

**RANCANG BANGUN APLIKASI PENDETEKSI  
KESAMAAN DOKUMEN MENGGUNAKAN METODE  
*DICE SIMILARITY***

**SKRIPSI**



**ARIS AKHYAR ABDILLAH**

**H071171505**

**PROGRAM STUDI SISTEM INFORMASI**

**DEPARTEMEN MATEMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**MEI 2023**

**RANCANG BANGUN APLIKASI PENDETEKSI  
KESAMAAN DOKUMEN MENGGUNAKAN METODE  
*DICE SIMILARITY***

**SKRIPSI**

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program  
Studi Sistem Informasi Departemen Matematika Fakultas Matematika dan Ilmu  
Pengetahuan Alam Universitas Hasanuddin

**ARIS AKHYAR ABDILLAH**

**H071171505**

**PROGRAM STUDI SISTEM INFORMASI  
DEPARTEMEN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN  
MAKASSAR  
MEI 2023**

**LEMBAR PERNYATAAN KEOTENTIKAN**

Saya yang bertanda tangan di bawah ini menyatakan dengan sungguh – sungguh  
bahwa skripsi yang saya buat dengan judul :

**RANCANG BANGUN APLIKASI PENDETEKSI KESAMAAN  
DOKUMEN MENGGUNAKAN METODE *DICE SIMILARITY*.**

Adalah benar hasil karya saya sendiri, bukan hasil plagiat dan belum pernah  
dipublikasikan dalam bentuk apa pun

Makassar, 12 Mei 2023



**Aris Akhyar Abdillah**

**NIM. H071171505**

# RANCANG BANGUN APLIKASI PENDETEKSI KESAMAAN DOKUMEN MENGGUNAKAN METODE *DICE SIMILARITY*

Disetujui Oleh :

Pembimbing Utama,

Pembimbing Pertama



Dr. Hendra S.Si., M. Kom.  
NIP. 19760102 200312 1 001



Edy Saputra R, S. Si., M. Si.  
NIP. 19910410 202005 3 001

Ketua Program Studi



Dr. Hendra S.Si., M. Kom.  
NIP. 19760102 200312 1 001



**HALAMAN PENGESAHAN**

Skripsi ini diajukan oleh :

Nama : Aris Akhyar Abdillah  
NIM : H071171505  
Program Studi : Sistem Informasi  
Judul Skripsi : Rancang Bangun Aplikasi Pendeteksi Kesamaan Dokumen  
Menggunakan Metode *Dice Similarity*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin

**DEWAN PENGUJI**

1. Ketua : Dr. Hendra, S.Si, M.Kom. 
2. Sekretaris : Edy Saputra Rusdi, S.Si, M.Si 
3. Anggota : Dr. Muhammad Hasbi, M.Sc 
4. Anggota : Ir. Eliyah Acantha Manapa Sampetoding,   
S.Kom, M.Kom.

Ditetapkan di : Makassar

Tanggal : 12 Mei 2023



## KATA PENGANTAR

*Assalamu'alaikum Warahmatullahi Wabarakatuh*

Segala puji bagi Allah Subhanallahu Wa Ta'ala atas segala nikmat serta limpahan rahmat dan hidayah-Nya yang telah diberikan kepada penulis sampai dapat menuntaskan penulisan skripsi ini. Shalawat dan salam serta doa senantiasa terlimpahkan kepada baginda Nabi besar, Nabi Muhammad Shallallahu 'Alaihi Wa sallam. Dengan izin dan kemurahan hatinya yang diberikan oleh Allah Subhanallahu Wa Ta'ala, penulis dapat menyelesaikan skripsi yang berjudul **“Rancang Bangun Aplikasi Pendeteksi Kesamaan Dokumen Menggunakan Metode Dice Similarity”** sebagai salah satu syarat memperoleh gelar sarjana pada Program Studi Sistem Informasi Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Dalam menyelesaikan skripsi ini, tentunya penulis mengalami banyak tantangan dan permasalahan selama proses penulisan skripsi ini. Hanya dengan karunia dan kehendak-Nya serta bimbingan dan pertolongan dari berbagai pihak baik secara langsung maupun tidak langsung, maka skripsi ini dapat diselesaikan. Untuk itu, penulis mengucapkan terima kasih yang sebesar-besarnya dan penghormatan yang setinggi-tingginya kepada berbagai pihak terkhusus kepada kedua orang tua penulis yaitu Ayahanda **Arham Aras** dan Ibunda **Ernawati Haruna** dan juga saudara satu-satunya yaitu **Nur Izzah Khairani**. Penghargaan yang tulus dan ucapan terima kasih dengan penuh keikhlasan juga penulis ucapkan kepada:

1. **Bapak Prof. Dr. Ir. Jamaluddin Jompa, M.Sc.**, selaku Rektor Universitas Hasanuddin beserta seluruh jajarannya.
2. **Bapak Dr. Eng. Amiruddin, S.Si., M.Si.**, yang menjabat sebagai Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin beserta seluruh jajarannya.
3. **Bapak Prof. Dr. Nurdin, S.Si., M.Si.**, yang menjabat sebagai Ketua Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin beserta seluruh jajarannya.

4. **Bapak Dr. Hendra S.Si., M. Kom.**, yang menjabat sebagai Ketua Program Studi Sistem Informasi dan juga sebagai pembimbing utama saya yang telah memberikan saya arahan, pengetahuan, motivasi dan bimbingan dengan sabar dan penuh perhatian di tengah kesibukannya.
5. **Bapak Edy Saputra R, S. Si., M. Si.**, yang menjabat sebagai Penasihat Akademik dan juga sebagai pembimbing pertama saya yang telah memberikan saya arahan, pengetahuan, motivasi dan bimbingan dengan sabar dan penuh perhatian ditengah kesibukannya.
6. **Bapak Dr. Muhammad Hasbi, M.Sc.**, dan **Bapak Ir. Eliyah Acantha Manapa Sampetoding, S. Kom, M. Kom.**, yang menjabat sebagai tim penguji yang telah memberikan saya saran dan kritikan yang konstruktif dalam penyempurnaan penyusunan tugas akhir ini.
7. **Bapak/Ibu Dosen** Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin yang telah memberikan saya ilmu dan pengetahuan selama saya belajar di sini.
8. Seluruh warga **KM FMIPA Unhas** dan **Himatika FMIPA Unhas** yang telah menjadi keluarga kedua saya yang memberikan saya kenangan, serta pelajaran hidup yang sangat berharga. Khususnya, seluruh saudara-saudara **DISKRIT 2017** dan **MIPA 2017**, terima kasih atas kebersamaan, loyalitas, pengalaman, dan semangat yang sangat berarti bagi saya.
9. Teman-teman **Sistem Informasi 2017** yang menjadi teman pertama dikampus yang telah memberi dukungan dan kerja samanya dalam menemani saya selama perkuliahan, yang banyak membantu saya dalam hal apa pun serta berbagi suka-duka dalam menuntut ilmu.
10. Kepada seluruh pihak yang tidak dapat saya sebutkan satu persatu, terima kasih sebesar-besarnya untuk segala dukungan dan partisipasi yang diberikan kepada saya semoga bernilai ibadah di sisi Allah Subhanahu Wa Ta'ala.

Penulis mengakui bahwa skripsi ini masih memiliki banyak kekurangan, untuk itu dengan segala kerendahan hati penulis meminta maaf atas kekurangan tersebut. Penutup, semoga tulisan ini dapat memberikan banyak manfaat untuk pembaca. *Wassalamu'alaikum Warahmatullahi Wabarakatuh.*

**PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR  
UNTUK KEPENTINGAN AKADEMIK**

---

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini :

Nama : Aris Akhyar Abdillah  
NIM : H071171505  
Program Studi : Sistem Informasi  
Departemen : Matematika  
Fakultas : Matematika dan Ilmu Pengetahuan Alam  
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Noneklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah yang berjudul:

**“Rancang Bangun Aplikasi Pendeteksi Kesamaan Dokumen  
Menggunakan Metode *Dice Similarity*”**

Beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar Pada tanggal 12 Mei 2023

Yang Menyatakan  
(Aris Akhyar Abdillah)

## ABSTRAK

Dengan adanya teknologi di saat ini, dapat dikatakan pekerjaan manusia semakin mudah. Seperti untuk akses informasi, sekarang sudah banyak sekali informasi yang tersebar di internet dan hampir semua yang dicari dapat ditemukan. Dengan semua kemudahan yang diberikan teknologi seperti contoh di atas, timbul suatu masalah yaitu mudahnya dilakukan plagiarisme. Plagiarisme berasal dari bahasa Latin yaitu “plagiare” yang berarti mencuri. Plagiarisme berasal dari kata plagiat yang berarti pengambilan karangan (pendapat dan sebagainya) orang lain dan menjadikannya seolah-olah karangan (pendapat dan sebagainya) sendiri, misalnya menerbitkan karya tulis orang lain atas nama dirinya sendiri. Untuk mengurangi masalah tersebut dibuatlah Aplikasi Pendeteksi Kesamaan Dokumen Menggunakan Metode *Dice Similarity*. Aplikasi ini dibangun dengan menggunakan metode *Prototype* yaitu salah satu metode untuk merancang aplikasi di mana aplikasi di analisa kebutuhannya baik secara kebutuhan fungsional dan non fungsional, kemudian merancang aplikasi tersebut kemudian diuji. Aplikasi ini juga diuji menggunakan metode *blackbox* di mana aplikasi diuji berdasarkan sudut pandang pengguna. Selain itu, aplikasi ini menggunakan arsitektur *Model View Controller* (MVC) yang mana mempartisi kode aplikasi ini menjadi beberapa bagian. Aplikasi ini telah berhasil diuji dengan hasil yang sesuai dengan pengujian.

**Kata Kunci** : aplikasi, plagiarisme, *prototype*, MVC, *blackbox*,

## ABSTRACT

With the existence of technology in the present, it can be said that human work is easier. For example, for access to information, there are now a lot of information scattered on the internet and almost everything that is searched can be found. With all the conveniences provided by technology such as the example above, a problem arises, namely the ease of plagiarism. Plagiarism comes from Latin, namely “plagiare” which means stealing. Plagiarism comes from the word plagiarism which means taking someone else’s work (opinions and so on) and making it as if it were one’s own work (opinions and so on), for example publishing someone else’s written work in one’s own name. To reduce this problem, an Application for Detecting Document Similarity Using the Dice Similarity Method was created. This application is built using the Prototype method which is one of the methods for designing applications where the application needs are analyzed both in terms of functional and non-functional requirements, then designing the application and then testing it. This application is also tested using the blackbox method where the application is tested based on the user’s point of view. In addition, this application uses the Model View Controller (MVC) architecture which partitions the application code into several parts. This application has been successfully tested with results that match the testing.

**Keywords** : application, plagiarism, prototype, MVC, blackbox

**DAFTAR ISI**

HALAMAN JUDUL.....	i
HALAMAN SAMPUL .....	ii
LEMBAR PERNYATAAN KEOTENTIKAN.....	iii
HALAMAN PERSETUJUAN PEMBIMBING .....	iv
HALAMAN PENGESAHAN.....	v
KATA PENGANTAR .....	vi
PERSETUJUAN PUBLIKASI TUGAS AKHIR .....	viii
ABSTRAK .....	ix
ABSTRACT .....	x
DAFTAR ISI.....	xi
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian .....	3
1.4 Manfaat Penelitian .....	3
1.5 Batasan Masalah .....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Penelitian Terkait.....	4
2.2 Landasan Teori .....	6
2.2.1 <i>Website</i> .....	6
2.2.2 <i>Hypertext Transfer Protocol Secure (HTTPS)</i> .....	6
2.2.3 Javascript .....	7
2.2.3.1 Karakteristik Javascript.....	7
2.2.3.2 NodeJS .....	9
2.2.3.3 ExpressJS .....	10
2.2.4 <i>Database</i> .....	10
2.2.4.1 SQL dan NoSQL.....	10
2.2.4.2 MongoDB.....	11
2.2.5 Jurnal.....	11
2.2.6 Plagiarisme.....	12
2.2.7 <i>Dice Similarity</i> .....	13
2.2.7.1 Implementasi <i>Dice Similarity</i> .....	14
BAB III METODE PENELITIAN.....	18

3.1	Prosedur Penelitian.....	18
3.2	<i>Timeline</i> Penelitian.....	18
3.3	Rancangan Sistem.....	19
3.3.1	Rancangan Antar Muka ( <i>Interface</i> ).....	19
3.4	Metode Penelitian.....	21
3.4.1	Metode Prototype.....	21
3.5	Instrumen Penelitian.....	22
BAB IV HASIL DAN PEMBAHASAN.....		23
4.1.	<i>Requirements Gatehering and Analysis</i> (Analisis Kebutuhan).....	23
4.2	<i>Quick Design</i> (Desain Cepat).....	24
4.2.1	<i>Use Case Diagram</i> .....	24
4.2.2	<i>Activity Diagram</i> .....	25
4.2.3	Entity Relationship Diagram (ERD).....	26
4.2.4	Tampilan Antar Muka ( <i>Interface</i> ).....	27
4.2.5	Desain Sistem.....	31
4.2.5.1	Algoritma Input Dokumen ke dalam <i>Database</i> .....	31
4.2.5.2	Algoritma Perbandingan Dokumen.....	32
4.2.5.3	Algoritma <i>Dice Similarity</i> .....	34
4.3	<i>Build Prototype</i> (Bangun Prototipe).....	37
4.3.1	Model View Controller (MVC).....	37
4.3.2	Implementasi Pembuatan Sistem.....	37
4.3.3	Sintaks <i>Dice Similarity</i> .....	44
4.4	<i>User Evaluation</i> (Evaluasi Pengguna).....	48
4.4.1	Rencana Pengujian.....	48
4.4.2	Tahap Pengujian.....	51
4.5	<i>Refining Prototype</i> (Memperbaiki Prototipe).....	58
4.6	<i>Implement Product and Maintain</i> (Implementasi dan Pemeliharaan).....	58
BAB V KESIMPULAN DAN SARAN.....		59
5.1	Kesimpulan.....	59
5.2	Saran.....	59
DAFTAR PUSTAKA.....		60
LAMPIRAN.....		62

**DAFTAR GAMBAR**

Gambar 2.1 Contoh Pembuatan Variabel..... 8

Gambar 2.2 Contoh Penggunaan Tipe Data..... 8

Gambar 2.3 Contoh Penggunaan *Synchronous* dan *Asynchronous* ..... 9

Gambar 2.4 Contoh Code Mongoddb ..... 11

Gambar 3.1 Rancangan Antar Muka Halaman Utama..... 19

Gambar 3.2 Rancangan Antar Muka Halaman *Login* ..... 20

Gambar 3.3 Rancangan Antar Muka Halaman Beranda..... 20

Gambar 3.4 Rancangan Antar Muka Halaman *Admin*..... 21

Gambar 4.1 *Use Case Diagram* ..... 25

Gambar 4.2 *Activity Diagram* ..... 26

Gambar 4.3 *Entity Relation Diagram* ..... 27

Gambar 4.4 Tampilan Halaman Utama ..... 28

Gambar 4.5 Tampilan Halaman *Login*..... 28

Gambar 4.6 Tampilan Halaman Beranda..... 29

Gambar 4.7 Tampilan Halaman Admin ..... 30

Gambar 4.8 Algoritma *Input* Dokumen ke dalam *Database* ..... 31

Gambar 4.9 Algoritma Perbandingan Dokumen..... 32

Gambar 4.10 Algoritma *Dice Similarity* ..... 35

Gambar 4.11 Contoh Sintaks *Model* ..... 38

Gambar 4.12 Contoh Sintaks *View* ..... 39

Gambar 4.13 Contoh Sintaks *Controller* ..... 41

Gambar 4.14 Contoh Sintaks *Middleware* ..... 42

Gambar 4.15 Contoh Sintaks *Routes* ..... 43

Gambar 4.16 Contoh Sintaks *Utils*..... 44

Gambar 4.17 Sintaks *Input* Dokumen ke Dalam *Database* ..... 45

Gambar 4.18 Sintaks *Dice Similarity*..... 46

Gambar 4.19 Pengujian Langsung Menggunakan Jurnal ..... 57

**DAFTAR TABEL**

Tabel 2.1 Perbandingan Data X dan Data Y .....	15
Tabel 2.2 Kamus Data.....	15
Tabel 2.3 Perhitungan Dalam Kamus Data.....	16
Tabel 3.1 <i>Timeline</i> Penelitian.....	18
Tabel 4.1 Pengujian Pengguna <i>Guest</i> .....	48
Tabel 4.2 Pengujian Pengguna <i>User</i> .....	49
Tabel 4.3 Pengujian Pengguna Admin.....	50
Tabel 4.4 Hasil Pengujian Pengguna <i>Guest</i> .....	52
Tabel 4.5 Hasil Pengujian Pengguna <i>User</i> .....	52
Tabel 4.6 Hasil Pengujian Pengguna <i>Admin</i> .....	54

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Teknologi merupakan sarana untuk menyediakan barang-barang yang diperlukan untuk mempermudah kegiatan atau kehidupan manusia. Penggunaan teknologi oleh manusia dimulai dengan pengubahan sumber daya alam menjadi alat-alat sederhana. Penemuan di masa lampau tentang kemampuan mengendalikan api telah menaikkan ketersediaan sumber-sumber pangan, sedangkan penciptaan roda telah membantu manusia dalam bepergian dari satu tempat ke tempat lainnya. Perkembangan teknologi terbaru dimasa sekarang seperti mesin cetak, telepon, dan internet telah mengubah manusia untuk bekerja maupun berinteraksi dalam menjalankan kehidupannya.

Dengan adanya teknologi di saat ini, dapat dikatakan pekerjaan manusia semakin mudah. Seperti untuk akses informasi, sekarang sudah banyak sekali informasi yang tersebar di internet dan hampir semua yang dicari dapat ditemukan. Tidak seperti dahulu, untuk mengakses informasi, orang biasa pergi ke perpustakaan untuk mendapatkan informasi yang diinginkan. Hal itu memungkinkan karena adanya digitalisasi dokumen yang dulunya hanya berupa *hardcopy* (salinan cetak) menjadi *softcopy* (salinan digital). Dengan hal tersebut kita bisa mengakses semua dokumen kapan saja dan di mana saja. Dengan semua kemudahan yang diberikan teknologi seperti contoh di atas, timbul suatu masalah yaitu mudahnya dilakukan plagiarisme yang merupakan suatu tindakan yang tidak baik.

Plagiarisme berasal dari bahasa Latin yaitu "*plagiare*" yang berarti mencuri. Plagiarisme berasal dari kata plagiat yang berarti pengambilan karangan (pendapat dan sebagainya) orang lain dan menjadikannya seolah-olah karangan (pendapat dan sebagainya) sendiri, misalnya menerbitkan karya tulis orang lain atas nama dirinya sendiri. Sehingga dapat diartikan plagiarisme merupakan tindakan mencuri gagasan hasil penelitian orang lain, untuk kemudian disajikan seolah-olah milik sendiri (Ridhatillah, 2003). Plagiarisme yang marak dilakukan di kalangan mahasiswa membuat mahasiswa menjadi malas berpikir dan mengembangkan kemampuan

sebagai kaum intelektual. Moral mahasiswa akan luntur karena dengan melakukan plagiarisme pemikiran mereka tidak dapat berkembang dengan maksimal. Sebab mahasiswa cenderung mencari kemudahan dengan mengambil karya orang lain dan mengakui sebagai karya pribadi (Arista & Listyani, 2015). Tindakan plagiarisme merupakan salah satu tindakan yang melanggar hak cipta. Hak Cipta itu sendiri merupakan hak eksklusif untuk Pencipta ataupun penerima hak buat mengumumkan ataupun perbanyak Ciptaannya ataupun membagikan izin buat itu dengan tidak kurangi pembatasan- pembatasan bagi peraturan perundang-undangan yang berlaku. Jika terjadi pelanggaran tersebut, dapat dikenai pelanggaran hak cipta di Pasal 72 ayat UUHC dengan dipidana dengan pidana penjara pendek selama 1 bulan serta / ataupun denda sangat sedikitnya Rp1.000.000,00, ataupun pidana penjara lama 7 tahun serta / ataupun denda sebanyak Rp5.000.000.000,00 (Lopes, 2013)

Untuk mengurangi hal tersebut, dapat dilakukan beberapa metode untuk mencegah plagiarisme seperti menggunakan cara manual ataupun menggunakan teknologi yang ada. Jika menggunakan cara manual, dilakukan dengan membandingkan sebuah dokumen yang akan dicek dengan dokumen yang akan dijadikan perbandingan dengan melihat kata yang digunakan ataupun tata penulisan dokumen tersebut. Cara ini kurang efektif karena jumlah dokumen yang akan dijadikan perbandingan jumlah tidak sedikit bisa mencapai ratusan bahkan ribuan dokumen lebih. Sedangkan untuk cepat menggunakan mesin sebagai pembanding.

Ada beberapa metode yang dapat digunakan seperti *Cosine Similarity*, TF-IDF, *Jaccard Similarity*, *Dice Similarity*, *Word2Vec* dan sebagainya. Dengan menggunakan salah satu metode di atas, dapat temukan persentase kesamaan sebuah dokumen. Salah satu metode yang dapat digunakan yaitu dengan metode *Dice Similarity*. *Dice Similarity* merupakan salah satu metode perbandingan antara dokumen dengan membandingkan kata yang ada di antara dokumen tersebut. Melihat permasalahan di atas, maka disusunlah tugas akhir ini dengan judul “RANCANG BANGUN APLIKASI PENDETEKSI KESAMAAN DOKUMEN MENGGUNAKAN METODE DICE SIMILARITY”. Dengan adanya penelitian ini, diharapkan dapat membangun sebuah aplikasi Web yang mampu memberikan persentase kesamaan dari sebuah dokumen terhadap dokumen lainnya yang

terdapat di dalam *database* guna mengurangi tindak plagiarisme yang sering terjadi saat ini.

### **1.2 Rumusan Masalah**

Rumusan masalah dalam penelitian ini yaitu bagaimana cara merancang sebuah aplikasi pendeteksi kesamaan teks berbasis *website* dengan mengimplementasikan metode *Dice Similarity* serta bagaimana cara menguji aplikasi tersebut menggunakan sebuah dokumen ?

### **1.3 Tujuan Penelitian**

Tujuan penelitian dalam penelitian yaitu merancang sebuah aplikasi pendeteksi kesamaan teks berbasis *website* dengan mengimplementasikan metode *Dice Similarity* serta menguji aplikasi tersebut menggunakan sebuah dokumen.

### **1.4 Manfaat Penelitian**

Penelitian ini merupakan pengimplementasian metode *Dice Similarity* pada sebuah aplikasi berbasis web guna mengetahui persentase kesamaan sebuah dokumen dengan dokumen lainnya sehingga dapat digunakan untuk mengurangi tindak plagiarisme dengan mudah.

### **1.5 Batasan Masalah**

Adapun beberapa batasan masalah pada penelitian ini yaitu :

1. Sistem ini dibuat menggunakan bahasa pemrograman Javascript dengan *runtime environment* Nodejs, *framework* Express JS, dengan database MongoDB.
2. Menggunakan data berupa jurnal yang diinput oleh user dan memiliki ekstensi pdf (*Portable Document Format*).

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terkait

Dalam Penelitian Mufti Ari Bianto, dkk (2018) dengan judul *Perancangan Sistem Pendeteksi Plagiarisme Terhadap Topik Penelitian Menggunakan Metode K-Means Clustering dan Model Bayesian*, hasil kemiripan diuji berdasarkan kesamaan pola kata dalam kalimat, dokumen dengan nilai *similarity* tertinggi diperoleh satu dokumen dengan persentase kemiripan sebanyak 100 %, dan jika berdasarkan kesamaan *term*, berpengaruh terhadap hasil dokumen mirip yang dihasilkan, sehingga diperoleh dua dokumen mirip. Meskipun uji berdasarkan kesamaan *term*, menghasilkan dokumen mirip yang lebih banyak namun belum cukup akurat menunjukkan adanya plagiasi, karena dalam menentukan plagiasi kesamaan rangkaian kalimat merupakan hal yang penting untuk diperhatikan (Bianto, Rahayu, Huda, & Kusriani, 2018).

Pada penelitian Joko Priambodo (2018) dengan judul *Pendeteksian Plagiarisme Menggunakan Algoritma Rabin - Karb dengan Meotde Rolling Hash*, pendeteksian plagiarisme menggunakan algoritma *Rabin-Karp* dengan metode *rolling hash* dari hasil pengujian 30 dokumen teks yang menghasilkan tingkat akurasi yang terbesar yaitu 47.58 %. Hasil persentase tersebut termasuk dalam kategori tingkat plagiat 15 - 50 %, berarti menandakan dokumen tersebut termasuk plagiat tingkat sedang. Sedangkan tingkat akurasi yang terkecil yaitu 19.28 %, berarti menandakan dokumen tersebut termasuk plagiat tingkat sedang. Selain itu, berdasarkan analisis proses pendeteksian tingkat plagiarisme menggunakan algoritma *Rabin-Karp* dengan metode *Rolling Hash* bisa membaca karakter berupa huruf, simbol seperti titik (.), koma (,), dan lain-lain (Priambodo, 2018).

Kemudian dipenelitian Pavel Stefanovic (2019) dengan judul *The N-Grams Based Text Similarity Detection Approach Using Self-Organizing Maps and Similarity Measures*, dengan pendekatan berdasarkan teks yang dipecah menjadi n-gram dan mengevaluasinya menggunakan *Self-Organizing Maps* (SOM) dan *Similarity Measure*. Deteksi teks serupa dilakukan dalam tiga langkah: (1) konversi kumpulan data teks ke numerik ekspresi menggunakan n-gram; (2) perhitungan

ukuran kesamaan; (3) visualisasi dataset teks menggunakan SOM dan representasi kesamaan di atasnya. Pada langkah pertama, fokus utamanya adalah membuat grup *n*-gram dari semua dataset. Berbagai jumlah kata dalam *n*-gram dianalisis. Selain itu, filter yang berbeda diterapkan seperti penghapusan angka dan tanda baca, frekuensi kata, transformasi huruf besar, *stemming* algoritma, dll. Analisis menunjukkan filter dan ukuran *n*-gram mempengaruhi hasil akhir. Untuk ini dataset, ukuran *n*-gram dipilih dan sama dengan tiga untuk penyelidikan eksperimental. Pada langkah kedua, empat ukuran kesamaan dihitung: *Cosinus*, *Dice*, *Extends Jaccard*, dan *Overlap*. Hasil akhir menunjukkan bahwa persentase kemiripan tertinggi diperoleh dengan menggunakan *overlap*. Tiga nilai ukuran lainnya selalu sama dan lebih kecil. Penggunaan SOM menunjukkan bahwa SOM membantu untuk melihat hasil ringkasan kesamaan semua teks dalam bentuk visual dengan cepat. Dia sangat mudah untuk memahami teks mana yang mirip satu sama lain atau tidak. Dalam kasus kumpulan data yang dianalisis, SOM membantu mendeteksi kesamaan, dan *cluster* yang terbentuk dikorelasikan dengan kategori yang diberikan deskripsi kumpulan data (Stefanovi, Kurasova, & Štrimaitis, 2019).

Sedangkan, pada penelitian Uswatun Hasanah, dkk (2019) dengan judul *Perbandingan Metode Cosine Similarity dan Jaccard Similarity Untuk Penilaian Otomatis Jawaban Pendek*, dinilai belum mampu memberikan jawaban yang memuaskan i dikarenakan kedua metode hanya menilai kemiripan berdasarkan susunan leksikalnya. Sementara itu, jawaban mahasiswa juga sangat bervariasi dan menggunakan kata-kata yang jauh berbeda dari jawaban kunci, walaupun pada dasarnya memiliki makna semantik yang sama. Oleh karena itu, pada penelitian selanjutnya diperlukan metode lain yang mampu menangani makna semantik pada jawaban. Bagaimanapun, metode *Cosine Similarity* dan *Jaccard Similarity* masih dapat dipertimbangkan untuk menilai jawaban pendek secara otomatis, dengan batasan bahwa pertanyaan yang digunakan mengharuskan jawaban dalam format *keyword* sehingga tidak memunculkan kata-kata lain yang mampu menurunkan nilai kemiripan (Hasanah & Mutiara, 2019).

## 2.2 Landasan Teori

### 2.2.1 Website

*Website* adalah kumpulan halaman web yang saling terhubung dan seluruh file saling terkait. Web terdiri dari *page* atau halaman dan kumpulan halaman yang dinamakan *homepage*. *Homepage* berada pada posisi teratas dengan halaman-halaman terkait berada di bawahnya. Biasanya, setiap halaman di bawah *homepage* (*child page*) berisi *hyperlink* ke halaman lain dalam web (Agung, 2000).

Sejarah *website* dimulai oleh Tim Berners Lee, yang mengembangkan *World Wide Web* (WWW) pada tahun 1989. Pada bulan Oktober tahun 1990, Tim menggagas tiga teknologi dasar untuk membangun sebuah *website*, yaitu HTML, URL, dan HTTP. Awalnya, *website* dirancang hanya untuk berbagi informasi di kalangan para ilmuwan di *Centre Europeen pour la Recherche Nucleaire* (CERN). Sampai akhirnya, dia melihat potensi *website* sebagai sarana yang bisa digunakan oleh siapa saja untuk berbagai tujuan. Kemudian, dia merilisnya secara resmi pada 6 Agustus 1991.

*Website* yang pertama dirilis saat itu baru sebatas menampilkan teks sederhana tanpa variasi *font*, video, maupun gambar. Saat ini, *website* sudah berkembang sangat pesat dan menjadikan *website* sebagai salah satu kebutuhan penting baik individu, organisasi, dan perusahaan di seluruh dunia hingga saat ini. Saat ini, *website* berkembang sangat pesat dengan muncul berbagai jenis *website* seperti pribadi, *e-commerce*, *blog*, dan media sosial.

### 2.2.2 Hypertext Transfer Protocol Secure (HTTPS)

*Hypertext Transfer Protocol Secure* (HTTPS) adalah sebuah *protocol* komunikasi dalam suatu jaringan internet dengan keamanan yang lebih terjamin. Disebut lebih aman karena suatu perintah atau data yang dikirim melalui HTTPS ini dilindungi dengan sistem enkripsi sehingga menyulitkan *hacker* untuk membobol atau mencurinya (Firmansyah, 2019). HTTPS merupakan tingkatan dari HTTP di mana yang membedakan di HTTPS terdapat (*Secure Socket Layer*) SSL dan (*Transport Layer Security*) TSL yang digunakan untuk mengamankan data yang disimpan atau yang akan dikirim. Menurut SSL Labs pada April 2018, 33.2% dari 1.000.000 situs web teratas Alexa menggunakan HTTPS sebagai *default*,

57.1% dari 137.971 situs web paling populer di Internet memiliki implementasi HTTPS yang aman, dan 70% dari pemuatan halaman (diukur oleh *Firefox Telemetry*) menggunakan HTTPS. Jadi saat ini *website* sekarang sudah mulai beralih dari sebelumnya HTTP ke HTTPS karena di HTTPS lebih aman daripada HTTP yang sudah lama.

### 2.2.3 Javascript

Javascript adalah sebuah bahasa pemrograman tingkat tinggi yang dinamis, *scripting*, *untyped*, dan *interpreter*. Javascript sendiri dibuat oleh Brendan Eich dari perusahaan Netscape pada tahun 1994 yang diberi nama Mocha pada saat itu, kemudian berganti menjadi Livescript. Karena saat itu browser yang populer adalah Netscape, Microsoft berusaha untuk mengalahkan popularitas browser tersebut dengan Internet Explorer dan melakukan *Reverse Engineering* terhadap Livescript dan terciptalah JScript pada tahun 1996. Karena terdapat dua browser yang besar yang berbeda, maka dibuatlah satu standar agar mempermudah pembuatan website saat itu dan dibuatlah ECMAScript. Menurut Douglas Rockford, "*Javascript, JScript, and ECMAScript 3 Silly Name for 1 Silly Language*" yang berarti bahwa ketiga nama tersebut adalah bahasa yang sama yaitu Javascript.

#### 2.2.3.1 Karakteristik Javascript

Untuk Ekstensi dari file Javascript menggunakan ekstensi \*.js. Contoh namaFile.js. Setiap akhir dari kode Javascript dapat menggunakan ";" ataupun tidak. Sedangkan *block scope* di Javascript menggunakan { }.

Javascript merupakan salah satu bahasa pemrogram yang *untyped / dynamicly typed* yang berarti tidak mendefinisikan terlebih dahulu tipe variabel yang akan didefinisikan. Untuk penamaan variabel, terdapat beberapa *keyword* yaitu var, let, dan const. Untuk var, dan let, nilai variabel dapat berubah atau diisi ulang, sedangkan const nilai variabelnya tidak dapat diubah. Perbedaan var dan let sendiri terletak pada *hoisting*. Sedangkan untuk penamaan variabel, hampir sama dengan beberapa bahasa pemrograman lainnya seperti karakter pertama variabel tidak boleh angka, menggunakan penulisan *camelCase*, dan untuk *keyword* const yang merupakan variabel const biasanya menggunakan *snake\_case* dan semuanya huruf kapital. Untuk contoh pembuatan variabel dapat dilihat pada Gambar 2.1.

```
var value = 10 // Contoh Penggunaan var
let nilai = 'Delapan' // Contoh Penggunaan let
const PI = 3.14 // Contoh Penggunaan const
```

Gambar 2.1 Contoh Pembuatan Variabel

Javascript memiliki beberapa tipe data seperti *String*, *Number* atau *Integer*, *Boolean*, *Array*, *Object*, dan *Undefined*. Untuk mengetahui tipe data dari sebuah variabel dapat menggunakan *keyword typeof*. Untuk contoh penggunaan tipe data dapat dilihat pada Gambar 2.2.

```
let name = 'Aris Akhyar Abdillah' // String
let age = 22 // Number atau Integer
let male = true // Boolean
let family = ['Father', 'Mother', 'Son', 'Girl'] // Array
let detail = {
  fullName : 'Aris Akhyar Addillah',
  nim : 'H071171505'
} // Object
let car = undefined // Undefined
```

Gambar 2.2 Contoh Penggunaan Tipe Data

*Synchronous* dan *Asynchronous*, Secara sederhana, *Synchronous* dan *Asynchronous* merupakan tahapan dalam mengeksekusi sebuah kode di mana *Synchronous* mengeksekusi sebuah kode per baris sesuai urutan kode yang dituliskan. Sedangkan *Asynchronous*, tidak selalu seperti *Synchronous*, tapi melihat waktu proses dari kode tersebut. Penggunaan *Asynchronous* tidak akan menunggu suatu kode selesai dijalankan, tetapi berlanjut ke kode selanjutnya. Untuk contoh penggunaan *Synchronous* dan *Asynchronous* dapat dilihat pada Gambar 2.3.

```

// Menggunakan Synchronous
console.log('What Is Your Name ?')
console.log('Hello World')
console.log('My Name is Aris')
// Output :
// What Is Your Name ?
// Hello World
// My Name is Aris

// Menggunakan Asynchronous
setTimeout(function() {
  console.log('What Is Your Name ?')
}, 1500)
console.log('Hello World')
setTimeout(function() {
  console.log('My Name is Aris')
}, 1000)
// Output :
// Hello World
// My Name is Aris
// What Is Your Name ?

```

Gambar 2.3 Contoh Penggunaan *Synchronous* dan *Asynchronous*

Jika melihat output dari dua program di atas antara menggunakan *Synchronous* dan *Asynchronous* terdapat perbedaan urutan dari hasil *output*, di mana *Synchronous* memiliki *output* sesuai dari urutan baris kode sedangkan *Asynchronous* berbeda di mana *output* yang dihasilkan berdasarkan jumlah waktu pengeksekusian dari kode tersebut yang disimulasikan menggunakan *setTimeout()* dari pengeksekusian paling cepat hingga yang paling lambat.

### 2.2.3.2 NodeJS

Node.js adalah *runtime environment* untuk Javascript yang bersifat *open-source* dan *cross-platform*. Dengan Node.js kita dapat menjalankan kode Javascript di mana pun, tidak hanya terbatas pada lingkungan *browser*. Seperti diketahui bahwa Javascript hanya dapat berjalan pada sebuah *web browser*, kemudian Ryan Dahl,

membuat sebuah *runtime environment* dengan mengeluarkan *engine* Javascript dari Chrome yaitu *V8 Javascript Engine* menggunakan bahasa C agar Javascript dapat dijalankan diluar *browser*. Akhirnya tercipta NodeJS pada tahun 2009. Dengan Begitu, Javascript yang sebelumnya hanya bisa di *client side* dengan adanya NodeJS bisa juga di *server side*. Beberapa fitur yang terdapat di NodeJS seperti *Asynchronous & Event-driven, Single Threaded but Highly Scalable* (Anonymous, NodeJS, 2022).

### 2.2.3.3 ExpressJS

Menurut website resmi dari Express JS adalah “*Fast, unopinionated, minimalist web framework for Nodejs*” (Anonymous, Express JS, 2022). Express JS sendiri merupakan salah satu *web framework* khusus untuk NodeJS di mana kita dapat membuat sebuah *website* yang cepat, sederhana, dengan struktur yang tidak ditentukan atau tergantung dari pengguna ExpressJS sendiri.

### 2.2.4 Database

*Database* atau basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data (Andaru, 2018). Berdasarkan definisi *database*, *database* berfungsi untuk menyimpan catatan atau sebuah data pada sebuah penyimpanan yang kemudian akan digunakan pada waktu lainnya. Pada umumnya, database sudah pasti memiliki *key* dan *value*, walaupun istilah ini berbeda di tiap-tiap jenis database yang ada. Adapun beberapa jenis database seperti *Operational Database, Database Warehouse, Distributed Database, Relational Database*, dan *End User Database*.

#### 2.2.4.1 SQL dan NoSQL

SQL dan NoSQL merupakan salah satu contoh dari *Relational Database* yang populer saat ini. SQL (*Structured Query Language*) merupakan bahasa yang digunakan untuk mengelola data secara *relational*. SQL sendiri memiliki ciri yaitu memiliki tabel yang terdiri dengan *row* atau *record* dan *field* yang bisa memiliki relasi dengan tabel lainnya. Untuk tiap data yang ada di dalam tabel, harus memiliki skema yang sama untuk tiap recordnya. Contoh dari SQL seperti MySQL, PostgreSQL, dan MariaDB. Berbeda dengan NoSQL (*Not Only Structured Query*

*Language*) yang merupakan bahasa untuk mengelola data secara *Non Relational*. Berbeda dengan SQL yang menggunakan tabel untuk menyimpan data. NoSQL memiliki banyak jenis tempat untuk menyimpan data seperti *Document Database*, *Key-Value Database*, dan *Graph Database*. NoSQL juga tidak memiliki skema sehingga untuk menyimpan data bisa secara flexibel. Contoh dari NoSQL seperti MongoDB, Redis, Neo4j, dan Cassandra.

#### 2.2.4.2 MongoDB

MongoDB merupakan salah satu contoh dari NoSQL yang dikembangkan menggunakan bahasa pemrograman C++ yang rilis pertama kali pada tanggal 11 Februari 2009. MongoDB adalah salah satu contoh *Document Database* yang di mana tiap - tiap datanya merupakan sebuah JSON (*Javascript Object Notation*) atau dalam MongoDB disebut BSON (*Binary JSON*). Hingga saat ini, MongoDB sudah digunakan lebih dari 85 Juta pengguna di seluruh dunia dan sudah banyak perusahaan besar yang menggunakan database ini seperti EBay, Google, Adobe, dan EA (Anonymous, What Is MongoDB ?, 2022). Salah satu contoh data yang terdapat di MongoDB :

```
{
  _id : '91829jw1j000jo2',
  firstName : 'Aris',
  lastName : 'Akhyar',
  age : 20
  hobbies : ['eat', 'drink']
}
```

Gambar 2.4 Contoh Code Mongodb

#### 2.2.5 Jurnal

Jurnal merupakan bagian dari jenis terbitan berseri yang ada diperpustakaan, adapun pengertian jurnal menurut *High Beam* “*Journal is the collection and periodic publication or transmission of news and the result of research through media*”, artinya bahwa jurnal merupakan suatu koleksi dan terbitan berkala atau transmisi mengenai berita dan hasil-hasil penelitian mengenai media. Jurnal sendiri terbagi atas dua format yaitu tercetak dan digital (*e-journal*). Untuk format digital

jurnal dikemas dalam dua format , yaitu bentuk CD-ROM dan dalam bentuk akses secara *online* melalui *internet*. *E-Journal* dipahami sebagai publikasi ilmiah dalam format elektronik dan mempunyai ISSN (*International Standard Serial Number*) yang format dokumennya menggunakan pdf (Rusydi, 2014).

Penggunaan kata jurnal untuk berbagai bidang juga memberi arti yang bervariasi, misalnya jurnal dalam bidang ekonomi menunjukkan sistem pembukuan rangkap. Jurnal dalam bidang pelayaran diartikan sebagai *logbook* berarti buku untuk mencatat semua kejadian selama pelayaran. Jurnal sebenarnya merupakan publikasi ilmiah yang memuat informasi tentang hasil kegiatan dalam bidang ilmu pengetahuan dan teknologi minimal harus mencakup kumpulan atau kumulasi pengetahuan baru, pengamatan empiris dan pengembangan gagasan atau usulan. Dengan demikian jurnal merupakan representasi dari pengetahuan baru tentang perkembangan ilmu pengetahuan yang dilaksanakan secara empiris dan biasanya merupakan gagasan yang terbaru.

### **2.2.6 Plagiarisme**

Plagiarisme berasal dari bahasa Latin "*plagiare*" yang berarti mencuri. Plagiarisme berasal dari kata plagiat yang berarti pengambilan karangan (pendapat dan sebagainya) orang lain dan menjadikannya seolah-olah karangan (pendapat dan sebagainya) sendiri, misalnya menerbitkan karya tulis orang lain atas nama dirinya sendiri. Sehingga dapat diartikan plagiarisme merupakan tindakan mencuri gagasan hasil penelitian orang lain, untuk kemudian disajikan seolah-olah milik sendiri (Ridhatillah, 2003).

Tindakan plagiarisme merupakan salah satu tindakan yang melanggar hak cipta. Hak Cipta itu sendiri merupakan hak eksklusif untuk pencipta ataupun penerima hak buat mengumumkan ataupun perbanyak Ciptaannya ataupun membagikan izin buat itu dengan tidak mengurangi pembatasan-pembatasan bagi peraturan perundang-undangan yang berlaku. Jika terjadi pelanggaran tersebut, dapat dikenai pelanggaran hak cipta di Pasal 72 ayat UUHC dengan dipidana dengan pidana penjara pendek selama 1 bulan serta / ataupun denda sangat sedikitnya Rp1.000.000,00, ataupun pidana penjara lama 7 tahun serta / ataupun denda sebanyak Rp5.000.000.000,00 (Lopes, 2013).

**2.2.7 Dice Similarity**

*Dice Similarity* atau *Sørensen–Dice coefficient* merupakan salah satu algoritma yang mengukur kesamaan antara dua set data (Khontoro, Andjarwirawan, & Yulia, 2021). Algoritma ini banyak digunakan dalam validasi algoritma segmentasi gambar yang dibuat dengan *Artificial Intelligence* atau AI, tetapi ini adalah konsep yang jauh lebih umum yang dapat diterapkan pada kumpulan data untuk berbagai aplikasi termasuk *Natural Language Processing* (NLP). Metode ini sering digunakan untuk mengukur kemiripan antara dua dokumen teks dengan menggunakan pendekatan N-gram. N-gram adalah cara untuk membagi dokumen menjadi potongan-potongan kecil yang terdiri dari N karakter kemudian divisualisasikan menjadi vektor.

*Dice Similarity* memiliki kesamaan terhadap *Jaccard Similarity* di mana keduanya merupakan cara untuk membandingkan kedua data set atau vektor menggunakan pendekatan matematika. Untuk rumus dari *Jaccard Similarity* dengan X dan Y sebagai vektor yaitu :

$$J = \frac{|X.Y|}{|X|^2 + |Y|^2 - |X.Y|} \tag{2.1}$$

Untuk memperoleh rumus dari *Dice Similarity*, dapat menggunakan persamaan yaitu :

$$DSC = \frac{2J}{J + 1} \tag{2.2}$$

Dengan DSC adalah *Dice Similarity* dan J adalah persamaan dari *Jaccard Similarity* yang terdapat pada persamaan 2.1. Kemudian, nilai dari variabel J diganti dengan persamaan *Jaccard Similarity* sebelumnya sehingga menjadi :

$$DSC = \frac{2 \frac{|X.Y|}{|X|^2 + |Y|^2 - |X.Y|}}{\frac{|X.Y|}{|X|^2 + |Y|^2 - |X.Y|} + 1} \tag{2.3}$$

$$DSC = \frac{2 \frac{|X.Y|}{|X|^2 + |Y|^2 - |X.Y|}}{\frac{|X.Y| + |X|^2 + |Y|^2 - |X.Y|}{|X|^2 + |Y|^2 - |X.Y|}} \tag{2.4}$$

$$DSC = \frac{2 \frac{|X.Y|}{|X|^2 + |Y|^2 - |X.Y|}}{\frac{|X|^2 + |Y|^2}{|X|^2 + |Y|^2 - |X.Y|}} \quad (2.5)$$

$$DSC = \frac{2 \frac{|X.Y|}{|X|^2 + |Y|^2 - |X.Y|}}{\frac{|X|^2 + |Y|^2}{|X|^2 + |Y|^2 - |X.Y|}} \quad (2.6)$$

$$DSC = \frac{2 |X.Y|}{|X|^2 + |Y|^2} \quad (2.7)$$

Sehingga diperoleh persamaan dari *Dice Similarity* menggunakan data *vektor*, dengan X dan Y sebagai *vektor*, maka persamaan akhirnya yaitu sesuai dengan persamaan 2.7. Metode *Dice Similarity* memiliki kelebihan yaitu tidak membutuhkan komputasi yang terlalu berat, namun memiliki kekurangan yaitu sensitif terhadap perbedaan panjang dokumen.

### 2.2.7.1 Implementasi *Dice Similarity*

Untuk pengimplementasian dari metode *Dice Similarity*, diambil contoh dari definisi dari komputer menurut para ahli yaitu Menurut Robert H. Blissmer, “komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas seperti menerima input, memproses input tadi sesuai dengan programnya, menyimpan perintah-perintah dan hasil pengolahan, serta menyediakan output dalam bentuk informasi”. Sedangkan menurut Arief Susanto, “komputer adalah sekelompok alat elektronik yang terdiri atas perintah input, alat yang mengolah input, dan peralatan output yang memberikan informasi serta bekerja secara otomatis” (Susanto, 2009).

Setelah mendapatkan kedua data di atas, kemudian dilakukan pembagian. Untuk pengertian dari Robert H. Blissmer akan menjadi data dari *database* sebagai X, kemudian untuk pengertian dari Arief Susanto akan menjadi data yang akan dibandingkan dengan data dari *database* sebagai Y. Sebelum diolah, kedua data di atas akan dilakukan penghapusan *Stopword* dan karakter yang tidak berguna dalam perhitungan seperti pada Tabel 2.1.

Tabel 2.1 Perbandingan Data X dan Data Y

Data X	Data Y
alat elektronik tugas menerima input memproses input sesuai programnya menyimpan perintah - perintah hasil pengolahan menyediakan output bentuk informasi	sekelompok alat elektronik perintah input alat mengolah input peralatan output informasi otomatis

Setelah dilakukan penghapusan *Stopword* dan karakter yang tidak berguna, kemudian dilakukan pembuatan kamus kata lalu dilakukan perhitungan jumlah kata terhadap kamus data yang ada seperti pada Tabel 2.2.

Tabel 2.2 Kamus Data

No	Kamus Data	Data X	Data Y
1	alat	2	1
2	bentuk	0	1
3	elektronik	1	1
4	hasil	0	1
5	informasi	1	1
6	input	2	2
7	memproses	0	1
8	menerima	0	1
9	mengolah	1	0
10	menyediakan	0	1
11	menyimpan	0	1
12	otomatis	1	0
13	output	1	1
14	pengolahan	0	1
15	perintah	1	2
16	peralatan	1	0
17	programnya	0	1
18	sekelompok	1	0
19	sesuai	0	1
20	tugas	0	1

Setelah itu, data di atas akan diubah menjadi vektor satu dimensi sehingga menjadi :

$$X = \{2, 0, 1, 0, 1, 2, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0\} \quad (2.8)$$

$$Y = \{1, 1, 1, 1, 1, 2, 1, 1, 0, 1, 1, 0, 1, 1, 2, 0, 1, 0, 1, 1\} \quad (2.9)$$

Kemudian kedua vektor di atas dimasukkan ke dalam rumus *Dice Similarity* pada persamaan 2.7. Untuk mempermudah perhitungan, dilakukan beberapa penambahan *row* dan *field* sehingga menjadi :

Tabel 2.3 Perhitungan Dalam Kamus Data

No	Kamus Data	Data X	Data Y	X . Y	X <sup>2</sup>	Y <sup>2</sup>
1	alat	2	1	2	4	1
2	bentuk	0	1	0	0	1
3	elektronik	1	1	1	1	1
4	hasil	0	1	0	0	1
5	informasi	1	1	1	1	1
6	input	2	2	4	4	4
7	memproses	0	1	0	0	1
8	menerima	0	1	0	0	1
9	mengolah	1	0	0	1	0
10	menyediakan	0	1	0	0	1
11	menyimpan	0	1	0	0	1
12	otomatis	1	0	0	1	0
13	output	1	1	1	1	1
14	pengolahan	0	1	0	0	1
15	perintah	1	2	2	1	4
16	perlatan	1	0	0	1	0
17	programnya	0	1	0	0	1
18	sekelompok	1	0	0	1	0
19	sesuai	0	1	0	0	1
20	tugas	0	1	0	0	1
<b>Total</b>		<b>12</b>	<b>18</b>	<b>11</b>	<b>16</b>	<b>22</b>

Setelah memperoleh hasil seperti di Tabel 2.3, kemudian nilai tersebut dimasukkan ke dalam rumus *Dice Similarity* sehingga menjadi :

$$DSC = \frac{2|X.Y|}{|X|^2 + |Y|^2} \quad (2.10)$$

$$DSC = \frac{2|11|}{16 + 22} \quad (2.11)$$

$$DSC = \frac{22}{38} \quad (2.12)$$

$$DSC = 0.5789473684 \quad (2.13)$$

Sehingga diperoleh nilai *Dice Similarity* adalah 0.5789473684 atau jika dipersentasekan menjadi 57.8 %. Jadi, persentase kesamaan antara Data X dengan Data Y adalah sebesar 57.8 %