an interesting topic of research.

Sentiment Analysis

Along with the social media popularity such as Facebook and Twitter, sentiment analysis has been one of current interesting research topic. Most of the subjects are about on how to determine customer's preference, and people interest on particular product and event. As described in the related works, there are several techniques that have been developed as depicted in Figure 2.

There are two basic techniques that commonly used in conducting Sentiment analysis: machine learning and lexical learning (Medhat and Hasan 2012). Both techniques have some advantages and disadvantages. Machine learning offers high accuracy in determining the polarity of a document, while it is required a well classified corpus as a training data, and a lot of effort to train the model. Supervised machine learning techniques that are commonly used such as Support Vector Machine (Xu *et al.* 2011), Neural Networks (Ghiassi *et al*. 2013), and Naive Bayes (Li and Wu 2010). All of them requires training documents or corpora that already labeled. In contrary, the unsupervised machine learning let say the well known K-Means (Li and Wu 2010) and hierarchical clustering (Zhai *et al.* 2011) does not require the training documents. Another thing that differs both techniques is that the supervised learning is used as a classifier, to define the unlabeled object or document to the designated label or class that already known. While the

unsupervised learning, the label or class are not known yet, therefore, a technique such as clustering is used to define the proper number of labels that suitable for the observed dataset.

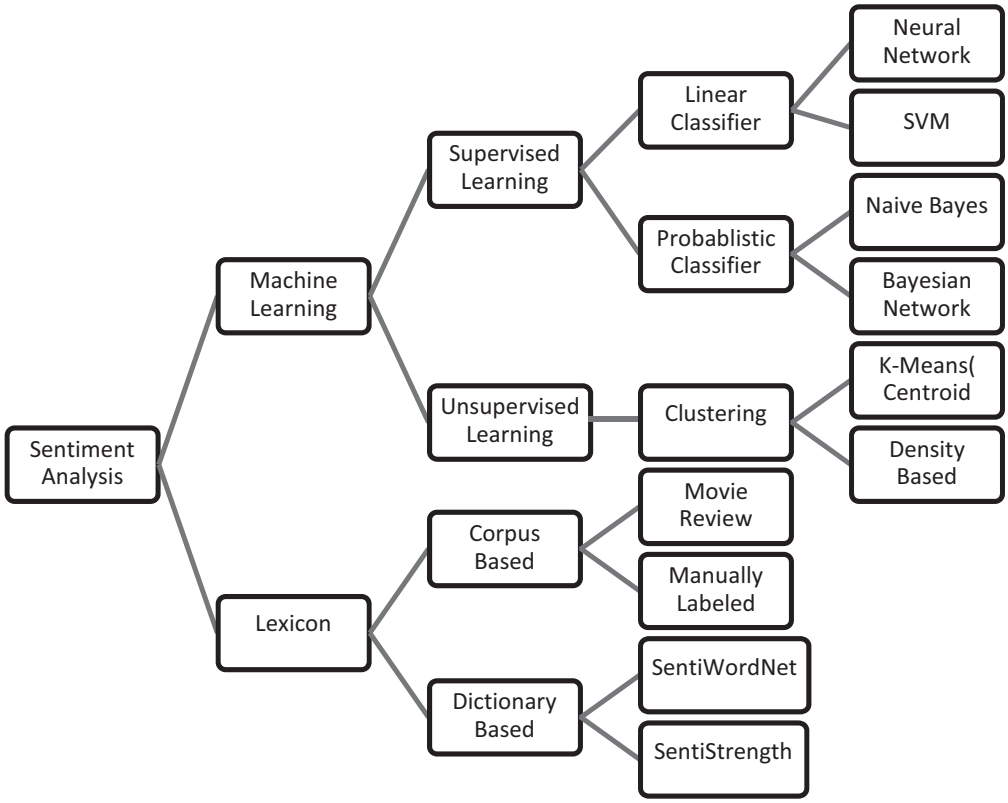


Figure 2 Sentiment analysis research method (Medhat and Hasan 2012)

In contrary, the lexical learning has lower accuracy in determining the polarity, but it offers efficiency to be used in a real time applications (Vinodhini and Chandrasekaran 2012). This is the reason why this research used the lexical based since there is no training data available and the desired model developed in this research is ready and immediately implemented in the real-time application to support the financing process. Lexical approach basically depends on a database contain terms, its part-of-speech, and its meaning. There are some available lexical resource available, however, after several searching attempts, there is only one lexical database that already succesfully utilized for Bahasa Indonesia using SentiWordNet 3.0 (Esuli and Sebastiani 2006). SentiWordNet has a structure that contains information about the terms in English as follows:

TermId | POS | Terms | Pos | Neg | Term Synonim | Term Usages

- TermId, contains the terms id number
- POS, contains tagsets. Namely, adverb (r), adjective (a), noun (n), verb (v)
- Terms, contains terms in English
- Pos, contains the positive polarity of the term
- Neg, contains the negative polarity of the term
- Term Synonim, contains several alternative terms or its synonim
- Term Usages, contains several terms example usages

There are various lexicon resources that can be utilized as a dictionary to determine the polarity of a term such as SentiWordNet (Esuli and Sebastiani 2006) which is derived from the well-known corpora namely WordNet, an English dictionary for word synonyms and antonyms. Next one is SentiStrength (Thelwall 2010) which is lexicon based technique, distributed as a desktop application tool that already combined with several popular supervised and unsupervised classifier algorithms: SVM, J48 classification tree, and Naive Bayes. Another kind is emoticon based lexicon, which is considered as the simplest one (Gonçalves *et al.* 2013). Unfortunately, most of the lexicon dictionaries and corpus resources are designated for English. Some efforts have been done to overcome this shortfall by translating either the observed corpus object (Denecke 2008) or translating the lexicon dictionary and the labeled corpus (Lunando and Purwarianti 2013). Moreover, the purpose of sentiment analysis is mostly dominated by on how businesses determine the opinion and judgment of their customers upon their products from open resources such as social media instead of performing sentiment analysis using a closed resources such as risk analyst opinion upon bank customers business.

Unlike in English, as of today there is only one international research publication utilized SWN 3.0 in Bahasa Indonesia that aimed to detect sarcasm in social media (Lunando and Purwarianti 2013). The translation problems were solved by utilizing tools and techniques such as Google Translate, Kateglo (Kamus Besar Bahasa Indonesia based dictionary). In addition, since the case study where this research was taken place is banking and finance, thus, We asked the banking experts for specific banking and finance terms that unavailable in the lexicon.

Part of Speech Tagging

Since the terms in SentiWordNet are labeled based on their position in the sentence, therefore it is a must to label the term so it can match its pair in the lexical database. This technique is called as Part of Speech Tagging (POS Tagging). In this research, a Hidden Markov Model (HMM) POS Tagging (Wicaksono and Purwarianti 2010) technique that used to label the terms. As reported in the paper, the accuracy on labeling the terms is about 96% higher than previously conducted POS Tagging research. The technique proposed in the paper is also equipped with the corpus resource, and Application Program Interface (API)[2] in Python programming that referred to the Wicaksono and Purwariantis paper and available for free download and use.

Singular Value Decomposition (SVD)

SVD is a factorization technique that decomposes matrix A that is a term-document matrix sized $t \times d$ dimension into three matrices, that formulate as follow:

$$A = U\Sigma V^T \quad (1)$$

Where U is orthonormal matrix sized $t \times t$ that represents the term concept, called as left singular value of A. Σ is a diagonal matrix sized sized $t \times d$, and the value of its diagonal is sorted decreasingly. The matrice D determines how big the dataset will reduced, ranging from 80% (Osinsky 2004) to 90% (Zhang and Dong 2004) scalar value of D will be used to obtain featured vectors from both matrix U

[2]Pebahasa-Bahasa Indonesia POS Tagger Python API based on Hidden Markov Model. https://github.com/pebbie/pebahasa

and matrix *V*. Matrix *V* is called as right singular value of *A*, that represents the document concept.

To get a better understanding on how SVD works for dimension reduction (Harikumar *et al.* 2012), here is some small example. Suppose there is a corpus contain several documents in Bahasa Indonesia d_1, d_2, d_3, d_4, d_5.

d_1: "*trend bisnis perumahan meningkat*"
d_2: "*minat masyarakat terhadap perumahan daerah abc meningkat*"
d_3: "*anak perusahaan pt xyz di abc mengalami kesulitan modal*"

Later, a term-documents binary matrix was constructed from above documents to represents the relation between the terms and the documents in terms of the presence of a term within a document. The presence of a term in a document will marked with *1* and the absence of a term will marked with *0*.

Table 1 Term-document matrix representation

	Documents		
Terms	**d1**	**d2**	**d3**
trend	*1*	*0*	*0*
bisnis	*1*	*0*	*0*
perumahan	*1*	*1*	*0*
meningkat	*1*	*1*	*0*
minat	*0*	*1*	*0*
masyarakat	*0*	*1*	*0*
Abc	*0*	*1*	*1*
anak	*0*	*0*	*1*
perusahaan	*0*	*0*	*1*
Xyz	*0*	*0*	*0*
kesulitan	*0*	*0*	*1*
modal	*0*	*0*	*1*

As seen in Table 1 a matrix with 16×5 sized to be used in SVD computation. The next step is to decompose the matrix into three matrices, the first matrix is U the left-singular value, that is an eigen vector from B that computed as $B = A^T A$. And the second one is Σ, the diagonal matrix or the square roots of eigenvalues of B. The last matrix is V, the right singular value or eigen vector from C obtained from the equation $B = AA^T$.

By implementing SVD, the expectation is to have a lower dimensional dataset that able to optimize the computation task such as clustering. To get the best low rank that represents threshold, denoted as $k - rank$ function of the original dataset A, below is the measurement comparing $\|A_k\|_F$ and $\|A_k\|$, Frobenius norm of sum of k items in Σ, and all items in Σ (Ozinski 2004; Zhang and Dong 2004).

$$k - rank(A, k) = \frac{\|A_k\|_F}{\|A\|_F} \quad (2)$$

Term Frequency-Inverse Document Frequency

In general, TF-IDF (Manning *et al.* 2009) is utilized to identify how important each term is in the corpus. TF or term frequency, reveals how important a term is used within a document by calculating its occurance. Denoted as $tf_{t,d}$, the frequency of term *t* in document *d* is defined in Formula 3. The more frequent a term used in a document means the term is important, or in other words, the term is the topic where the author would like to discuss. However, there may be a question regarding against the term high frequency usage, that is what if that term also used in many documents? This situation indicates that the term are less important, since it is not specifically mentioned in a document. Therefore, the IDF or Inverse Document Frequency calculation overcomes this problem by calculating the occurrence of a term compared to the number of documents where the term occurred.

$$tf_{t,d} = \frac{f_{t,d}}{argmax(tf_d)} \tag{3}$$

Denoted as idf_i, inverse document frequency for term *t* in the corpus *D* defined as formula (4).

$$idf_t = log_2 \frac{N}{n_t} \tag{4}$$

Where *N* is number of document available in corpus, n_t is occurrence number of term *t* in all documents in the corpus. There was a little modification in implementing the formula above. The term in the basic TF-IDF is selected distinctly based only on how the term spelled, and disregard the term preposition in sentence. Since the SWN 3.0 is also based on the term preposition, the term position in the term list was need to be added, which is obtained in the POS Tagging task.

K-Means Clustering

K-Means clustering (Mac Queen 1967), a widely used centroid based on a partitioning algorithm which is used in order to find how many risk cluster exist in the corpus. The algorithm is considered as a fast, simple, and effective to solve many partitioning problem. However, K-Means suffered from initiating centroids, where the centroids are selected randomly and tend to have an instability to the cluster calculation. The second problem is, the number of clusters must observed thoroughly to obtain the best cluster solution. Thus, in this research we proposed a strategy by using a modification of K-Means clustering, by implementing centroid optimization using Pillar Algorithm, and obtain the best cluster solution by evaluating the quality of cluster using Silhouette function and Sum of Squared Error.

Centroid Optimization using Pillar Algorithm

The original proposal of K-Means clustering used random centroid selection in the beginning of iteration. This is not an issue when it computes a small size of dataset. But dealing with a large size of data it could take a lot of time to have the best centroid selection.

The Pillar algorithm was inspired by the function of pillars of a building or a construction. It is a common reasoning that a pillar in a building is deployed at each edge or corner in a building, so the mass of the building is concentrates in each pillar. The same idea is adopted for the clustering task that the best initial centroids are presumed exist in the edge of the dataset, or in other word, those *k*-farthest objects in the dataset is selected as initial centroids, where *k* is number of clusters to be observed. Complete steps from original Pillar Algorithm paper is described in Figure 3.

1. Set $C=\emptyset$, $SX=\emptyset$, and $DM=[]$
2. Calculate $D \leftarrow \text{dis}(X,m)$
3. Set number of neighbors $nmin = \alpha \,.\, n / k$
4. Assign $dmax \leftarrow argmax(D)$
5. Set neighborhood boundary $nbdis = \beta \,.\, dmax$
6. Set i=1 as counter to determine the i-th initial cer
7. $DM = DM + D$
8. Select ж $\leftarrow x_{argmax(DM)}$ as the candidate for i-th centroids
9. SX=SX ∪ ж
10. Set D as the distance metric between X to ж.
11. Set $no \leftarrow$ number of data points fulfilling $D \leq nb$
12. Assign DM(ж)=0
13. If $no < nmin$, go to step 8
14. Assign $D(SX)$=0
15. $C = C \cup$ ж
16. $i = i + 1$
17. If $i \leq k$, go back to step 7
18. Finish in which C is the solution as optimize centroids

Figure 3 Complete steps of Pillar Algorithm (Barakbah 2009)

From the line 1 to line 3 the algorithm started with initialization variables: *C* is to stores the selected centroids*, SX* is to stores candidate for the initial centroid*,* and *DM* to is to stores the distance between objects and the mean variable *m.* The next step is to calculate distances between the object in *X* and *m*, and stores it into *D* which is represented as a list sorted descendingly. The objective of this sorting mechanism is to select the farthest objects that later can be used as an initial centroid candidates.

However, these candidates also determined by two variables that must be carefully selected: α and β. α is a variable to determine how many objects that surround the centroid candidate, this calculation is done by multiplying α with *n/k* (line 2) and stores it into *nmin* variable, where *n* is the number of objects in the dataset, and *k* is the number of clusters. This multiplication is intended to get the proportional number of objects in a clusters, and to avoid the selection of an outlier as a centroid. Other variable is β that determined how far the neighbor objects are that surrounds the centroid. This achieved by multiplying β with *dmax*, that is the farthest objects in *D* (line 5), and the result is stores into variable denoted as *nbdis*.

After the initialization is completed, the algorithm continued by iterating each objects within *D*. The current centroid is marked with Ж (Capital Zhe) stored termporarily in *SX* (line 9), and the algorithm will observed how many objects are

feasible as the neighbors of Ж by observing objects in *DM* list, that stores the distances between object in *X* and Ж. The number of feasible neighbors must be greater than *nmin* and with the distance that at least equal or less than *nbdis*. Once these two criteria are satisfied, then Ж is selected as centroid candidate, and stored into *C* list. Otherwise, the algorithm will continued the iteration in *D* list to select the next farthest object.

Once the number of centroids in *C* list is equal to *k*, the algorithm will stopped and present the centroid candidates that used later in the K-Means clustering task.

Cosine Similarity

In this research, cosine similiarity is used to determine the similarity between the query terms entered by users and the documents within clusters described in Formula 7. Where d_i is the *i-th* from document vector collection denoted as $\boldsymbol{d} = (\overrightarrow{d_1}, \overrightarrow{d_2}, \overrightarrow{d_3}, \dots, \overrightarrow{d_m})$, where *m* is the number of documents. Each document vector is about to be compared against the term query vector collection denoted as $\boldsymbol{q} = (\overrightarrow{q_1}, \overrightarrow{q_2}, \overrightarrow{q_3}, \dots, \overrightarrow{q_n})$, where *n* is the number of terms obtained from the preprocessing task. The weight of d_i is obtained from right singular value *U* in Formula 1, as well as was obtained from the left singular value *V* (Thomo 2015;Nguyen 2015). However, not all of vectors in q are involved in the calculation, only vectors that represent the terms that entered by the users are involved.

The lower $cos\ (d_i, q_j)$ value indicates that the *j-th* query terms are strongly related to the *i-th* document. Otherwise, the higher $cos\ (d_i, q_j)$ value indicates the query terms are less related. An example of how the SVD, and cosine similarity works can be followed in Appendix 5.

$$cos\ (d_i, q_j) = \frac{d_i . q_j}{\|d_i\| \|q_j\|} \tag{7}$$

Cluster Evaluation using Silhouette Function

After performing the clustering task, the cluster evaluation was done by using silhouette function (Rousseeuw 1986). The formulation of silhouette function is described in Formula 8.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{8}$$

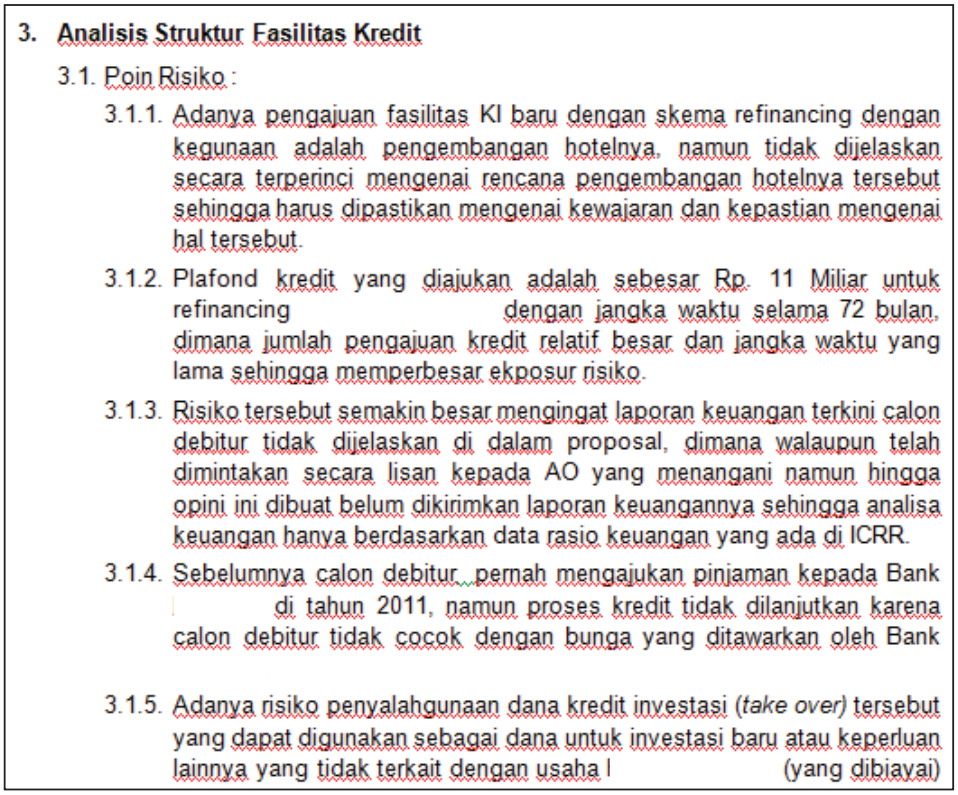
Where *s(i)* is the silhouette score of object *i*, *a(i)* is the average distance between object *i* against all objects within the same cluster of object *i*. *b(i)* is the average distance between object *i* against all objects in other clusters. By using silhouette function, it will be easy to understand how well is an object placed in a cluster, therefore the quality of a clustering task is ensured for the risk documents.

The purpose of silhouette function to replace the usage of variance analysis in the original paper of Pillar Algorithm since the variance analysis cannot describe the quality level of cluster result just like silhouette has, that is $s \in [-1.00, 1.00]$.

3 METHODS

Research Framework

There are 519 loan risk analysis documents collected since 2013 until early 2014, written by risk analyst in the Risk Management Division. All of the documents is written in Microsoft Word format. The documents consist of seven risk analysis parts, those are (1) ICRR (2) Financial Performance (3) Proposed Loan Facility (4) Business Performance (5) Repayment Ability (6) Legal Analysis and (7) Foreign Exchange. All of the parts are analyze based on 5Cs Credit Criteria (Character, Capacity, Capital, Condition, Collateral). Here in Figure 4, we depict an example of one risk analysis document.



3. Analisis Struktur Fasilitas Kredit

3.1. Poin Risiko :

3.1.1. Adanya pengajuan fasilitas KI baru dengan skema refinancing dengan kegunaan adalah pengembangan hotelnya, namun tidak dijelaskan secara terperinci mengenai rencana pengembangan hotelnya tersebut sehingga harus dipastikan mengenai kewajaran dan kepastian mengenai hal tersebut.

3.1.2. Plafond kredit yang diajukan adalah sebesar Rp. 11 Miliar untuk refinancing dengan jangka waktu selama 72 bulan, dimana jumlah pengajuan kredit relatif besar dan jangka waktu yang lama sehingga memperbesar ekposur risiko.

3.1.3. Risiko tersebut semakin besar mengingat laporan keuangan terkini calon debitur tidak dijelaskan di dalam proposal, dimana walaupun telah dimintakan secara lisan kepada AO yang menangani namun hingga opini ini dibuat belum dikirimkan laporan keuangannya sehingga analisa keuangan hanya berdasarkan data rasio keuangan yang ada di ICRR.

3.1.4. Sebelumnya calon debitur pernah mengajukan pinjaman kepada Bank di tahun 2011, namun proses kredit tidak dilanjutkan karena calon debitur tidak cocok dengan bunga yang ditawarkan oleh Bank

3.1.5. Adanya risiko penyalahgunaan dana kredit investasi *(take over)* tersebut yang dapat digunakan sebagai dana untuk investasi baru atau keperluan lainnya yang tidak terkait dengan usaha I (yang dibiayai)

Figure 4 An example of risk analysis document

Each of risk analysis documents consist of analysis and mitigation of following parts: 1) Internal Credit Risk Rating, 2)Financial Performance, 3)Proposed Loan Facility, 4)Business Performance, 5) Repayment Ability and Cash Flow, 6) Legal Analysis, and 7)Foreign Exchange Analysis (Optional). However, not all of those criteria are always commented by the analysts. For instance, the Foreign Exchange Analysis are not required to be analyzed for the business that only using Rupiah for their daily transactions.

As seen in Figure 5, the research framework is divided into 4 parts, those are 1) Preprocessing 2) Risk Clustering 3) Risk Measurement and 4) Evaluation.

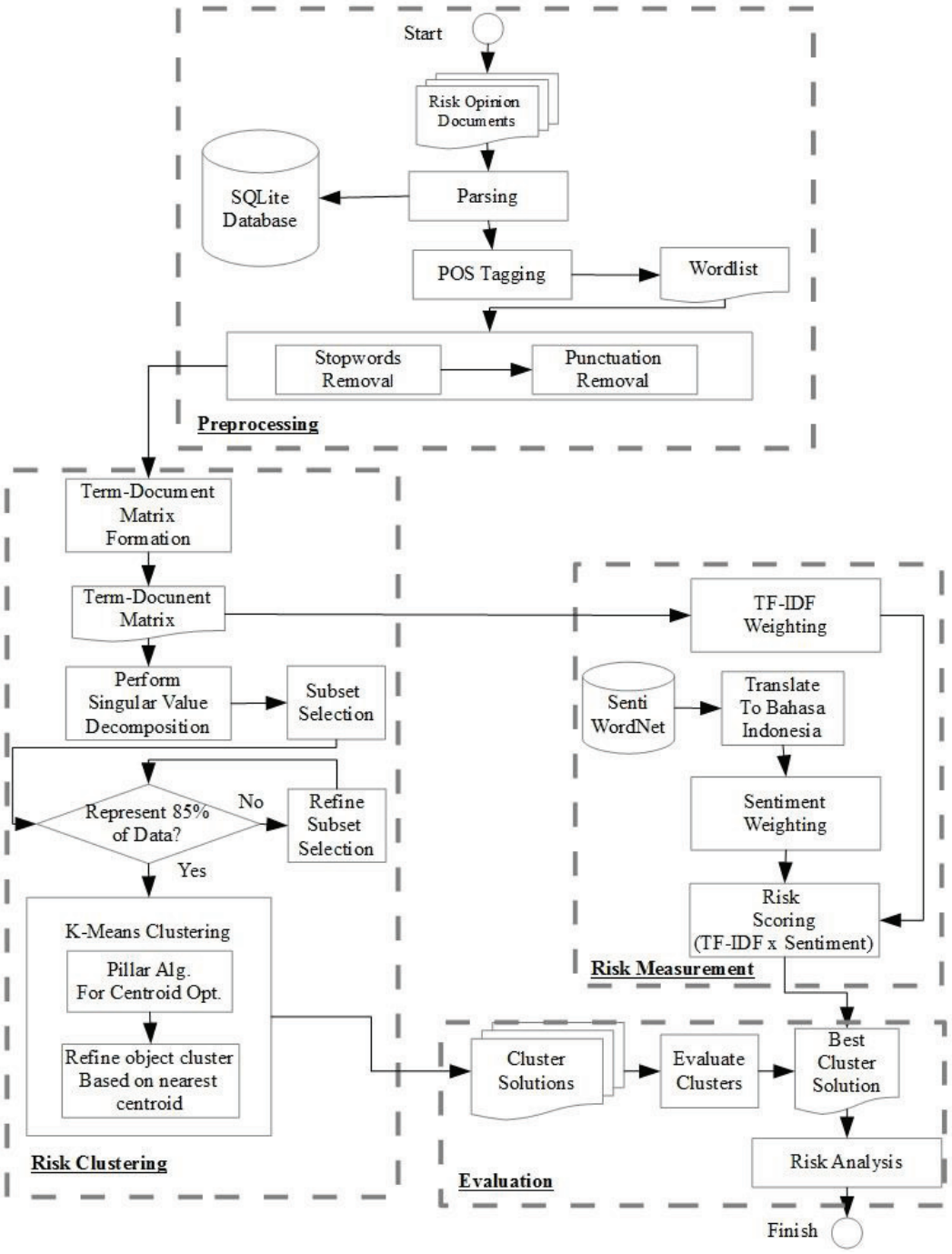


Figure 5 Research Famework

Parsing The Risk Opinion from The Documents

Processing all parts and its content in a risk analysis document is unnecessary since there might be one or more part which do not contain information about risk such as the opening and the closing section. We observed that there are three major

parts in risk analysis documents: (1) Opening (2) Opinion and mitigation (3) Closing and signature. Since what this research really need is the opinion, thus, we only retrieved the risk opinion and mitigation part by parsing the documents, extracting the sentence, and perform term tokenization using "re" API, a regular expression library from Python. We noticed that all part of opinion are always started with the same and specific header. For instance, risk opinion in the bank's credit scoring system always marked with the "Pengisian ICRR" header, opinion for financial condition market with "Aspek Keuangan" header. The complete list of header that marked each part of opinion are marked in Table bla below. After all of opinion were parsed and retrieved from the documents, then those were stored in a database management system called SQLite (Hipp 2000) table namely mst_opini_mitigasi_raw, with a structure as listed in Table 2.

Tabel 1 Table structure of mst_opini_mitigasi_raw

Field Name	Field Type	Description
Id	INTEGER	Document Id
opini_mitigasi	INTEGER	An identification for the content, filled by 1 if the content is an opinion, and filled by 2 if the content os a mitigation
Part	INTEGER	An identification for the part of analysis. 1 is for "Pengisian ICRR" 2 is for "Aspek Keuangan" 3 is for "Analisis Aspek Bisnis" 4 is for "Analisis Kemampuan Pembayaran" 5 is for "Analisis Struktur Kredit" 6 is for "Analisis Terkait Aspek Lain" 7 is for "Analisis Terkait Kurs"
content	TEXT	Content of the document

And in Figure 6 shows how the content of the document is stored in the mst_opini_mitigasi_raw table.

Table: mst_opini_mitigasi_raw

	id	opini_mitigasi	part	cluster_task	content
1	0	1	1	1	benchmark sektor : perdagangan - retail - mobil berdasarkan pe...
2	0	2	1	1	:dilakukan evaluasi atas kelayakan debitur maupun key person s...

Figure 6 A snapshot from table mst_opini_mitigasi_raw

POS Tagging and Term Tokenization

As already mentioned earlier, the SentiWordNet uses the position or tag of a term, thus a POS Tagging is an inevitable process. Apart from tagging or labeling the terms, this process is also combined with the tokenization process which is a convertion process that transforms the terms into feature to be used later in clustering and sentiment measurement. Obviously, not all of terms are selected for the feature, this selection process was completed by performing the stopwords and punctuation removal. List of stopword and punctuation can be seen in Appendix 3.

And in the Results and Discussions we will show how the POS Tagging works in labeling terms within a sentence.

```
1  benchmark|n      6  berdasarkan|v  11  lain|a
2  sektor|n         7  perhitungan|n  12  credit|n
3  perdagangan|n    8  rasio|n        13  rating|n
4  retail|n         9  keuangan|n     14  posisi|n
5  mobil|n          10 data|n         15  perusahaan|n
```

Figure 7 An example of content from the f_wordlist file

All terms that selected for the next process are stored in a plain text file called as f_wordlist, in the form as shown in Figure 7 above, where a term are stored in a row followed by its POS Tag in a sentence that founded during the POS Tagging process.

Table 2 A matrix of term-document illustration

	Documents				
Terms	d_1	d_2	d_3	d_4	d_5
trend|n	*1*	*0*	*0*	*0*	*0*
bisnis|n	*1*	*0*	*0*	*0*	*0*
perumahan|n	*1*	*1*	*0*	*0*	*0*
meningkat|v	*1*	*1*	*0*	*0*	*1*
minat|n	*0*	*1*	*0*	*0*	*0*
masyarakat|n	*0*	*1*	*0*	*0*	*0*
abc|n	*0*	*1*	*1*	*1*	*1*
anak|n	*0*	*0*	*1*	*0*	*0*
perusahaan|n	*0*	*0*	*1*	*0*	*0*
xyz|n	*0*	*0*	*0*	*0*	*0*
kesulitan|a	*0*	*0*	*1*	*0*	*0*
modal|n	*0*	*0*	*1*	*0*	*0*
track-record|a	*0*	*0*	*0*	*1*	*0*
pemilik|n	*0*	*0*	*0*	*1*	*0*
kriminal|n	*0*	*0*	*0*	*1*	*1*
kasus|n	*0*	*0*	*0*	*0*	*1*

Later, those terms that listed in f_wordlist file will be used to construct a term-document matrix with size of $t \times d$ as seen in Table 3, where t is the number of terms (row) and its tag that founded in number of d documents (column). All of these process starting from tagging, stopwords removal, and term-document matrix construction are performed in a single Python file namely i_preprocess.py, that listed in Appendix 6.

Singular Value Decomposition

The term-document matrix that we have mentioned earlier, are used to obtain a reduced dataset in order to minimize the computation time. Since the objective of this research is to cluster the documents, we used the right singular value V^T from the Equation (1). The $t \times d$ term-document matrix is represents the matrix A, where

the dimension we would like to reduce. To perform the task, we utilize a Python library namely "numpy" which is has a function to perform SVD. The function is obtained by simply executing the following command:

```
SA, EA, UA = np.linalg.svd(A, full_matrices=True)
```

Where SA represents the left singular value, EA represents the singular value, and UA represents the right singular value that will used later for the clustering task.

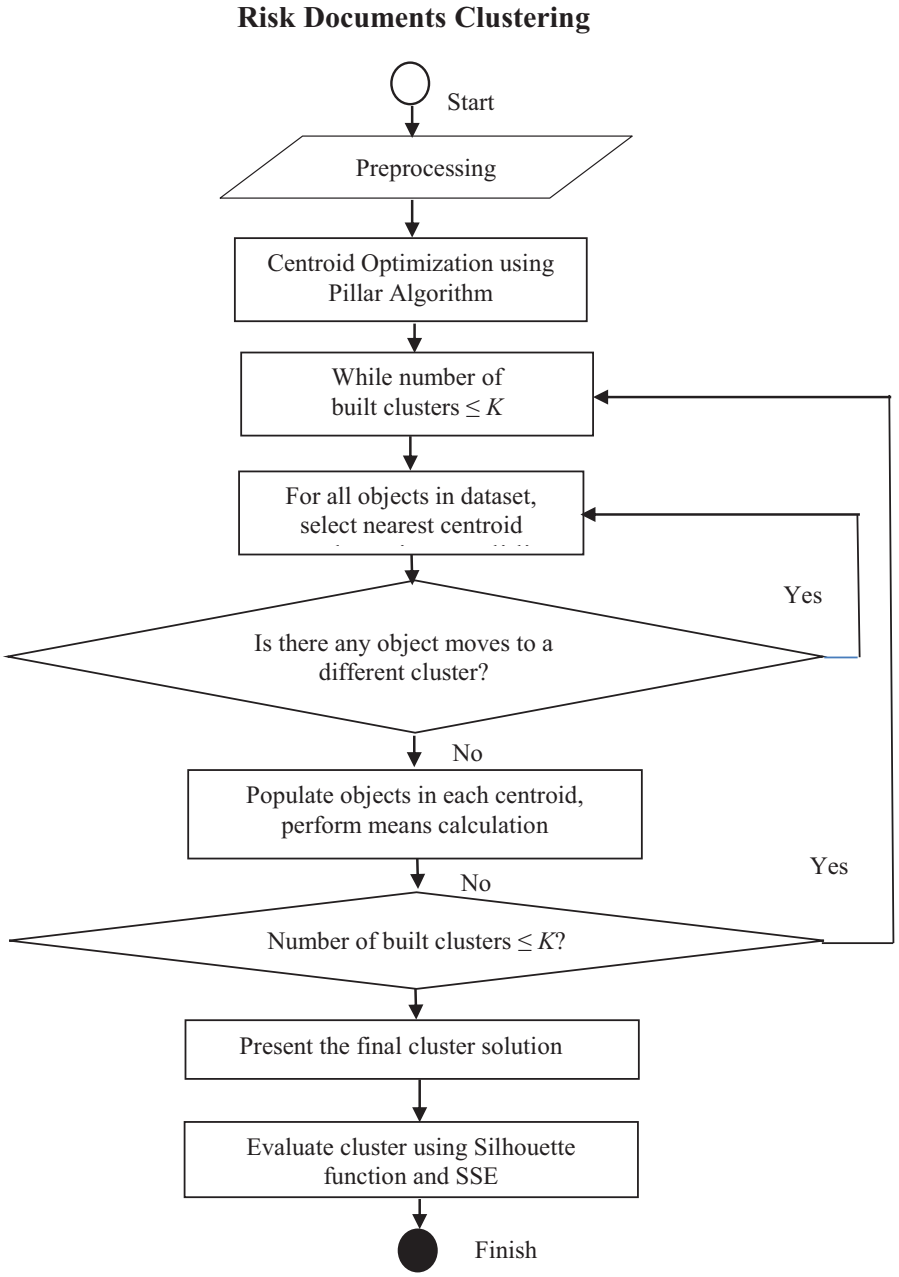


Figure 8 Workflow of Clustering process using K-Means Clustering that optimized by Pillar Algorithm and evaluated by Silhouette function and SSE

The centroid optimization will performed before starting K-Means, and both of evaluations are performed each time K-Means present the final cluster solution.

```
1 Pillar_Algorithm(P, K, alpha, beta)
2       m = GetMeanFromEachVariable(P)
3       Distances = []
4       n = length(P)
5       MaximumIteration = n
6       nmin = (alpha * n) / K //The minimum number of neighbors
7       for i = 1 to n
8               m[i] = Sum(P[i]) / NumberOfVariables
9               d = EuclidianDistance(P[i] - m[i])
10              Distances←d //Stores the distance to mean

11
12      D = SortDescendingly(Distances) //Get the farthest objects
13      DM = D
14
15      while (NumberOfCentroid < K and iteration < MaximumIteration)
16              dmax = DM[0]
17              nbdis = beta * dmax //The farthest distance to fulfilled
18
19              for x = 0 to n-1    //Iterates the objects
20                      if DM[x] not in SX //If object not in SX list
21                              Ж = P[DM[x]]
22                              SX← Ж //Stores to SX list
23
24              for x = 0 to n:
25                      d = EuclidianDistance(P[i] - Ж) //Calculate distance between
                                                        //object with Ж
26                      DTemp1←d //Stores into DTemp1
27
28              D = SortDescendingly(DTemp1) //Sort descendingly
29
30              no = 0
31              for x = 0 to n: //Calculate the number of neigbors in radius of nbdis
32                      if D[0] <= nbdis //Check if the distance < nbdis
33                              no++    //Add value of no. variable fulfilled max distance
34
35              if no >= nmin //Check if the number of neigbors >= nmin
36                      NumberOfCentroid++ //Add the number of centroids
37                      C.append(Ж)
38
39                      for x = 0 to n
40                              d = EuclidianDistance(P[x] - Ж)
41                              DTemp2←d
42                      DM = SortDescendingly(DTemp2)
43              else //Otherwise, continue exploration through other objects
44                      iteration++
45              if iteration == MaximumIteration //If it has reach max iteration allowed
46                      return                   //then exit
47      return C
```

Figure 9 Pseudocode of Pillar Algorithm developed from algorithm execution steps in the original paper (Barakbah 2009)

The order tasks of centroid optimization, K-Means clustering, and cluster evaluation will be repeated for some possible numbers of cluster, where in this case is *K*, as seen in Figure 8. After performing the optimized centroid initialization, the

algorithm will search the best centroid of each objects based on the Euclidian distance.

The Euclidian distance used in this research is based on below formulation:

$$\forall \boldsymbol{c} \in C, D = \sqrt{\sum_{t=1}^{n} (\boldsymbol{w}_t - \boldsymbol{c}_t)^2} \quad (6)$$

Where D is distance between weight vector $\boldsymbol{w}$ for term t to certain selected centroid. The algorithm will continue to iterates until all of the centroids are convergent. The source code implementation of this algorithm in Python can be found in Appendix 1.

Since the number of clusters are not defined yet, thus, We have to observe every possibilities regarding the number of clusters from K=2 to K=10. The observation also applied to find the optimum value of Pillar Algorithm's parameters α and β respectively as mentioned in the pseudocode in Figure 9. Thus, for each value of K, We set the combination value of α and β. The complete pseudocode to find the best cluster solution is described in Figure 10.

```
1 alpha = 0.4
2 beta = 0.6
3 while (K ≥ 2 and K ≤ 10)
4         while (alpha ≥ 0.4 and alpha ≤ 1.0)
5                 while (beta ≥ 0.6 and beta≤ 1.0)
6                         Centroids = Pillar_Algorithm(P, K, alpha, beta)
7                         Solution = K_Means(K, Centroids)
8                         Silhouette_Score = Silhouette_Function(Solution)
9                         SSE_Score = SSE(Solution)
```

Figure 10 Pseudocode to find the best cluster solution, combines Pillar Algorithm, K-Means clustering, Silhouette Function, and Sum Squared of Errors to get the best cluster solution

Term Frequency-Inverse Document Frequency

The TF-IDF calculation utilized the term-document matrix to measures the TF-IDF scores. We implement the TF-IDF calculation as mentioned in the Literature Review section, in a Python program named i_preprocess.py, and the result of this calculation is a $t \times d$ matrix stored in a flat text file named f_bag_of_weighted_terms. This file are later to be used to retrieved the result from TF-IDF calculation, and combine it with the Sentiment Score, instead of recalculate it that would takes some time.

Translating SentiWordNet

As mentioned before, that the SentiWordNet only available in English, thus, a translation is needed in order to utilized the lexicon for Bahasa Indonesia. The translation tool used here was the Google Translate which provide an API. One of available API for Python programming is the goSlate API that available freely online. Before the translation is began, the lexicon is imported into a table SQLite database. Then, by using a concise Python program the translation begin by iterating through all terms available in SentiWordNet lexicon and translate those terms to Bahasa Indonesia.

The SentiWordNet is available as a text file with a format that we have explained earlier. The aim of importing the content of SentiWordNet into a SQLite

database is to minimizing the possibility of losing data while translating terms that caused by network failures or any unpredicted problems. And once the translation process is stopped, we are able to trace and continue from the last term that successfully translated. There are 205,624 terms that we have been translated using the goSlate API and stored into a table called ref_sentiwordnet_nodesc with the structure described in Table 3, that is similar with the original SentiWordNet file as depicted in Figure 11, except the description field.

Table 3 The structure of SQLite table ref_sentiwordnet_nodesc

Field Name	Field Type	Description
word_type	TEXT	Term position in a sentence
word_index	TEXT	The term index in SentiWordNet
Pos	NUMERIC	Positive polarity score
Neg	NUMERIC	Negative polarity score
word_en	TEXT	Term in English
word_id	TEXT	Term in Bahasa Indonesia

As seen in the Figure 11, “risk” has several synonim: “peril”, “risk”, “jeopardy”, and “hazard", those four terms are not stored in the same rows as in the SentiWordNet lexicon file, instead, those term are stored separately in the different rows to make the query process in measuring the sentiment polarity becomes more efficient. Those terms are stored in a file namely ref_sentiwordnet_nodesc as depicted in Figure 12.

```
1  n   14541852    0   0.25    risk#1 peril#1 jeopardy#1 hazard#1
   endangerment#1   a source of danger; a possibility of incurring loss or
   misfortune; "drinking alcohol is a health hazard"
2  n   14542579    0   0   moral_hazard#1  (economics) the lack of any
   incentive to guard against a risk when you are protected against it (as
   by insurance); "insurance companies are exposed to a moral hazard if the
   insured party is not honest"
3  n   14542320    0.625   0.25    health_hazard#1 hazard to the health of
   those exposed to it
4  n   14542441    0   0   biohazard#1 hazard to humans or the environment
   resulting from biological agents or conditions
```

Figure 11 A snapshot from SentiWordNet lexicon file

	word_type	word_index	pos	neg	word_en	word_id
1	n	00802238	0	0.625	risk	risiko
2	n	00802238	0	0.625	peril	bahaya
3	n	00802238	0	0.625	danger	bahaya

Figure 12 An example of some terms stored in ref_sentiwordnet_nodesc

Since in the document we found a mix usages of Bahasa Indonesia and English, the strategy to find the best matching term in SentiWordNet is by performing a look up into both SentiWordNet in Bahasa Indonesia and English. We split both languages into separated files, the SentiWordNet file in Bahasa Indonesia are stored in files with the first alphabet and tag as the prefix and “_id” as the suffix.

SentiWordNet file in English in files with the first alphabet and tag as the prefix and "_en" as the suffix. For instance, adjective terms in Bahasa Indonesia e.g. "abadi","acak", and "ahli" are stored in file named "aa_id". Otherwise, adjective terms in English e.g. "capital","cashable", and "categoric: are stored in file named "ca_en". The aim of storing these terms into separated files is to reduce the query time from the database.

Sentiment Scoring

Technique that used to retrieve the terms in SentiWordNet is as seen below in Figure 13.

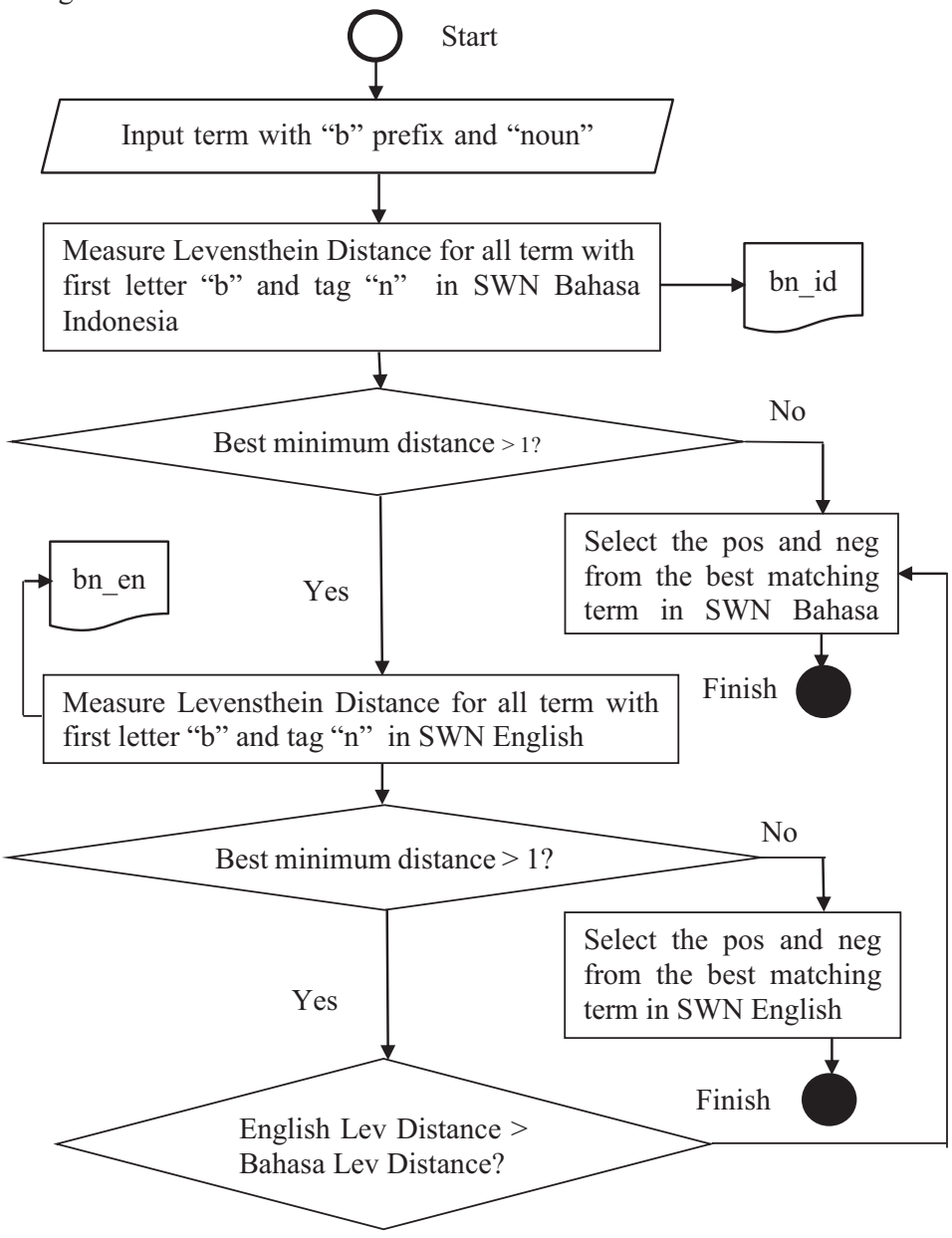


Figure 13 Flowchart on how to select the best matching term in SentiWordNet

Basically, the look up is started with look up in Bahasa Indonesia files, where the risk analysis documents based on to this language. If the look up found that the best Levensthein distance is greater than 1, the iteration will continued to the English files. Then, the best distance found in English compared to the best distance in Bahasa Indonesia, if the English's best distance is less than the Bahasa Indonesia's best distance, term in English will be selected, otherwise, term Bahasa Indonesia will be selected.

Equipments

There are two type of equipment used in this research. The first one is software consists of programming tools, libraries, and packages. Microsoft Visual Studio 2013 Community Edition was chosen as Integrated Development Environment (IDE) to develop a C# program equipped with Microsoft Office .NET native library. Both were used to make it easier to import the corpus since the risk analysis documents is in Microsoft Word (both doc and docx extension). After the documents have been read by the program, those imported into a SQLite Database.

Then, the process continued by proceeding the main tasks, those are preprocessing, clustering, term and sentiment scoring, and eventually ended by the evaluation. All of the tasks are used the Python version 2.7 programming equipped by some packages that commonly used in computational task such as numpy, and scipy. The tool used for the Python programming is Sublime Text version 3. The second equipment is the hardware, a notebook with AMD A6 Processor, 4 GB DDR3 Memory, and 750 GB Harddisk Drive.

4 RESULTS AND DISCUSSIONS

Risk Analysis

Risk analysis is necessary and required by central bank as described in the regulation regarding to implementation of risk management (BI 2003). It is clearly stated that the conventional bank must effectively implement risk management on running their business, and in our case, including risk management for financing process. Conceptually, the risk management is consist of (1) Surveillance from the board of commissioner and board of director (2) Sufficiency of policies, procedures, and limit definition (3) Sufficiency of identification, measurement, surveillance control, and system information (4) A comprehensive internal control. What we want to highlight are point number two and number three, where a bank must has a identification, measurement, mechanism, and information system implementation to analyze and mitigate the risk.

As an implication Indonesia's conventional bank and must fulfills following criteria that required and must be reported monthly to the central bank through debtor information system (BI 2007): (1) The debtor (2) The business owner 3) Capital 4) Collateral 5) Guarantor 6) Financial quality. Technically, the best practice risk management for financing in banking industry is commonly implemented as credit scoring system in quantitative fashion, and conceptually generated from analysts mind in qualitative fashion (Soares *et al* 2011). Thus, if the

conventional bank in Indonesia willing to implement both models, they must to make sure that their implementation are already fulfill above criteria.

Preprocessing

In the preprocessing, we divide the task into three parts: (1) Parsing the documents, the aim is to retrieve the opinions from the documents, (2) POS Tagging, the is to give the label into each term, and (3) Construct term-document matrix and reduce it using the dimension using SVD (Nguyen 2015), to get a feature with lower dimension.

Parsing The Documents

There are two major task in preprocessing, those are removing the stopwords and labeling the Part of Speech position of each term. In stopwords removal the stopword list used in this research retrieved from one of GitHub page along with the POS Tagging API[1]. There are many stopwords in Bahasa Indonesia available in the web, mostly addressed for particular field or study and not for general purpose. The reason why the stopword list from this page was chosen is simply because it was made for a general Natural Language Processing research in Bahasa Indonesia. There are 124 stopword terms used to remove the less important terms in the risk analysis documents, Most terms here are conjunctive terms such as “sekedar”, “sekadar”, “ketika”, “melainkan”, “bagaikan”, and “beserta”.

Part of Speech Tagging

The preprocessing task continued with POS Tagging. Since there are only four term position labels in SentiWordNet as mentioned in the Literature Review section, and the POS Tagging API provide 37 position labels, then a mapping was needed to find the best pair between term in the corpus and in the SentiWordNet lexicon. Here in Table 4 is the mapping between the both from POS Tagging API and SentiWordNet.

Table 4 POS Tag transformation between tags in SentiWordNet and tags in POSTag API

SentiWordNet Label	POS Tagging API Label
A (Adverb)	JJ, JJR, JJS, CDC, CDI, CDO, CDP
V (Verb)	VB, VBD, VBT, VBG, VBN
R (Additional Adverb)	RB, RBR, RBS, NEG, SC
N (Noun)	NN, NNS, NNP, NNPS, NNG, FW, MD, NNG, NNPP, WP

An example result of POS Tagging would be a string variable as seen as below.

“terdapat/VBT agunan/NN fix/NN asset/NN (/OP tanah/NN dan/CC bangunan/NN)/CP dan/CC tagihan/NN sehingga/SC perlu/MD dipastikan/VBT mengenai/VBT legalitas/NN pengikatan/NN ,/, mekanisme/NN penanganan/NN

dan/CC monitoring/NN atas/IN kualitas/NN dan/CC nilai/NN terkini/NN dari/IN agunan/NN tersebut.eksekusi/JJ atas/IN jaminan/NN yang/SC berupa/VBT non/NEG fixed/NN asset/NN (/OP peralatan/NN)/CP bukan/NEG merupakan/VBT hal/NN yang/SC mudah/JJ dan/CC sangat/RB rentan/JJ terhadap/IN penurunan/NN nilai/NN //GM depresiasi/NN (/OP terlihat/VBI dari/IN aset/NN tetap/JJ dari/IN tahun/NN 2010-2102/CDP yang/SC nilainya/NN terus/RB menurun/VBI)/CP"

Afterwards, the stopwords and punctuation removal were performed to remove the unecessary terms such as "dan", "sehingga", "merupakan", "yang", "dari", and "2010-2012". The remaining terms are used to construct term-document matrix as seen in Table 1 that used as a feature vector later on the clustering and sentiment weighting.

Dimension Reduction using SVD

According to Figure 5, the preprocessing task produced a dataset containing about 3290 terms. These terms were used to construct the term-document matrix sized 3290×519. Here, the SVD plays its role to minimize the execution time and obtain the most prominent terms available in the corpus. As described in the literature review section, by computing SVD, there will be three matrices. The left singular matrix, *U*, the term concept was generated with the size of 3290×300. The diagonal matrix $\sum$ generated with the size of 519×300, and the last matrix *V* that is the document concept matrix sized 519×519.

Actually, by utilizing SVD, the dimension of the dataset is already reduced, but we tried to get a smaller dataset by To get the best *k-rank* that represent the entire corpus dataset, by using formula (2). In this research we set threshold q=0.98 and the result is, selected *k-rank* is 300. Yet, there is no standard on what is the best threshold for the best *k-rank*. Earlier, Zhang Dong(2004) defined that the best *k-rank* is 0.8, Osinzki(2004) defined that the best *k-rank* is 0.9.

The dimension reduction objective was achieved by utilizing the document concept matrix *V* and the diagonal matrix $\sum$ (Thomo 2015). Actually, document clustering can be achieved by only performing SVD (Zhang 2004). However, the number of document groups is considered still too vast for the bank to get the proper risk model, since there are 300 concepts found, in other words there are 300 risk levels and concepts that can be obtained. For a comparison, in particular bank, the quantitative model currently has only 20 risk level. By narrowing the number of clusters the expectation is that the bank could be easy to determine in what level of risk is a proposal can be accepted.

Therefore, by using the reduced dataset came an idea to perform clustering task using K-Means. The expectation is to get the desired number of clusters at least equal with the quantitative model.

Risk Clustering

Afterwards, the next task is clustering the documents by utilizing the dataset that already preprocessed by stopwords-punctuation removal, POS Tagging, and SVD. We will explain how the clustering process step-by-step, starting with optimizing the centroids initialization.

Optimizing Centroid

Obviously, K-Means clustering has a classic issue in optimizing the initial centroid since K-Means is relied on randomized object selection for the initial centroid. Even though the dimension reduction was already performed, the randomized centroid selection will generate a non-persistent cluster solution in each iteration. Thus, it will suffered from computation time execution, since a number of try and error experiment will occurred. For instance, in the case study, there are about 519 documents to be clustered. Let say there are about 5 clusters to compute, by using permutation, $n!/(n-k)!$, where n is the number of documents, and k is the number clusters, there would be 36,935,551,782,360 centroid probabilities. Hence, an optimization is needed to overcome this problem.

Overall, for each number of k, $\{\forall k \in \mathbb{Z} | 2 \leq k \leq 10\}$ to be observed. The number of k used in the computation is the representation of the number of business risk levels in the bank, so that we expect that the each cluster group represents the level of risk, ranging from the lowest to the highest. The iteration was started from measuring the mean of the feature vectors, which is the term and sentiment weight that previously measured. Below is the formula we use to measure the mean available in the corpus:

$$\forall t \in T, m(t) = \frac{1}{N}\sum_{i=1}^{n} w(t_i) \quad (5)$$

In Formula 5 the mean calculation for each term t that is available terms in the term list T, N is number of documents, w is term and sentiment weight that previously measure, and $\boldsymbol{m}$ is the mean vector of the term. After we obtain the mean of all terms as a starting point of iteration, the algorithm select the k farthest distance objects from m, defined as Ж, as initial centroids and check whether Ж already exist in SX list, otherwise, it will be stored to SX. The selection method is simply by sorting the distance matrix dataset that containing each term vector distance to the mean. The distance formula we used here is the basic euclidian distance measurement similar to Formula 6.

Starting from the farthest object from the mean, the computation is followed by exploring the number of neighbors of selected objects, defined as $\alpha \times N$ stored in *nmin* variable, inside the boundary of $\beta \times D(\boldsymbol{d}, \boldsymbol{m})$ which is stored in *nbdis* variable. D is the distance between the object term vector of document object $\boldsymbol{d}$ and $\boldsymbol{m}$. This exploration is intended to avoid selecting an outlier object as centroid. Our experiment explore a set of combination α and β, where α, $\{\forall \propto \in \mathbb{R} | 0.05 \leq \propto \leq 0.95\}$ and $\{\forall \beta \in \mathbb{R} | 0.05 \leq \beta \leq 0.95\}$ to search the best cluster solution in the search space. The reason we were not exploring the α out of the given threshold was because it did not satisfies the Pillar algorithm criteria, that the selected

centroids must have minimum number of neighbors in given distance that already set in the beginning of iteration.

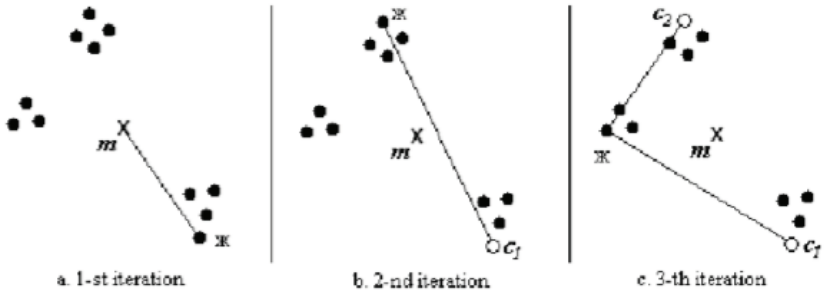


Figure 14 Centroid selection for the first-3 interation (Barakbah 2009)

After having the first centroid, the next object to be selected is the farthest object from the first centroid, this intended to have a set of centroids that not concentrate in one particular area. Figure 14 describe the centroid selection from the first three iteration. Then the iteration will stop when *i*, the number of objects selected as centroid is equal to *k*.

Risk Measurement

In financing industry, measuring risk in the form of grading system is necessary and required to enable the decision makers in comparing risk exposures and having further analysis upon the customers business (Gumparthi 2010). Therefore, in this research a measuring mechanism must be deployed against the cluster model that has been built. We adopted the grading system that commonly implemented in credit scoring system by labeling each cluster with the level of risk exposure, ranging from the lowest to the highest, and the number of levels are depend on the number of cluster obtained from the clustering task.

From the previous risk clustering step, the cluster that has been modeled represent the interconnectivity of risk opinion between each risk analysis documents. Yet, it is not represents the level of risk that has been mentioned by the risk analysts through the documents. Thus, a process to measure the level of the risk is needed, namely risk measurement. In this research, we analogously refer the risk with sentiment. However, the level of the risk were not only determined by the importance of the terms within the document, but it also determined by the polarity.

To define which term that emphasized or categorized as important some works were previously done using TF-IDF (Paltoglou and Thelwall 2010) that adapted for sentiment analysis. Using TF-IDF are useful to determine on which topic are the document is on to, instead of using binary unigram weighting. However, using only TF-IDF in sentiment analysis, cannot be used to determine the polarity of a document. Thus, sentiment weighting using a lexicon database are used to determine the level of sentiment or in this particular research is the level of the risk. Li and Wu (2010) use lexicon database to determine the polarity of term by simply query the value of positivity and negativity of a term to the HowNet database and set the value either 1 if a term is tend to positive, or -1 if a term is tend to negative. The same rule are implemented in this research to determine the risk

polarity using SentiWordNet lexicon database. The mechanism will be explained in the following Sentiment Weighting section.

As explained before, Term Importance Weighting using TF-IDF and Sentiment Weighting using were combined to define risk level in each cluster generated from the Risk Clustering process. The idea was came to find out on how the risk analyst emphasis the usages of terms by calculating its importance using TF-IDF. An example to perform a TF-IDF calculation can be followed in Appendix 6. And the sentiment weighting used to calculate the polarity, whether a term is tend to positive or negative. Hence, the formulation of both calculation for each term described as below:

$$w(t) = tf.idf(t,d) \times s(t) \quad (7)$$

Where *w* is the total weight of the term generated from the TF-IDF calculation *tf.idf(t,d)* that consists of *tf* and *idf* calculation as can be seen on formula (1), multiplied by the polarity score, *s(t)* retrieved from SentiWordNet lexicon.

In order to find the best matching sentiment score in SentiWordNet lexicon, alongside querying the terms POS Tag, a term approximation technique using Levensthein distance measurement also used to compute the edit-difference between two pair of strings. The order of steps as seen in Figure 15 (refers to the flowchart in Figure 13 that described earlier), was used to find the best sentiment score preceded by the regular Structured Query Language (SQL) that collect the terms in SentiWordNet ordered by the first alphabet and written those terms into group of files that arranged the first letter and the POS Tag labels. Or in other words, this also called as an indexing technique. The SQL syntax we used to retrieve the terms is divided into two syntax. The first SQL syntax is to retrieve terms in Bahasa Indonesia, and the second is for the English. The purpose is to make the term look up more efficient, rather than scanning through all terms in a single files.

The file labeled with the format <alphabet><pos_tag>_<language>, such as file “an_id” contains terms in Bahasa Indonesia with first letter“a” and has POS Tag label “noun”, file “av_id” contains terms with first letter “a” and has POS Tag label “verb”, and so does the other terms. For English terms, the files label would be look like “ba_en” that contains terms with first letter “b” and “adjective” POS Tag label. The aim of this division was to handle the usage of English terms in the documents, so that if the pairing was not found in Bahasa Indonesia the look up will search in English. Therefore the SentiWordNet lexicon indexed and transformed 204 files.

After the indexing is done, the next step performed is the terms look up. For instance, for the term “asuransi” (insurance) that labeled as “noun”, the program will search in the file “an_id” for the first best matching term in SentiWordNet lexicon. If the Levensthein distance is higher than 0, the query will continued in the “an_en” which is the english file for the term with prefix “a”. And if the Levensthein distance measurement does not better than in the Bahasa Indonesia, the sentiment score from Bahasa Indonesia will returned, otherwise, the score from the English lexicon is returned.

Find The Best Matching Terms in SentiWordNet

```
1 TermWeighting(Term, PosLabel)
2       SentimentScore = 0
3       SwnDbId[a,b,c ... z][a, n, v, r]
4       SwnDbEn[a,b,c ... z][a, n, v, r]
5       CurrentDistanceId = 0, CurrentDistanceEn = 0
6       MinimumDistanceId = 0, MinimumDistanceEn = 0
7       PositiveScoreId = 0, NegativeScoreId = 0
8       PositiveScoreEn = 0, NegativeScoreEn = 0
9
11      for each TermId, PositiveScore, NegativeScore  in SwnDbId[Term[0]][PosLabel]
10              CurrentDistanceId = LevenstheinDistance(Term, TermId)
12              if CurrentDistanceId < MinimumDistanceId then
13                      MinimumDistanceId = CurrentDistanceId
14                      PositiveScoreId = PositiveScore
15                      NegativeScoreId = NegativeScore
16                      if CurrentDistanceId = 0 then break
17
18      if CurrentDistanceId > 0 then
19              for each TermEn, PositiveScore, NegativeScore
                in SwnDbEn[Term[0]][PosLabel]
20              CurrentDistanceEn = LevenstheinDistance(Term, TermEn)
21                      if CurrentDistanceEn < MinimumDistanceEn then
22                              MinimumDistanceEn = CurrentDistanceEn
23                              PositiveScoreEn = PositiveScore
24                              NegativeScoreEn = NegativeScore
25                              if CurrentDistanceEn = 0 then break
26
27      if MinimumDistanceId ≤ MinimumDistanceEn then
28              if PositiveScoreId ≤ NegativeScoreId then
29                      SentimentScore = 1
30              else
31                      SentimentScore = 0
32      else
33              if PositiveScoreEn ≤ NegativeScoreEn then
34                      SentimentScore = 1
35              else
36                      SentimentScore = 0
```

Figure 15 Pseudocode to find the best matching term in SentiWordNet lexicon

Sentiment Weighting

The weighting process is basically simply by comparing the negative score and the positive score as defined in logical formulation below. The reason is because the value positive and negative value varied from 0, 0.123, 0.5, 0.625, 0.75, 1.

$$sentiment = \begin{cases} 1, pos \geq 0 \\ -1, neg < 0 \end{cases} \quad (6)$$

Unfortunately, not all terms are correctly found, this was as shown from the Levensthein distance (How Levensthein distance works is as seen in Appendix 7) that the distance was > 0. There are 1666 terms that wrongly paired to the

SentiWordNet. Thus, a further observation against these terms were conducted, by several steps.

1. If it is a typos then a correction directly made up against the term. For example,“penggatian”→”penggantian”,“meningkatakan”→”meningkatkan”, and “menggangu” →”mengganggu”. The correction were later to be used to look up in the SentiWordNet lexicon to find the proper sentiment score.
2. If it is not a typo, then check whether it is a financial and banking terms or not. Suggestions from a banking expert are used to define this kind of terms. Terms like “likuidasi”, “refinance”, “wanprestasi” are obviously considered as banking terms. There are also common terms that categorized as name of people or company, such as “husada”, “manunggal”, and “investama”. These terms are also categorized as a banking terms since they were specificly mentioned by the risk analyst regarding the loan proposal. Specifically for the banking terms, the sentiment scoring was done manually, and again, this is done under consultation with a banking expert. For example, for previously mentioned terms “likuidasi”, “refinance”, and “wanprestasi” were automatically given the negative sentiment score = -1, since those indicating a bad condition of a business. Other terms
3. If it is not a banking terms, then the root words need to be found and used to look up the synonim in SentiWordNet lexicon.
4. The last is that if its not one of above conditions, then the look up is performed by using Kateglo (Kamus, Thesaurus, and Glossarium) tools that is a dictionary tool for Bahasa Indonesia.that available online.

Evaluation

To ensure that the cluster that we have obtained is the proper solution, we performed some evaluation calculation using Silhouette performance and Sum of Squared Error. For sentiment analysis evaluation, we performed analytically or manual analysis and figure out the logical explanation for terms that present most frequently.

Silhouette Performance

From the experiments that have been conducted, the value of α and β plays significant role in silhouette score. We noticed that the lowest value of α and β are 0.4 and 0.6, feasible to $2 \leq K \leq 10$. Any combination value lower than those combination are only feasible to $K=2$. There are 715 cluster solution, thus, it is hard to observe all the cluster solution so we decided to pick up the cluster solution with the highest silhouette score as listed in Table 5.

Table 5 The best cluster solution based on the Silhouette score

K	*α*	*β*	*Silhouette Score (s)*	*Number of Empty Clusters*
2	0.85	0.9	0.494237	0
3	0.75	0.85	0.660766	0
4	0.95	0.85	0.660766	1
5	0.65	0.75	0.642436	2
6	0.75	0.75	0.642436	3
7	0.7	0.8	0.701234	1
8	0.55	0.75	0.70496	1
9	0.95	0.75	0.574014	4
10	0.7	0.75	0.624935	3

From Table 5, it seems like K=8 is the best cluster solution to model the risk document based on the term relationship. But, when it has a closer look, it has a cluster without any member in it, so we can conclude that the clustering task did not place the document properly.

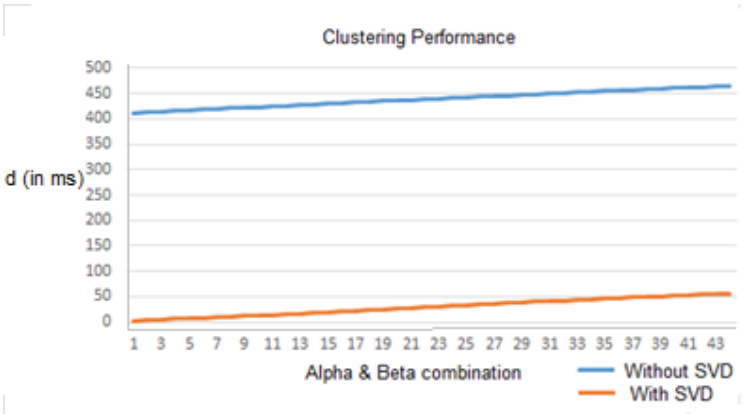


Figure 16 Clustering performance comparison (by execution time in miliseconds) on dataset that decomposed with SVD and without SVD

We also performed some comparison between two clustering task where the first task was performed without SVD decompisition, and the second task was performed with SVD decomposition. From the result we can see that by reduce the dimension using SVD we can save execution time. As seen in Figure 16, the performance of clustering task with SVD is surpass the other task that not using SVD. The comparison was taken for these following parameters: $6 \leq K \leq 7$, $0.5 \leq \alpha \leq 1.0$, and $0.5 \leq \beta \leq 1.0$.

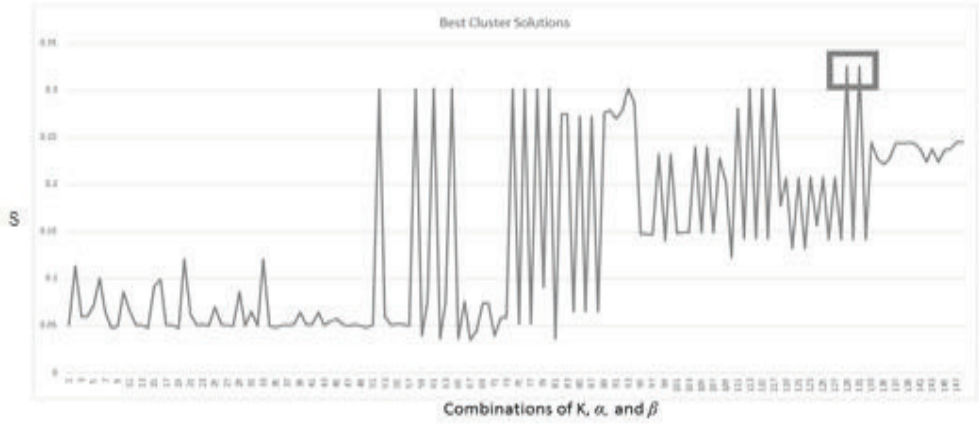


Figure 17 Best cluster solutions that fulfills additional criteria

Then, the exploration continued with additional condition to select the best cluster solution. The code implementation in Python can be found in Appendix 2. Additional conditions defined as follow, the best cluster solution is only selected (1) if it has no empty cluster and, (2) if it has no negative average silhouette score. After both criteria were deployed, the cluster solutions were narrowed into only 148 solutions. From here, it was founded that the best cluster solution which is has the highest silhouette score is cluster with the silhouette score, $s \approx 0.32$, as highlighted in the rectangle in Figure 17. The explanation on how this cluster was selected, will be described later in the next section.

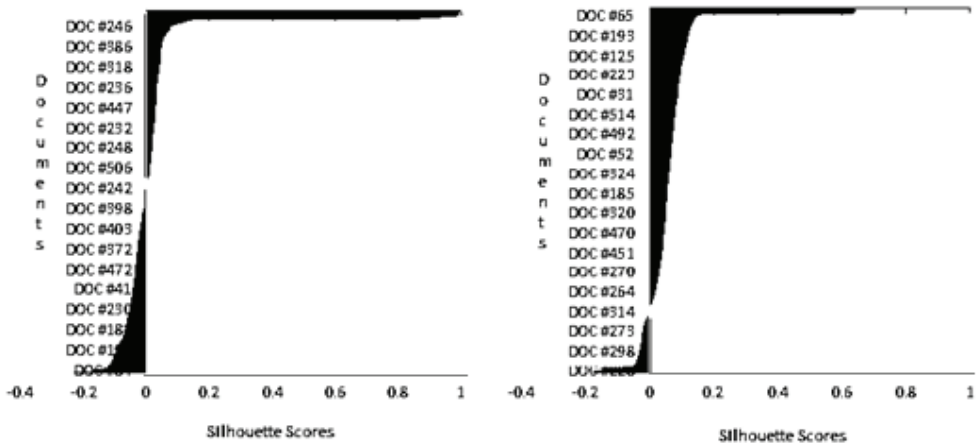


Figure 18 Comparison between one of bad cluster solution with negative average silhouette score (left), and the best cluster solution K=6, without average negative silhouette score (right)

The second additional condition has been added because high silhouette score does not always represent a solution with good quality for each cluster within. For an example, for $K = 7$, $\alpha = 0.65$, and $\beta = 0.8$, from the silhouette score, it may considered as one of best cluster solution, but when it observed deeper, most of its object has negative silhouette score. Our exploration found that the best cluster solution is $K = 6$, $\alpha = 0.9$, and $\beta = 0.9$ visualized as the right graph in Figure 18, because it has the highest silhouette score, s =0.32559, and fulfills both additional criteria.

Table 6 List of the SSE of first top 5 cluster solution for $K = 6$, and the best cluster solution

α	β	*Number of Clusters with Negative Avg Silhouette*	*Number of Empty Clusters*	Σ *SSE*
0.75	0.75	1	3	38,390.313
0.7	0.75	1	3	39,011.439
0.6	0.8	2	0	36,790.285
0.55	0.8	2	0	36,790.285
0.65	0.8	2	1	37,082.906
0.9	**0.9**	**0**	**0**	**35,413.153**

Sum of Squared Error

This evaluation also help to understand the nature of the cluster solution. It has been noticed that the greater the value of α and β in a cluster solution, then the lower SSE it resulted. Table 6 empowers our reason to add the additional condition since the cluster solutions that have not satisfy both additional condition, tend to have higher SSE. Thus, those are not recommended as the best solution despite of they have high silhouette score. The illustration on how we filter the best cluster solution using Microsoft Excel can be found in Appendix 4.

Risk Cluster Analysis

The sortation was limited up to 200 terms which are mostly appear in the documents and represent the character of the cluster, and Table 7 shows of the most weighted terms (by selecting the term with negative polarity then accumulate those TF-IDF and sentiment weight) in each cluster.

To get proportional measurement, the Risk Score is obtained by dividing the total term score by total number of documents in the cluster. For instance, in cluster 1, total term weight is -1078.607, and total number of documents is 42, thus, the Risk Score is -25.681.

To indicate how big the risk is, the sum of TF-IDF score and negative sentiment score were accumulated in each cluster. While each cluster has unique characteristic, in contrary, each of them has variation sector of business. This indicates that specific business does not always has specific risk, so, the bank may be more aware and more thoroughly in analyzing every loan proposal that come in, not treating it in the same way they analyze previous proposal with the same business sector.

The result also shown that the type of risk found in the cluster solution is related to four of 5Cs (Character, Capacity, Capital, Condition, Collateral) Credit criteria (Gumparthi 2010), that is commonly used to make lending decision. The most founded criteria in the corpus is Capacity, while Character and Capital is not considered as not too significant as the top ranked terms did not reflect this criteria.

Table 7 Risk cluster analysis and its corresponding 5Cs criteria

Cluster (Rank)	Risk Analysis	Number Documents	Risk Score	Corresponding 5C Criteria
1(3)	Related to business segment's condition, e.g. "manajemen", "kelancaran","kewajaran" ,"perhotelan", "real estate", "restaurant".	42	-25.681	Condition
2(2)	Related to business segment's condition, e.g. "komersialisasi" (commercialization), "cabbotage", "wanprestasi"(failed to repay), "pelanggan"(customers).	206	-26.654	Condition
3(6)	Related to business capital, e.g. "solvabilitas", "ketidakcukupan"(insufficiency), "mengcover"(to cover)	34	-24.491	Capital
4(1)	Related both to business capital, e.g. "sewa"(rent), "kadaluarsa"(expired), "tunggakan"(arrears), "denda"(penalty).	2	-33.815	Capital
5(4)	Related to business condition, e.g. "pesaing"(competitor), "investasi"(investment), "pengalaman"(experience)"penyewa"(tenant),	232	-25.601	Condition
6(5)	Related to production capacity, e.g. "pengembangan"(development), "distribusi" (distribution), "profitabilitas" (profit), "optimalisasi" (optimization)	3	-24.498	Capacity

Loan Proposal Documents Assesment

The loan proposal documents assesment includes the whole proposed process above as seen in Figure 19, where the result of document suppression combined with the result risk measurement are utilized to retrieve the best matching documents and the risk level respectively. After the average of keywords are

obtained, it used to calculate the cosine similarity against 519 documents, where the vector d_i from formula (7) of each document is obtained from the matrix V, the right singular value from SVD. Then, the cosine similarity scores are sorted ascendingly to get the best matching documents at the top list. The risk level also provided and visualized as a bar at the right side of the query result table. The more redish the color is indicate that the document has higher risk, and the more greenish the color is indicate that the document has a lower risk. The snapshot of the user interface as seen in Figure 20. This risk leveling were inspired by the credit scoring in quantitative model where the result of credit score is divided into several level, such as low, medium, and high. Each risk level from each document is retrieved from the risk measurement process.

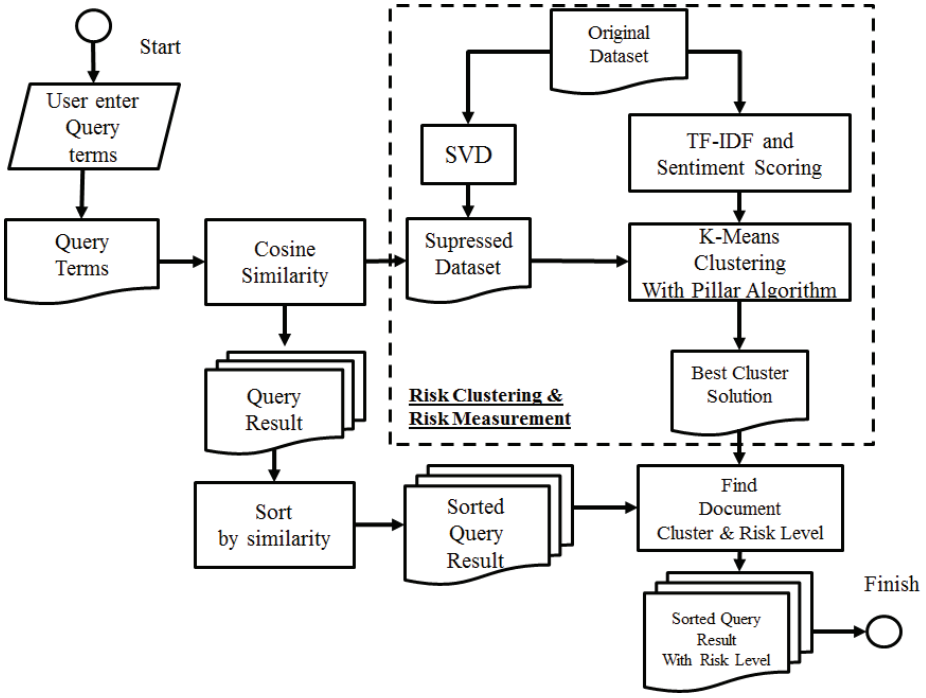


Figure 19 A complete step of document query process

The program flow is begin with the user entering the keywords, let say "apartemen property". At first, the keyword must labeled using the POS Tagging API to get the proper term POS tag. On this example, the keywords labeled as "apartemen|n" and "property|n", since both terms are noun. The next step is to lookup the vector value of both keywords from matrix U, and calculates the average using Formula (8), where n is the number of keywords.

$$q = \frac{\overline{q_1}+\overline{q_2}+\cdots+\overline{q_n}}{n} \tag{8}$$

Since the risk analysts perspective are based on the business segmentation, the performance evaluation of this search engine were based on the business segmentation such as "rumah sakit, obat" (hospital, medicine), "apartemen, properti" (apartment, property), "garmen, pakaian" (garment, apparel), and "mobil, kendaraan" (car, vehicle). The search results then manually analyzed to determine whether the context is related to the keywords.

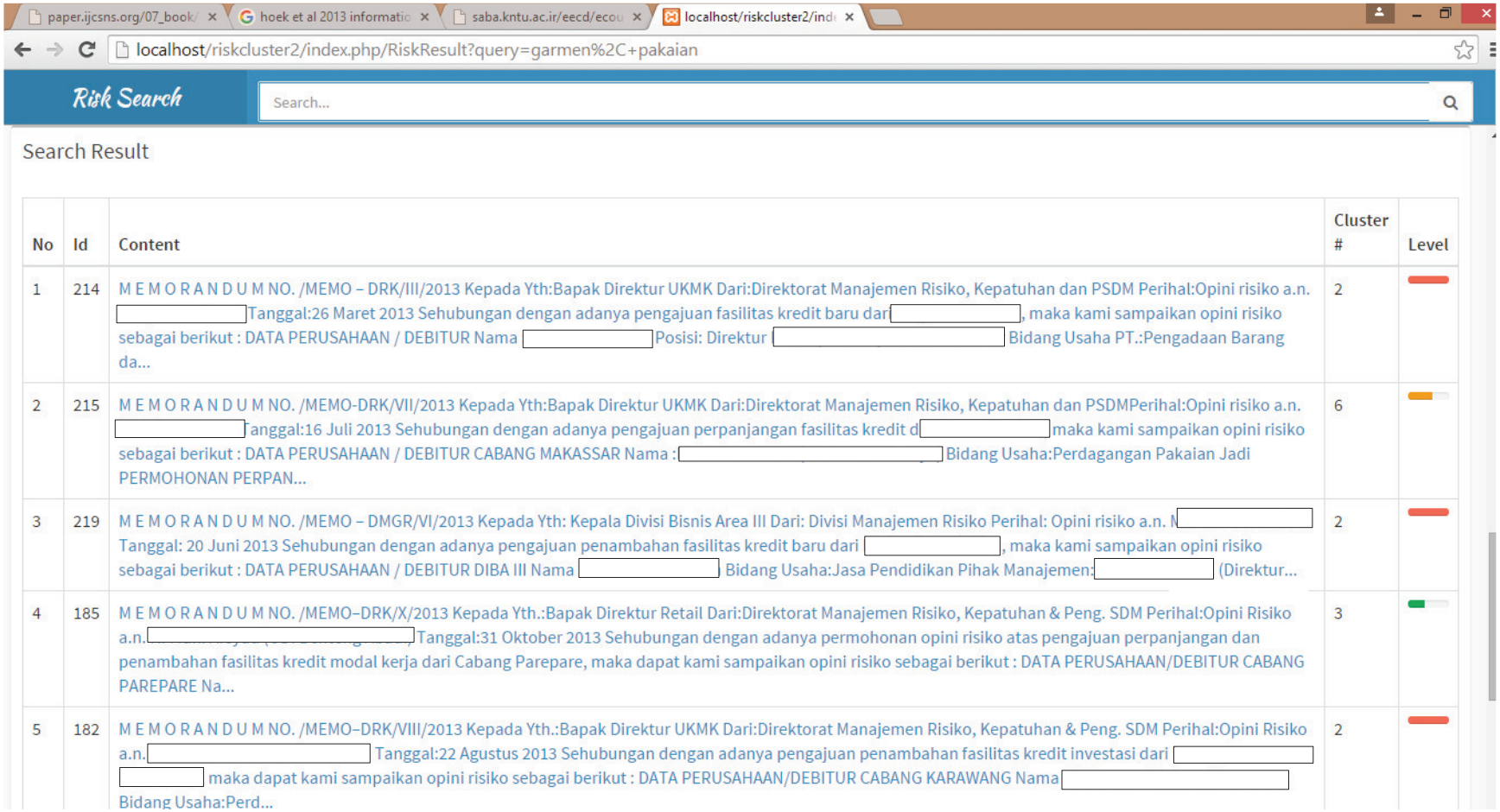


Figure 20 The user interface of the search engine prototype. Contain information about the analysis from risk analysts, cluster, and risk level (indicated by the colors) from the document.

Since the SVD decomposes the documents based on its concept, not all of the top results are explicitly contain the entered keywords (Thomo 2015). However, at least the semantic concept of the top ten results are relatively related to the entered keywords. This result would be useful to help the risk analysts to learn about the risk exposures and determine the risk mitigation based on the previous analysis documents.

A simple search engine prototype also has been created to help the analysts search the risk according the keyword they entered. The implementation of this search engine are based on cosine similarity as seen in formulation (7). The design of this search engine as seen in Figure 19. The first program is the web-based user interface program that allows user to enter the keyword and displays the search result that contains the fragment of the documents, cluster id, and the risk level. The color in the most right column represents the risk level of a document, the more reddish, then the higher the risk, and the more greenish, then the lower the risk. The program was developed using PHP 5.3 programming language, one of famous and reliable web programming language. And the second program is the query processor that developed using Python that preprocesses the query, computes the cosine similarity formulation, and queries the documents from the SQLite database.

The risk cluster model could assist the risk analyst in making analysis regarding the loan proposal. By clustering the risk and retrieving previous analysis from document database, the analyst could understand in the level of the risk from a customer and its risk exposure based on 5Cs credit criteria. Furthermore, it could be expected to help the decision makers to be more objective in making the decision, since the decision is based on a criteria as well as the quantitative model. In the other side, the risk analysis documents would well managed and not become useless.

In contrary, there are further challenges and disadvantages from this model if the bank would seriously follow up on what this research has been proposed. First, since this model would change the behaviour of risk analysis and loan decision making by modifying the SOP, the bank would ask the agreement the central bank (Bank Indonesia) to implement this model. Secondly, if the bank would seriously implement this model in loan proposal processing, they might need to integrating this model into their LOS, and it will required some effort such as providing time, cost, and human resource to enhance the LOS application. Another drawback is the model is lack of mechanism to adapt newly inputted documents that will affect the existing cluster and sentiment computation.

5 CONCLUSIONS AND RECOMMENDATIONS

Conclusion

In this research, the clustering task shows that there were six clusters that represent the risk exposures in SME business financing, which were previously analyzed by risk analysts in a national private bank in Indonesia during 2013 to early 2014. The process of clustering task is performed by utilizing K-Means clustering algorithm optimized by Pillar algorithm iterating some possible number of clusters ranging from $K = 2$ to $K = 10$. We also tried to fed the Pillar algorithm by some possible neighborhood parameters combination. These parameters namely α and β, were affected the cluster quality result. By assigning value ranging from 0.05 to 0.95, for each α and β, we found that the best combination are $\alpha = 0.9$ and

$\beta = 0.9$ for $K = 6$, thus we conclude that the best cluster solution for the risk analysis documents that we have collected is six.

This research also shows that sentiment analysis which is now dominated by industry to gain information from the market reception upon their products, can be utilized to measure the risk level despite of its limitation such as only available in English. By sorting top 200 terms which mostly appear in each cluster, we measure the negative sentiment that represents the risk level. The cluster with the highest risk score was the fourth cluster represents Capital criteria, with risk score equal to -33.815, while the cluster with the lowest risk score was the third represents Capital criteria with risk score equal to -24.491.

By performing evaluation task, we found that the best cluster solution that we have mention earlier, has Silhouette function score $s \approx 0.32$. Although this score was not the highest, however, our conclusion was based on a consideration that for $\alpha = 0.9$, $\beta = 0.9$, and $K = 6$ satisfied some criteria that we have defined: (1) has no empty clusters (2) has no average Silhouette score. Moreover, this combination also has less SSE score which is equal to 35,413.153, compared to another cluster solution, with $\alpha = 0.65$, $\beta = 0.8$, and $K = 6$, which has better Silhouette score $s \approx 0.46$, but has worse SSE score, equal to 37,082.906. The second evaluation is the logical analysis against the risk document's cluster. We found that each cluster does represent risk analysis criteria, despite of not all of the 5C's criteria are represented. The first, second, and fifth cluster represent the Condition criteria, while the third, and fourth cluster represent the Capital criteria, as well as the sixth cluster that represents Capacity criteria.

Recommendation

For future works, the cluster model can be integrated with existing Loan Originating System, and expected to empower the decision making. A lot of effort will need to implement this qualitative model in terms of the classification task that not covered in this research. However, by utilizing the outsourcing labor to develop the system, may reduce time and cost. The biggest effort that the bank may need to do is to provide resource of people to conduct the preprocessing. Since it was the process that required a lot of time. Frequent update the data source are also required to enrich the knowledge base and the information regarding risks in SME business, since this research only observed documents from 2013 to the beginning of 2014. And this can be used as a challenge for the future works, that to find the best efficient and effective method in adding new information when there are new documents added.

REFERENCES

Barakbah AR, Kiyoki Y. 2009. A Pillar Algorithm for K-Means Optimization by Distance Maximization for Initial Centroid Designation. IEEE Symposium on Computational Intelligence and Data Mining (CIDM). 1:61-68. doi:10.1109/CIDM.2009.4938630.

[BI] Bank Indonesia. 2003. Peraturan Bank Indonesia nomor 5/8/PBI/2003 tentang Penerapan Manajemen Risiko Bagi Bank Umum.

[BI] Bank Indonesia. 2007. Peraturan Bank Indonesia nomor 9/14/PBI/2007 tentang Sistem Informasi Debitur.

[BI] Bank Indonesia. 2012. Peraturan Bank Indonesia nomor 14/22/PBI/2012 tentang Pemberian Kredit atau Pembiayaan oleh Bank Umum dan Bantuan Teknis dalam rangka pengembangan UMKM.

Denecke K. 2008. Using SentiWordNet for Multilingual Sentiment Analysis. International Council for Open and Distance Education Conference. doi: 10.1109/ICDEW.2008.4498370

Esuli & Sebastiani, 2006. Sentwordnet: A publicly. International Conference on Language Resources and Evaluation (LREC). 1:417-422.

Ghiassi M, Skinner J, Zimbira D. 2013. Tw itter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. Expert Systems with Applications. 40(16):6266-6282.

Gonçalves P, Benevenuto F, Araujo M, Cha M. 2013. Comparing and Combining Sentiment Analysis Methods. Conference on Online Social Networks (COSN). 1:27-38.

Gumparthi S. 2010. Risk Assessment Model for Assessing NBFCs' (Asset Financing) Customers. International Journal of Trade, Economics and Finance, 1(1):121-130.

Harikumar R, Vijayakumar T, Sreejith MG. 2012. Performance Analysis of SVD and K-Means Clustering for Optimization of Fuzzy Outputs in Classification of Epilepsy Risk Level from EEG Signals. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). 1:1:4.

Hipp R, 2000. SQLite Database Management System. 2014. Available for download from https://www.sqlite.org/.

Kou G, Peng Y, Wang G. 2014. Evaluation of clustering algorithms for financial risk analysis using MCDM methods. Information Sciences. 275:1-12.

Lunando E, Purwarianti A. 2013. Indonesian Social Media Sentiment Analysis with Sarcasm Detection. Advanced Computer Science and Information Systems (ICACSIS). 1:195 - 198.

Manning CD, Prabakhar R, Schütze H. 2009. An Introduction of Information Retrieval. Cambridge(EN): Cambridge University Press.

Medhat W, Hassan A, Korashy H. 2014. Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, pp. 1093-1113.

Newey R. 2014. Management of Risk. London(EN): Royal Institution of Chartered Surveyors (RICS).

Nguyen TD. 2015. An approach to look up documents in a library using singular value decomposition. IJCSNS International Journal of Computer Science and Network. 15(1):88-97.

Osinski SL, Stefanowski J, Weiss D. 2004. Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition [thesis]. Poznan (PL): Poznan University of Technology.

Li N, Wu DD. 2010. Using text mining and sentiment analysis for online forums hotspot detection. Decision Support Systems. 48:354-368.

Paltoglou G, Thelwall M. 2010. A study of Information Retrieval weighting schemes for sentiment analysis. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. 1386:1395.

Rousseeuw PJ. 1986. Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. Computational and Applied Mathematics. 20:53-65.

Soares J, Pina J, Ribeiro M, Catalao-Lopes M. 2011. Quantitative vs. Qualitative Criteria for Credit Risk Assessment. Frontiers in Finance and Economics, 8(1):68-87.

Thelwall M. 2010. Sentiment Strength Detection in Short Informal Text. Journal of the American Society for Information Science and Technology, Volume Statistical Cybermetrics Research Group, School of Technology, University of Wolverhampton, Wulfruna Street, Wolverhampton WV1 1SB, UK. 1:2544–2558.

Thomo A. Latent Semantic Analysis (Tutorial). 2015. Available for download from http://www.engr.uvic.ca/~seng474/svd.pdf .

Tekchandani P, Dhole A. 2015. Overview and Preliminary Results For Opinion Mining Using Fuzzy String Searching. International Journal of Advanced Research in Computer Science and Software Engineering, 5(6):1367-1372.

Vinodhini G, Chandrasekaran RM. 2012. Sentiment Analysis and Opinion Mining: A Survey. International Journal of Advanced Research in Computer Science and Software Engineering, 2(6):282-292.

Xu K, Shaoyi S, Li J, Yuxia S. 2011. Mining comparative opinions from customer reviews for Competitive Intelligence. Decision Support Systems. 50:743-754.

Zhai Z, Liu B, Xu H, Jia P. 2011. Clustering Product Features for Opinion Mining Proceedings of the fourth ACM international conference on Web search and data mining. ACM. 347:354. doi: 10.1145/1935826.1935884.

Zhang D, Dong Y. 2004. Semantic, Hierarchical, Online Clustering of Web Search Results. Advanced Web Technologies and Applications, 3007:69-78.

Appendix 1 Pillar Algorithm in Python

```
def Optimized_Centroids(K, P, alpha, beta):
  #=====================================================#
  #         Adopted from Ali Ridha Barakbah's Paper Pillar Algorithm(2005)
  #         Code written by Irfan Wahyudin
  #=====================================================#
  n = len(P)
  NumAttr = len(P[0])

  # alpha = 0.9
  # beta = 0.8
  Centroid = []
  C = None
  SX = None
  DM = []
  nmin = int((alpha * n) / K)
  maxiter = n
  iters = 0
  print alpha, "*", n, "/",K,"=",nmin

  #=====================================
  # Cari mean
  #=====================================
  SumP = [sum(i) for i in zip(*P)]
  m = []
  for sp in SumP:
          temp = float(sp) / n
          m.append(temp)
  # print m

  Distance = []
  DistanceDM = []
  i = 0
  for p in P:
          d = np.linalg.norm(np.array(p) - np.array(m))
          Distance.append([i, d])
          i+=1

  D = sorted(Distance, key=itemgetter(1), reverse=True)

  # print Distance
  iDist = 0
  SX = []
  DM = []
  print DM
```

```
i = 0
  no = 0.0
  C = []
  DM = D
  while (i<K and iters < maxiter):
        print "back to first object...", i, K, iters, maxiter
        dmax = DM[0][1]
        dmax_index = DM[0][0]
        nbdis = beta * float(dmax)

        for x in range(0,n-1):
              try:
                    if DM[x][0] not in SX:
                          zhe = P[DM[x][0]]
                          SX.append(DM[x][0])
                          break
              except:
                    return


        iDist += 1
        j=0
        Distance = []
        for p in P:
              d = np.linalg.norm(np.array(p) - np.array(zhe))
              Distance.append([j, d])
              j+=1
        D = sorted(Distance, key=itemgetter(1), reverse=True)

        no = 0
        for dy in D:
              if dy[1] <= nbdis:
                    no +=1

        if no >= nmin:
              i+=1
              C.append(zhe)
              print "Centroid #",str(i),"Jumlah objek terdekat: ", no, "dari",
              nmin, "yang diperbolehkan"
              iters = 0
              for p in P:
                    d = np.linalg.norm(np.array(p) - np.array(zhe))
                    DistanceDM.append([i, d])

              DM = sorted(DistanceDM, key=itemgetter(1), reverse=True)
  return C
```

Appendix 2 Silhouette Function in Python

```
cluster_population = Populate_Cluster()
        c = 0
        s = 0.0
        c1 = 0
        c2 = 0
        s_list = []
        s_list_all = []
        sum_s = 0.0
        iters = 0
        D = []
        distance_matrix_file = open('f_distance_matrix')
        distance_matrix = distance_matrix_file.read()

        for doc in distance_matrix.split("\n"):
                item_dist = []
                for dist in doc.split(","):
                        item_dist.append(dist)
                D.append(item_dist)
        for curr_cluster in cluster_population:
                s_list = []
                ac = 0
                num_population = float(len(curr_cluster))
                for i in curr_cluster:
                        a = 0.0
                        for member in cluster_population[c1]:
                                if member != i:
                                        # a += abs(np.linalg.norm(np.array(D[member])
                                                - np.array(D[i])))
                                        try:
                                                a += float(D[member][i])
                                        except:
                                                print "member", member, "i",i
                                                return
                        a = float(a) / float(num_population) #Count Average

                        c2 = 0
                        num_population_neigh = 0
                        b_list = []
                        b_temp = 0.0
                        for neigh_cluster in cluster_population:
                                num_population_neigh = 0
                                b_temp = 0.0
                                if c2 != c1:
                                        for member in neigh_cluster:
                                                b_temp += float(D[member][i])
                                                num_population_neigh+=1
```

```
                        b_temp = float(b_temp) /
                        float(num_population_neigh)
                    b_list.append(b_temp)
                    c2+=1

                    b = min(b_list)
                    # print "a", a,"b", b
                    if max(b,a) == 0:
                        s = 0.0
                    else:
                        s = (b - a) / max(b,a)
                    # print "s = ", s
                    s_list.append(s)
                    sum_s += s
                    iters+=1
                    print "s", s
            s_list_all.append(reduce(lambda x, y: x + y,s_list) / len(s_list))
            c1+=1
    print "sum_s",sum_s, sum_s / float(iters)
    return s_list_all
```

Appendix 3 Stopword List

saya
aku
gue
kamu
kau
dia
ia
kami
kalian
beliau
elo
lo
loe
lu
elu
ini
itu
mereka
yang
apa
kapan
kemana
dimana
kenapa
mengapa
siapa
bagaimana
gimana
telah
suatu
karena
dan
atau
tapi
tetapi
dari
ke
di
hingga
selama
pun
sehingga
untuk
dengan
antara
sebelum
sesudah
lagi
maka
jika
sekali
semua
lebih
kurang
paling
beberapa
bukan
hanya
bila
sangat
sama
harus
sekarang
kemarin
yaitu
seperti
bak
yg
dalam
pada
oleh
sebab
memang
tak
tidak
ya
iya
emang
berapa
begitu
juga
begini
gini
gitu
sana
sini
yakni
misalnya
sendiri
justru
tersebut
merupakan
adalah
amat
terlalu
sekian
demikian
menjadi
kepada
makin
semakin
kata
ujar
bagai
bagaikan
melainkan
padahal
sedang
sedangkan
sekedar
sekadar
&
ketika
namun
bisa
saja
beserta
telanjur
belum
sudah
sempat
akhirnya
nyaris
atas
mo
vii
no
xa

Appendix 4 Filtering Clusters Silhouette results on Microsoft Excel

K	Alpha	Beta	Silhouette(S) Overall	S Cluster 1	S Cluster 2	S Cluster 3	S Cluster 4	S Cluster 5	S Cluster 6	Empty Cluster Validation											idation
2	0.05	0.55	0.051045258	0.003818344	0.098272173					1	1	0	0	0	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
2	0.05	0.7	0.113631389	0.001943106	0.225319672					1	1	0	0	0	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
2	0.05	0.75	0.059928748	0.040326584	0.079530911					1	1	0	0	0	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
2	0.05	0.8	0.059928748	0.040326584	0.079530911					1	1	0	0	0	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
2	0.05	0.85	0.070521989	0.019568364	0.121475615					1	1	0	0	0	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
2	0.05	0.9	0.100637397	0.128189477	0.073085316					1	1	0	0	0	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
2	0.05	0.95	0.064414006	0.052781476	0.076046537					1	1	0	0	0	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
2	0.1	0.8	0.047240671	0.069886621	0.024594721					1	1	0	0	0	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
2	0.1	0.85	0.049743901	0.082276242	0.017211559					1	1	0	0	0	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
2	0.1	0.9	0.086566496	0.151205688	0.021927305					1	1	0	0	0	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
2	0.1	0.95	0.06503289	0.057221743	0.072844038					1	1	0	0	0	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
2	0.15	0.65	0.051045258	0.003818344	0.098272173					1	1	0	0	0	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
2	0.15	0.8	0.050795327	0.083360869	0.018229784					1	1	0	0	0	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
2	0.15	0.85	0.047634576	0.075040819	0.020228333					1	1	0	0	0	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
5	0.65	0.95	0.148860671	0.190643705	0.032985767	0.055030235	0.444150382	0.021493268		1	1	1	1	1	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
5	0.7	0.95	0.148860671	0.190643705	0.032985767	0.055030235	0.444150382	0.021493268		1	1	1	1	1	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
5	0.75	0.9	0.240478351	0.092308267	0.078989429	0.010484294	1	0.020609764		1	1	1	1	1	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
5	0.75	0.95	0.148860671	0.190643705	0.032985767	0.055030235	0.444150382	0.021493268		1	1	1	1	1	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
5	0.8	0.9	0.240478351	0.092308267	0.078989429	0.010484294	1	0.020609764		1	1	1	1	1	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
5	0.8	0.95	0.148860671	0.190643705	0.032985767	0.055030235	0.444150382	0.021493268		1	1	1	1	1	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
5	0.95	0.9	0.228156981	0.022423366	0.086532085	0.014344051	1	0.017485401		1	1	1	1	1	0	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
6	0.4	0.9	0.202512539	0.061537911	0.051220848	0.013645059	1	0.024306298	0.064365117	1	1	1	1	1	1	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
6	0.4	0.95	0.122593504	0.057060789	0.057496522	0.083874522	0.447399855	0.029046833	0.060682503	1	1	1	1	1	1	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
6	0.45	0.9	0.280986635	0.52253737	0.068750002	0.01247239	1	0.018103616	0.064056431	1	1	1	1	1	1	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
6	0.45	0.95	0.142489136	0.148001823	0.085674199	0.093110909	0.445987571	0.023563073	0.058597243	1	1	1	1	1	1	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
6	0.5	0.9	0.302056294	0.636305217	0.070668289	0.014940027	1	0.019468226	0.070956007	1	1	1	1	1	1	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
6	0.5	0.95	0.142489136	0.148001823	0.085674199	0.093110909	0.445987571	0.023563073	0.058597243	1	1	1	1	1	1	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive
6	0.55	0.9	0.302056294	0.636305217	0.070668289	0.014940027	1	0.019468226	0.070956007	1	1	1	1	1	1	0	0	0	0	VALID	0 0 0 0 0 0 0 0 0 0 Fully Positive

ppendix 5 An SVD and LSI Tutorial (Thomo 2015; Nguyen 2006)
Suppose there are five documents:

d1: "Krisis-moneter dan Perbankan“
d2: "Perbankan adalah pilar perekonomian",
d3: "Krisis-moneter diyakini akan menghancurkan perekonomian",
d4: "Menambah hutang diyakini oleh pemerintah akan berdampak buruk",
d5: "Pemerintah serius menyehatkan ekonomi negara“

Step 1: Perform prerprocessing by remove less important terms and construct a 5×8 term-document matrix *A*:

	d1	*d2*	*d3*	*d4*	*d5*
krisis-moneter	1	0	1	0	0
Perbankan	1	1	0	0	0
Pilar	0	1	0	0	0
Perekonomian	0	1	1	0	0
Menambah	0	0	0	1	0
Diyakini	0	0	1	1	0
Hutang	0	0	0	1	0
Pemerintah	0	0	0	1	1

Step 2: Perform SVD, in this research we used Python Library to construct SVD

```
SA, EA, UA = np.linalg.svd(A, full_matrices=True)
```

Where *SA* is right singular value, contains term concept of matrix *A*:

```
SA = [-3.961 -3.142 -1.782 -4.383 -2.638 -5.240 -2.638 -3.263]
     [ 2.800  4.495  2.689  3.685 -3.459 -2.464 -3.459 -4.596]
     [-5.711  4.105  4.973  1.287  1.457 -3.386  1.457  3.170]
     [ 4.496  5.130 -2.569 -5.773  4.748 -2.728  4.748  2.372]
     [-1.018  2.039  4.305 -2.196  4.174  1.547  4.174 -7.248]
     [-7.805  7.805  4.144 -4.925 -4.914  5.706 -7.911  1.942]
     [ 2.798 -2.798  2.431  3.669 -4.041 -3.165  7.206  2.775]
     [3.7602 -3.760  5.913 -2.153  4.561 -1.606 -2.954 -3.608]
```

EA is diagonal matrix of singular values, that utilized to get the define *k* subset that represent some represents a percentage numbers of entire dataset. This can be obtained by using Frobenius norm from matrix *A*. Suppose set the threshold *q*=0.8

```
EA = [ 2.285  2.010  1.360  1.118  0.796]
```

$$q(A,2) = \frac{\|A\|_2}{\|A\|}$$

$$q(A,2) = \frac{\sqrt{2.28^2 + 2.01^2}}{\sqrt{2.28^2 + 2.01^2 + 1.36^2 + 1.11^2 + 0.79^2}} = 0.84$$

UA is left singular value of *A*, contains document concept of matrix *A*:

```
UA = [-0.310 -0.407 -0.594 -0.603 -0.142]
```

```
[ 0.362  0.540  0.200 -0.695 -0.228]
[-0.118  0.676 -0.659  0.198  0.232]
[ 0.860 -0.287 -0.358  0.053  0.212]
[ 0.128  0.034 -0.209  0.332 -0.909]
```

Step 3: So with k=2 it is represent about 0.84 of entire dataset. The next step is multiply singular values matrix EA_2 with right singular value SA_2, and left singular value UA_2, we obtained collection of matrices that represents terms:

```
krisis-moneter    = [-0.905,  0.562]
perbankan         = [-0.718,  0.903]
pilar             = [-0.407,  0.540]
perekonomian      = [-1.001,  0.740]
menambah          = [-0.603, -0.695]
diyakini          = [-1.197, -0.495]
hutang            = [-0.603, -0.695]
pemerintah        = [-0.745, -0.924]
```

The same thing also applied to the left singular value UA_2, we obtained collection of matrices that represents documents:

```
d1    = [-0.710,  0.729]
d2    = [-0.930,  1.087]
d3    = [-1.358,  0.402]
d4    = [-1.378, -1.397]
d5    = [-0.326, -0.459]
```

This left singular value UA_2, is used later as a feature in clustering documents using K-Means with Pillar algorithm.

Step 4: Cosine similarity is used to retrieve the information and most related documents when a user supplies query terms against the document collection. For example, if a user queries the terms perbankan and perekonomian, the first task is looking up to the SA_2 for both terms, in this case:

```
perbankan         = [-0.718,  0.903]
perekonomian      = [-1.001,  0.740]
```

Afterwards, it calculate the average value of both terms:

$$qx = \frac{\begin{bmatrix}-0.718\\0.903\end{bmatrix} + \begin{bmatrix}-1.001\\0.740\end{bmatrix}}{2}$$

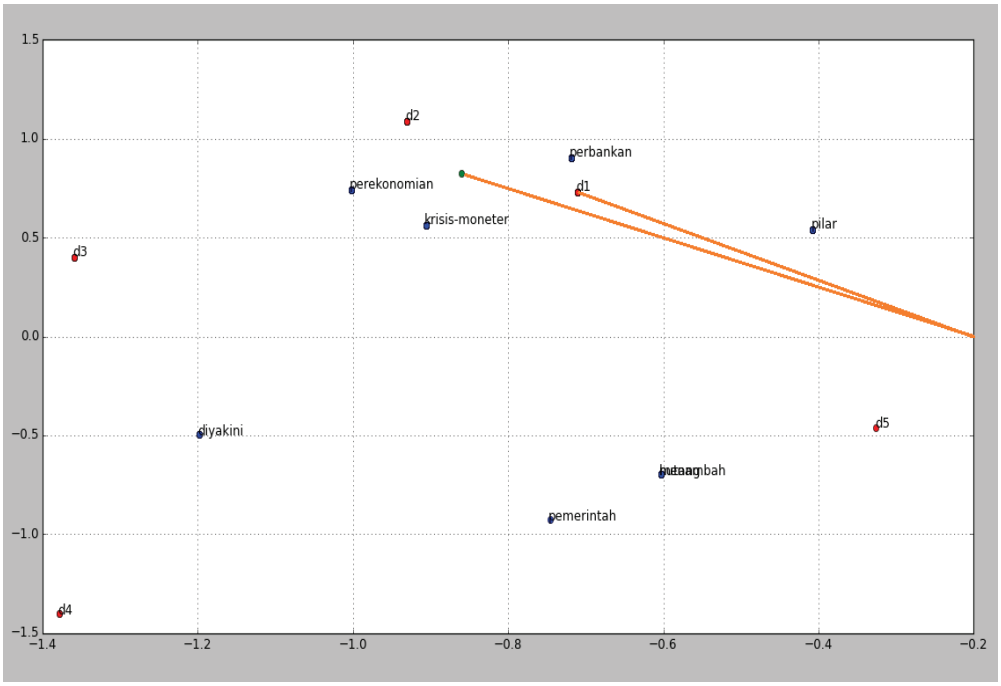
$$qx = \begin{bmatrix}-0.859\\0.822\end{bmatrix}$$

Finally, the average value of query terms qx is used to measure the similarity to the existing documents $d_1,\ldots,d_5$, and the result is:

$\cos(qx, d_1) = 0.999$
$\cos(qx, d_2) = 0.995$
$\cos(qx, d_3) = 0.889$
$\cos(qx, d_4) = 0.015$

cos(qx, d_5) = -0.145

The calculation summarized that the terms perekonomian, and perbankan are closely related to the d_1, and less related to d_5



Appendix 6

Supposed we have a term-document matrix as follow:

	d_1	d_2	d_3	d_4	d_5
krisis-moneter	1	0	1	0	0
Perbankan	1	1	0	0	0
Pilar	0	1	0	0	0
Perekonomian	0	1	1	0	0
Menambah	0	0	0	1	0
Diyakini	0	0	1	1	0
Hutang	0	0	0	1	0
Pemerintah	0	0	0	1	1

The Tf-Idf score for the term “krisis-moneter” is:

Tf(krisis-moneter, d1) = 1

Idf(krisis-moneter) $= \log\left(\frac{5}{2}\right)$

$= 0.397$

Tf-Idf(krisis-moneter, d1) $= Tf \times Idf$

$= 1 \times 0.397$

$= 0.397$

Appendix 7

Levensthein distance measure between two strings *a* with length *i* and *b* with length *j* is mathematically denoted as follow (Tekchandani and Dhole 2015):

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + 1 \\ \mathrm{lev}_{a,b}(i,j-1) + 1 \\ \mathrm{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

The algorithm will measures by comparing the edit distance between each character in both strings. If it is found that the *i-th* and *j-th* character are different, then it will assign the edit variable with 1, and later to be used to found the minimum value between deletion, insertion, and substitution (the first, second, and third row in "otherwise" block).

Supposed there are 2 strings to compare:
t_1: "perbankan"
t_2: "perburuan"

	-	p	e	r	b	a	n	k	a	n
-	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
p	**1**	0	1	2	3	4	5	6	7	8
e	**2**	1	0	1	2	3	4	5	6	7
r	**3**	2	1	0	1	2	3	4	5	6
b	**4**	3	2	1	0	1	2	3	4	5
u	**5**	4	3	2	1	1	2	3	4	5
r	**6**	5	4	3	2	2	2	3	4	5
u	**7**	6	5	4	3	3	3	3	4	5
a	**8**	7	6	5	4	3	4	4	3	4
n	**9**	8	7	6	5	4	3	4	4	**3**

Levensthein Distance = 3

BIOGRAPHY

The author was born in July 1st, 1983, as the third son of from Zakaria Nurdin, and Romelah. He was graduated from SMA Negeri 2 Surabaya in 2001. Continued to study in Mathematic Department, Institut Teknologi Sepuluh Nopember (ITS) Surabaya starting from 2002, and graduated as Bachelor of Science in 2007. In 2013, attended to pursuit the master degree in Computer Science Department, Bogor Agricultural University (IPB) Bogor.

The author also has a career experience as a software engineer from 2006 to 2008 in a software developer company with expertise in API programming for electricity metering device. And in 2008, jumped in to banking industry as a system analyst in a national private bank in Indonesia until 2014. In 2014, the author also managed to certified as Project Manager based on PMBOK 2008. Later in 2015 the author promoted as a system architect to design and lead the software engineer team to develop the IT service for banking industry based on Service Oriented Architecture (SOA).

Some skills and expertises that author has are project management in software development, and programming in RPG LE for IBM AS/400 iSeries, Microsoft .NET, PHP, as well as Python that used to solve most of the challenges in this research.