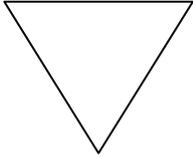


	<p>Rangka Dari Sebuah Dokumen</p>	<p>Rangka dokumen diindikasikan dengan pemberian nomor dokumen pada sudut sebelah kanan simbol dokumen</p>
	<p>Arsip</p>	<p>Arsip dokumen yang disimpan secara manual. Huruf yang tercantum dalam simbol mengindikasikan urutan pengarsipan. N=urut nomor, A=urut abjad, T=urut tanggal.</p>

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan cepatnya pertumbuhan informasi digital, mengakibatkan semakin besarnya dimensi data atau informasi yang ada. Dimensi data yang besar dapat memicu informasi yang penting menyebar terlalu luas, sehingga menjadi kurang informatif. Hal tersebut juga mengakibatkan berkembangnya teknologi informasi yang dibutuhkan oleh pengguna sehingga mengakibatkan munculnya suatu cabang ilmu baru dalam teknologi informasi yaitu *information retrieval (IR)*[1].

Information Rretrieval System (IRS) atau sistem temu kembali informasi adalah suatu proses untuk mengidentifikasi, kemudian memanggil suatu dokumen dari suatu simpanan. Pengertian lain menyatakan bahwa sistem temu balik informasi merupakan proses yang berhubungan dengan refresentasi, penyimpanan, pencarian dengan pemanggilan informasi yang relevan dengan kebutuhan informasi yang diinginkan pengguna [2]. *Information Rretrieval System (IRS)* akan diterapkan untuk memudahkan dalam proses pencarian jurnal.

Saat ini perkembangan ilmu pengetahuan dan teknologi tidak terlepas dari banyaknya penelitian-penelitian yang dilakukan, oleh individu-individu atau institusi-institusi baik dari kalangan pendidikan, swasta, dan pemerintahan. Kemudian hasil dari penelitian tersebut berupa tugas akhir, makalah, artikel, dan jurnal yang dipublikasikan baik melalui media cetak ataupun media elektronik. Di media elektronik selain melauai radio dan televisi, hasil dari penelitian tersebut bisa

juga dipublikasikan di internet yaitu melalui *website* jurnal *online*. Selain itu, banyaknya jumlah jurnal yang dipublikasikan tidak disertai dengan pertumbuhan jumlah informasi atau pengetahuan yang dapat diambil dari dokumen-dokumen elektronik tersebut. Maka diperlukan sebuah aplikasi yang dapat membantu untuk mempermudah dalam proses pencarian jurnal tentunya dengan menerapkan *Information Rretrieval System (IRS)*. Dalam penelitian ini akan dibuat sebuah aplikasi pencarian jurnal dengan menerapkan *Information Rretrieval System (IRS)* maka metode yang digunakan dalam penelitian ini yaitu dengan menggunakan metode *Levenshtein Distance*.

Levenshtein Distance atau yang biasa disebut dengan *edit distance* adalah suatu metode yang dapat digunakan untuk mengatasi terjadinya kesalahan ejaan. Kesalahan ejaan terjadi apabila kata yang diketik oleh pengguna tidak terdapat pada daftar kamus Bahasa Indonesia. Fungsi metode *Levenshtein Distance* yaitu untuk menghitung jarak kedekatan dari dua buah string melalui penambahan karakter, pengubahan karakter, dan penghapusan karakter hingga kedua string tersebut cocok [3]. Metode *Levenshtein Distance* telah diterapkan pada beberapa penelitian. Pada penelitian yang dilakukan oleh Isbalaikana Larasati dan Fitri Marisa (2019) dengan judul “Penerapan Algoritma *Levenstein Distance* Dalam Aplikasi Pengolahan Surat”, maka ditarik kesimpulan bahwa dapat membantu dalam mempermudah proses pencarian arsip surat pada aplikasi pengolahan surat di CV. Khasanah Konsultama. Metode *algoritma levenshtein distance* menggunakan matrik perbandingan dengan rumus $[j+1]$, $[i+1]$, $[+edit]$. Dari hasil perhitungan dengan menggunakan metode *algoritma levenshtein distance*

memiliki hasil perbandingan kata yang benar akan ditampilkan [4]. Selanjutnya, pada penelitian yang dilakukan oleh M. Julianda (2020) dengan judul “Sistem Informasi Pendeteksi Kemiripan Isi Teks Dokumen Sebagai Acuan Penentuan Kelayakan Judul Skripsi Menggunakan Metode *Levenstein Distance*”, maka ditarik kesimpulan bahwa dengan adanya sistem informasi pendeteksi kemiripan teks ini, kaprodi akan dimudahkan khususnya dalam proses penyeleksian, dimana proses penyeleksian akan menjadi lebih cepat efisien. Dengan adanya sistem informasi pendeteksi kemiripan teks ini, mahasiswa juga akan diuntungkan dengan cepatnya mereka mendapatkan informasi hasil seleksi proposal judul yang mereka ajukan, dan dengan demikian maka mahasiswa juga akan cepat untuk menyelesaikan skripsinya. Dengan adanya sistem informasi pendeteksi kemiripan teks ini, akan dapat mengurangi tindak penjiplakan atau plagiarisme yang dilakukan mahasiswa saat mengajukan proses pengajuan proposal judul [5]. Berdasarkan penelitian tersebut, maka ingin mencoba menerapkan metode *Levenshtein Distance* pada aplikasi pencarian jurnal berbasis *web*.

Berdasarkan latar belakang yang telah dipaparkan, maka penelitian ini diberi judul sebagai berikut “**Rancang Bangun Aplikasi Pencarian Jurnal Dengan Menerapkan *Information Rretrieval System* Menggunakan Metode *Levenshtein Distance*”.**

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang dipaparkan, dapat dirumuskan sebagai berikut : “Bagaimana merancang dan membangun sebuah aplikasi

pencarian jurnal berbasis *web* dengan menerapkan *Information Rretrieval System* menggunakan metode *Levenshtein Distance* ?”.

1.3 Tujuan Penelitian

Tujuan kegiatan penelitian tugas akhir ini adalah merancang dan membangun sebuah aplikasi pencarian jurnal berbasis *web* dengan menerapkan *Information Rretrieval System* menggunakan metode *Levenshtein Distance*.

1.4 Batasan Masalah

Pada pembangunan sistem ini dibuat beberapa batasan masalah agar pembahasan lebih terfokus pada masalah yang diteliti sesuai dengan tujuan yang akan dicapai. Adapun batasan masalahnya adalah sebagai berikut :

1. Aplikasi pencarian yang dibuat untuk mencari jurnal berupa file dengan format *Portable Document Format (PDF)*.
2. Aplikasi menggunakan Metode *Levenshtein Distance* dengan pendekatan *Information Retrieval System*.
3. Input aplikasi pencarian berupa *Stopword* dalam bentuk kata, dan berupa jurnal dalam format file *PDF*.
4. Aplikasi ini dibuat berbasis *web* dengan menggunakan bahasa pemrograman *PHP* dan database *MySQL*.

1.5 Manfaat Penelitian

Manfaat yang ingin dicapai dalam implementasi tugas akhir ini adalah :

1. Pengguna dapat menemukan jurnal yang relevan berdasarkan kata kunci yang dimasukkan tidak sama persis dengan judul jurnal.

2. Diharapkan proses pencarian jurnal dengan *Information Retrieval System (IRS)* dan menerapkan metode *Levenshtein Distance* pada sistem pencarian jurnal dapat menghemat waktu.
3. Pengguna dapat membaca dan menemukan jurnal yang dibutuhkan dengan membuka aplikasi pencarian.

1.6 Sistematika Penulisan

Sistematika penulisan dari tugas akhir ini terdiri dari pokok-pokok permasalahan yang dibahas pada masing-masing yang diuraikan menjadi beberapa bagian :

BAB 1 PENDAHULUAN

Berisi latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan.

BAB 2 LANDASAN TEORI

Bab ini berisi teori-teori yang digunakan pada penelitian ini. Teori-teori yang berhubungan dengan rancang bangun, aplikasi, jurnal, *Information Retrieval System*, *Natural Language Processing (NLP)*, *Text Mining*, Metode *Levenshtein Distance*, alat bantu perancangan program, alat bantu perancangan aplikasi dan penelitian terdahulu.

BAB 3 METODOLOGI PENELITIAN

Bab ini berisi kerangka penelitian yang diusulkan dalam pengembangan sistem dengan tujuan mampu menjadi pemandu didalam pengembangan proyek, dan menyediakan solusi kepada *statement* masalah.

BAB 4 ANALISA DAN PERANCANGAN

Bab ini berisi analisa dan perancangan aplikasi pencarian jurnal berbasis *web* dengan menerapkan *Information Rretrieval System* menggunakan metode *Levenshtein Distance*.

BAB 5 IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi hasil rancangan kode program dan hasil pengujian perangkat lunak, serta analisa terhadap hasil pengujian.

BAB 6 PENUTUP

Bab ini berisi rangkuman dari hasil penelitian yang telah dilakukan dan saran-saran untuk pengembangan aplikasi atau penelitian selanjutnya.

BAB 2

LANDASAN TEORI

Bab ini berisi landasan teori sebagai parameter rujukan untuk dilaksanakannya penelitian ini. Adapun landasan teori tersebut adalah tentang rancang bangun, aplikasi, jurnal, *Information Retrieval System*, *Natural Language Processing (NLP)*, *Text Mining*, Metode *Levenshtein Distance*, alat bantu perancangan program, alat bantu perancangan aplikasi dan penelitian terdahulu.

2.1 Rancang Bangun

Kata “rancang” merupakan kata kerja dari “merancang”, yakni mengatur segala sesuatu (Sebelum bertindak, mengerjakan, atau melakukan sesuatu) atau merencanakan. Sedangkan perancangan merupakan kata benda yang memiliki arti proses, perbuatan merancang. Sedangkan “rancang bangun” dapat bermakna sebagai merancang atau mendesain suatu bangunan Pusat bahasa depdiknas dalam.

Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan. Sedangkan pengertian bangun atau pembangunan sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian [6].

2.2 Aplikasi

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan

mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya, aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user* [7].

2.3 Jurnal

Majalah ilmiah atau yang sering disebut dengan jurnal biasanya memuat hasil hasil penelitian, ungkapan tersebut di perjelas bahwa jurnal merupakan suatu terbitan yang memuat informasi, dan merupakan dokumentasi untuk kemajuan ilmu pengetahuan. Ungkapan senada juga diungkapkan oleh Sulistyio Basuki, bahwa jurnal merupakan terbitan berseri literatur sumber primer yang memuat sumber informasi ilmiah muktahir sebagai sarana informasi formal. Serta pendapat lain juga menjelaskan bahwa jurnal adalah majalah publikasi yang memuat karya tulis ilmiah yang secara nyata mengandung data dan informasi yang mengajukan iptek dan ditulis sesuai dengan kaidah-kaidah penulisan ilmiah serta diterbitkan secara berkala. Menurut Ensiklopedia Nasional Indonesia majalah ilmiah adalah suatu penerbitan khusus berisi artikel mengenai ilmu pengetahuan berdasarkan hasil penelitian yang tidak jarang mengandung uraian bersifat teknis, biasanya dikelola dan diterbitkan oleh lembaga-lembaga ilmiah, perguruan tinggi dan organisasi profesi [8].

2.4 *Information Retrieval System*

Pengambilan informasi menunjukkan proses pencarian informasi yang diperlukan. *Information retrieval (IR)* umumnya berkaitan dengan pencarian dan pengambilan informasi berbasis pengetahuan sistem *information retrieval (IR)*. Salah satu penerapan prinsip relevansi yang sejak dahulu digunakan dalam pengembangan sistem. *Information Retrieval System* menemukan informasi yang biasanya dalam bentuk dokumen dari sebuah data yang tidak terstruktur dalam bentuk teks untuk memenuhi kebutuhan informasi dari koleksi data yang sangat besar umumnya tersimpan dalam database komputer [9].

Information Retrieval terbagi dari beberapa bagian yang dijabarkan sebagai berikut [10] :

1. *Text Operations*, meliputi pemilihan katakata dalam query maupun dokumen (*term selection*) dalam proses transformasi dokumen atau *query* menjadi *term index* (indeks kata-kata).
2. *Query formulation*, memberi bobot pada indeks kata-kata *query*.
3. *Ranking*, mencari dokumen-dokumen yang relevan terhadap *query* dan mengurungkan dokumen tersebut berdasarkan kesesuaiannya dengan *query*.
4. *Indexing*, membangun basis data indeks dari koleksi dokumen dilakukan terlebih dahulu sebelum pencarian dokumen dilakukan.

2.5 *Natural Language Processing (NLP)*

Natural Language Processing (NLP) adalah kombinasi dari ilmu komputer dan bidang kecerdasan buatan yang terkait dengan linguistik. *NLP* berkaitan dengan bagaimana mesin memahami bahasa manusia untuk saling berinteraksi. Dengan adanya *NLP*, komputer dapat belajar dan memahami bahasa manusia, sehingga komputer dapat berkomunikasi dengan manusia. Bahasa manusia adalah hal yang unik karena dibuat khusus untuk menyampaikan suatu makna. Untuk membuat komputer dapat mengerti bahasa manusia adalah tugas yang sulit, karena bahasa manusia memiliki struktur yang kompleks. Selain itu, setiap bahasa memiliki keunikannya sendiri dan mungkin memiliki makna ganda. Sebagai contoh dapat dilihat dari kalimat berikut, “*Look at the dog with one eye*”, di mana kalimat tersebut dapat memiliki arti “melihat anjing dengan satu mata” atau “melihat anjing yang mempunyai mata satu” [11].

Dalam melakukan pengolahan bahasa alami, sistem membutuhkan beberapa komponen untuk menganalisis makna kalimat [12].

1. *Parser*

Sistem mengambil kata-kata yang telah di-input oleh user, lalu mengelompokkannya berdasarkan gramatikal. Dalam skripsi ini, penulis membaginya menjadi subject, verb, adv, noun, adj, conj, atau ip.

2. Sistem Representasi Pengetahuan

Sistem menganalisis kumpulan kata tersebut guna menentukan maknanya.

Dalam skripsi ini, penulis menyebut makna tersebut sebagai “*intention*”.

3. *Output Translator*

Setelah melakukan analisis, sistem memberikan balasan kepada pengguna.

Output dapat berupa bahasa alami atau output lainnya.

4. *Pattern Matching*

Pattern Matching adalah metode pencocokan pola yang digunakan dalam kecerdasan buatan. Dalam ilmu komputer, *pattern matching* digunakan untuk memeriksa urutan token yang ada di dalam suatu kalimat. Salah satu kegunaan *pattern matching* adalah untuk mencari komponen pola yang cocok, kemudian menggantinya dengan urutan token lainnya. Contoh dari *pattern matching* ialah *sequence pattern*. Dalam *sequence pattern*, pencarian pola terjadi dengan menggunakan *regular expression (regex)* dan *algoritma backtracking*.

2.6 *Text Mining*

Banyaknya informasi yang ada di dunia maya membuat upaya-upaya pengembangan terhadap penggalian informasi dari basis data daring semakin pesat, salah satunya *text mining*. *Text mining*, yang juga disebut sebagai Teks Data Mining (*TDM*) atau *Knowledge Discovery in Text (KDT)*, secara khusus dikembangkan untuk proses ekstraksi informasi dari dokumen-dokumen teks tak terstruktur (*unstructured*). *Text mining* memiliki definisi menambang data berupa teks di mana sumber data biasanya didapatkan dari dokumen dan tujuannya adalah untuk mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisis keterhubungan antar dokumen [13].

Tahapan-tahapan *Text Mining* [14] :

a. *Case Folding*

Mengubah semua huruf dalam dokumen menjadi huruf kecil (*lowercase*).

Dalam tahap ini juga karakter selain huruf dihilangkan.

b. *Tokenizing*

Memotong tiap kata dalam kalimat atau parsing dengan menggunakan spasi sebagai delimiter yang akan menghasilkan token berupa kata.

c. *Filtering*

Menyaring kata yang didapat dari proses *tokenizing* yang dianggap tidak penting atau tidak memiliki makna dalam proses *text mining* yang disebut *stoplist*. Tiap kata yang diperoleh dari *tokenizing* akan dicocokkan dalam kamus *stopword* di dalam database, jika kata tersebut cocok dengan salah satu kata dalam *stopword* maka kata tersebut akan dihilangkan, sementara yang tidak cocok akan dianggap cocok dan diproses ke tahap selanjutnya.

d. *Stemming*

Mengembalikan kata-kata yang diperoleh dari hasil *filtering* ke bentuk dasarnya, menghilangkan imbuhan awal (*prefix*) dan imbuhan akhir (*suffix*) sehingga di dapat kata dasar.

e. *Tagging*

Merubah kata dalam bentuk lampau (*past tense*) menjadi bentuk sekarang (*future tense*).

f. *Analyzing*

Keterhubungan antar kata dalam dokumen akan ditentukan dengan menghitung frekuensi term pada dokumen atau lebih sering dikenal dengan tahap pembobotan.

2.7 *Metode Levenshtein Distance*

Levenshtein Distance atau yang biasa disebut dengan *edit distance* adalah suatu metode yang dapat digunakan untuk mengatasi terjadinya kesalahan ejaan. Kesalahan ejaan terjadi apabila kata yang diketik oleh pengguna tidak terdapat pada daftar kamus Bahasa Indonesia. Fungsi metode *Levenshtein Distance* yaitu untuk menghitung jarak kedekatan dari dua buah string melalui penambahan karakter, pengubahan karakter, dan penghapusan karakter hingga kedua string tersebut cocok [3].

Pada algoritma ini terdapat 3 macam operasi karakter yaitu [15] :

1. Operasi pengubahan

Operasi ini merupakan operasi yang digunakan untuk mengubah suatu karakter dengan karakter lain. Contoh terdapat kata 'dalsm' kata tersebut akan diubah menjadi kata 'dalam' maka karakter 's' akan diganti dengan karakter 'a'.

2. Operasi penambahan

Operasi ini merupakan operasi untuk menambahkan suatu karakter pada suatu kata. Contoh operasi ini adalah ketika terdapat kata 'kpan' yang akan diubah menjadi kata 'kapan' maka akan dilakukan penambahan karakter 'a' setelah karakter 'k'.

3. Operasi penghapusan

Operasi penghapusan karakter merupakan operasi yang digunakan untuk menghapus suatu karakter pada kata. Contoh dari operasi ini adalah ketika terdapat kata 'glass' yang akan diubah menjadi kata 'glas', maka karakter 's' di akhir akan dihilangkan.

Berikut ini adalah langkah-langkah algoritma *Levenshtein Distance* dalam mendapatkan nilai *distance* [16] :

Langkah 1 : Inisialisasi

- a) Hitung panjang S dan T, misalkan m dan n.
- b) Buat matriks berukuran 0...m baris dan 0...n kolom.
- c) Inisialisasi baris pertama dengan 0...n.
- d) Inisialisasi kolom pertama dengan 0...m.

Langkah 2 : Proses

- a) Periksa $S[i]$ untuk $1 < i < n$.
- b) Periksa $T[j]$ untuk $1 < j < m$.
- c) Jika $S[i] = T[j]$, maka entrinya adalah nilai yang terletak pada tepat didiagonal atas sebelah kiri, yaitu $d[i,j] = d[i-1, j-1]$.
- d) Jika $S[i] \neq T[j]$, maka entrinya adalah $d[i,j]$ minimum dari :
 - Nilai yang terletak tepat di atasnya, ditambah satu, yaitu $d[i,j-1]+1$.
 - Nilai yang terletak tepat dikirinya, ditambah satu, yaitu $d[i-1,j]+1$.
 - Terletak pada tepat didiagonal atas sebelah kirinya, ditambah satu, yaitu $d[i-1,j-1]+1$.

Langkah 3 : Hasil entri matriks pada baris ke-i dan kolom ke j, yaitu $d[i,j]$.

Langkah 4 : Diulang hingga entri $d[m,n]$ ditemukan.

2.8 Website

Pengertian *Website* adalah ”kumpulan dari halaman *web* yang sudah dipublikasikan di jaringan internet dan memiliki domain/*URL (Uniform Resource Locator)* yang dapat diakses semua pengguna internet dengan cara mengetikkan alamatnya. Hal ini dimungkinkan dengan adanya teknologi *World Wide Web (WWW)*. Halaman *website* biasanya berupa dokumen yang ditulis dalam format *Hyper Text Markup Language (HTML)*, yang bisa diakses melalui *HTTP, HTTPS* adalah suatu protokol yang menyampaikan berbagai informasi dari server *website* untuk ditampilkan kepada para user atau pemakai melalui *web browser* [17].

2.9 Alat Bantu Perancangan Aplikasi

2.9.1 Flowchart

Flowchart adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan *flowchart* akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah, disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek. *Flowchart* membantu memahami urutan-urutan logika yang rumit dan panjang. *Flowchart* membantu mengkomunikasikan jalannya program ke orang lain (bukan pemrogram) akan lebih mudah [18].

2.9.2 Context Diagram

Context Diagram adalah gambaran umum tentang suatu sistem yang terdapat didalam suatu organisasi yang memperlihatkan batasan (*boundary*) sistem, adanya interaksi antara eksternal *entity* dengan suatu sistem dan informasi

secara umum mengalir diantara *entity* dan sistem. *Context Diagram* merupakan alat bantu yang digunakan dalam menganalisa sistem yang akan dikembangkan. Simbol-simbol yang digunakan di dalam *Context Diagram* hampir sama dengan simbol-simbol yang ada pada *DFD*, hanya saja pada *Context Diagram* tidak terdapat simbol file [19].

2.9.3 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) merupakan gambaran sistem secara logika yang tidak tergantung pada perangkat keras, lunak, struktur data dan organisasi file. Keuntungan dari *DFD* adalah untuk memudahkan pemakai yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan atau dikembangkan [20].

2.9.4 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah suatu cara untuk menjelaskan kepada para pemakai tentang hubungan antar data dalam basis data secara logis dengan persepsi bahwa *real world* terdiri dari objek-objek dasar yang saling berhubungan dengan cara memvisualisasikan ke dalam bentuk simbol-simbol grafis [21].

2.10 Alat Bantu Pembuatan Aplikasi

2.10.1 Basis Data (Database)

Beberapa defenisi tentang database dari beberapa orang ahli, database adalah sebagai berikut [22]:

- a. Database adalah sekumpulan data *store* yang tersimpan dalam *magnetic disck*, *optical disck*, *magnetic drum* atau media penyimpanan sekunder lainnya.
- b. Database adalah sekumpulan program- program aplikasi umum yang mengeksekusi dan memproses data secara umum seperti pencarian data, peremajaan data, penambahan dan penghapusan data. Database terdiri dari data yang akan digunakan atau diperuntukkan terhadap *user*, dimana masing-masing *user* akan menggunakan data tersebut sesuai dengan tugas dan fungsinya, dan *user* lain juga dapat menggunakan data tersebut dalam waktu yang bersamaan.

2.10.2 My Structure Query Language (MySQL)

MySQL adalah suatu perangkat lunak database relasi (*Relation Database Management Sistem/RDMS*) seperti halnya Oracle, PostgreSQL, Microsoft SQL. *MySQL* jangan sama diartikan dengan *SQL (Structure Query Language)* yang didefinisikan sebagai sintaks perintah-perintah tertentu dalam bahasa (program) yang digunakan untuk mengelola suatu database. Kelebihan *MySQL* [23] :

1. *MySQL* Merupakan sebuah database yang mampu menyimpan data berkapasitas sangat besar hingga berukuran *gigabyte* sekalipun.
2. *MySQL* didukung oleh sever ODBC, artinya database *MySQL* dapat di akses membangun aplikasi apa saja termasuk berupa visual JavaNeatbeans.
3. *MySQL* adalah database dapat menggunakan enkripsi password

4. *MySQL* merupakan server database multi user artinya database ini dapat digunakan oleh banyak orang.
5. *MySQL* dapat menciptakan lebih dari 16 kunci per table dan satu kunci memungkinkan belasan *fields*.

2.10.3 *Hypertext Markup Language (HTML)*

HTML adalah Bahasa pemrograman yang fleksibel di mana kita bisa meletakkan scrip dari bahasa pemrograman lain seperti JAVA, Visual Basic, C dan lain-lain. Jika *HTML* tersebut tidak dapat mendukung suatu perintah pemrograman tertentu. Browser tidak akan menampilkan kotak dialog "Syntax Error" jika terdapat penulisan kode yang keliru pada scrip *HTML* sepanjang kode-kode yang kita tuliskan merupakan kode-kode *HTML* tanpa penambahan kode-kode dari luar seperti java. Oleh karena itu, jika terjadi *syntax error* pada skrip *HTML*, efek yang paling jelas adalah *HTML* tersebut tak akan ditampilkan pada halaman jendela browser [24].

2.10.4 *Hypertext Preprocessor (PHP)*

PHP adalah singkatan dari "*PHP: Hypertext Preprocessor*", yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs *web* dan bisa digunakan bersamaan dengan *HTML*. *PHP* diciptakan oleh Rasmus Lerdorf pertama kali tahun 1994. Pada awalnya *PHP* adalah singkatan dari "*Personal Home Page Tools*". Selanjutnya diganti menjadi FI ("*Forms Interpreter*"). Sejak versi 3.0, nama bahasa ini diubah menjadi "*PHP: Hypertext Preprocessor*" dengan singkatannya "*PHP*". *PHP* versi terbaru adalah versi ke-5. Berdasarkan survey Netcraft pada bulan Desember

1999, lebih dari sejuta site menggunakan PHP, di antaranya adalah NASA, Mitsubishi, dan RedHat [25].

Beberapa kelebihan *PHP* dari bahasa pemrograman *web*, antara lain [26] :

1. Bahasa pemrograman *PHP* adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *PHP* memiliki tingkat akses yang lebih cepat.
3. *PHP* memiliki tingkat *lifecycle* yang cepat sehingga selalu mengikuti perkembangan teknologi internet.
4. *PHP* juga mendukung akses ke beberapa database yang sudah ada baik yang bersifat *free/gratis* ataupun komersial. Database itu antara lain : *MySQL*, *PostgreSQL*, *Informix*, dan *MicrosoftSQL Server*. Web server yang mendukung *PHP* dapat ditemukan dimana mana dari mulai *Apache*, *IIS*, *AOserver*, *phhttp*, *Fhttp*, *PWS*, *Lighttpd* hingga *Xitami* dengan konfigurasi yang relative mudah.

2.10.5 JavaScript

JavaScript adalah bahasa yang berbentuk kumpulan skrip berjalan pada suatu dokumen *HTML*. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap *HTML* dengan mengizinkan pengekseskuan perintah-perintah disisi user variabel atau fungsi dengan nama *TEST* berbeda dengan variabel dengan nama *test* dan setiap instruksi diakhiri dengan artinya disisi *browser* bukan disisi *server web*. *JavaScript* adalah bahasa yang “*case sensitive*” artinya memnedakan penamaan variabel dan fungsi yang menggunakan huruf besar dan huruf kecil, contoh karakter titik koma [26].

2.10.6 Cascading Style Sheets (CSS)

CSS adalah kependekan dari *Cascading Style Sheet*. *CSS* merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan/layout halaman *web* supaya lebih elegan dan menarik. *CSS* adalah sebuah teknologi internet yang direkomendasikan oleh *World Wide Web Consortium* atau W3C pada tahun 1996. Awalnya, *CSS* dikembangkan di SGML pada tahun 1970, dan terus dikembangkan hingga saat ini. *CSS* telah mendukung banyak bahasa markup seperti *HTML*, *XHTML*, *XML*, *SVG* (*Scalable Vector Graphics*) dan *Mozilla XUL* (*XML User Interface Language*) [27].

2.10.7 Bootstrap

Bootstrap adalah sebuah framework untuk *CSS* dan berupa produk *open source* yang dibuat oleh Mark Otto dan Jacob Thornton. Pada awalnya *Bootstrap* ini dibuat untuk membuat standarisasi front end untuk semua programmer di perusahaannya. *Bootstrap* telah berubah dari yang sebelumnya adalah *CSS-Driven* proyek ke sebuah host dari *JavaScript plugins* dan ikon yang dapat dengan mudah digunakan untuk formulir dan tombol [28].

2.10.8 XAMPP

XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *XAMPP* merupakan perangkat lunak bebas, yang mendukung banyak sistem operasi, dan merupakan kompilasi dari beberapa program. Seperti Apache, *MYSQL*, *PHPP*, dan Perl. *XAMPP* adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket. Dalam paket *XAMPP* sudah terdapat

Apache (*Web Server*), *MySql (Database)*, *PHP (server side scripting)*, *Perl*, *FTP server*, *PhpMyAdmin*, dan berbagai pustaka bantu lainnya [29].

Kepanjangan dari *XAMPP* yaitu :

- X : Program ini dapat dijalankan dibanyak sistem operasi, seperti *Windows*, *Linux*, *Mac OS* dan juga *Solaris*.
- A : Apache merupakan aplikasi *web server*. Tugas utama dari Apache adalah menghasilkan halaman *web* yang benar kepada *user* berdasarkan kode *PHP* yang dituliskan oleh pembuat *web* atau *user*.
- M : *MySql*, merupakan aplikasi data server. Perkembangannya disebut juga *SQL* yang merupakan kepanjangan dari *Structured Query Language*. *SQL* merupakan bahasa terstruktur yang digunakan untuk mengolah database.
- P : *PHP*, merupakan bahasa pemrograman *web*, dimana *user* dapat menggunakan bahasa pemrograman ini untuk membuat *web* yang bersifat *server-side scripting*.
- P : Perl, yaitu merupakan bahasa pemrograman untuk segala keperluan, dan dikembangkan pertama kali oleh Larry Wall di mesin Unix.

2.10.9 Sublime Text Editor

Sublime Text Editor adalah editor teks untuk berbagai bahasa pemrograman termasuk pemrograman *PHP*. *Sublime Text Editor* merupakan *editor text lintas-platform* dengan *Python Application Programming Interface (API)*. *Sublime Text Editor* juga mendukung banyak bahasa pemrograman dan bahasa *markup*, dan fungsinya dapat ditambah dengan *plugin*, dan *Sublime Text Editor* tanpa lisensi perangkat lunak. *Sublime Text Editor* pertama kali dirilis pada tanggal 18

Januari 2008, sekarang versi *Sublime Text Editor* sudah mencapai versi 3 yang dirilis pada tanggal 29 Januari 2013. *Sublime Text Editor* mendukung *operation system* seperti *linux*, *Mac Os X*, dan juga *windows*. Sangat banyak fitur yang tersedia pada *Sublime Text Editor* diantaranya minimap, memuka *script* secara *side by side*, *bracket highlight* sehingga tidak bingung mencari pasangannya, kode snippets, drag and drop direktori ke sidebar terasa mirip dengan *TextMate* untuk *Mac OS* [30].

2.10.10 Web Browser

Web browser merupakan suatu program komputer yang dirancang untuk menerjemahkan kode-kode perintah *HTML* menjadi sebuah tampilan *web* di komputer yang berupa teks, gambar dan multimedia yang bisa dilihat langsung dan dinikmati secara langsung, dengan tampilan yang dihasilkan dapat dibaca dan dimengerti oleh orang awan sekalipun [31].

2.11 Penelitian Terdahulu

Berikut ini merupakan penelitian terdahulu yang dapat menjadi acuan pada tugas akhir :

Tabel 2.1 Penelitian Terdahulu

No	Penulis dan Tahun	Judul	Metode	Hasil
1.	Isbalaikana Larasati dan Fitri Marisa (2019)	Penerapan Algoritma <i>Levenstein Distance</i> Dalam Aplikasi Pengolahan Surat	<i>Levenstein Distance</i>	1. Dapat membantu dalam mempermudah proses pencarian arsip surat pada aplikasi pengolahan surat di CV. Khasanah Konsultama. 2. Metode algoritma <i>levenshtein distance</i> menggunakan matrik perbandingan dengan rumus

				[j+1], [i+1], [+edit]. 3. Dari hasil perhitungan dengan menggunakan metode algoritma <i>levenshtein distance</i> memiliki hasil perbandingan kata yang benar akan ditampilkan.
2.	Aprillianda Pasaribu, Marto Sihombing dan Relita Buaton (2020)	Perancangan Sistem Pendeteksi Berita Hoax Menggunakan Algoritma <i>Levenstein Distance</i> Berbasis <i>PHP</i>	<i>Levenstein Distance</i>	Sistem yang dibangun dapat melakukan penerapan algoritma TF-IDF dan algoritma <i>levenshtein distance</i> yang mendeteksi berita hoaks dan menghasilkan nilai keakurasian hasil deteksi berita.
3.	Welly Kuswanto (2020)	Implementasi Algoritma <i>Levenstein Distance</i> Dengan <i>Restful Web Service</i> Pada Kata Bahasa Indonesia Ke Bahasa Jawa Berbasis <i>Web</i>	<i>Levenstein Distance</i>	1. Pengujian dilakukan terhadap dua belas <i>query</i> , didapatkan hasil persentase nilai <i>precision</i> sebesar 95,32%, nilai <i>recall</i> 100%, nilai <i>accuracy</i> 99,97%, dan nilai <i>f-measure</i> 97,51%. 2. Penggunaan metode <i>Levenshtein Distance</i> pada <i>query</i> sebanyak dua belas <i>query</i> berdasarkan input sumber dan <i>output target</i> dikatakan efektif karena mendapatkan persentase yang cukup tinggi. 3. Tidak terdapat banyak kesalahan pada proses, sehingga secara fungsional dapat menampilkan hasil yang sesuai harapan.
4.	M. Julianda (2020)	Sistem Informasi Pendeteksi Kemiripan Isi Teks Dokumen Sebagai Acuan Penentuan Kelayakan Judul Skripsi	<i>Levenstein Distance</i>	1. Dengan adanya system informasi pendeteksi kemiripan teks ini, kaprodi akan dimudahkan khususnya dalam proses penyeleksian, dimana proses penyeleksian akan

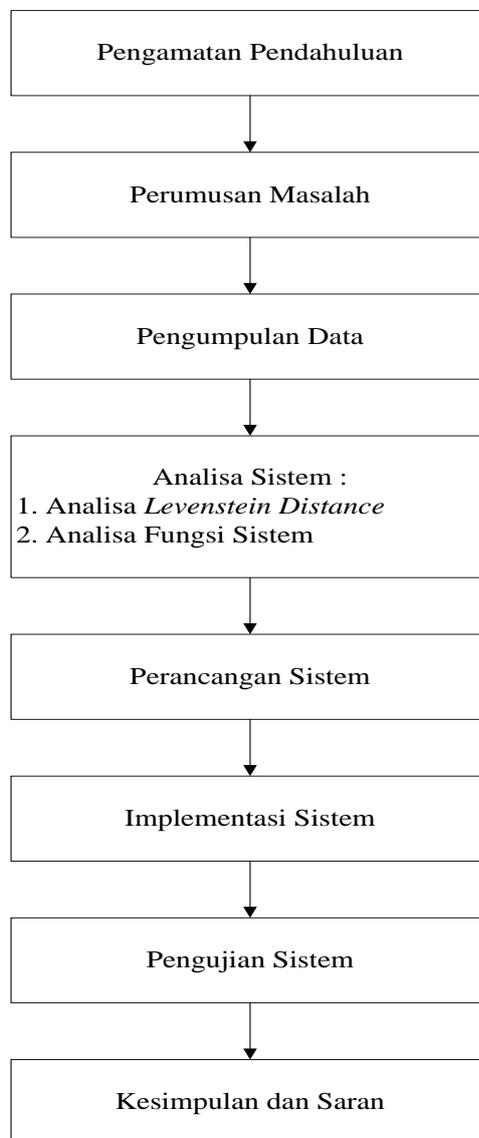
		Menggunakan Metode <i>Levenstein Distance</i>		menjadi lebih cepat efisien. 2. Dengan adanya sistem informasi pendeteksi kemiripan teks ini, mahasiswa juga akan diuntungkan dengan cepatnya mereka mendapatkan informasi hasil seleksi proposal judul yang mereka ajukan, dan dengan demikian maka mahasiswa juga akan cepat untuk menyelesaikan skripsinya. 3. Dengan adanya sistem informasi pendeteksi kemiripan teks ini, akan dapat mengurangi tindak penjiplakan atau plagiarisme yang dilakukan mahasiswa saat mengajukan proses pengajuan proposal judul.
5.	Bei Harira Irawan, Manase Sahat H Simarankir dan Erlinna (2021)	Deteksi Kemiripan Judul Skripsi Menggunakan Algoritma <i>Levenstein Distance</i> Pada Kampus STMIK MIC Cikarang	<i>Levenstein Distance</i>	Dari hasil sampel pengajuan sebuah judul skripsi untuk perhitungan multi target dengan pengujian antara data sumber dengan 6 kategori data target dalam database dan penentuan threshold ≥ 4 serta penentuan bobot similarity $>25\%$ dinyatakan mirip untuk ditampilkan, maka disimpulkan data yang ditampilkan adalah dari kategori judul perancangan system dengan distance 4 dan kemiripan 42,85% dan dari kategori aplikasi berbasis web dengan distance 5 dan kemiripan 28,5%. Sedangkan 4 kategori lain memiliki distance lebih dari 5 dan

				<p>persen bobot similarity kurang dari 25% sehingga tidak dianggap mirip dan tidak ditampilkan. Untuk mendapatkan hasil lebih akurat, disarankan ada penelitian terkait untuk melanjutkan penelitian ini dengan pemberian nilai threshold lebih rendah dan penentuan bobot similarity lebih besar. Dapat juga dikembangkan dengan penambahan proses stemming untuk membuang kata-kata yang tidak diperlukan.</p>
--	--	--	--	--

BAB 3

METODOLOGI PENELITIAN

Penelitian ini dilakukan dengan melaksanakan tahapan demi tahapan yang berhubungan. Tahapan-tahapan tersebut dijabarkan dalam metode penelitian. Metode penelitian diuraikan kedalam bentuk skema yang jelas, teratur, dan sistematis. Berikut tahapan-tahapan penelitian dapat dilihat pada gambar 3.1:



Gambar 3.1 Tahapan Metodologi Penelitian

Penjelasan dari tahapan – tahapan penelitian pada gambar 3.1 dapat dilihat pada penjelasan di bawah ini :

3.1 Pengamatan Pendahuluan

Pengamatan pendahuluan merupakan tahapan awal yang dilakukan dalam penelitian ini bertujuan untuk mengamati penelitian yang menggunakan metode *Levenstein Distance* yang dijadikan sebagai penelitian studi pustaka dalam penelitian tugas akhir ini. Hasil dari pengamatan pendahuluan ini berupa penelitian sebelumnya yang melakukan penelitian terkait dengan metode *Levenstein Distance*.

3.2 Perumusan Masalah

Berdasarkan hasil dari tahapan pendahuluan sebelumnya, maka tahapan selanjutnya adalah tahapan perumusan masalah. Pada tahapan perumusan masalah akan dirumuskan masalah yang dianggap sebagai penelitian dalam tugas akhir ini. Permasalahan – permasalahan yang menjadi rumusan masalah dalam penelitian ini didapatkan dari penelitian terkait data pengamatan pendahuluan sebelumnya. Solusi yang didapatkan pada tahapan perumusan masalah ini yang akan menjadi judul penelitian tugas akhir ini “Rancang Bangun Aplikasi Pencarian Jurnal Dengan Menerapkan *Information Rretrieval System* Menggunakan Metode *Levenstein Distance*” .

3.3 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang berhubungan dengan penelitian dan pembuatan sistem, yaitu dengan :

3.3.1 Observasi (Pengamatan)

Pengamatan secara langsung pada aplikasi pencarian jurnal yang telah ada sebelumnya.

3.3.2 Studi Kepustakaan

Studi kepustakaan dilakukan dengan cara mempelajari buku-buku, jurnal-jurnal dan artikel-artikel di internet yang berhubungan dengan permasalahan yang dibahas.

3.4 Analisa Sistem

Tahapan selanjutnya adalah melakukan analisa metode sistem dari penelitian Tugas Akhir ini. Adapun tahapan analisa dalam penelitian ini adalah sebagai berikut :

3.4.1 Analisa Metode Levenstein Distance

Tahap ini adalah proses dimana langkah-langkah pengolahan data jurnal menerapkan *Information Rretrieval System* menggunakan metode *Levenstein Distance*.

3.4.2 Analisa Fungsi Sistem

Setelah melakukan tahapan analisa terhadap metode *Levenstein Distance* maka selanjutnya adalah analisa fungsional sistem yang akan dibangun. Adapun tahapan – tahapan analisa fungsional yaitu dalam pembuatan *flowchart*, *context diagram*, *data flow diagram* (DFD), *entity relationship diagram* (ERD), dan perancangan *user interface*.

3.5 Perancangan Sistem

Setelah tahapan analisa selesai dilakukan, maka tahapan selanjutnya adalah perancangan sistem. Tahapan perancangan sistem terdiri dari :

1. Perancangan struktur menu yang akan digunakan pada sistem yang akan dibangun.
2. Tahapan rancangan database beserta atribut yang dibutuhkan.
3. Tahapan perancangan *user interface* atau antar muka pengguna terhadap sistem yang akan digunakan.

3.6 Implementasi Sistem

Implementasi sistem merupakan suatu konversi dari desain aplikasi yang telah dirancang kedalam sebuah program komputer dengan berbasis *web* dengan menggunakan bahasa pemrograman *PHP* dan database *MySQL*.

3.7 Pengujian Sistem

Pengujian (*testing*) yaitu uji coba yang dilakukan terhadap sistem yang dibangun apakah telah sesuai dengan yang diharapkan atau tidak. Pengujian yang dilakukan terdiri dari:

1. Pengujian *blackbox*, digunakan untuk menguji tingkat kemampuan *user interface* terhadap sistem yang dibangun.
2. Pengujian *User Acceptance Test (UAT)*.

3.8 Kesimpulan dan Saran

Tahapan terakhir adalah menarik kesimpulan dari hasil penelitian yang didapatkan dalam aplikasi pencarian jurnal berbasis *web* dengan menerapkan

Information Rretrieval System menggunakan metode *Levenstein Distance*. Pada tahapan ini juga berisikan saran penelitian bagi pembaca untuk melakukan pengembangan terhadap penelitian ini kedepannya.