RANGKUMAN NATURAL LANGUANGE DAN MACHINE LANGUANGE

Disusun oleh :

Jelyan Ananda Herdiyatma

(24241175)

PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI

1. **Natural language:**

Natural Language Processing (NLP) merupakan salah satu cabang ilmu AI yang berfokus pada pengolahan bahasa natural. Bahasa natural adalah bahasa yang secara umum digunakan oleh manusia dalam berkomunikasi satu sama lain. Bahasa yang diterima oleh komputer butuh untuk diproses dan dipahami terlebih dahulu supaya maksud dari user bisa dipahami dengan baik oleh komputer.

Ada berbagai terapan aplikasi dari NLP. Diantaranya adalah Chatbot (aplikasi yang membuat user bisa seolah-olah melakukan komunikasi dengan computer), Stemming atau Lemmatization (pemotongan kata dalam bahasa tertentu menjadi bentuk dasar pengenalan fungsi setiap kata dalam kalimat), Summarization (ringkasan dari bacaan), Translation Tools (menterjemahkan bahasa) dan aplikasi-aplikasi lain yang memungkinkan komputer mampu memahami instruksi bahasa yang diinputkan oleh user.

I. NLP Area

Pustejovsky dan Stubbs (2012) menjelaskan bahwa ada beberapa area utama penelitian pada field NLP, diantaranya:

1. Question Answering Systems (QAS). Kemampuan komputer untuk menjawab pertanyaan yang diberikan oleh user. Daripada memasukkan keyword ke dalam browser pencarian, dengan QAS, user bisa langsung bertanya dalam bahasa natural yang digunakannya, baik itu Inggris, Mandarin, ataupun Indonesia.
2. Summarization. Pembuatan ringkasan dari sekumpulan konten dokumen atau email. Dengan menggunakan aplikasi ini, user bisa dibantu untuk mengkonversikan dokumen teks yang besar ke dalam bentuk slide presentasi.
3. Machine Translation. Produk yang dihasilkan adalah aplikasi yang dapat memahami bahasa manusia dan menterjemahkannya ke dalam bahasa lain. Termasuk di dalamnya adalah Google Translate yang apabila dicermati semakin membaik dalam penterjemahan bahasa. Contoh lain lagi adalah BabelFish yang menterjemahkan bahasa pada real time.
4. Speech Recognition. Field ini merupakan cabang ilmu NLP yang cukup sulit. Proses pembangunan model untuk digunakan telpon/komputer dalam

mengenali bahasa yang diucapkan sudah banyak dikerjakan. Bahasa yang sering digunakan adalah berupa pertanyaan dan perintah.

5. Document classification. Sedangkan aplikasi ini adalah merupakan area penelitian NLP Yang paling sukses. Pekerjaan yang dilakukan aplikasi ini adalah menentukan dimana tempat terbaik dokumen yang baru diinputkan ke dalam sistem. Hal ini sangat berguna pada aplikasi spam filtering, news article classification, dan movie review.

II. Terminologi NLP

Perkembangan NLP menghasilkan kemungkinan dari interface bahasa natural menjadi knowledge base dan penterjemahan bahasa natural. Poole dan Mackworth (2010) menjelaskan bahwa ada 3 (tiga) aspek utama pada teori pemahaman mengenai natural language:

1. Syntax: menjelaskan bentuk dari bahasa. Syntax biasa dispesifikasikan oleh sebuah grammar. Natural language jauh lebih daripada formal language yang digunakan untuk logika kecerdasan buatan dan program komputer
2. Semantics: menjelaskan arti dari kalimat dalam satu bahasa. Meskipun teori semantics secara umum sudah ada, ketika membangun sistem natural language understanding untuk aplikasi tertentu, akan digunakan representasi yang paling sederhana.
3. Pragmatics: menjelaskan bagaimana pernyataan yang ada berhubungan dengan dunia. Untuk memahami bahasa, agen harus mempertimbangan lebih dari hanya sekedar kalimat. Agen harus melihat lebih ke dalam konteks kalimat, keadaan dunia, tujuan dari speaker dan listener, konvensi khusus, dan sejenisnya.

Contoh kalimat di bawah ini akan membantu untuk memahami perbedaan diantara ketiga aspek tersebut di atas. Kalimat-kalimat ini adalah kalimat yang mungkin muncul pada bagian awal dari sebuah buku Artificial Intelligence (AI):

1. This book is about Artificial Intelligence
2. The green frogs sleep soundly
3. Colorless green ideas sleep furiously
4. Furiously sleep ideas green colorless

Kalimat pertama akan tepat jika diletakkan pada awal sebuah buku, karena tepat secara sintaks, semantik, dan pragmatik. Kalimat kedua tepat secara sintaks dan semantic, namun kalimat tersebut akan menjadi aneh apabila diletakkan pada awal sebuah buku AI, sehingga kalimat ini tidak tepat secara pragmatik. Kalimat ketiga

tepat secara sintaks, tetapi tidak secara semantik. Sedangkan pada kalimat keempat, tidak tepat secara sintaks, semantik, dan pragmatik.

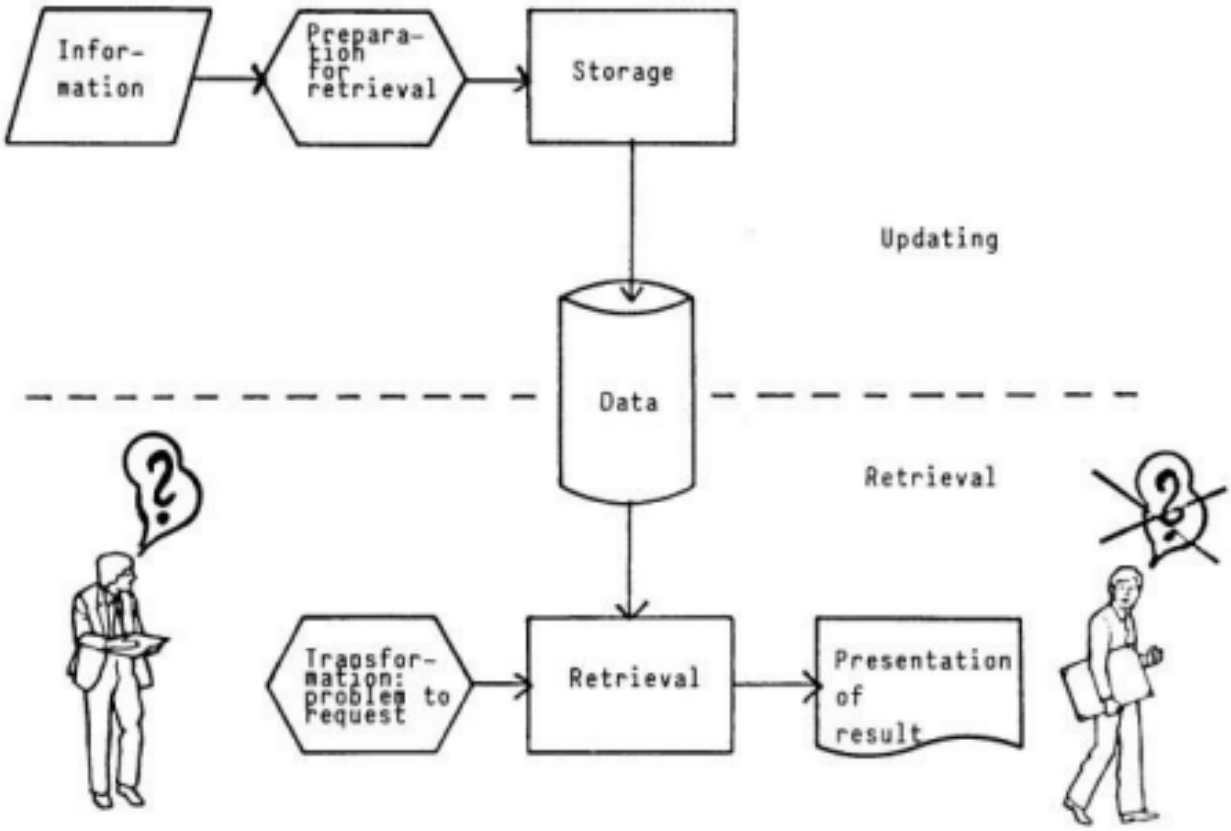
Selain daripada ketiga istilah tersebut ada beberapa istilah yang terkait dengan NLP, yaitu:

- Morfologi. Adalah pengetahuan tentang kata dan bentuknya sehingga bisa dibedakan antara yang satu dengan yang lainnya. Bisa juga didefinisikan asal usul sebuah kata itu bisa terjadi. Contoh : membangunkan –> bangun (kata dasar), mem- (prefix), -kan (suffix)
- Fonetik. Adalah segala hal yang berhubungan dengan suara yang menghasilkan kata yang dapat dikenali. Fonetik digunakan dalam pengembangan NLP khususnya bidang speech based system

III. Information Retrieval

Information Retrieval (IR) adalah pekerjaan untuk menemukan dokumen yang relevan dengan kebutuhan informasi yang dibutuhkan oleh user. Contoh sistem IR yang paling popular adalah search engine pada World Wide Web. Seorang pengguna Web bisa menginputkan query berupa kata apapun ke dalam sebuah search engine dan melihat hasil dari pencarian yang relevan. Karakteristik dari sebuah sistem IR (Russel & Norvig, 2010) diantaranya adalah:

- A corpus of documents. Setiap sistem harus memutuskan dokumen yang ada akan diperlakukan sebagai apa. Bisa sebagai sebuah paragraf, halaman, atau teks multipage.
- Queries posed in a query language. Sebuah query menjelaskan tentang apa yang user ingin peroleh. Query language dapat berupa list dari kata-kata, atau bisa juga menspesifikasikan sebuah frase dari kata-kata yang harus berdekatan
- A result set. Ini adalah bagian dari dokumen yang dinilai oleh sistem IR sebagai yang relevan dengan query.
- A presentation of the result set. Maksud dari bagian ini adalah tampilan list judul dokumen yang sudah di ranking.



Gambar 2. Proses dari Information Retrieval

IV. Morphological Analysis

Proses dimana setiap kata yang berdiri sendiri (individual words) dianalisis kembali ke komponen pembentuk mereka dan token nonword seperti tanda baca dsb dipisahkan dari kata tersebut.

Contohnya apabila terdapat kalimat:

"I want to print Bill's .init file"

Jika morphological analysis diterapkan ke dalam kalimat di atas, maka:

- Pisahkan kata "Bill's" ke bentuk proper noun "Bill" dan possessive suffix ""s"
- Kenali sequence ".init" sebagai sebuah extension file yang berfungsi sebagai adjective dalam kalimat.

Syntactic analysis harus menggunakan hasil dari morphological analysis untuk membangun sebuah deskripsi yang terstruktur dari kalimat. Hasil akhir dari proses

ini adalah yang sering disebut sebagai parsing. Parsing adalah mengkonversikan daftar kata yang berbentuk kalimat ke dalam bentuk struktur yang mendefinisikan unit yang diwakili oleh daftar tadi.

Hampir semua sistem yang digunakan untuk syntactic processing memiliki dua komponen utama, yaitu:

- Representasi yang deklaratif, yang disebut juga sebagai Grammar, dari fakta sintaktis mengenai bahasa yang digunakan
- Procedure, yang disebut juga sebagai Parser, yang membandingkan grammar dengan kalimat yang diinputkan untuk menghasilkan struktur kalimat yang telah di parsing

Cara yang paling umum digunakan untuk merepresentasikan grammar adalah dengan sekumpulan production rule. Rule yang paling pertama bisa diterjemahkan sebagai "Sebuah Sentence terdiri dari sebuah Noun Phrase, diikuti oleh Verb Phrase", garis vertical adalah OR, sedangkan ε mewakili string kosong.

Proses parsing menggunakan aturan-aturan yang ada pada Grammar, kemudian membandingkannya dengan kalimat yang diinputkan. Struktur paling sederhana dalam melakukan parsing adalah Parse Tree, yang secara sederhana menyimpan rule dan bagaimana mereka dicocokkan satu sama lain. Setiap node pada Parse Tree berhubungan dengan kata yang dimasukkan atau pada nonterminal pada Grammar yang ada. Setiap level pada Parse Tree berkorespondensi dengan penerapan dari satu rule pada Grammar.

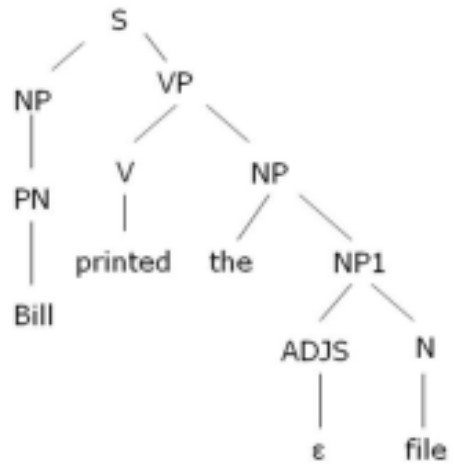
Contoh:

Terdapat Grammar sebagai berikut:

- S → NP VP
- NP → the NP1
- NP → PRO
- NP → PN
- NP → NP1
- NP1 → ADJS N
- ADJS → ε | ADJ ADJS
- VP → V

- P → V NP
- N → file | printer
- PN → Bill
- PRO → I
- ADJ → short | long | fast
- V → printed | created | want

Maka, apabila terdapat kalimat "Bill printed the file", representasi Parse Tree nya akan menjadi:



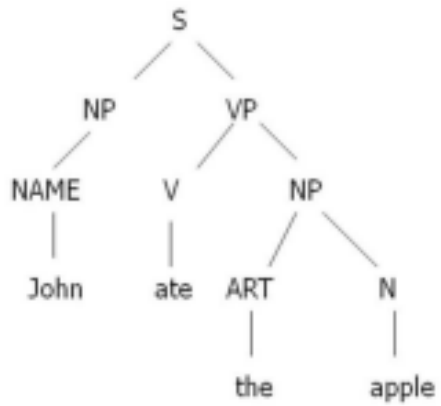
Pembangunan Parse Tree ini didasarkan pada Grammar yang digunakan. Apabila Grammar yang digunakan berbeda, maka Parse Tree yang dibangun harus tetap berdasarkan pada Grammar yang berlaku.

Contoh:

Terdapat Grammar sebagai berikut:

- S → NP VP
- VP → V NP
- NP → NAME
- NP → ART N
- NAME → John
- V → ate
- ART→ the
- N → apple

Maka Parse Tree untuk kalimat "John ate the apple" akan menjadi:



V. Stemming & Lemmatization

Stemming merupakan sebuah proses yang bertujuan untuk mereduksi jumlah variasi dalam representasi dari sebuah kata (Kowalski, 2011). Resiko dari proses stemming adalah hilangnya informasi dari kata yang di-stem. Hal ini menghasilkan menurunnya akurasi atau presisi. Sedangkan untuk keuntungannya adalah, proses stemming bisa meningkatkan kemampuan untuk melakukan recall. Tujuan dari stemming sebenarnya adalah untuk meningkatkan performace dan mengurangi penggunakan resource dari sistem dengan mengurangi jumlah unique word yang harus diakomodasikan oleh sistem. Jadi, secara umum, algoritma stemming mengerjakan transformasi dari sebuah kata menjadi sebuah standar representasi morfologi (yang dikenal sebagai stem).

Contoh:

"comput" adalah stem dari "computable, computability, computation, computational, computed, computing, compute, computerize"

Ingason dkk. (2008) mengemukakan bahwa lemmatization adalah sebuah proses untuk menemukan bentuk dasar dari sebuah kata. Nirenburg (2009) mendukung teori ini dengan kalimatnya yang menjelaskan bahwa lemmatization adalah proses yang bertujuan untuk melakukan normalisasi pada teks/kata dengan berdasarkan pada bentuk dasar yang merupakan bentuk lemma-nya. Normalisasi disini adalah dalam artian mengidentifikasikan dan menghapus prefiks serta suffiks dari sebuah kata. Lemma adalah bentuk dasar dari sebuah kata yang memiliki arti tertentu berdasar pada kamus.

Contoh:

- Input: "The boy's cars are different colors"
- Transformation: am, is, are à be
- Transformation: car, cars, car's, cars' à car
- Hasil: "The boy car be differ color"

Algoritma Stemming dan Lemmatization berbeda untuk bahasa yang satu dengan bahasa yang lain.

VI. Contoh Aplikasi NLP

Penelitian yang dikerjakan oleh Suhartono, Christiandy, dan Rolando (2013) adalah merancang sebuah algoritma lemmatization untuk Bahasa Indonesia. Algoritma ini dibuat untuk menambahkan fungsionalitas pada algoritma Stemming yang sudah

pernah dikerjakan sebelumnya yaitu Enhanced Confix-Stripping Stemmer (ECS) yang dikerjakan pada tahun 2009. ECS sendiri merupakan pengembangan dari algoritma Confix-Stripping Stemmer yang dibuat pada tahun 2007. Pengembangan yang dikerjakan terdiri dari beberapa rule tambahan dan modifikasi dari rule sebelumnya. Langkah untuk melakukan suffix backtracking juga ditambahkan. Hal ini untuk menambah akurasi.

Secara mendasar, algoritma lemmatization ini tidak bertujuan untuk mengembangkan dari metode ECS, larena tujuannya berbeda. Algoritma lemmatization bertujuan untuk memodifikasi ECS, supaya lebih tepat dengan konsep lemmatization. Namun demikian, masih ada beberapa kemiripan pada proses yang ada pada ECS. Ada beberapa kasus yang mana ECS belum berhasil untuk digunakan, namun bisa diselesaikan pada algoritma lemmatization ini.

Pengujian validitas pada algoritma ini adalah dengan menggunakan beberapa artikel yang ada di Kompas, dan diperoleh hasil sebagai berikut:

	FULL					UNIQUE				
Category	**T**	**V**	**S**	**E**	**P**	**T**	**V**	**S**	**E**	**P**
Business	6344	5627	5550	77	0.98632	1868	1580	1559	21	0.98671
Regional	6470	4802	5846	81	0.98313	1213	1011	995	16	0.98417
Education	4165	5927	3598	32	0.99460	868	637	623	14	0.97802
Science	6246	5504	5398	73	0.98674	874	643	630	13	0.97978
Sports	6231	3242	5522	42	0.98705	838	608	604	4	0.99342
International	10953	3630	9917	75	0.97934	2037	1593	1575	18	0.98870
Megapolitan	3998	5471	3214	28	0.99488	610	302	297	5	0.98344
National	5499	5564	4764	38	0.99317	559	326	324	2	0.99387
Oasis	6087	9992	5462	42	0.99580	820	528	524	4	0.99242
Travel	8379	7502	7457	45	0.99400	892	611	607	4	0.99345
All	**64372**	**57261**	**56728**	**533**	**0.99069**	**10579**	**7839**	**7738**	**101**	**0.98712**

Hasil dari pengujian menunjukkan bahwa akurasi yang diperoleh sekitar 98.71%.

T = Total data count

V = Valid test data count

S = Successful lemmatization

E = Error / Kegagalan

P = Precision

Aplikasi NLP yang lainnya adalah seperti penerjemah bahasa, chatting dengan komputer, meringkas satu bacaan yang panjang, pengecekan grammar dan lain sebagainya.

2. **Machine language:**

3. *Bahasa mesin, atau kode mesin, adalah bahasa tingkat rendah yang terdiri dari digit biner (yang dan nol).*
4. *Bahasa tingkat tinggi, seperti Swift dan C ++ harus dikompilasi ke dalam bahasa mesin sebelum kode dijalankan pada komputer.*
5. *Karena komputer adalah perangkat digital, mereka hanya mengenali data biner.*
6. *Setiap program, video, gambar, dan karakter teks diwakili dalam biner.*
7. *Data biner ini, atau kode mesin, diproses sebagai input oleh CPU.*
8. *Output yang dihasilkan dikirim ke sistem operasi atau aplikasi, yang menampilkan data secara visual.*
9. *Misalnya, nilai ASCII untuk huruf "A" adalah 01000001 dalam kode mesin, tetapi data ini ditampilkan sebagai "A" di layar.*
10. *Suatu gambar mungkin memiliki ribuan atau bahkan jutaan nilai biner yang menentukan warna setiap piksel.*

11. *Sementara kode mesin terdiri dari 1s dan 0s, arsitektur prosesor yang berbeda menggunakan kode mesin yang berbeda.*
12. *Misalnya, prosesor PowerPC, yang memiliki arsitektur RISC, membutuhkan kode yang berbeda dari prosesor Intel X86, yang memiliki arsitektur CISC.*
13. *Kompiler harus mengkompilasi kode sumber tingkat tinggi untuk arsitektur prosesor yang benar agar program dapat dijalankan dengan benar.*
14. *Bahasa mesin vs bahasa perakitan bahasa mesin dan bahasa perakitan adalah bahasa tingkat rendah, tetapi kode mesin di bawah perakitan dalam hierarki bahasa komputer.*
15. *Bahasa perakitan mencakup perintah yang dapat dibaca manusia, seperti MOV, ADD, dan SUB, sementara bahasa mesin tidak mengandung kata atau bahkan huruf.*
16. *Beberapa pengembang secara manual menulis bahasa perakitan untuk mengoptimalkan suatu program, tetapi mereka tidak menulis kode mesin.*
17. *Hanya pengembang yang menulis kompiler perangkat lunak yang perlu khawatir tentang bahasa mesin.*
18. *Catatan: Sementara kode mesin secara teknis terdiri dari data biner, itu juga dapat diwakili dalam nilai heksadesimal.*
19. *Misalnya, huruf "Z," yang 01011010 dalam biner, dapat ditampilkan sebagai 5a dalam kode heksadesimal.*
20. *Salin*

21. Definisi dan Faktor Penggunaan Istilah Kata *Machine Language*

22. Gambar Definisi Machine Language Berdasarkan Sumber Rujukan Relevan Terpercaya Serta Menurut Para Pakar Dan Ahli Serta Faktor Atau Tingkat Penggunaan Dari Istilah Teknologinya
23. Agar kita dapat lebih mendalami arti penjelasan serta arti dari *acronym* atau kata terkait lingkup "Software Terms" di atas, pastinya kita perlu mengenali lebih lanjut terkait apa itu definisi dari ***Bahasa mesin*** ini.
24. Sebelumnya, akan Kami jelaskan terlebih dahulu bahwa untuk menguraikan artinya sendiri, kita perlu mendasarinya berdasarkan penjelasan sumber terkait, relevan, dan terpercaya, baik itu yang berasal situs *Technopedia Dictionary*, Wikipedia, atau kamus sejenis bidang teknologi, maupun secara langsung berasal dari pengertian menurut para ahli dan pakar di bidang tersebut.
25. Faktor kata terkait penggunaannya Kami beri dengan nilai "8", di mana itu merupakan terminologi lanjutan dan biasanya cukup sulit untuk dipahami oleh pengguna biasa. Kalian dapat membaca lebih lanjut tentang Faktor Penggunaan Kata dalam Kamus Teknologi RM Digital.
26. Seperti yang dapat kita semua mengerti, arti dari definisi itu sendiri adalah sebuah batasan (*limit* atau *limitation*) yang fungsi dan maknanya adalah sebagai pembatas serta juga sebagai penerangan tentang arti dari suatu kata.
27. Definisi yang Kami bahas dalam pengertian di atas diartikan dengan sebuah kata istilah, terminologi, akronim, atau jargon yang memberikan penggambaran, serta memberitahukan tentang sebuah pemaknaan, arti, ataupun karakteristik utama dari kata, baik itu terkait fungsi, proses, kegiatan, ataupun seseorang.
28. Seperti yang dapat kalian baca, dalam pengertian dan definisinya di atas, secara bahasa (arti literal, harfiah, atau aslinya), khususnya secara terjemahannya, kata **"Machine Language"** ini didefinisikan sebagai **"Bahasa mesin"** dalam bahasa Indonesia dan merupakan kata yang terkait dengan Istilah Software.
29. Lebih lanjutnya, kata ini juga merupakan salah satu dari kumpulan kamus, istilah, akronim (jargon) dalam bidang teknologi yang memiliki awalan huruf M.

NOTASI KALIMAT DESKRIPTIF

PADA ALGORITMA

Notasi deskriptif pada algoritma adalah cara untuk menjelaskan langkah-langkah dalam algoritma menggunakan bahasa natural atau pseudo-code yang mudah dimengerti, tanpa terikat pada sintaks pemrograman tertentu.

CONTOH :

Berikut adalah contoh notasi kalimat deskriptif dalam algoritma, yang menggambarkan logika secara jelas dalam bentuk narasi (deskriptif), tanpa sintaksis formal dari bahasa pemrograman:

Contoh Algoritma Menghitung Nilai Rata-Rata dari 5 Angka:

1. Mulai.
2. Inisialisasi sebuah variabel jumlah dengan nilai 0.
3. Tentukan 5 angka yang akan dijumlahkan.
4. Untuk setiap angka:

• Tambahkan angka tersebut ke dalam variabel jumlah.

5. Setelah semua angka dijumlahkan, bagi nilai jumlah dengan 5 untuk mendapatkan rata-rata.
6. Cetak hasil rata-rata tersebut.
7. Selesai.

Deskripsi Kalimat dalam Algoritma:

a. "Mulai": Menunjukkan permulaan algoritma.
b. "Inisialisasi sebuah variabel": Artinya menetapkan nilai awal ke variabel yang akan digunakan.
c. "Tentukan 5 angka": Menentukan data input yang akan diproses.
d. "Untuk setiap angka": Melakukan iterasi atau pengulangan untuk setiap elemen angka.
e. "Cetak hasil rata-rata tersebut": Menampilkan hasil yang diperoleh dari perhitungan.
f. Contoh ini menunjukkan bagaimana langkah-langkah algoritma dituliskan secara naratif dan bisa diterjemahkan menjadi kode pemrograman apa pun sesuai logika yang diuraikan.