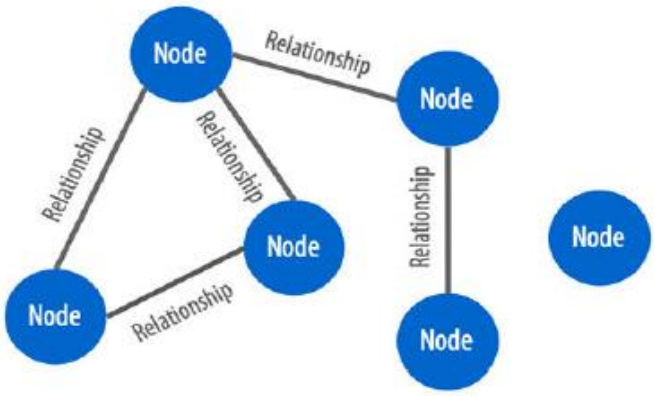
# BAB III
# LANDASAN TEORI

## 3.1 Teori graf

### 3.1.1 Pengertian

Graf merupakan struktur diskrit yang terbentuk dari sebuah tuple yaitu himpunan simpul (*vertices*) dan himpunan sisi (*edges*) yang menghubungkan simpul-simpul pada graf tersebut. Notasi Graf adalah G = (V,E), dimana V adalah himpunan simpul dan E adalah himpunan sisi. Dalam istilah lain, simpul juga disebut *node* dan menghubungkan antar simpul disebut *relationship* (Daniel & Taneo, 2019) (Wilson, 1972).

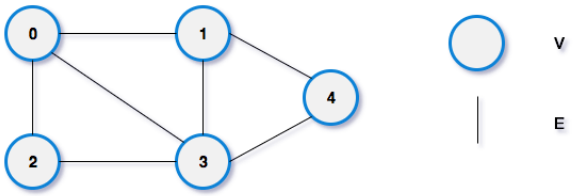



Gambar 3. 1 Representasi graf (Needham & Hodler, 2019)

### 3.1.2 *Orientasi Graf*

1. Graf Tak-Berarah (*Undirected Graph*)

   Graf yang sisinya (E) tidak memiliki orientasi arah. Pada graf tak berarah, urutan pasangan simpul yang dihubungkan oleh sisi tidak di perhatikan. Artinya (u,v) = (v,u) adalah sisi yang sama. Graf tak berarah direpresentasikan pada Gambar 3. 2.




Gambar 3. 2 Graf tak berarah

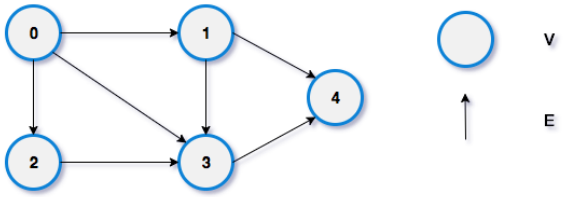
Pada Gambar 3. 2 apabila dibentuk himpunan, sebagai berikut:

$$V = \{0, 1, 2, 3, 4\}$$

$$E = \begin{Bmatrix}\{0,1\},\{0,2\},\{0,3\},\{1,0\},\{1,3\},\{1,4\},\{2,0\},\\ \{2,3\},\{3,0\},\{3,1\},\{3,2\},\{3,4\},\{4,1\},\{4,3\}\end{Bmatrix}$$

2. Graf Berarah (*Directed Graph*)

Graf yang sisinya (E) memiliki orientasi arah. Sisi berarah disebut sebagai arch (busur). Pada graf berarah, (u,v) dan (v,u) menyatakan dua buah busur yang berbeda. Untuk simpul (u,v), simpul u dinamakan simpul asal dan simpul v disebut sebagai simpul terminal.




Gambar 3. 3 Graf berarah

Pada Gambar 3. 3 apabila dibentuk himpunan, sebagai berikut:

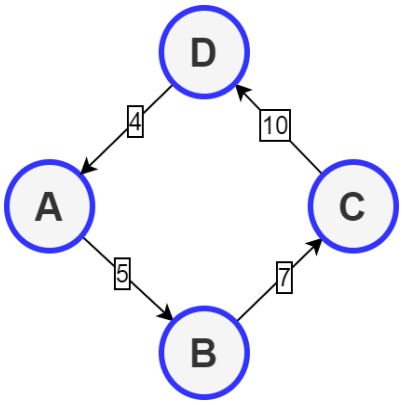
$$V = \{0, 1, 2, 3, 4\}$$

$$E = \{(0,1), (0,2), (0,3), (1,3), (1,4), (2,3), (3,4)\}$$

### 3.1.3 Graf Berarah dan Berbobot (*Weighted Directed Graph*)

Sebuah graf jika setiap busur mempunyai nilai yang menyatakan hubungan antara dua buah simpul, maka busur tersebut dinyatakan memiliki bobot. Bobot pada graf terkadang juga disebut *cost* atau biaya.

Maka graf berarah dan berbobot ialah graf yang sisinya (E) memiliki orientasi arah dan sisi tersebut memiliki bobot.




Gambar 3. 4 Graf berarah dan berbobot

Pada Gambar 3. 4 apabila dibentuk himpunan, sebagai berikut:

$V = \{A, B, C, D\}$

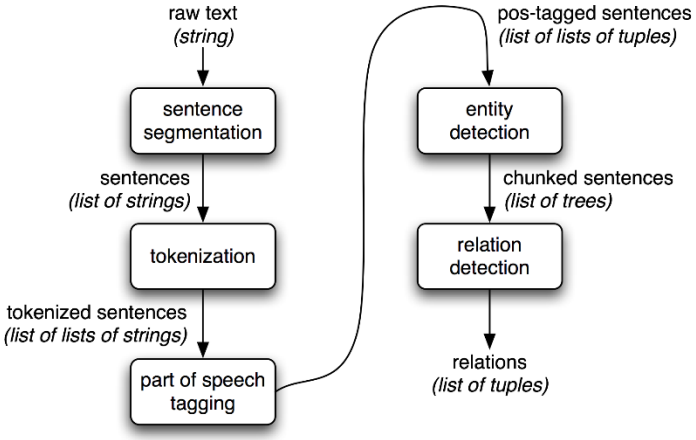
$E = \{(A, B, 5), (B, C, 7), (C, D, 10), (D, A, 4)\}$

Sehingga dapat dibaca simpul A ke B memiliki bobot 5, B ke C memiliki bobot 7, C ke D memiliki bobot 10 dan D ke A memiliki bobot 4.

## 3.2 *Natural Language Processing* (NLP)

*Natural Language Processing* (NLP) merupakan cara komputer untuk memproses bahasa manusia dengan cara yang bermakna. NLP pada umumnya digunakan menurunkan pola semantik dari sebuah teks atau ucapan dan dilakukan pengubahan format yang lebih terstruktur agar dapat diproses oleh komputer.(Al-Moslmi, Gallofre Ocana, L. Opdahl, & Veres, 2020).

Salah satu fungsi NLP dapat digunakan untuk information extraction (IE). Tujuan utama IE adalah mengekstrak informasi terstruktur dari teks tidak terstruktur atau semi terstruktur (Azeroual, 2019). Langkah proses IE dijelaskan pada Gambar 3. 5.




Gambar 3. 5 Proses information extraction

Pada langkah pertama, segmentasi kalimat, teks yang belum diproses dipecah menjadi kalimat menggunakan karakter akhir kalimat seperti ".", "!", "?". *Tokenization* merupakan proses pemisahan kalimat hingga menjadi bagian-bagian kecil baik berupa kata, angka, simbol dan tanda baca yang disebut *tokens* (S & R, 2016). Proses *tokenization* dilakukan secara bertahap mulai dari pemisahan (*split*) berdasar spasi (*whitespace*). Part of speech tagging adalah proses memberi label pada setiap kata dalam kalimat dengan POS atau tag yang sesuai dengan kelas kata seperti kata kerja, kata keterangan, kata sifat, dan lainnya (Nadkarni et al., 2011). Entity Detection atau Named Entity Recognition (NER) adalah tugas mengidentifikasi dan mengkategorikan informasi kunci (entitas) dalam teks (Yadav & Bethard, 2019). Entitas dapat berupa kata atau rangkaian kata apa pun yang secara konsisten mengacu pada hal yang sama. Setiap entitas yang terdeteksi diklasifikasikan ke dalam kategori yang telah ditentukan. Pada langkah terakhir, relation detection ialah hubungan antara entitas yang ditemukan dalam satu teks.

### 3.3 *Knowledge Graph*

Sebuah knowledge graph adalah grafik berlabel terarah di mana label memiliki arti yang terdefinisi dengan baik. Graf berlabel berarah terdiri dari node, edge, dan label. Sebuah *knowledge graph* (KG) dapat merepresentasikan data semantik yang disusun dengan 3 komponen seperti *subject* (s), *predicate*

(p), *object* (o).(Al-Moslmi *et al*., 2020). *Knowledge Graph* juga bisa disebut sebuah perpaduan *knowledge base* dengan teori graf.


Gambar 3. 6 Knowledge pada graf

Sedangkan menurut (Jesús Barrasa et al., 2021), Knowledge Graphs adalah kumpulan fakta yang saling terkait yang menggambarkan entitas, peristiwa, atau benda dunia nyata dan keterkaitannya dalam format yang dapat dimengerti manusia dan mesin.

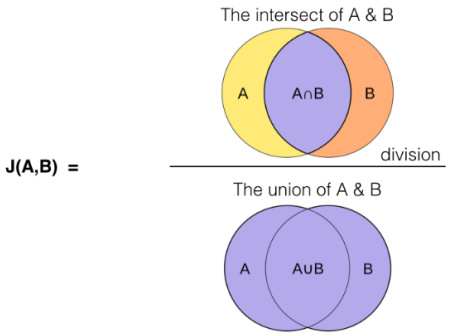
Sebuah knowledge graph terdiri dari tiga komponen utama: node, edge, dan label. Objek, tempat, atau orang apa pun bisa menjadi simpul. Sebuah edge mendefinisikan hubungan antara node. Seperti yang ditunjukkan pada Gambar 3. 6, node A mewakili subjek, edge B mewakili predikat dan node C mewakili objek.

Data semantik disimpan dengan model graf pada *graph* database seperti Neo4J, Janus*Graph*, TigerGraph. Database ini dirancang untuk menyimpan data graf dan melakukan operasi query data (Webber & Bruggen, 2020).

## 3.4 Algoritme graf Jaccard Similarity

Algoritme graf adalah bagian dari graf analitik yang digunakan untuk kebutuhan menganalisis data yang saling terhubung atau dalam model graf dengan menggunakan kalkulasi matematika yang dikhususkan untuk pemecahan masalah pada teori graf (Needham & Hodler, 2019). Beberapa analitik pada graf diantaranya pathfinding untuk menentukan lintasan terpendek, centrality untuk memahami node mana yang lebih penting dalam jaringan, community detection berguna sebagai algoritme media sosial untuk menemukan orang-orang dengan minat yang sama dan menjaga mereka tetap terhubung dengan erat (Hodler & Needham, 2021).

Algoritme *graph similarity* digunakan membandingkan antar graf sehingga mendapatkan tingkat kesamaan atau kemiripan dari dua graf, salah satunya *algoritme Jaccard Similarity Coeficient* atau *Jaccard Index*. *Jaccard Index* (*Jaccard* 1901) mengukur kemiripan dua himpunan dengan menghitung besar irisan (*intersection*) dibagi dengan besar gabungan (*union*) dari himpunan (Fletcher & Islam, 2018). Jaccard Similarity mudah diterapkan di berbagai domain karena kesederhanan dalam proses penghitungan (Bag et al., 2019). Pada Gambar 3. 7 menjelaskan bahwa menghitung *similarity* dengan cara irisan (*intersection*) himpunan A dan B dibagi dengan gabungan (*union*) himpunan A dan B. Dengan demikian dapat dituliskan dalam notasi matematika sebagai berikut (Verma & Aggarwal, 2020):




Gambar 3. 7 Representasi *graph similarity* dalam himpunan

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

dimana:

A dan B adalah dua himpunan

∩ adalah irisan (*intersection*) dari dua himpunan A dan B

∪ adalah gabungan (*union*) dari himpunan A dan B