UNIVERSITI TEKNOLOGI MARA

SEARCHING INTERMEDIARY CLOSELY RELATED LANGUAGES ON BIG GRAPH DATA TO FIND MEDIATOR FOR CONFLICT RESOLUTION BETWEEN INDONESIAN TRIBES

SHELLA ELDWINA FITRI

BACHELOR OF SCIENCE (Hons.)
COMPUTER AND MATHEMATICAL SCIENCES

JANUARY 2021

Universiti Teknologi MARA

Searching Intermediary Closely Related Languages on Big Graph Data to Find Mediator for Conflict Resolution between Indonesian Tribes

Shella Eldwina Fitri

Report submitted in fulfillment of the requirements for Bachelor of Science (Hons.)
Faculty of Computer and Mathematical Sciences

January 2021

SUPERVISOR'S APPROVAL

SEARCHING INTERMEDIARY CLOSELY RELATED LANGUAGES ON BIG GRAPH DATA TO FIND MEDIATOR FOR CONFLICT RESOLUTION BETWEEN INDONESIAN TRIBES

By

SHELLA ELDWINA FITRI
2020682678

This report was prepared under the direction of supervisors, Dr. Rizauddin Saian and Dr. Arbi Haza Nasution, M.IT. It was submitted to the Faculty of Computer and Mathematical Sciences and was accepted in partial fulfillment of the requirements for the degree of Bachelor of Sciences (Hons.).

Approved by:

Dr. Rizauddin Saian	Dr. Arbi Haza Nasution
Supervisor	Supervisor

January 22, 2021

STUDENT'S DECLARATION

I certify that this report and the research to which it refers are the product of my own work and that any ideas or quotation from the work of other people, publish or otherwise are fully acknowledge in accordance with the standard referring practices of the discipline.

SHELLA ELDWINA FITRI
2020682678

JANUARY 22, 2021

ACKNOWLEDGEMENTS

Alhamdulillah, praises and thanks to Allah because of His Almighty and His utmost blessing, I was able to finish this research within the time duration given. Shalawat and salam are conveyed to the Prophet Muhammad shallallahu ‘alaihiwassalam, by saying *Allahumma shali’ala Muhammad wa ‘ala ali Muhammad Assalamualaika ya Rasulullah*, may Allah’s blessing be always upon him, his family, his companions, and his followers.

This research would not be possible without the contribution of various people. Firstly, my special thanks go to my supervisors, Dr. Rizauddin Saian and Dr. Arbi Haza Nasution for the continuous support to the writer’s study and research, for their patience, motivation, enthusiasm, and immense knowledge. Their guidance helped the writer in all the time of research and writing of this final year project.

Besides the advisors, the writer to would like to grateful acknowledgment the contributors; Akmar Efendi, S.Kom., Ana Yulianti, ST, M.Kom., Madam Diana Simaryunie Binti Mohd Nasir, and all Informatics Engineering lecturers at Universitas Islam Riau for their suggestions, knowledge, motivations, and encouragements.

The writer would also like to grateful her beloved parents and family for their prayers, supports, and motivations in accomplishing this research. I would like to thank my dearest father Edwin Saputra, my mother Dianitika, my sister Bella Windiana Putri, and my brother Bagus Aldio Saputra.

I express deep and sincere gratitude to my beloved friend, M. Eggy Al Fachrezy for his motivation, prayers and their sincere help during my studies, and all of my dearest friend from Universiti Teknologi MARA and Informatics Engineering, Universitas Islam Riau who were with me and supported me through thick and thin. Last but not least thanks for everything.

ABSTRACT

Indonesia is a country with diverse ethnic and cultural backgrounds. However, this diversity sometimes creates social problems in society, such as conflicts between tribes. With such differences in tribal languages, it will be difficult for deliberations to resolve conflicts. To address these issues, we aim to find intermediary closely related languages on big graph data using the best-performing pathfinding algorithms to be able to find mediators in resolving inter-tribe conflicts in Indonesia. We analyze the performance of three pathfinding algorithms, which are Dijkstra, A*, and Yen's K algorithm by comparing execution time, geographical distance, and the total lexical distance of the intermediary languages as cost. The research findings show that even though Dijkstra and Yen's K algorithm have equal total cost and geographical distance for all cases, Yen’s K is the fastest at searching for intermediary languages that are closely related while Dijkstra is the slowest. Even though A* always has the highest total cost due to the consideration of additional information in the form of a heuristic function that is geospatial distance, the A* algorithm shows a promise by getting one case with the smallest total geographical distance between languages. Therefore, the A* has the potential to be used to select a mediator with a distance closer to the conflicting languages, while in the general case, Yen’s K algorithm can be considered.

Keywords: Closely related languages, conflict resolution, Indonesian tribe languages, pathfinding algorithms.

TABLE OF CONTENTS

CONTENTS **PAGE**

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ASJP	Automated Similarity Judgement Program
LDND	Levenshtein Distance Normalized Divided

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Indonesia with the motto “Bhinneka Tunggal Ika” (Unity in Diversity) clearly states that Indonesia is a country with diverse ethnic and cultural backgrounds. However, this diversity sometimes creates social problems in society, such as conflicts between tribes that have occurred in several areas. Some conflicts were large and claimed many lives, such as the conflict in Sambas between the Malay and Madurese tribes in 1999, a conflict that occurred in 2001 between the Dayak and Madurese tribes, and another conflicts like between the Batak and Nias tribes in Kampar Kiri, Riau. In many cases, conflicts and violence with ethnic nuances, religions that break out in the community are more motivated by social, economic, and political conditions than differences in belief (Pelly, 1999).

Conflict resolution can be carried out by deliberation to get an agreement between the conflicting parties. However, if people from tribe A can only communicate using language A, and people from tribe B can only communicate using language B, then communication cannot occur. For that, it is necessary to have someone from another party who acts as a mediator. The mediator can come from tribe C who communicates using the C language which is closely related to both A language and B language, thus the conflicts are resolved. So that, to choose a mediator, one can refer to the similarity of language used by the mediator with the conflicting parties. For that, it is necessary to know what is related to this research, such as comparative linguistics, Automated Similarity Judgment Program (ASJP), graph theory, and pathfinding algorithm.

Comparative linguistics is a branch of historical linguistics that is concerned with language comparison to determine historical relatedness and to construct language families (Lehmann, 2013). The genetic relationship of languages is used to classify languages into language families. Closely-related languages are those that come from the same origin or proto-language and belong to the same language family. Swadesh List is a classic compilation of basic concepts for historical-comparative linguistics.

The ASJP, open-source software was proposed by Holman et al. (2008) with the main goal of developing a database of Swadesh lists for all of the world's languages from which lexical similarity or lexical distance matrix between languages can be obtained by comparing the word lists. The lexical similarity or lexical distance is useful, for instance, for classifying a language group and for inferring its age of divergence. The classification is based on a 100-item reference list of Swadesh and further reduced to 40 most stable items (Holman et al., 2008). The item stability is a degree to which words for an item are retained over time and not replaced by another lexical item from the language itself or a borrowed element. Words resistant to replacement are more stable. Stable items have a greater tendency to yield cognates (words that have a common etymological origin) within groups of closely related languages.

Graph theory has demonstrated its impact on machine learning applications such as classification, prediction, and recommendation (Hou et al., 2019). Computing the path in big data generally consumes a lot from the computation perspective and makes computation much more complex (Selim & Zhan, 2016). The real-world graph can be both structurally large and complex (Yamazaki et al., 2019). Pathfinding algorithms build on top of graph search algorithms and explore routes between nodes, starting at one node and traversing through relationships until the destination has been reached (Needham & Hodler, 2019). Some pathfinding algorithms to compute the shortest path between a pair of nodes are Dijkstra, A*, and Yen's K.

1.2 Problem Statement

With the diversity of tribes in Indonesia, conflicts between tribes often occur. Each tribe has a different language. With the diversity of tribes in Indonesia, conflicts between tribes often occur. Each tribe has a different language. However, it will be difficult to resolve a conflict that occurs between two tribes if they both communicate using different languages. Therefore, a person from another party who speaks a language that is closely related to both tribal languages is needed to act as a mediator and resolve the conflict. So then, our research wants to determine the right mediator based on the similarity of communication language to the two conflicting tribes and to compare the best performance of the Dijkstra, A*, and Yen's K pathfinding algorithms in finding intermediary closely related languages on big graph data as mediators to resolve tribal conflicts in Indonesia.

1.3 Objectives

The main objective is to find the closely related language mediators in resolving inter-tribe conflicts in Indonesia using a pathfinding algorithm.

Specific objectives:

1. To simulate 3 pairs of conflicting tribes and to find the closely related languages to get the mediators.
2. To compare the performance of 3 pathfinding algorithms which are Dijkstra, A*, and Yen's K based on execution time, total cost, and distance between locations on maps.

1.4 Scope and Limitations of The Study

The scope and limitation of the study is the intermediate language search is performed by comparing the performance of only three pathfinding algorithms which are Dijkstra, A*, and Yen’s K. In this study, three pairs of tribal languages are used in the experiment, they are:

1. Bali and Buginese

2. Ambonese Malay and Karo Batak

3. Yogyakarta and Mandar

The three pairs of tribal languages used are not actually conflicting tribes, hereinafter referred to as a simulation of inter-tribe conflict in Indonesia.

1.5 Significance of The Study

As a multi-tribe country, Indonesia has a higher probability of the inter-tribe conflict occurring as a social problem. Every tribe in Indonesia has a different language. Looking back at several tribal conflicts that have occurred in Indonesia, it is possible that these tribal conflicts can recur, whether it is from tribes that have had conflicts before or between tribes that have never been suspected before. Most of the conflicts that occur are motivated by economic problems, beliefs, and disparities between the tribes.

However, the situation will worsen if there is a misunderstanding of the tribal people in Indonesia, whether they are in conflict or this could be the cause of conflict between tribes due to miscommunication, where tribe A only understands and can communicate in language A, and tribe B only understands and can communicate in B language. So, we need a mediator who can understand and communicate using language A and language B, where the mediator serves as a mediator in resolving conflicts between these tribes.

Therefore, the purpose of this study is to help resolve social problems by finding intermediary closely related languages on big graph data using the best-performing pathfinding algorithms in determining which mediators can communicate using similar languages to languages of conflicting tribes to help resolve inter-tribe conflicts in Indonesia if it occurs in the future.

CHAPTER 2
LITERATURE REVIEW

2.1 Tribe Diversity

Indonesia is composed of graphical population composition, according to nationality, religion, ethnicity, and language. According to the 2010 statistical data, there are about 1,340 ethnic groups that spread throughout Indonesia.

Tribe groups are ethnic groups and community cultures that are formed from generation to generation, as part of the community's cultural system. The tribe identity and attributes of a community group will be inherited by the next generation. Culturally, tribe identity and attributes are directly attached to each person, according to the parents' tribes (Na'im & Syaputra, 2011).

According to Mulyana (2008), tribes in Indonesia usually have their own locality identity, for example, the Sundanese are in West Java, the Javanese are in Central and East Java, the Bataknese in North Sumatra, the Ambonese in Maluku, the Buginese in South Sumatra and so on.

2.2 Tribal Conflict

Conflict is sometimes unavoidable in the life of society or organization. Mismatches in social processes can be the cause of a conflict. Theoretically, conflict is often defined as a condition that indicates a dispute between one party and one or more other parties who have different views or interests. Conflict is also a form of struggle to achieve rare things such as value, status, power, authority, and so on, where the conflicting parties are not only in conflict to gain benefits for themselves but also to subdue rivals as the goal of the conflict (Nieke, 2017).

Conflict is a natural occurrence inherent in human life that cannot be avoided. Humans are faced with choices in life to fulfill their needs, which in practice can be contrary to conscience (intrapersonal) and with other humans (interpersonal), causing conflict. Conflict becomes a problem when individuals have a negative view which causes an inability to manage conflict which tends to lead to violent behavior (Sartika, 2017).

In the Law of the Republic of Indonesia Number 7 of 2012 concerning Social Conflict Handling, it is stated that social conflict, hereinafter referred to as conflict, is a physical clash and/or clash with violence between two or more groups of people that occurs within a certain time and has a wide impact resulting in insecurity and social disintegration, thereby disrupting national stability and hindering social development.

Conflicts can occur at any time, can involve anyone and for any cause. A person can become involved in the conflict that is happening around him, whether because of misunderstanding, differences of opinion, customs, culture, tradition, and even ethnic differences.

By having around 1,340 tribes in Indonesia, Indonesia is a country that is rich in diversity. However, sometimes the tribe diversity triggers the emergence of social problems such as tribal conflicts. Tribal conflicts are motivated by various reasons. Starting from welfare inequality, economic problems, and politics.

According to Mulyana (2008), the occurrence of tribal conflicts is closely related to historical aspects where historical writing is unification and uniformity of monocultural nationalism. The government enforces centralization which results in the loss of local identity. There is a kind of indoctrination for understanding nationalism. The formation of a nation should start from the local ethnic dynamics that occur. Local events that occur must be positioned as events that are autonomous and unique but become the basis for the formation of a nation. The values of nationalism were then questioned when several ethnic conflicts emerged, such as those in Sampit, Maluku, Poso, and even ethnic resistance to central power.

2.2.1 Characteristics and Stages of Conflict

According to Wijono (1993) the characteristics of conflict are as follow:

1. The existence of a conflicting interaction involving two or more parties, either individually or in groups.
2. At least there will be a conflict between two or more parties, either individually or in groups in achieving goals, playing roles, and being ambitious or the existence of conflicting values or norms.
3. The emergence of behavioral symptoms that are planned to eliminate, reduce, and suppress other parties which indicate the appearance of interactions that have a positive impact on status, position, responsibility, the fulfillment of various physical needs such as clothing, material, and welfare or certain benefits: cars, houses, bill, or the fulfillment of socio-psychological needs such as security, confidence, love, appreciation, and self-actualization.
4. The emergence of opposite actions as a result of protracted conflict.
5. The emergence of imbalances as a result of the efforts of each party related to the position, social status, rank, class, authority, power, self-respect, pretensions, and so on.

The following are the stages of conflict development (Wijono, 1993) :

1. The conflict is still hidden. Various kinds of emotional conditions were felt as normal and unquestionable as things that bothered him.
2. The conflict that precedes (antecedent conditions) such as the emergence of different goals and values, different roles and so on which indicates a stage of change from what is felt that is not disclosed and does not interfere with itself, the group, or the organization as a whole.
3. Perceived conflicts arise due to unresolved antecedent conditions.

4. Something that manifests in behavior is as a conflict is usually viewed. Various self-defense mechanisms through behaviors that tend to be owned by individuals, groups, and organizations as an effort to anticipate the emergence of conflicts and the causes and consequences of these conflicts.

5. Conflict resolution with various strategies or vice versa are two actions that need to be taken in dealing with a conflict that occurs.

6. In conflict resolution, some consequences must be faced which depend on how the conflict is resolved. If the right and effective strategy are used in resolving conflicts, it can lead to satisfaction and have a positive impact on all parties. However, if the conflict is resolved oppositely, it can be negative for both parties and can affect work productivity.

2.2.2 Tribal Conflict Resolution

Every domestic social conflict resolution does not necessarily depend on the national law enforcement institutions and apparatus, but it is necessary to seriously open up space and involve local community participation in the conflict resolution process. However, its implementation is not always easy, especially when examined from cross-cultural communication, because each party with different cultural backgrounds must have their frame in responding and solving a problem. In this context, cross-cultural communication in resolving conflicts is very important (Bahari, 2008).

In the Law of the Republic of Indonesia Number 7 of 2012 concerning Social Conflict Handling, it is stated that what is meant by conflict resolution is a series of activities carried out in a systematic and planned manner in situations and events both before, during, and after a conflict which includes conflict prevention, stopping conflict, and post-conflict recovery. Conflict prevention is a series of activities carried out to prevent conflict by increasing institutional capacity and early warning systems. Cessation of conflict is a series of activities to end violence, save victims, limit the expansion and escalation of the conflict, and prevent the increase in the number of victims and property loss.

Conflict management requires skills such as effective communication, problem-solving, and function that can encourage increased productivity if the conflict can be managed properly. Resolving a conflict is not a simple matter. Whether a conflict is resolved quickly or not depends on the willingness and openness of the disputing parties to resolve the conflict, the severity or level of the conflict, and the ability to intervene (intervene) by third parties who are also trying to resolve the conflict that arises (Sumaryanto, 2010).

Here's how to resolve conflicts according to Wahyudi (2015) :

1. Refer.
   It is an endeavor to approach and desire for cooperation and to have a better relationship, for the common interest.

2. Persuasion.
   Attempts to change the position of the other party, by showing the harm that may arise, with factual evidence, and by showing that our proposal is favorable and consistent with the norms and standards of justice that apply.

3. Bargaining.
   A settlement acceptable to both parties, by exchanging acceptable concessions. In this way, indirect communication can be used, without making explicit promises.

4. Integrated problem-solving.
   Attempts to solve problems by combining the needs of both parties. The process of exchanging information, facts, feelings, and needs takes place openly and honestly. Generating mutual trust by formulating alternative solutions together with balanced benefits for both parties.

5. Withdrawal.

   A problem solving, in which one or both parties withdraw from the relationship. This method is effective when in the task the two parties do not need to interact and is ineffective when the tasks depend on each other.

6. Coercion and suppression.

   This method forces and presses the other party to surrender. It will be more effective if one party has formal authority over the other party. If there is no difference in authority, threats or other forms of intimidation can be used. This method is often less effective because one party has to give in and give up forcefully.

7. Third-party intervention.

   If the disputing parties are not willing to negotiate or the two parties try to reach a dead end, then a third party can be involved in conflict resolution. Several ways that can be used there are arbitration, mediation, and consultation. Arbitration is when a third party hears the complaints of both parties and serves as a judge seeking binding solutions. This method may not benefit both parties equally, but it is considered better than mutual aggression or destructive actions. Mediation is using an invited mediator to mediate a dispute. Mediators can help gather facts, establish broken communication, clarify and clarify problems and pave the way for integrated problem-solving. The effectiveness of enforcement also depends on the talent and behavioral traits of the mediator. For a consultation, the aim is to improve relations between the two parties as well as develop their capacity to resolve conflicts. The consultant has no discretionary power and does not attempt to mediate. He uses a variety of techniques to increase the perception and awareness that the behavior of both parties is disturbed and malfunctioning, thus hindering the process of resolving the problem that is the subject of the dispute.

2.3 Closely-Related Language

Language is a system of arbitrary sound symbols used by a community to cooperate, interact, and identify themselves. So the language here is a means of communication in social life, both written and oral. Without language, humans cannot interact with other humans.

Closely related languages are those that have the same origin or protolanguage and usually belong to the same language family. According to Gooskens et al (2018), linguistic diversity can lead to communication problems that might only be reconciled with sufficient knowledge about the language situation at hand. The principle of receptive multilingualism is based on the fact that some language pairs are so closely related that the speakers can communicate each using their language without prior language instruction. This strategy is widely used for communication among speakers of the three mainland Scandinavian languages, there are Danish, Swedish, and Norwegian (Bø, 1978; Delsing & Lundin Åkesson, 2005; Maurud, 1976). For example, Danish tourists traveling to Sweden will often speak their mother tongue, Danish, to the Swedes they meet at the camping site or on the street (Abraham & Chapelle, 1992). The Swedes will often react with some hesitation at first, but will often discover that it is possible and even easier to stick to their mother tongue, Swedish, when talking to a Dane.

Lexicostatistics comparisons explain the historical relationships between languages by estimating the percentage of related words together in language pairs. For example, Germanic languages are more closely related to each other than to Romance languages, and vice versa. In the lexicostatistical approach, the percentage of cognates shared by two languages is estimated based on cognacy judgments by experts (Schepens et al., 2013).

The vocabulary used for such cognacy judgments often consists of translation pairs from Swadesh lists. Swadesh lists are small sets of universal culture-free meanings that are robust to changes in meaning and appearance over time. The meaning of items in Swadesh lists is considered to be resistant to borrowings or chance resemblances between languages. Quantifications of the percentage of shared cognates in Swadesh lists can accurately predict language-relatedness (Dyen et al., 1992).

Therefore, it can be concluded that closely related language is a language that has similarities with the target language where closely related languages can be used and understood by the intended language. Closely related language in this study is calculated based on a high similarity value which is useful for finding mediators in resolving conflicts between tribes.

2.4 Automated Similarity Judgement Program (ASJP)

The Automated Similarity Judgment Program (henceforth ASJP) aims to include 40-word lists from all languages of the world. Obtaining lexical distance by comparing lists of words is useful, for example for classifying a language group and for inferring the ages of differences.

ASJP is a project dedicated to the diachronic analysis of the world's linguistic diversity, including the specific task of language classification. Holman et al (2008) a set of 40 highly stable lexical items was selected and subsequently, a large database of wordlists with translational equivalents of these 40 items (or, minimally 70% of the items) in the majority of the world's languages was assembled (Søren & Taraka, 2018). The word lists are transcribed in a simplified ASCII representation already described in several papers (Brown et al., 2008, 2013). Since 2008, the preferred approach to computing distances among languages for further input to various analyses has been a modified version of the Levenshtein or 'edit' distance called LDND (Bakker et al., 2009).

In research conducted by Müller et al (2010), graphically, the world language tree illustrates relative degrees of lexical similarity holding among 4350 of the world's languages and dialects (henceforth, languages) currently found in the ASJP database. Four factors influence lexical similarity registered in the tree: (1) genetic or genealogical relationship of languages, (2) diffusion (language borrowing), (3) universal tendencies for lexical similarity such as onomatopoeia, and (4) random variation (chance). Languages branched closely together on the tree may be so because of strong lexical similarity produced by any one or a combination of the four factors.

2.5 Neo4J

Neo4j is a native graph database, built from the ground up to leverage not only data but also data relationships. Neo4j connects data as it's stored, enabling queries never before imagined, at speeds never thought possible.

In Mathematical terms, a graph is simply a collection of elements - typically called Nodes (also called Vertices or Points) - that are joined together by Edges. Each node represents some piece of information in the Graph, whereas each edge represents some connection between two nodes (Cox, 2017). Neo4j has Cypher language, which is a graph query language and the links between them are known as relationships, links, or edges.

A graph database is a non-relational database that provides an effective and efficient solution for the storage of information in current scenarios, where data is increasingly interconnected. The storage mechanisms of graph databases are optimized for graphing, for the way they store adjacent records linked by direct references. In this adjacency list, each vertex maintains references to their adjacent vertices, forming an index species for the vertices on the neighborhood. This property is known as index-free adjacency (Robinson et al., 2013).

A graph database can store any kind of data using a few simple concepts, there are nodes, relationships, and properties. Nodes or vertices are objects that make up a graph.

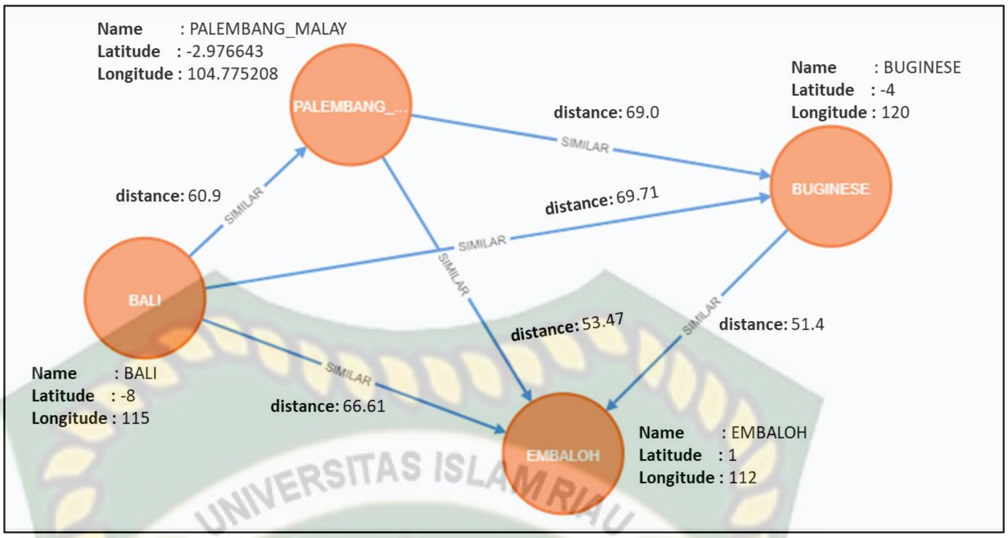


Figure 2.1 Graph Example

There are four nodes in Figure 2.1. Nodes can have a label for classification. The label associates a set of nodes. A label marks a node as part of a group. Properties are attributes of nodes and relationships. A relationship is relating nodes by type and direction. In here, every node has 3 properties, that are *Name*, *Latitude*, and *Longitude* and also has a relationship that is *Similar* with one property named *distance*.

The main characteristics and advantages of a graph database are as follow (Robinson et al., 2013):

- Information search is far more optimized compared to relational databases since it takes advantage of the proximity data from one or more root (main nodes) of the graph database.
- Quite intuitive, due to their natural form of information representation – the graphs.
- Support the data storage in the order of petabytes.
- They are very agile in development since they can be easily adapted over time, either in the insert or in the deletion of information.
- Allow new types of data.
- Suitable for data connected to each other, usually involved in real-world cases.
- Optimized for data mining operations.

Neo4J has many competitive advantages, which makes this software one of the most used ones in this area. The major features of Neo4J are as follow (Fernandes & Bernardino, 2018):

- Cloud-enabled,
- Exporting of query data to JSON and XLS format,
- Most active graph community in the world,
- High-performance thanks to native graph storage and processing,
- Easy to learn and to use,
- Easy to load data into the software,
- Whiteboard-friendly data modeling to simplify the development cycle.

2.6 Pathfinding Algorithms

The pathfinding algorithm is built on the graph search algorithm by tracing the route from one node to another node, traversing the route associated with other nodes until it reaches the destination node. Pathfinding algorithm is used to identify optimal routes that can be used for logistics planning, call routing, or low-cost IP, including game simulations (Needham & Hodler, 2019).

Pathfinding is a study to find out how to get from a source to a destination in a graph. A graph consists of several arcs connecting some nodes. A graph with labels can have more than one description attached to each node which differentiates between the nodes in the graph. Dijkstra is the most common pathfinding algorithm in the computer science literature. A weighted graph is given for Dijkstra to solve the problem of finding the path in the graph with the total weight between a pair of nodes. There are several other algorithms developed for variants of the problem including directed and undirected edges. Graph search is divided into blind search and heuristic search (Cui & Shi, 2011). In this study, Dijkstra, A*, and Yen's K are the pathfinding algorithms used to calculate the shortest path between a pair of nodes.

2.6.1 Dijkstra Algorithm

The Dijkstra algorithm calculates the shortest (weighted) path between a pair of nodes. In this category, Dijkstra's algorithm is the most well-known. It is a real-time graph algorithm that can be used as part of the normal user flow in a web or mobile application.

Dijkstra's algorithm visits vertices in the graph one by one starting with the object's starting point. It then examines the closest vertex which is yet to be examined and this process runs in an outer loop which terminates when either the vertex examined happens to be the target or else if the target is not found even after all the vertices have been examined. Otherwise, the closest vertices to the examined vertex are then added to the collection of vertices to be examined. In this fashion, it expands outwards from the starting point until it reaches the goal. When the target is found, the loop terminates and then the algorithm backtraces its way to the start remembering the required pathfinding the Dijkstra starting from the starting point to the destination point is how Dijkstra's algorithm works. However, this algorithm is not recommended for use to find a target because this algorithm must examine some nodes which results in spending extra time and resources. After all, the number of nodes to be checked will continue to increase. However, if there is a target or destination to look for, this algorithm will serve as the quickest option in finding the shortest path (Anita et al., 2018).

Dijkstra which is useful for finding the optimal route between a node and the destination node is widely used to find the shortest path between locations, for example finding the shortest path from a company to the hospital. In this case, finding the shortest pathway is useful for efficient travel time so that the time needed to get to the hospital is less.

Example use cases include (Needham & Hodler, 2019):

1. Finding directions between locations. The Dijkstra algorithm is applied to Google Maps to provide directions and find the shortest path that connects the starting location to the intended location.

2. Finding the degrees of separation between people in social networks. For example, when viewing someone's profile on LinkedIn, it will show how many people separated someone on the graph, as well as list reciprocal connections. As another example, on Facebook, where when visiting a friend's profile on Facebook, we can see other people's suggested Facebook accounts, where the account is a friend of our friend on Facebook. Facebook will find the possibility for us to also know that person, this is called a friend of a friend.

3. Finding the number of degrees of separation between an actor and Kevin Bacon based on the movies they've appeared in (the Bacon Number). Bacon Number is a Google feature that shows the actor or actress relationship with Kevin Bacon with the assumption that every actress or actor has been linked to Kevin through other actors or actresses.

2.6.2 A* Algorithm

The A* shortest path algorithm improves on Dijkstra's by finding the shortest paths more quickly. This algorithm includes additional information that is used as a heuristic function that becomes a reference for determining the next path to be explored. The heuristic function is geospatial distance. The algorithm was invented by Peter Hart, Nils Nilsson, and Bertram Raphael and described in their 1968 paper "*A Formal Basis for the Heuristic Determination of Minimum Cost Paths*". To reach the destination node, The A* algorithm determines a partial path which is then expanded on each iteration of its main loop based on the estimated cost (heuristic) remaining (Needham & Hodler, 2019).

"*What makes the A* algorithm so appealing is that it is guaranteed to find the best path between any initial point and any ending point, assuming that a path exists."* (Bourg & Seemann, 2004).

The A* is a generic search algorithm that can be used to find solutions for several problems, pathfinding is one of them (Barnouti et al., 2016).

According to Cui & Shi (2011), A* is a generic search algorithm that can be used to find solutions for many problems, pathfinding just being one of them. For pathfinding, the A* algorithm checks repeatedly for the most promising locations it has seen, which are unexplored. The algorithm will finish exploring the location if a location is a destination. Otherwise, it records all neighboring locations for further exploration. A* is probably the most popular path finding algorithm in game AI (Artificial Intelligence) (Patel, n.d.).

2.6.3 Yen’s K Algorithm

The Yen's K-Shortest Paths algorithm is similar to the Dijkstra algorithm, however, the difference is that the algorithm does not only find the closest path between pairs of nodes. This algorithm can calculate the shortest path as many as K paths. This algorithm was invented by Jin Y. Yen in 1971 which he described in "Finding the K Shortest Loopless Paths in a Network". the utility of this algorithm is to get the second, third, and so on shortest paths as much as K which is useful as an alternative path when the first shortest path is not the only desired destination. It is very helpful when needing more than one backup plan (Needham & Hodler, 2019).

2.7 Summary

The literature review supports this research by explaining that having many ethnic groups in Indonesia could increase the potency of conflicts between tribes. For this reason, it is necessary to resolve conflicts between tribes by deliberation involving mediators who are selected based on closely related language. To do that, the word list is obtained from ASJP, then Neo4J is used as a graph database and the pathfinding algorithm will run on it.

CHAPTER 3
RESEARCH METHODOLOGY

This study was conducted to find intermediary closely related languages on big graph data. To do this, the pathfinding algorithm that produces the closest and best performing distance can be used. This chapter will discuss several stages in finding intermediary closely related languages on big graph data including data preparation, experiment design by simulating conflicts between tribes in Indonesia, and algorithm comparisons. Figure 3.1 shows the phase of research required in this study.

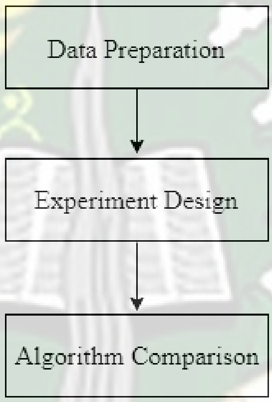


Figure 3.1: The flow of the research phase

3.1 Data Preparation

This study uses a data set from research conducted by Nasution and Murakami (2019). This research conducted visualization of language similarity clusters by utilizing ASJP to generate language similarity. The dataset consists of 119 Indonesian ethnic languages as shown in Table 3.1.

Table 3.1: 119 Indonesian ethnic languages

No	Language	No	Language
1	Abung Sukanda Lampung Nyo	61	Malang
2	Aceh	62	Malay
3	Adumanis Ulu Komering	63	Mambae
4	Ambonese Malay	64	Mandar

5	Anaiwoi Bajau	65	Manggarai
6	Bajoe Bajau	66	Menggala Tulang Bawang Lampung
7	Bali	67	Minangkabau
8	Banggai	68	Mongondow
9	Banjarese Malay	69	Moramo ajau
10	Baree	70	Muna
11	Basemah	71	Ngaju Baamang
12	Batak Angkola	72	Ngaju Oloh Mangtangai
13	Batak Mandailing	73	Ngaju Oloh Mangtangani
14	Belalau Lampung Api	74	Ngaju Pulopetak
15	Betawi	75	Nias Northern
16	Bima	76	Ogan
17	Boepinang Bajau	77	Old Or Middle Javanese
18	Buginese	78	Padei Laut Bajau
19	Coastal Konjo	79	Palembang Malay
20	Daya Lampung Api	80	Perjaya Ulu Komering
21	Delang	81	Pitulua Bajau
22	Ende	82	Pubian Lampung Api
23	Gayo	83	Ramau Lampung Api
24	Gorontalo	84	Rejang
25	Ilir Komering	85	Sadam
26	Indonesian	86	Salako Badamea
27	Indonesian Bajau	87	Samihim
28	Jabung Lampung Api	88	Sangir
29	Jambi Malay	89	Sasak
30	Kadatua	90	Savu
31	Kaleroang Bajau	91	Selayar
32	Kalianda Lampung Api	92	Sika
33	Kambera	93	Sindue Tawaili
34	Kapuas Kahayan	94	Soppeng Buginese
35	Karo Batak	95	Southern Kambera
36	Katingan	96	Sukau Lampung Api
37	Kayu Agung Asli Komering	97	Sumbawa
38	Kayuadi Bajau	98	Sundanese
39	Kerinci	99	Sungkai Lampung Api
40	Kolo Bawah Bajau	100	Tae
41	Komering	101	Talang Padang Lampung Api

42	Konjo	102	Tamuan
43	Kota Agung Lampung Api	103	Tara
44	Krui Lampung Api	104	Tetun
45	Lakaramba Bajau	105	Toba Batak
46	Lakoena Bajau	106	Tolaki
47	Lamaholot Ile Mandiri	107	Tolaki Asera
48	Lampung	108	Tolaki Konawe
49	Lampung Nyo Ambung Kotabumi	109	Tolaki Laiwui
50	Lampung Nyo Melinting	110	Tolaki Mengkongga
51	Langgara Laut Bajau	111	Tolaki Wiwirano
52	Lapulu Bajau	112	Tontemboan
53	Lauru Bajau	113	Tukang Besi Northern
54	Lemo Bajau	114	Tukang Besi Sothern
55	Lewa Kambera	115	Uab Meto
56	Lio	116	Umbu Ratu Nggai Kambera
57	Lom	117	Way Kanan Lampung Api
58	Luwuk Bajau	118	Way Lima Lampung Api
59	Madurese	119	Yogyakarta
60	Makasar		

(Source : Nasution & Murakami, 2019)

In Table 3.1, there are 119 tribal languages in Indonesia, each of which represents a node labeled Language. Each language node has 16 properties. The link between nodes is called a relation. The relation has 2 properties, namely similarity, which means the similarity between the two nodes and the distance which is equal to 100-similarity value. In this study, only 4 properties were selected as important properties there are Distance, Name, Latitude and Longitude.

Distance is the first important property. This property exists in the relationship between nodes. In finding the shortest path between a pair of nodes, the distance selected is the shortest distance. Languages that are close together have a big similarity. However, in the pathfinding algorithm, the algorithm will read that the shortest distance between a pair of nodes is the one with the smallest distance. For this reason, the distance property is used as a property that will be used to measure the cost of finding a similar intermediate language.

Another property is the latitude and longitude at each language location. This property will be useful as additional information or heuristic functions in the A* algorithm. Only the A* algorithm uses the longitude and latitude properties. Lastly, the Name property stores information about the name of the language stored in the dataset.

For Dijkstra and Yen's K algorithms, the Distance and Name properties are used to get the result, while the Latitude and Longitude properties are used after the results are obtained to determine the location of the language on the map.

3.2 Experiment Design

The pathfinding algorithm is useful for finding the shortest path between connected nodes. Types of pathfinding algorithms that can be used to find the shortest path between a pair of nodes are the Dijkstra, A*, and Yen's K shortest path algorithms. Based on this, the three algorithms can be used in finding inter-tribal languages in Indonesia that are closely assembled to find a mediator to resolve tribal conflicts in Indonesia. However, only the algorithm that has the best performance will be selected to be used. This study is divided into 2 trials; one focuses on finding for the language of the mediator without geographical proximity which is the result of Dijkstra and Yen's K algorithms, and the other focuses on finding for the language of the mediator with geographic proximity which is the result of A* algorithm.

Calculating the Levenshtein distance between translated words from the Swadesh list, then taking the average value from the calculation is a way to get the similarity value between languages. Levenshtein distance (LD) is a measure of the similarity between two strings measured from the number of deletions, insertions, or substitutions required.

Mathematically, the Levenshtein distance between two strings a, b is given by lev($length(a)$,$length(b)$) where:

$$lev(i,j) = \begin{cases} \max(i,j)\,, \min(i,j) = 0 \\ min \begin{cases} lev(i-1,j) + 1 \\ lev(i,j-1) + 1\,, \min(i,j) \neq 0 \\ lev(i-1,j-1+eq(i,j) \end{cases} \end{cases} \tag{3.1}$$

Where eq(i, j) returns 1 if $a[i] = b[j]$ and 0 otherwise. For example, the Levenshtein distance between string "kitten" and "sitting" is 3. The steps are: (1) replace "k" with "s", (2) replace "e" with "i", and (3) insert "g" at the end (Siregar et al., 2014). Levenshtein distance algorithm is shown in Table 3.2.

Table 3.2: Levenshtein distance algorithm

Step	Description
1	Set n to be the length of s. Set m to be the length of t. If n = 0, return m and exit. If m = 0, return n and exit. Construct a matrix containing 0..**m rows** and **0..n columns**.
2	Initialize the first row to 0..n. Initialize the first column to 0..m.
3	Examine each character of s (i from 1 to n).
4	Examine each character of t (j from 1 to m).
5	If s[i] equals t[j], the **cost is 0**. If s[i] doesn't equal t[j], the **cost is 1.**
6	Set cell d[i,j] of the matrix equal to the minimum of: a. The cell immediately **above plus 1:** **d[i-1,j] + 1.** b. The cell immediately to the **left plus 1:** **d[i,j-1] + 1.** c. The cell **diagonally above and to the left plus the cost: d[i-1,j-1] + cost**.
7	After the iteration steps (3, 4, 5, 6) are complete, the **distance** is found in cell **d[n, m]**.

(Source: Gilleland, 2006)

In this study, we make the similarity value in the form of a relation property that can be calculated in the algorithm. Similarity property is the similarity between nodes or between languages. The greater the similarity, the higher the level of lexical similarity of the language. On contrary, the smaller the similarity, the lower the level of lexical similarity of the language.

Figure 3.2 shows a graph of the example that refers to the formalization of a graph in research conducted by Nasution & Murakami (2019), where a node represents a language and an edge represents a language similarity between the two languages. The thickness of an edge represents how similar the two languages are. For example, in Figure 3.2, there are two paths for L_A to be connected to L_Z, which are L_A-L_B-L_Z and L_A-L_C-L_Z. Node L_A and node L_B have a similarity of 40, which means the lexical similarity level value is 40. Node L_A and node L_C have a similarity of 30, which means the lexical similarity level value is 30. The same goes for the similarity of node L_B and node L_Z which is 10 and the similarity of node L_C and node L_Z which is 40. The total similarity of path L_A-L_B-L_Z is 50 and the total similarity of path L_A-L_C-L_Z is 70.

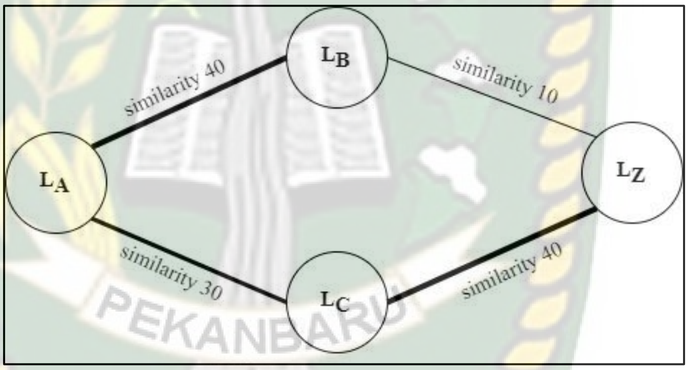


Figure 3.2: Example of a language similarity graph

The pathfinding algorithm works by selecting the path with the shortest cumulative distance from node L_A to node L_Z. In fact, we wanted to find an intermediate languages that was as similar as possible to the source language and target language, meaning the paths with the highest cumulative similarity. Therefore, in this study, we created a property called *distance* with the following formula:

$$distance = 100 - similarity \tag{3.2}$$

Cypher projection is used in this research for the Dijkstra, the A*, and the Yen's K Shortest Path algorithms. In this study, the tribal language used in the experiment is not actually a conflicted tribe, hereinafter referred to as a simulation of conflict between tribes in Indonesia.

```
MATCH (start:Language {name: "BALI"}), (end:Language {name: "BUGINESE"})

CALL gds.alpha.shortestPath.stream({
        nodeQuery:'MATCH(n:Language) RETURN id(n) AS id',
        relationshipQuery:'MATCH(n:Language)-[r:SIMILAR]-(m:Language) WHERE
        r.distance < 61 RETURN id(n) AS source, id(m) AS target, r.distance AS cost',
        startNode: start, endNode: end, relationshipWeightProperty:"cost"
})

YIELD nodeId, cost

RETURN gds.util.asNode(nodeId).name AS Language, cost as Cost;
```

Figure 3.3 : Cypher projection of Dijkstra algorithm

Figure 3.3 shows the Cypher projection of the Dijkstra algorithm. This algorithm declares a start node and an end node representing the source language and the target language. The algorithm works by tracing the path connecting the two nodes. The algorithm will return the path with the minimum distance (relation property value) in tabular form.

```
MATCH (start:Language {name: "BALI"}), (end:Language {name: "BUGINESE"})

CALL gds.alpha.shortestPath.astar.stream({
        nodeQuery: 'MATCH (l:Language) RETURN id(l) AS id, l.latitude AS
        latitude, l.longitude AS longitude',
        relationshipQuery: 'MATCH (l1:Language)-[r:SIMILAR]->(l2:Language)
        WHERE r.distance<61 RETURN id(l1) AS source, id(l2) AS target, r.distance
        AS cost',
        startNode: start, endNode: end,  relationshipWeightProperty: 'cost',
        propertyKeyLat: 'latitude', propertyKeyLon: 'longitude'
})

YIELD nodeId, cost

RETURN gds.util.asNode(nodeId).name AS Language, cost as Cost
```

Figure 3.4: Cypher projection of A* algorithm

Figure 3.4 shows the Cypher projection of the A* algorithm. Just like Dijkstra, this algorithm declares a start node and an end node representing the source language and the target language. The algorithm works by tracing the path connecting the two nodes. In this algorithm, determining the closest distance between the two nodes is not only by calculating the cumulative lexical distance, but also considering a heuristic function in the form of geographical location utilizing longitude and latitude properties. The algorithm will return the path with the minimum total cost in tabular form.

```
MATCH (start:Language{name:"BALI"}), (end:Language{name:"BUGINESE"})

CALL gds.alpha.kShortestPaths.stream({
        nodeQuery:'MATCH(n:Language) RETURN id(n) as id',
        relationshipQuery:'MATCH (n:Language)-[r:SIMILAR]->(m:Language)
        WHERE r.distance<61 RETURN id(n) as source, id(m) as target, r.distance as
        cost',
        startNode: start, endNode: end, relationshipWeightProperty:"cost", k: 1
})

YIELD index, sourceNodeId, targetNodeId, nodeIds, costs, path

RETURN index, [node in gds.util.asNodes(nodeIds[1..-1]) | node.name] AS via,
reduce(acc=0.0, cost in costs | acc + cost) AS totalCost;
```

Figure 3.6: Cypher projection of Yen’s K shortest path

Figure 3.5 shows the Cypher projection of the Yen's K shortest path algorithm. Just like Dijkstra and A* algorithm, at the beginning of the Yen’s K algorithm, the start node and end node are declared representing the source language and the target language. The algorithm works by tracing the path connecting the two nodes. The algorithm will return the path with the minimum distance (relation property value) in tabular form. Unlike the Dijkstra and A* algorithms, the Yen's K algorithm has a variable K, where the value of the K variable determines the number of shortest paths that can connect the two nodes. The value of K is used as a solution to find alternative connected paths. The value of K can be adjusted depending on the needs of an alternative paths to be obtained. However, in this experiment, only the best path is needed, so that the K value is set to 1.

In Figure 3.4, Figure 3.5 and Figure 3.6, distance (property) used is less than 61, which means that the similarity taken between connected languages is more than 39.

3.3 Algorithm Comparison

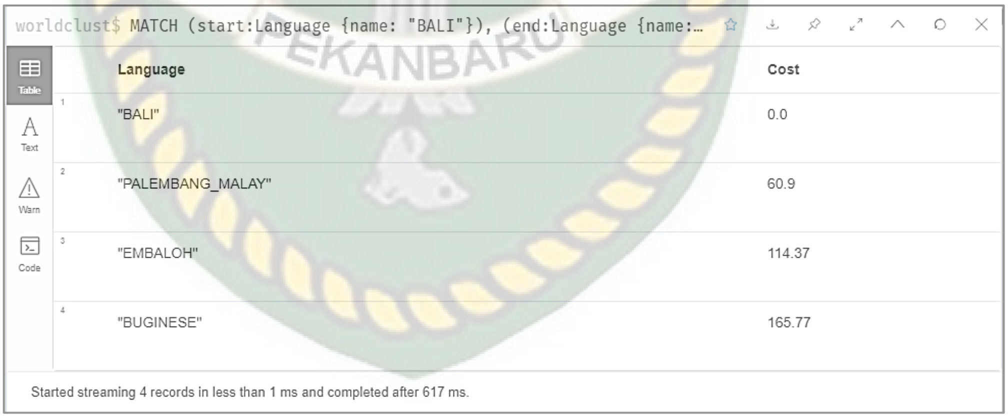
The pathfinding algorithm for both Dijkstra, A*, and Yen's K will return the smallest distance property value that shows the magnitude of the lexical similarity of the two languages. The next step is to compare which algorithm is most suitable for use to find closely related languages in order to find a mediator to resolve inter-tribal conflicts in Indonesia. Algorithm comparison is done by comparing the performance of each algorithm on execution time, total cost, and distance between location on maps as comparison parameters. The distance between location on maps is calculated from the coordinate points obtained from ASJP which are formalized as latitude and longitude properties of each node. On Google Maps, these points will be connected to one another to calculate the distance between the location points.

CHAPTER 4
RESULTS AND DISCUSSIONS

This chapter will discuss the results obtained from the pathfinding algorithms, which are Dijkstra, A*, and Yen's K to find intermediary closely related languages on big graph data. The tribal language used is a simulation of the conflict that can occur. There are three pairs of languages in the simulation, which are Bali-Buginese, Ambonese Malay-Karo Batak, and Yogyakarta-Mandar. The performance of the algorithm will be compared based on execution time, total cost, and distance between location on maps.

4.1 Result of Dijkstra Algorithm

Figure 4.1 shows the results of the Neo4j Cypher projection from Bali to Buginese using the Dijkstra algorithm with a distance property of less than 61.


worldclust$ MATCH (start:Language {name: "BALI"}), (end:Language {name:…

	Language	Cost
1	"BALI"	0.0
2	"PALEMBANG_MALAY"	60.9
3	"EMBALOH"	114.37
4	"BUGINESE"	165.77

Started streaming 4 records in less than 1 ms and completed after 617 ms.

Figure 4.1: The result of Bali to Buginese using Dijkstra

The execution time is 617 ms with a total cost of 165.77 and a route from Bali to Palembang Malay to Embaloh then to Buginese. Then, we calculated the distance between locations on maps based on the coordinates of the location of each language.

Table 4.1: Coordinates and language locations of Bali to Buginese using Dijkstra

Language	Coordinate	Location
BALI	8°20'S, 115°15'E	Buahan Kaja, Payangan, Kabupaten Gianyar, Bali
PALEMBANG_MALAY	2°58'35.9"S, 104°46'30.8"E	Palembang, Lawang Kidul, Kec. Ilir Tim. II, Kota Palembang, Sumatera Selatan 30111
EMBALOH	1°00'00.0"N 112°00'00.0"E	Pulau Majang, Badau, Kabupaten Kapuas Hulu, Kalimantan Barat
BUGINESE	4°00'00.0"S 120°00'00.0"E	Danau Buaya, Danau Tempe, Kabupaten Wajo, Sulawesi Selatan

Table 4.1 shows the location of each language connected from Bali to Buginese based on its coordinates. The geographical location distance calculation on maps is shown in Figure 4.2 where the total distance is 3256.34 km.

Figure 4.2: The distance between locations of Bali (Buahan Kaja) to Buginese (Danau Buaya) on maps using Dijkstra

The results from Bali to Buginese based on execution time, total cost, and distance between locations on maps using the Dijkstra algorithm are presented in Table 4.2.

Table 4.2: The result of Bali to Buginese using Dijkstra

Execution Time (ms)	Total Cost	Distance between Locations on Maps (km)
617	165.77	3256.34

The next language pair is Ambonese Malay to Karo Batak as shown in Figure 4.3.

Figure 4.3: The result of Ambonese Malay to Karo Batak using Dijkstra

With distance property less than 60, the execution time for Ambonese Malay to Karo Batak is 800 ms with a total cost of 72.94 and a route from Ambonese Malay to Ternate Pasar then to Karo Batak.

Table 4.3: Coordinates and language locations of Ambonese Malay to Karo Batak using Dijkstra

Language	Coordinate	Location
AMBONESE_MALAY	3°45'55.8"S 128°08'55.1"E	Nusaniwe, Kota Ambon, Maluku
TERNATE_PASAR	1°00'01.1"S 128°20'09.8"E	Pulau Damar, Kukupang, Kepulauan Joronga, Kabupaten Halmahera Selatan, Maluku Utara
KARO_BATAK	3°00'00.0"N 98°00'00.0"E	Liang Jering, Tanah Pinem, Kabupaten Dairi, Sumatera Utara

Table 4.3 shows the location of each language connected from Ambonese Malay to Karo Batak based on its coordinates. According to the latitude and longitude location of the Ternate Pasar in ASJP, the language is located at Halmahera sea, so we shift the location point to the nearest land. The geographical location distance calculation on maps is shown in Figure 4.4.

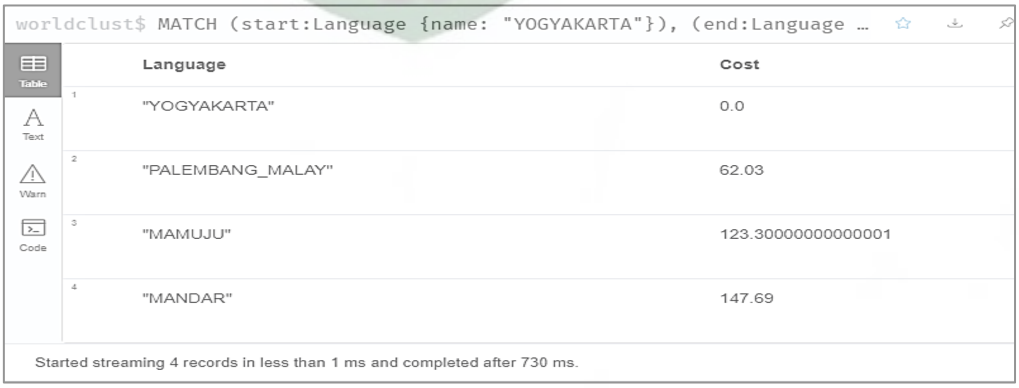
Figure 4.4: The distance between locations of Ambonese Malay (Nusaniwe) to Karo Batak (Liang Jering) using Dijkstra on maps

Figure 4.4 shows the location of each language connected from Ambonese Malay to Karo Batak where a total distance is 3707.23 km. The results from Ambonese Malay to Karo Batak based on execution time, total cost, and distance between locations on maps using the Dijkstra algorithm are presented in Table 4.4.

Table 4.4: The result of Ambonese Malay to Karo Batak using Dijkstra

Execution Time (ms)	Total Cost	Distance between Locations on Maps (km)
800	72.94	3707.23

The last language pair is Yogyakarta to Mandar as shown in Figure 4.5 with a distance property less than 63.


worldclust$ MATCH (start:Language {name: "YOGYAKARTA"}), (end:Language …

	Language	Cost
1	"YOGYAKARTA"	0.0
2	"PALEMBANG_MALAY"	62.03
3	"MAMUJU"	123.30000000000001
4	"MANDAR"	147.69

Started streaming 4 records in less than 1 ms and completed after 730 ms.

Figure 4.5: The result of Yogyakarta to Mandar using Dijkstra

The execution time for Yogyakarta to Mandar is 730 ms with a total cost of 147.69 and a route from Yogyakarta to Palembang Malay to Mamuju then to Mandar.

Table 4.5: Coordinates and language locations of Yogyakarta to Mandar using Dijkstra

Language	Coordinates	Location
YOGYAKARTA	7°00'00.0"S 110°00'00.0"E	Kranggan, Madugowongjati, Kec. Gringsing, Kabupaten Batang, Jawa Tengah
PALEMBANG_MALAY	2°58'35.9"S, 104°46'30.8"E	Palembang, Lawang Kidul, Kec. Ilir Tim. II, Kota Palembang, Sumatera Selatan 30111
MAMUJU	2°00'05.9"S 119°14'18.0"E	Tumbu, Topoyo, Kabupaten Mamuju, Sulawesi Barat
MANDAR	2°19'26.6"S 119°07'52.3"E	Sampaga, Kabupaten Mamuju, Sulawesi Barat

Table 4.5 shows the location of each language connected from Yogyakarta to Mandar based on its coordinates. The geographical location distance calculation on maps is shown in Figure 4.6.

Figure 4.6: The distance between locations of Yogyakarta (Kranggan) to Mandar (Sampaga) using Dijkstra on maps

Figure 4.6 shows the location of each language connected from Yogyakarta to Mandar where a total distance is 2381.18 km. The results from Yogyakarta to Mandar based on execution time, total cost, and distance between locations on maps using the Dijkstra algorithm are presented in Table 4.6.

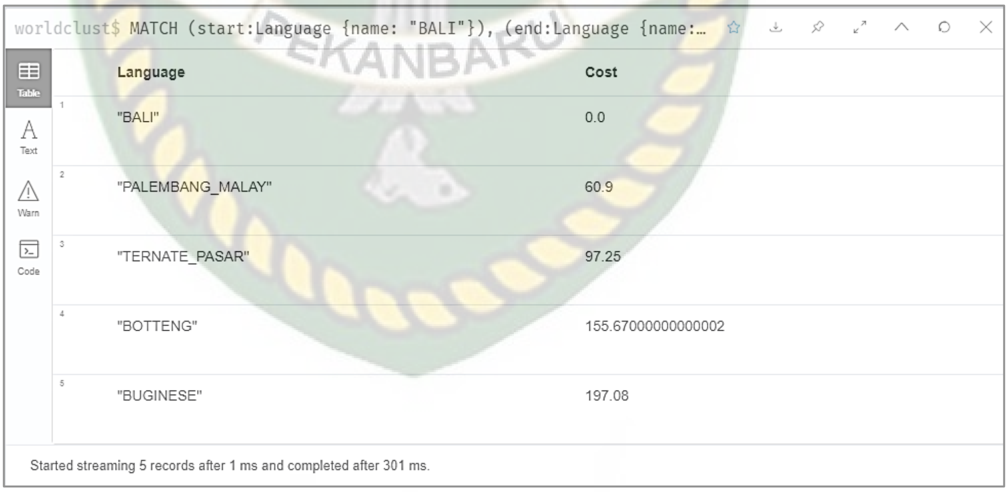
Table 4.6: The result of Yogyakarta to Mandar using Dijkstra

Execution Time (ms)	Total Cost	Distance between Locations on Maps (km)
730	147.69	2381.18

Based on the presentation in Table 4.2, Table 4.4, and Table 4.6 related to the results of Dijkstra's algorithm, it can be concluded that the Bali and Buginese language pairs are fastest for execution time, Ambonese Malay and Karo Batak have the smallest total cost, and Yogyakarta and Mandar have the smallest distance between location on maps.

4.2 Result of A* Algorithm

Figure 4.7 shows the results of the Neo4j Cypher projection from Bali to Buginese using A* algorithm with a distance property of less than 61.


worldclust$ MATCH (start:Language {name: "BALI"}), (end:Language {name:…

	Language	Cost
1	"BALI"	0.0
2	"PALEMBANG_MALAY"	60.9
3	"TERNATE_PASAR"	97.25
4	"BOTTENG"	155.67000000000002
5	"BUGINESE"	197.08

Started streaming 5 records after 1 ms and completed after 301 ms.

Figure 4.7: The result of Bali to Buginese using A*

The execution time for Bali to Buginese is 301 ms with a total cost of 197.08 and a route from Bali to Palembang Malay to Ternate Pasar to Botteng then to Buginese. Then, we calculated the distance between locations on maps based on the coordinates of the location of each language.

Table 4.7 shows the location of each language connected from Bali to Buginese based on its coordinates. According to the latitude and longitude location of the Ternate Pasar in ASJP, the language is located at Halmahera sea so we shift the location point to the nearest land. The geographical location distance calculation on maps is shown in Figure 4.8.

Table 4.7: Coordinates and language locations of Bali to Buginese using A*

Language	Coordinate	Location
BALI	8°20'S, 115°15'E	Buahan Kaja, Payangan, Kabupaten Gianyar, Bali
PALEMBANG_MALAY	2°58'35.9"S, 104°46'30.8"E	Palembang, Lawang Kidul, Kec. Ilir Tim. II, Kota Palembang, Sumatera Selatan 30111
TERNATE_PASAR	1°00'01.1"S 128°20'09.8"E	Pulau Damar, Kukupang, Kepulauan Joronga, Kabupaten Halmahera Selatan, Maluku Utara
BOTTENG	2°55'12.0"S 119°00'00.0"E	Bela, Tapalang, Kabupaten Mamuju, Sulawesi Barat
BUGINESE	4°00'00.0"S 120°00'00.0"E	Danau Buaya, Danau Tempe, Kabupaten Wajo, Sulawesi Selatan

Figure 4.8: The distance between locations of Bali (Buahan Kaja) to Buginese (Danau Buaya) using A* on maps

Figure 4.8 shows the location of each language connected from Bali to Buginese where a total distance is 5154.94 km. The results from Bali to Buginese based on execution time, total cost, and distance between locations on maps using the A* algorithm are presented in Table 4.8.

Table 4.8: The result of Bali to Buginese using A*

Execution Time (ms)	Total Cost	Distance between Locations on Maps (km)
301	197.86	5154.94

The next language pair is Ambonese Malay to Karo Batak as shown in Figure 4.9.

Figure 4.9: The result of Ambonese Malay to Karo Batak using A*

With a distance property less than 60, the execution time for Ambonese Malay to Karo Batak is 304 ms with a total cost of 95.81 and a route from Ambonese Malay to Banjarese Malay then to Karo Batak.

Table 4.9: Coordinates and language locations of Ambonese Malay to Karo Batak using A*

Language	Coordinate	Location
AMBONESE_MALAY	3°45'55.8"S 128°08'55.1"E	Nusaniwe, Kota Ambon, Maluku
BANJARESE_MALAY	1°S, 116°30'E	Riko, Penajam, Kabupaten Penajam Paser Utara, Kalimantan Timur
KARO_BATAK	3°00'00.0"N 98°00'00.0"E	Liang Jering, Tanah Pinem, Kabupaten Dairi, Sumatera Utara

Table 4.9 shows the location of each language connected from Ambonese Malay to Karo Batak based on its coordinates. The geographical location distance calculation on maps is shown in Figure 4.10.

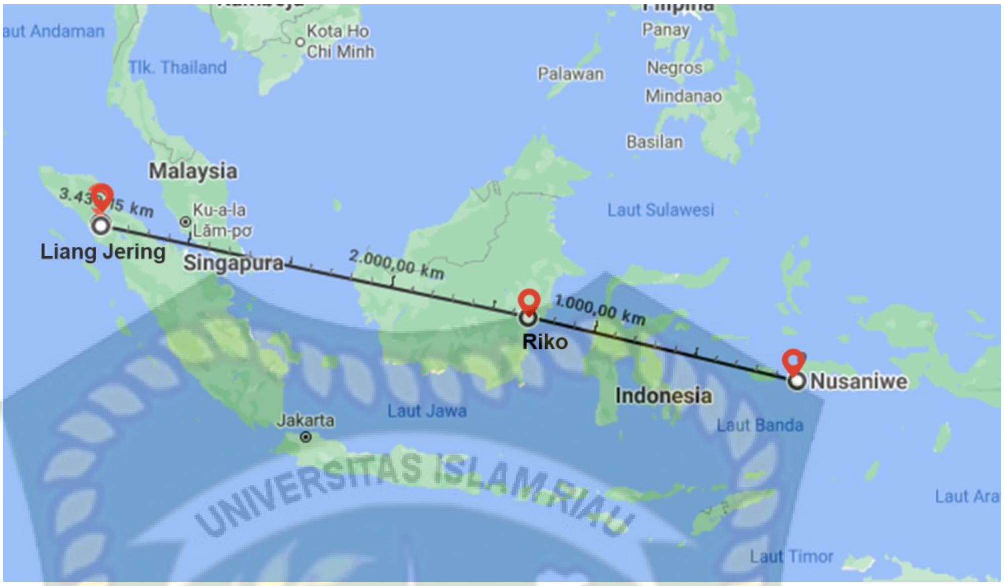


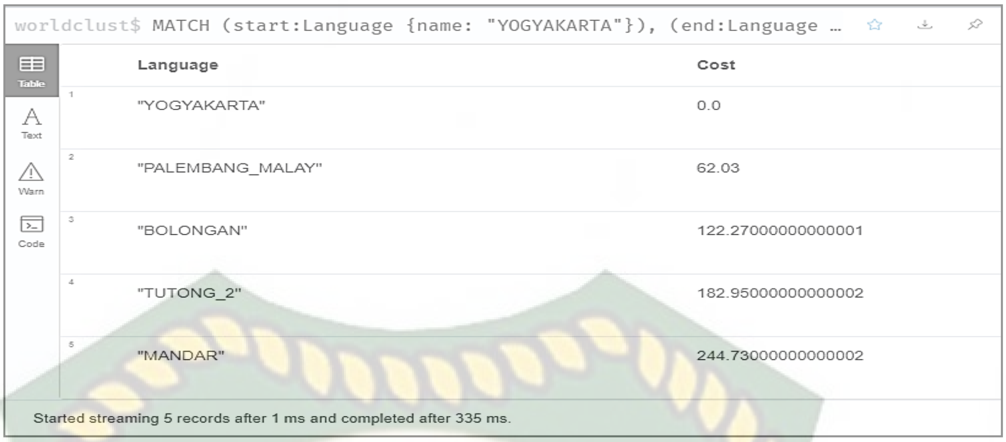
Figure 4.10: The distance between locations of Ambonese Malay (Nusaniwe) to Karo Batak (Liang Jering) on maps using A*

Figure 4.10 shows the location of each language connected from Ambonese Malay to Karo Batak where a total distance is 3439.15 km. The results from Ambonese Malay to Karo Batak based on execution time, total cost, and distance between locations on maps using the A* algorithm are presented in Table 4.10.

Table 4.10: The result of Ambonese Malay to Karo Batak using A*

Execution Time (ms)	Total Cost	Distance between Locations on Maps (km)
304	95.81	3439.15

The last language pair is Yogyakarta to Mandar as shown in Figure 4.11 with a distance property less than 63.


worldclust$ MATCH (start:Language {name: "YOGYAKARTA"}), (end:Language …

	Language	Cost
1	"YOGYAKARTA"	0.0
2	"PALEMBANG_MALAY"	62.03
3	"BOLONGAN"	122.27000000000001
4	"TUTONG_2"	182.95000000000002
5	"MANDAR"	244.73000000000002

Started streaming 5 records after 1 ms and completed after 335 ms.

Figure 4.11: The result of Yogyakarta to Mandar using A*

The execution time for Yogyakarta to Mandar is 335 ms with a total cost of 244.73 and a route from Yogyakarta to Palembang Malay to Bolongan to Tutong 2 then to Mandar.

Table 4.11: Coordinates and language locations of Yogyakarta to Mandar using A*

Language	Coordinate	Location
YOGYAKARTA	7°00'00.0"S 110°00'00.0"E	Kranggan, Madugowongjati, Kec. Gringsing, Kabupaten Batang, Jawa Tengah
PALEMBANG_MALAY	2°58'35.9"S, 104°46'30.8"E	Palembang, Lawang Kidul, Kec. Ilir Tim. II, Kota Palembang, Sumatera Selatan 30111
BOLONGAN	3°N, 117°30'E	Salim Batu, Kec. Tj. Palas Tengah, Kabupaten Bulungan, Kalimantan Utara
TUTONG_2	4°47'32.7"N 114°37'17.1"E	Pekan Tutong, Brunei Darussalam
MANDAR	3°00'00.0"S 119°00'00.0"E	Lombang, Kec. Malunda, Kabupaten Majene, Sulawesi Barat

Table 4.11 shows the location of each language connected from Yogyakarta to Mandar based on its coordinates. The geographical location distance calculation on maps is shown in Figure 4.12.

Figure 4.12: The distance between locations of Yogyakarta (Kranggan) to Mandar (Lombang) on maps

Figure 4.12 shows the location of each language connected from Yogyakarta to Mandar where a total distance is 4992.86 km. The results from Yogyakarta to Mandar based on execution time, total cost, and distance between locations on maps using the A* algorithm are presented in Table 4.12.

Table 4.12: The result of Yogyakarta to Mandar using A*

Execution Time (ms)	Total Cost	Distance between Locations on Maps (km)
335	244.73	3657.28

Based on the presentation in Table 4.8, Table 4.10, and Table 4.12 related to the results of A* algorithm, it can be concluded that the Bali and Buginese pairs are faster for execution time, Ambonese Malay and Karo Batak have the smallest total cost and distance between location on maps.

4.3 Result of Yen's K Shortest Path Algorithm

As discussed in the previous chapter, Yen's K algorithm is different from the Dijkstra algorithm and A* because there is a K value that can be adjusted as needed. In this study, to measure the best algorithm performance, the K value used is 1, meaning that there is only 1 shortest path returned. However, we will show the results of using K = 4 for the first language pair, Bali to Buginese in Figure 4.13.

worldclust$ MATCH (start:Language{name:"BALI"}), (end:Language{name:"BU...

	index	via	totalCost
1	0	["PALEMBANG_MALAY", "REMUN", "BOTTENG"]	196.1
2	1	["PALEMBANG_MALAY", "TERNATE_PASAR", "BOTTENG"]	197.08
3	2	["PALEMBANG_MALAY", "TAMUAN", "BOTTENG"]	205.79
4	3	["PALEMBANG_MALAY", "TERNATE_PASAR", "SANGIL"]	210.25

Started streaming 4 records after 16 ms and completed after 275 ms.

Figure 4.13: The result of Bali to Buginese using Yen's K where K=4

In the results shown in Figure 4.13, according to the K value used, there are 4 routes selected. The first route with an index of 0 is from Bali to Palembang Malay to Remun to Botteng then to Buginese with a total cost of 196.1. The second route with an index of 1 is from Bali to Palembang Malay to Ternate Pasar to Botteng then to Buginese with a total cost of 197.08. The third route with an index of 2 is from Bali to Palembang Malay to Tamuan to Botteng then to Buginese with a total cost of 205.79. The last route with an index of 3 is from Bali to Palembang Malay to Ternate Pasar to Sangil then to Buginese with a total cost of 210.25. The execution time required to obtain these 4 pathways in Yen's K algorithm is 275 ms.

Next, we will show the results of the Yen's K algorithm execution for three language pairs using the value of K = 1 to find only the shortest path. Figure 4.14 shows the results of the Neo4j Cypher projection from Bali to Buginese using Yen's K algorithm with distance property less than 62.

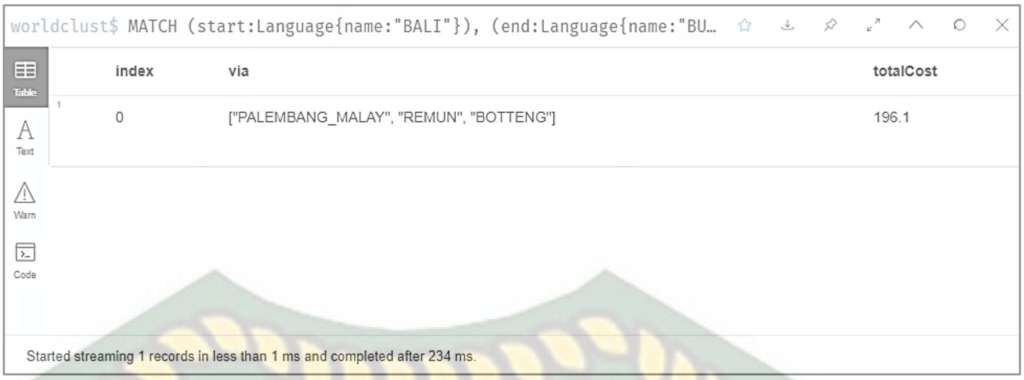


Figure 4.14: The result of Bali to Buginese using Yen's K

The execution time for Bali to Buginese is 234 ms with a total cost of 196.1 and a route from Bali to Palembang Malay to Remun to Botteng then to Buginese. Then, we calculated the distance between locations on maps based on the coordinates of the location of each language.

Table 4.13: Coordinates and language locations of Bali to Buginese using Yen's K

Language	Coordinate	Location
BALI	8°20'S, 115°15'E	Buahan Kaja, Payangan, Kabupaten Gianyar, Bali
PALEMBANG_MALAY	2°58'35.9"S, 104°46'30.8"E	Palembang, Lawang Kidul, Kec. Ilir Tim. II, Kota Palembang, Sumatera Selatan 30111
REMUN	1°05'24.0"N 110°40'12.0"E	Divisi Serian, Sarawak, Malaysia
BOTTENG	2°55'12.0"S 119°00'00.0"E	Bela, Tapalang, Kabupaten Mamuju, Sulawesi Barat
BUGINESE	4°00'00.0"S 120°00'00.0"E	Danau Buaya, Danau Tempe, Kabupaten Wajo, Sulawesi Selatan

Table 4.13 shows the location of each language connected from Bali to Buginese based on its coordinates. The geographical location distance calculation on maps is shown in Figure 4.15.

Figure 4.15: The distance between locations of Bali (Buahan Kaja) to Buginese (Danau Buaya) using Yen's K on maps

Figure 4.15 shows the location of each language connected from Bali to Buginese where a total distance is 3285.62 km. The result from Bali to Buginese based on execution time, total cost, and distance between locations on maps using the Yen's K algorithm are presented in Table 4.14.

Table 4.14: The result of Bali to Buginese using Yen's K

Execution Time (ms)	Total Cost	Distance between Locations on Maps (km)
243	196.1	3285.62

The next language pair is Ambonese Malay to Karo Batak as shown in Figure 4.16.

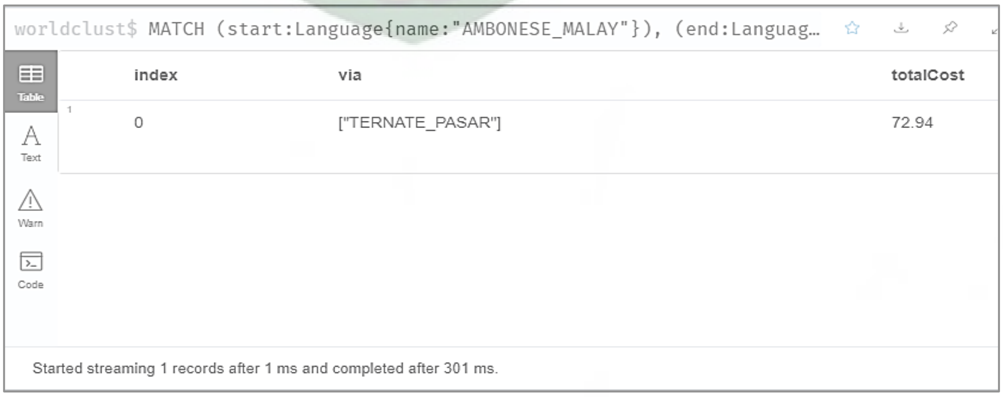


Figure 4.16: The result of Ambonese Malay to Karo Batak using Yen's K

With distance property less than 60, the execution time for Ambonese Malay to Karo Batak is 301 ms with a total cost of 72.94 and a route from Ambonese Malay to Ternate Pasar then to Karo Batak.

Table 4.15: Coordinates and language locations of Ambonese Malay to Karo Batak using Yen's K

Language	Coordinate	Location
AMBONESE_MALAY	3°45'55.8"S 128°08'55.1"E	Nusaniwe, Kota Ambon, Maluku
TERNATE_PASAR	1°00'01.1"S 128°20'09.8"E	Pulau Damar, Kukupang, Kepulauan Joronga, Kabupaten Halmahera Selatan, Maluku Utara
KARO_BATAK	3°00'00.0"N 98°00'00.0"E	Liang Jering, Tanah Pinem, Kabupaten Dairi, Sumatera Utara

Table 4.15 shows the location of each language connected from Ambonese Malay to Karo Batak based on its coordinates. The geographical location distance calculation on maps is shown in Figure 4.17.

Figure 4.17: The distance between locations of Ambonese Malay (Nusaniwe) to Karo Batak (Liang Jering) using Yen's K on maps

Figure 4.17 shows the location distance that connects from Ambonese Malay to Karo Batak where a total distance is 3707.23 km. The result from Ambonese Malay to Karo Batak based on execution time, total cost, and distance between locations on maps using the Yen's K algorithm are presented in Table 4.16.

Table 4.16: The result of Ambonese Malay to Karo Batak using Yen's K

Execution Time (ms)	Total Cost	Distance between Locations on Maps (km)
301	72.94	3707.23

The last language pair is Yogyakarta to Mandar as shown in Figure 4.18 with a distance property less than 63.

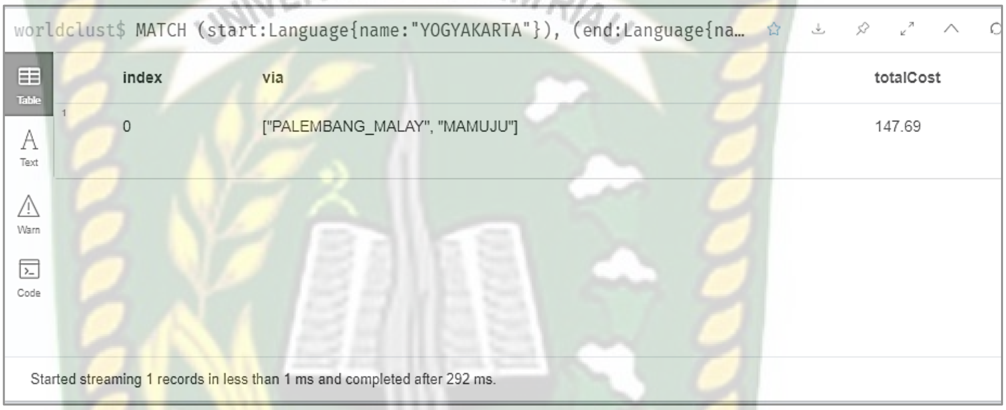


Figure 4.18: The result of Yogyakarta to Mandar using Yen's K

The execution time for Yogyakarta to Mandar is 292 ms with a total cost of 147.69 and a route from Yogyakarta to Palembang Malay to Mamuju then to Mandar.

Table 4.17: Coordinates and language locations of Yogyakarta to Mandar using Yen's K

Language	Coordinate	Location
YOGYAKARTA	7°00'00.0"S 110°00'00.0"E	Kranggan, Madugowongjati, Kec. Gringsing, Kabupaten Batang, Jawa Tengah
PALEMBANG_MALAY	2°58'35.9"S, 104°46'30.8"E	Palembang, Lawang Kidul, Kec. Ilir Tim. II, Kota Palembang, Sumatera Selatan 30111
MAMUJU	2°00'05.9"S 119°14'18.0"E	Tumbu, Topoyo, Kabupaten Mamuju, Sulawesi Barat
MANDAR	2°19'26.6"S 119°07'52.3"E	Sampaga, Kabupaten Mamuju, Sulawesi Barat

Table 4.17 shows the location of each language connected from Yogyakarta to Mandar based on its coordinates. The geographical location distance calculation on maps is shown in Figure 4.19.

Figure 4.19: The distance between locations of Yogyakarta (Kranggan) to Mandar (Sampaga) using Yen’s K on maps

Figure 4.19 shows the location distance that connects from Yogyakarta to Mandar where a total distance is 2381.38 km. The result from Yogyakarta to Mandar based on execution time, total cost, and distance between locations on maps using the Yen’s K algorithm are presented in Table 4.18.

Table 4.18: The result of Yogyakarta to Mandar using Yen’s K

Execution Time (ms)	Total Cost	Distance between Locations on Maps (km)
292	147.69	2381.18

Based on the presentation in Table 4.14, Table 4.16, and Table 4.18 related to the results of Yen's K algorithm, it can be concluded that the Bali and Buginese pairs are faster for execution time, Ambonese Malay and Karo Batak have the smallest total cost, and Yogyakarta and Mandar have the smallest distance between location on maps.

4.4 Performance Comparison

Algorithm performance comparison refers to the execution time, total cost, and geographical location distance on the maps. Algorithms are run on a PC with 4GB RAM, an Intel® Core ™ i5-8250U processor, and a 64-bit Windows 10 operating system. Figure 4.20, Figure 4.21, and Figure 4.22 show a comparison of the three parameters.

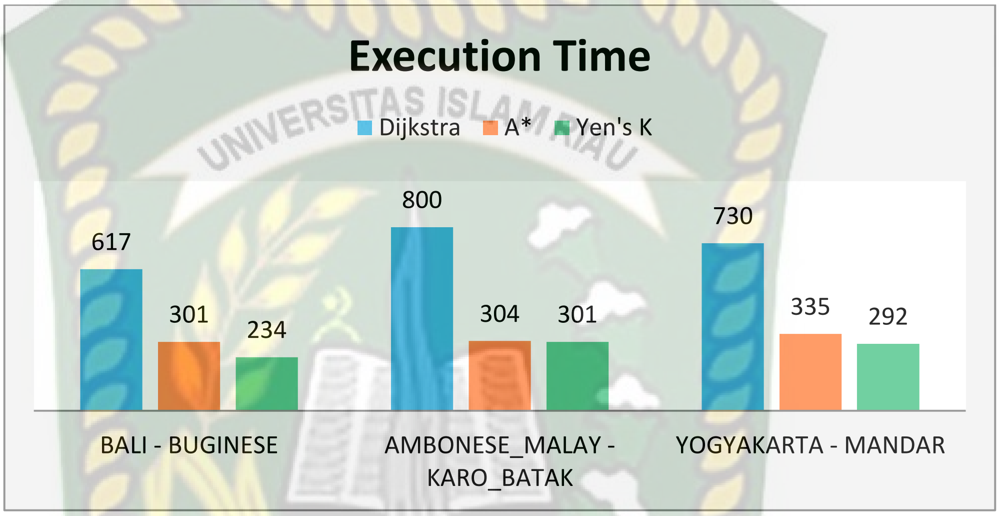


Figure 4.20: Performance comparison by the execution time

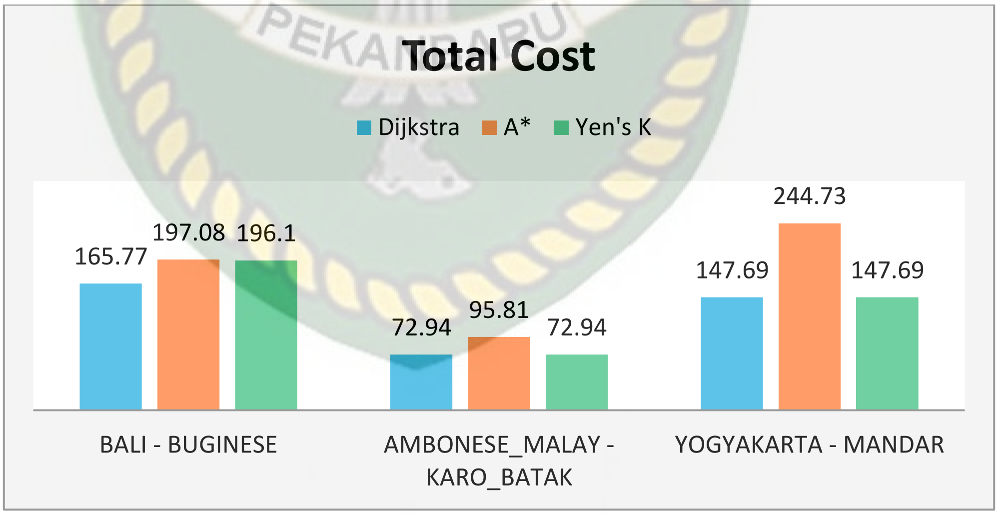


Figure 4.21: Performance comparison by the total cost

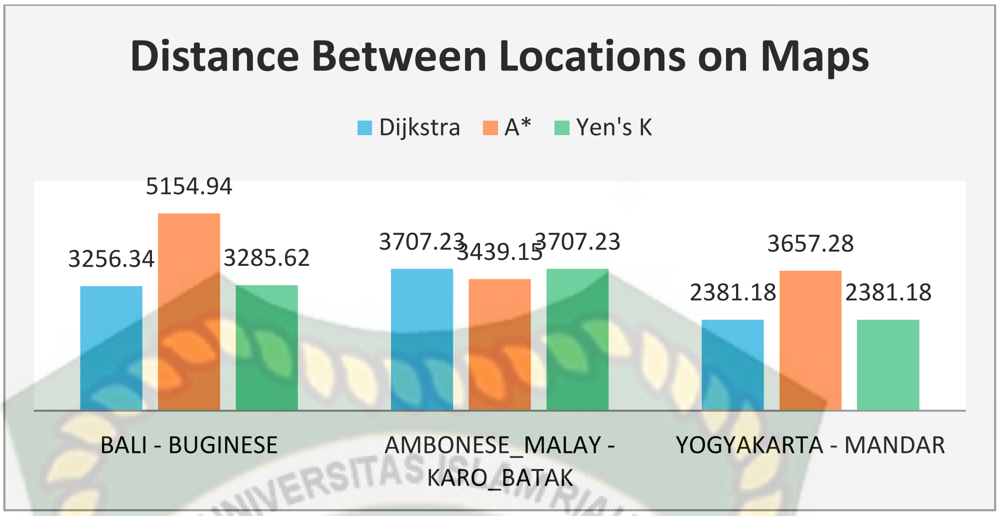


Figure 4.22:Performance comparison by the distance between locations on maps

It can be seen that the Dijkstra and Yen's K have the same results for the total cost and distance between locations. However, the Yen's K algorithm has a faster execution time than the Dijkstra algorithm with an average of 160% higher performance. Based on the execution time, the A* algorithm takes just a little bit longer time than Yen's K algorithm but the difference is less than 100 ms.

For total cost and distance between location on maps, the results show that the A* algorithm underperformed the other algorithms, except for the Ambonese Malay and Karo Batak language pair where it slightly outperformed the other algorithms.

CHAPTER 5
CONCLUSIONS AND RECOMMENDATIONS

This chapter discusses the conclusion and the recommendations for further study regarding this topic.

5.1 Conclusions

The research findings show that even though Dijkstra and Yen's K algorithm have equal total cost and geographical distance for all cases onsidered for Indonesian tribes languages, Yen’s K is the fastest at searching for closely related intermediate languages while Dijkstra is the slowest. Although A* always has the highest total cost due to the consideration of additional information in the form of a heuristic function that is geospatial distance, the A* algorithm shows a promise by getting one case with the smallest total geographical distance between languages. Therefore, the A* has the potential to be used to select a mediator with a distance closer to the conflicting languages, while in the general case, Yen’s K algorithm can be considered.

Mediators from the three pairs of languages used to simulate conflict in this study have been found. Based on the results of the Yen's K algorithm which is considered the best algorithm overall, for the first language pair, Bali and Buginese has three mediators, they are the mediator who uses Palembang Malay, Remun, and Botteng languages. The language pair Ambonese Malay and Karo Batak, the chosen mediator is a mediator who uses the Ternate Pasar language and for the last language pair, that is Yogyakarta and Mandar, mediators with Palembang Malay and mediators with Mamuju languages can be used.

5.2 Recommendations

Based on the results obtained from this study, there are several recommendations for further research. Firstly, both pathfinding algorithm, Yen's K and A*, should be explored further with a larger dataset and additional comparison parameters to determine the best algorithm for a general case. Furthermore, the model used in this study can be implemented to search closely related languages in resolving inter-tribes conflicts in Indonesia based on the level of language similarity, and the distance between the languages to help solve social problems that occur. In the future, this research can also be used to support presidential elections, especially when the candidates want to campaign in areas that still use tribal languages. By using a mediator who uses a similar language as a speaker in the campaign, the number of speakers used does not need to be adjusted to a large number of tribal languages in the area.

REFERENCES

Abraham, R. G., & Chapelle, C. A. (1992). The Meaning of Cloze Test Scores: An Item Difficulty Perspective. *The Modern Language Journal*, *76*(4), 468–479.

Anita, Verma, N., & Bansal, A. (2018). A Review Paper on Examination of Dijkstra's and A* Algorithm to Find the Shortest Path. *International Journal of Creative Research Thoughts*, *6*(2), 635–641.

Bahari, Y. (2008). Model Komunikasi Lintas Budaya Dalam Resolusi Konflik Berbasis Pranata Adat Melayu Dan Madura Di Kalimantan Barat. *Jurnal Ilmu Komunikasi UPNYK*, *6*(1), 103867.

Bakker, D., Müller, A., Velupillai, V., Wichmann, S., Brown, C. H., Brown, P., Egorov, D., Mailhammer, R., Grant, A., & Holman, E. W. (2009). Adding typology to lexicostatistics: A combined approach to language classification. *Linguistic Typology*, *13*(1), 169–181.

Barnouti, N. H., Al-Dabbagh, S. S. M., & Naser, M. A. S. (2016). Pathfinding in Strategy Games and Maze Solving Using A* Search Algorithm. *Journal of Computer and Communications*, *04*(11), 11.

Bø, I. (1978). *Ungdom og naboland [Youth and Neighboring Country]* (rapport 4). Rogalandsforskning.

Bourg, D. M., & Seemann, G. (2004). *AI for Game Developers*. O'Reilly Media, Inc.

Brown, C. H., Holman, E. W., & Wichmann, S. (2013). *Sound Correspondences in The World's Languages*. 4–29.

Brown, C. H., Holman, E. W., Wichmann, S., & Velupillai, V. (2008). Automated Classification of the World's Languages: A Description of the Method and Preliminary Results. *STUF - Sprachtypologie Und Universalienforschung*, *61*(4), 285–308. https://doi.org/10.1524/stuf.2008.0026

Cox, G. (2017). *Introduction to Graph Databases*. https://www.compose.com/articles/introduction-to-graph-databases/

Cui, X., & Shi, H. (2011). A * -based Pathfinding in Modern Computer Games. *International Journal of Computer Science and Network Security*, *11*(1), 125–130.

Delsing, L.-O., & Lundin Åkesson, K. (2005). Håller språket ihop Norden? En forskningsrapport om ungdomars förståelse av danska, svenska och norska. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9).

Dyen, I., Kruskal, J. B., & Black, P. (1992). An Indoeuropean Classification: A Lexicostatistical Experiment. *American Philosophical Society*, *82*(5), iii–132.

Fernandes, D., & Bernardino, J. (2018). Graph databases comparison: Allegrograph, arangoDB, infinitegraph, Neo4J, and orientDB. *DATA 2018 - Proceedings of the 7th International Conference on Data Science, Technology and Applications*, *Data*, 373–380. https://doi.org/10.5220/0006910203730380

Gilleland, M. (2006). *Levenshtein Distance, in Three Flavors*. Merriam Park Software. https://people.cs.pitt.edu/~kirk/cs1501/Pruhs/Spring2006/

Gooskens, C., van Heuven, V. J., Golubović, J., Schüppert, A., Swarte, F., & Voigt, S. (2018). Mutual intelligibility between closely related languages in Europe. *International Journal of Multilingualism*, *15*(2), 169–193. https://doi.org/10.1080/14790718.2017.1350185

Holman, E. W., Wichmann, S., Brown, C. H., Velupillai, V., Müller, A., & Bakker, D. (2008). Explorations in automated language classification. *Folia Linguistica*, *42*(3–4), 331–354.

Hou, Y., Chen, H., Li, C., Cheng, J., & Yang, M. C. (2019). A representation learning framework for property graphs. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *July*, 65–73. https://doi.org/10.1145/3292500.3330948

Lehmann, W. P. (2013). *Historical linguistics: an introduction* (third). Routledge.

Maurud, Ø. (1976). *Nabospråkforståelse i Skandinavia: en undersøkelse om gjensidig forståelse av tale- og skriftspråk i Danmark, Norge og Sverige [Mutual Intelligibility of Neighbouring Languages in Scandinavia. A Study of the Mutual Understanding of Written and Spoken Lang*. Nordiska Rådet.

Müller, A., Wichmann, S., Velupillai, V., Brown, C. H., Brown, P., Sauppe, S., Holman, E. W., Bakker, D., List, M., Egorov, D., Belyaev, O., Mailhammer, R., Geyer, H., & Grant, A. (2010). ASJP World Language Tree of Lexical Similarity : Version 3 (July 2010) by. *Molecular Biology and Evolution*, *3*(July).

Mulyana. (2008). Hubungan Etnis Dalam Pendidikan Sejarah di Indonesia. *Seminar Antarabangsa Hubungan Etnik UKM*, 1–17.

Na'im, A., & Syaputra, H. (2011). *Kewarganegaraan, Suku Bangsa, Agama, dan Bahasa Sehari-hari Penduduk Indonesia*. Badan Pusat Statistik.

Nasution, A. H., & Murakami, Y. (2019). *Kluster Kesamaan : Studi Kasus Bahasa Etnis Indonesia*. 49–59.

Needham, M., & Hodler, A. E. (2019). *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*. O'Reilly Media.

Nieke, N. (2017). Manajemen Dan Resolusi Konflik Dalam Masyarakat. *Jurnal Ilmiah Pendidikan Lingkungan Dan Pembangunan*, *12*(2), 51–60. https://doi.org/10.21009/plpb.122.04

Patel, A. (n.d.). *Introduction to A**. Retrieved December 1, 2020, from http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html

Pelly, U. (1999). Akar Kerusuhan Etnis di Indonesia : Suatu Kajian Awal Konflik dan Disintegrasi Nasional di Era Reformasi. *Antropologi Indonesia*, *58*.

Robinson, I., Webber, J., & Eifrem, E. (2013). *Graph Databases: New Opportunities for Connected Data* (M. Loukides & N. Jepson (eds.); 1st ed.). O'Reilly Media, Inc.

Sartika, R. (2017). Persepsi Mahasiswa Terhadap Konflik Dalam Pembelajaran Mata Kuliah Pendidikan Resolusi Konflik. *Edutech*, *16*(1), 85. https://doi.org/10.17509/e.v16i1.7111

Schepens, J., Dijkstra, T., Grootjen, F., & Heuven, W. J. B. Van. (2013). *Cross-Language Distributions of High Frequency and Phonetically Similar Cross-Language Distributions of High Frequency and Phonetically Similar Cognates*. *May 2014*. https://doi.org/10.1371/journal.pone.0063006

Selim, H., & Zhan, J. (2016). Towards shortest path identification on large networks. *Journal of Big Data*, *3*(1). https://doi.org/10.1186/s40537-016-0042-7

Siregar, T. D. K., Bandung, I. T., & Bandung, J. G. (2014). *Levenshtein Distance Calculation Using Dynamic Programming for Source Code Plagiarism Checking*. 1–6.

Søren, W., & Taraka, R. (2018). 4. Jackknifing the Black Sheep : ASJP Classification Performance and Austronesian. *Senri Ethnological Studies = Senri Ethnological Studies*, *98*, 39–58.

Sumaryanto. (2010). Manajemen Konflik Sebagai Salah Satu Solusi Dalam Pemecahan Masalah. *OPPEK Dosen UNY*, *September*, 1–17. http://staff.uny.ac.id/sites/default/files/pengabdian/dr-sumaryanto-mkes/6-manajemen-konflik-sebagai-salah-satu-solusi-dalam-pemecahan-masalah.pdf

Wahyudi, A. (2015). Konflik, Konsep Teori Dan Permasalahan. *Jurnal Publiciana*, *8*(1), 1–15.

Wijono, S. (1993). *Konflik dalam organisasi / industri dengan strategi pendekatan psikologis*. Satya Wacana.

Yamazaki, K., Shohei, M., Hiroaki, S., & Hiroyuki, K. (2019). Fast RankClus Algorithm via Dynamic Rank Score Tracking on Bi-type Information Networks. *Proceedings of the 21st International Conference on Information Integration and Web-Based Applications & Services*, 110–117.