# BAB I PENDAHULUAN

## Latar Belakang

Plagiarisme adalah salah satu tindak kejahatan yang sering terjadi di dunia akademik. Plagiarisme dapat terjadi baik sengaja maupun tidak sengaja. Menurut Rito (Pratama dkk., 2019), plagiarisme diklasifikasikan sebagai tindak kejahatan dengan mengambil gagasan, mengambil hasil riset, mengakui hasil riset, ataupun merangkum karya tulis tanpa menyebutkan sumbernya. Tindakan plagiarisme menjadi lebih mudah seiring berkembangnya teknologi yang mempermudah akses informasi. Terutama akses informasi jurnal daring yang memungkinkan dokumen digital diunduh dimana saja dan kapan saja. Kemudahan inilah yang mendorong terjadinya plagiasi, apalagi dokumen digital dapat disalin dan diedit menggunakan komputer. Beberapa tahun terakhir berbagai sistem deteksi plagiarisme telah dibangun untuk menangani permasalahan ini. Pada umumnya sistem beroperasi dengan tiga tahapan: (1) melakukan ekstraksi teks dari dokumen, (2) mencari teks atau bagian dokumen yang mirip, dan (3) menghitung skor plagiarisme. Hasil penelitian yang dilakukan Indriyanto (Indriyanto & Sumitra, 2019) menyimpulkan bahwa karya ilmiah dalam bahasa Indonesia dapat dideteksi plagiarismenya menggunakan vector space model dan algoritma cosine similarity.

Meskipun begitu berbagai trik telah dilakukan untuk menghindari proses deteksi plagiarisme. Upaya dasar yang dilakukan adalah dengan menyerang proses pertama yaitu ekstraksi teks dari dokumen. Menurut penelitian James (Heather, 2010), jika teks tidak dapat diekstrak dari dokumen dengan benar tanpa mempengaruhi tampilannya, maka sistem plagiarisme tidak dapat mengidentifikasi bagian plagiat dari dokumen. Misalnya dengan menggunakan font yang telah dimodifikasi sehingga karakter yang tampak visual tidak berkaitan dengan karakter sebenarnya. Hal ini menyebabkan sistem deteksi plagiarisme tidak dapat melakukan proses ekstraksi teks dengan benar. Selain itu terdapat berbagai trik lain seperti mengubah teks menjadi berwarna putih, mengonversi teks paragraf menjadi gambar, menyembunyikan teks dibalik gambar, melakukan kesalahan ketik dengan sengaja, mengubah teks menjadi bézier curve, dll.

Cara untuk mencurangi sistem plagiarisme terus berkembang. Sehingga diperlukan solusi untuk menyelesaikan permasalahan tersebut dengan cara mendeteksi atau menghindari masing-masing trik. Pada penelitian ini, cara curang yang akan diteliti adalah penggunaan font dengan glyph yang telah dipetakan ulang. Misalnya kata “membaca” ketika di-parsing menjadi kata “mxmbaca” karena huruf “x” memiliki huruf tampak visual “e”.

Oleh karena itu, penelitian ini mencoba menerapkan metode berbasis computer vision dengan menggunakan optical character recognition (OCR) untuk menyelesaikan masalah tersebut. Penggunaan OCR bertujuan untuk mencoba menafsirkan dokumen sama seperti bagaimana manusia membaca (Heather, 2010). Sehingga data tampak visual dapat dibandingkan dengan data hasil parsing. Selain itu penelitian ini juga akan menguji metode tersebut berdasarkan dokumen hasil ciptaan otomatis secara acak. Nilai metrik yang akan digunakan nilai akurasi, presisi, recall, dan f1-score. Nilai-nilai metrik tersebut dihitung berdasarkan tabel confusion matrix.

## Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka rumusan masalah yang akan diangkat pada penelitian ini antara lain:

1. Bagaimana cara mendeteksi penggunaan font modifikasi dengan glyph yang telah dipetakan ulang menggunakan optical character recognition.
2. Bagaimana performa metode berbasis optical character recognition dalam mendeteksi penggunaan font yang telah dimodifikasi.

## Tujuan Penelitian

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka tujuan dari penelitian ini antara lain:

1. Untuk melakukan deteksi penggunaan font modifikasi dari dokumen sebelum masuk ke dalam sistem deteksi plagiarisme.
2. Untuk mengetahui kinerja dan performa penerapan optical character recognition dalam mendeteksi penggunaan font modifikasi.

## Batasan Masalah

Dalam penelitian ini, ruang lingkup permasalahan dibatasi oleh batasan sebagai berikut:

1. Berkas dokumen yang menjadi alat pengujian dalam format PDF dan berbahasa Indonesia.
2. Dokumen skripsi dan tugas akhir memiliki topik atau berkategori electrical engineering and system science, dan computer science.
3. Dokumen pengujian dibuat secara acak dan otomatis.
4. Font yang digunakan adalah “Times New Roman” dengan rentang karakter alfanumerik.
5. Penelitian terbatas pada proses deteksi atau identifikasi manipulasi sehingga tidak termasuk proses pemulihan dokumen ke kondisi awal.

## Manfaat Penelitian

Manfaat dari penelitian ini diharapkan mampu membantu sistem deteksi plagiarisme dalam mengatasi permasalahan manipulasi dari dokumen itu sendiri. Sehingga sistem plagiarisme dapat berfungsi secara maksimal tanpa manipulasi dokumen secara visual.

## Sistematika Penulisan

Sistematika penulisan yang disusun untuk penelitian ini dijelaskan sebagai berikut:

**BAB I: PENDAHULUAN**

Bab ini berisikan latar belakang, rumusan masalah, tujuan dan manfaat penelitian, serta batasan masalah dari sistem deteksi manipulasi pra-plagiarisme berbasis optical character recognition.

**BAB II: KAJIAN PUSTAKA**

Bab ini menjelaskan pengertian serta konsep dasar tentang variabel dari topik yang diangkat. Selain itu menjelaskan teknologi yang digunakan untuk membentuk sistem pra-plagiarisme.

**BAB III: METODE PENELITIAN**

Bab ini menjelaskan tentang metode yang digunakan untuk mendeteksi manipulasi pra-plagiarisme, karakteristik jenis manipulasi, serta penjelasan lebih lanjut dari metode tersebut.

**BAB IV: HASIL DAN PEMBAHASAN**

Bab ini menjelaskan tentang analisis dari hasil serta pembahasan hasil eksekusi sistem pra-plagiarisme. Selain itu menjelaskan bagaimana hasil dari metode yang digunakan oleh sistem.

**BAB V: KESIMPULAN**

Bab ini berisikan penutup penelitian yang menjelaskan mengenai kesimpulan dan saran dari hasil penelitian.

# 

# BAB II KAJIAN PUSTAKA

## Sistem Deteksi Plagiarisme

Tindakan plagiarisme menjadi ancaman serius terhadap integritas akademik (Meo & Talha, 2019). Plagiarisme merupakan tindakan mengambil gagasan, mengambil hasil riset, mengakuisisi hasil riset, dan meringkas suatu tulisan tanpa menyebutkan sumbernya (Pratama dkk., 2019). Sehingga hasil plagiarisme cenderung memiliki kemiripan dengan naskah aslinya. Sebuah perangkat lunak dapat dikembangkan untuk menganalisis kemiripan naskah tersebut. Saat ini telah terdapat banyak penelitian mengenai sistem plagiarisme (Hambi & Benabbou, 2020). Pada umumnya sistem plagiarisme bekerja dengan membandingkan dan mengukur kemiripan dokumen baru dengan dokumen yang ada pada database. Semakin mirip suatu dokumen, maka semakin besar kemungkinan bahwa dokumen tersebut merupakan hasil plagiarisme.

Penelitian yang dilakukan oleh Ristu (Saptono dkk., 2018) menggunakan cosine similarity untuk melakukan deteksi plagiarisme. Metode yang digunakan adalah merepresentasikan dokumen menjadi vektor di dalam ruang vektor. Sehingga kemiripan antar dokumen dapat dihitung berdasarkan vektornya menggunakan cosine similarity. Data yang digunakan berasal dari skripsi mahasiswa Universitas Sebelas Maret (UNS) yang dikoleksi dari repositori perpustakaan pusat UNS. Selanjutnya dilakukan proses parsing berkas pdf untuk mendapatkan isi teks dari dokumen jurnal. Tahap preprocessing yang dilakukan pemecahan paragraf, case folding, tokenizing, filtering, dan stemming. Tahap ekstraksi fitur menggunakan pembobotan TF-IDF yang menghitung frekuensi kemunculan suatu kata. Terakhir adalah proses perhitungan kemiripan menggunakan cosine similarity. Hasil dari penelitian ini menunjukkan bahwa metode pemodelan ruang vektor dan cosine similarity dapat digunakan dengan nilai rata-rata precision sebesar 54,37% dan nilai rata-rata recall sebesar 100%.

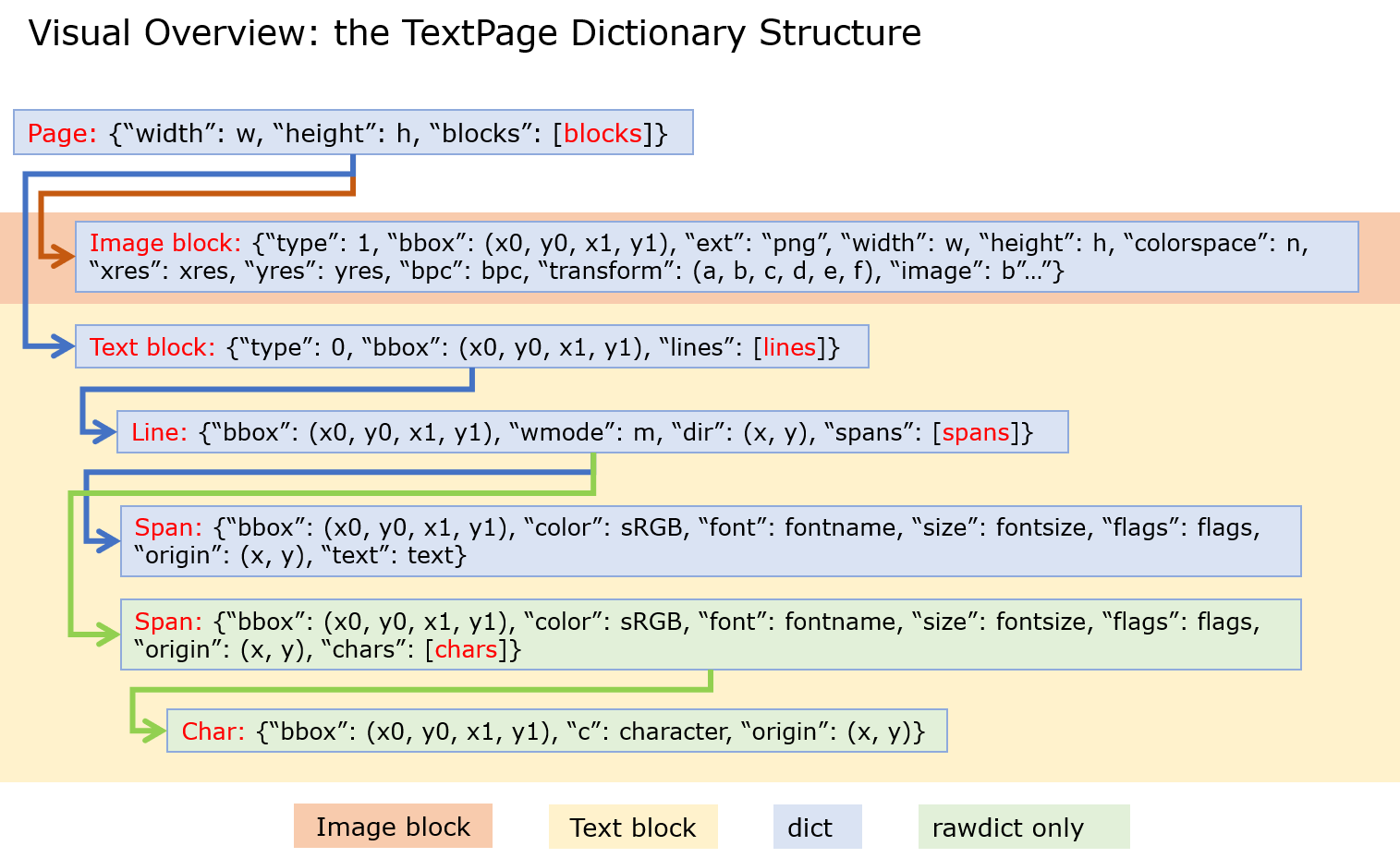
Perangkat lunak komersial yang sering digunakan untuk mendeteksi plagiarisme adalah Turnitin. Alat ini telah digunakan oleh institusi akademik di seluruh dunia. Turnitin dikembangkan sejak tahun 1998 dan menjual lisensi kepada berbagai institusi akademik untuk menggunakan layanannya. Misi Turnitin adalah untuk memastikan integritas pendidikan global dan meningkatkan hasil pembelajaran secara bermakna[[1]](#footnote-1). Turnitin memiliki cara kerja yang hampir sama yaitu mengecek konten dari dokumen dengan konten dokumen original yang telah ada. Penelitian yang dilakukan Sultan (Meo & Talha, 2019) menyimpulkan bahwa Turnitin bukanlah alat deteksi plagiarisme, namun merupakan alat pencocokan teks yang memberikan skor similarity dengan karya yang sudah diterbitkan.

## Ekstraksi Dokumen PDF

Pada masa kini berbagai karya ilmiah dibagikan secara gratis maupun berbayar dalam format PDF. Portable Document Format atau disingkat PDF memiliki berbagai keunggulan salah satunya adalah sifatnya yang fleksibel sehingga dapat dibuka di perangkat apapun. Sebuah berkas PDF merupakan gabungan dari beberapa obyek dasar seperti teks, vektor grafik, dan gambar (Gemelli dkk., 2022). Berbagai library high level dapat digunakan untuk mengekstrak informasi tersebut. Sedangkan informasi mengenai lokasi berbagai elemen, embedded font, dan metadata lainnya membutuhkan library low level.

Pada penelitian ini library yang akan digunakan PyMuPDF karena memiliki performa yang sangat cepat (Dong dkk., 2021), dokumentasi yang lengkap (Gemelli dkk., 2022), dan akurasi yang tinggi. Library tersebut akan digunakan untuk melakukan ekstraksi embedded font yang ada di dalam berkas PDF. Selain itu library ini juga akan digunakan untuk mencari lokasi teks atau karakter yang menggunakan font tersebut. Sehingga di akhir proses deteksi dapat diketahui dimana letak manipulasinya secara akurat.

Struktur hasil parsing library PyMuPDF dapat dilihat Gambar 2.1 dimana setiap dokumen PDF terdiri dari beberapa page/halaman. Setiap page terdiri atas kumpulan image block dan text block. Image block memiliki berbagai informasi mengenai lokasi, ekstensi, ukuran dll. Sedangkan text block masih terdiri atas struktur bersarang antara lain line/baris, span, dan char. Setiap struktur yang ada di dalam text block juga memiliki informasi metadata terutama lokasi bounding box dan nama font yang digunakan.

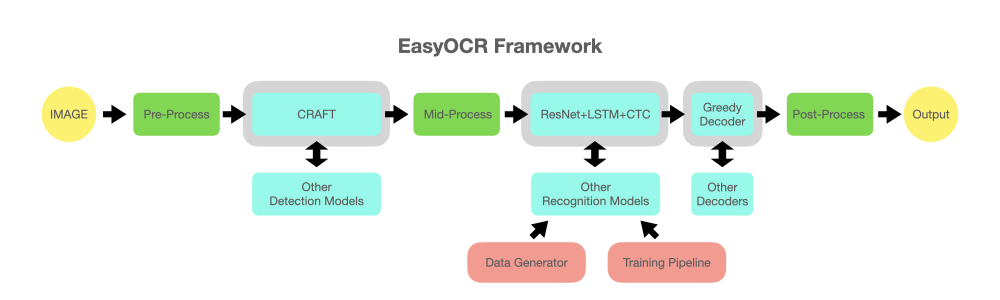


Gambar . Struktur Hasil Parsing PyMuPDF[[2]](#footnote-2)

## Optical Character Recognition

Optical character recognition atau disingkat OCR adalah sebuah sistem yang mengenali teks dalam bentuk citra gambar menjadi teks dalam bentuk machine-encoded format (Memon dkk., 2020). Data citra tersebut dapat berasal dari hasil render, hasil scan, maupun hasil potret menggunakan kamera. Keakuratan proses OCR sangat bergantung dari kualitas gambar itu sendiri. Semakin cerah, kontras, dan minim noise suatu input citra, maka semakin akurat pula teks yang dikenali. Pada saat ini berbagai teknologi OCR telah digunakan untuk pengenalan isi teks pada cek bank (Srivastava dkk., 2019), pengenalan nomor pelat kendaraan (Dalarmelina dkk., 2019), dan sebagainya.

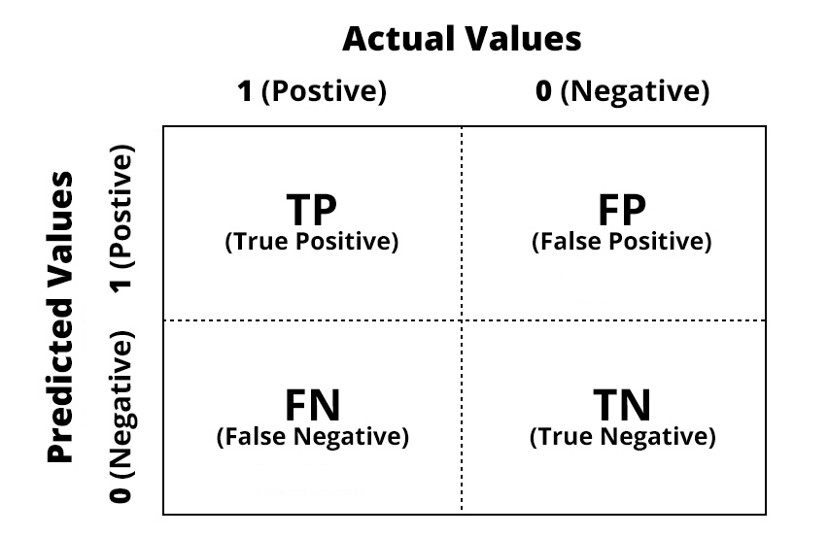
Library OCR yang akan digunakan pada penelitian ini adalah EasyOCR. Pada dasarnya terdapat beberapa tahapan di dalam sistem EasyOCR, antara lain: pre-processing, segmentasi, ekstraksi fitur, klasifikasi, dan post-processing. Tahapan pre-processing bertujuan untuk membersihkan input citra dari noise sehingga meningkatkan kualitas dari input itu sendiri. Selanjutnya tahap segmentasi dimana area teks dideteksi dan dilakukan proses orientasi horizontal agar teks mudah diproses. Pada tahap ketiga ekstraksi teks merupakan pengambilan bagian citra yang hanya memuat teks. Di tahap selanjutnya proses klasifikasi merupakan tahapan pengenalan dari masing-masing gambar teks. Secara default tahapan klasifikasi menggunakan model deep learning PyTorch. Pada tahap terakhir yaitu post-processing dimana dilakukan koreksi teks yang dikenali karena bisa jadi klasifikasi mengalami kesalahan sehingga diperlukan koreksi berdasarkan tata bahasa yang dipilih. Diagram alir proses pengenalan teks dari library EasyOCR dapat dilihat pada Gambar 2.2.



Gambar . Diagram Alir Library EasyOCR[[3]](#footnote-3)

## Confusion Matrix

Evaluasi metode pada penelitian ini dilakukan dengan mengamati kinerja metode melalui nilai accuracy, precision, recall, dan F-measure. Nilai-nilai di atas didapat dengan cara menghitung confusion matrix. Berikut contoh tabel confusion matrix yang disajikan pada Gambar 2.3.



Gambar . Contoh Tabel Confusion Matrix

Confusion matrix membandingkan hasil prediksi dengan hasil sebenarnya menjadi 4 kategori. Berikut adalah penjelasan dari masing-masing kategori:

1. **True Positive** : data positif yang diprediksi benar
2. **True Negative** : data negatif yang diprediksi benar
3. **False Positive** : data negatif namun diprediksi sebagai data positif
4. **False Negative** : data positif namun diprediksi sebagai data negatif

Accuracy adalah metrik performa untuk mengukur seberapa akurat model memprediksi benar dari total pengujian.

Precision adalah metrik performa untuk mengukur keakuratan model dengan menghitung seberapa banyak data yang benar-benar positif dari total terprediksi positif.

Recall adalah metrik performa untuk mengukur seberapa akurat model memprediksi data positif dari keseluruhan data positif. Recall sebaiknya memiliki skor setinggi mungkin.

F-Measure adalah metrik performa untuk menggabungkan dan menyeimbangkan antara precision dan recall menggunakan harmonic mean.

## State Of The Art

Terdapat beberapa penelitian terkait manipulasi dokumen menggunakan teknik pemetaan ulang character map. Pada Tabel 2.1 dapat dilihat hasil penelitian relevan yang dijelaskan berdasarkan judul penelitian, metode, dan hasilnya.

Tabel . Penelitian Relevan

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Judul Penelitian** | **Metode** | **Hasil** |
| 1 | Detecting visual spoofing using classical cryptanalysis methods in  plagiarism detection systems (Chow dkk., 2016) | Analisis kriptografi/ Cryptanalysis | Penelitian menyimpulkan bahwa metode cryptanalysis dapat mendeteksi anomali dari dokumen yang dimanipulasi. Hal ini disebabkan karena dokumen tersebut memiliki karakteristik distribusi unigram dan bigram yang terdistorsi dan tidak sesuai dengan bahasa yang digunakan. |
| 2 | Turnitoff: identifying and fixing a hole in current plagiarism detection software (Heather, 2010) | Optical Character Recognition | Penelitian ini tidak melakukan pengujian terkait metode yang diusulkan. Namun hanya memberikan solusi yaitu menggunakan OCR untuk melakukan ekstraksi teks dari dokumen PDF. |

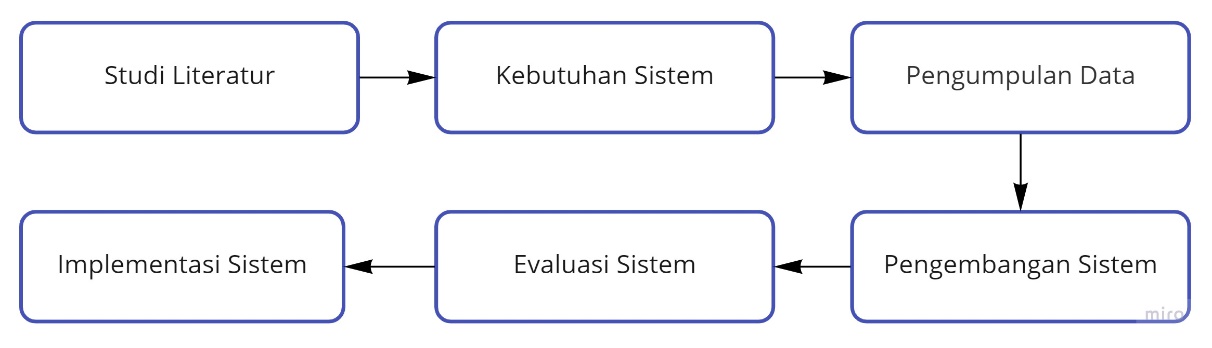
Pada penelitian pertama, metode yang digunakan adalah cryptanalysis dengan menganalisis distribusi karakter yang digunakan. Namun penelitian ini juga menyimpulkan adanya batasan karena metode yang digunakan berdasarkan informasi tekstual hasil ekstraksi dokumen. Dimana pengukuran bergantung dari seberapa besar anomali distribusi dibanding dengan pola yang diharapkan. Selain itu metode ini juga bergantung pada bahasa yang digunakan dimana tidak efektif apabila dokumen memiliki bahasa yang berbeda.

Pada penelitian kedua, metode yang digunakan adalah dengan melakukan ekstraksi teks di setiap halaman menggunakan OCR. Sedangkan teknologi OCR hingga saat ini memiliki reputasi yang buruk karena hasil deteksi yang tidak akurat. Selain itu metode ini juga melakukan pemindaian menggunakan OCR pada setiap halaman yang membutuhkan sumber daya komputasi yang besar.

Pada penelitian ini metode yang diusulkan adalah memanfaatkan OCR untuk melakukan verifikasi pada font yang di-embed ke dalam berkas PDF. Dengan membuat rendering teks per-karakter dan memvalidasinya, kesalahan deteksi OCR dapat diminimalisir. Hal ini disebabkan teknologi OCR yang bekerja baik pada gambar berkualitas baik.

# BAB III METODE PENELITIAN

Tahapan penelitian yang digunakan terdiri dari berbagai langkah runtut. Tahapan pertama adalah studi literatur dengan mengidentifikasi permasalahan dan menentukan deskripsi masalah serta pendekatan solusi. Tahap kedua adalah memenuhi kebutuhan sistem yang diperlukan untuk mengembangkan solusi. Tahap ketiga adalah pengumpulan data untuk pengujian sistem deteksi pra-plagiarisme. Tahap keempat adalah pengembangan dari sistem deteksi itu sendiri. Tahap kelima adalah evaluasi sistem dengan menganalisis hasil dan memperbaiki sistem. Tahap yang terakhir adalah implementasi sistem ke dalam container docker. Adapun langkah-langkah penelitian divisualkan pada Gambar 3.1 di bawah ini.



Gambar . Tahapan Penelitian

## Studi literatur

Pada tahapan ini terdiri dari serangkaian sub tahapan, antara lain: mempelajari permasalahan yang ada, mendefinisikan permasalahan secara rinci, mencari solusi yang ada saat ini, mempelajari penelitian terkait sebelumnya, dan merancang solusi terbaru yang akan ditawarkan.

## Kebutuhan Sistem

Pada tahapan ini yang dilakukan adalah mendefinisikan kebutuhan sistem yang harus dipenuhi. Kegagalan dalam memenuhi standar ini dapat berakibat pada permasalahan performa hingga permasalahan fatal. Berikut analisis kebutuhan sistem yang diperlukan:

### Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam penelitian ini dijabarkan pada Tabel 3.1 di bawah ini.

Tabel . Spesifikasi Perangkat Keras

|  |  |
| --- | --- |
| **Kategori** | **Spesifikasi** |
| Processor | Intel® Core™ i9 |
| Memory | 32 GB DDR4 |
| Storage | 512 GB SSD NVMe |
| CUDA-Enabled GPU | GeForce RTX 3060 |

### Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam penelitian dijelaskan sebagai berikut:

* Windows 11

Windows 11 adalah sistem operasi yang dikembangkan oleh Microsoft. Sistem operasi ini yang akan menjadi basis dari semua perangkat lunak yang dibutuhkan pada penelitian ini.

* Microsoft Word 2021

Microsoft Word 2021 adalah perangkat lunak pengelola teks yang populer dan merupakan bagian dari ekosistem Microsoft Office 2021. Aplikasi ini digunakan untuk membuat dokumen pengujian.

* Visual Studio 2022

Visual Studio 2022 adalah Integrated Development Environment (IDE) untuk bahasa pemrograman Visual Basic dan C#. IDE ini digunakan karena dukungan penuh framework .NET yang memungkinkan untuk mengakses API dari Microsoft Word.

* DataSpell

DataSpell adalah Integrated Development Environment untuk data science yang mendukung Jupyter Notebook dan bahasa Python. Software ini digunakan untuk melakukan riset hingga pengujian.

* Anaconda

Anaconda adalah paket distribusi bahasa Python untuk komputasi ilmiah seperti data science, machine learning, pemrosesan big data. Anaconda digunakan karena memiliki fitur untuk membuat environment lokal.

* Python

Python adalah bahasa pemrograman yang populer untuk data science dan machine learning. Bahasa ini sering digunakan sebagai alat bantu untuk menyelesaikan masalah di berbagai bidang.

* Anaconda & Python Packages

Berbagai package dari Anaconda dan Python akan digunakan antara lain: PyMuPDF, EasyOCR, PyTorch, Numpy, Pandas, Pillow, Fonttools, PyMongo, PyNSQ, dan Matplotlib.

## Pengumpulan Data

Pengumpulan data dilakukan dengan mengoleksi dokumen skripsi dan tugas akhir di Departemen Teknik Elektro Universitas Negeri Malang. Namun karena sedikitnya jumlah tampered data yang tersedia, maka digunakanlah data dummy untuk menyimulasikan dokumen skripsi dan tugas akhir. Dokumen ini akan sengaja dimanipulasi yang kemudian akan dideteksi oleh sistem.

## Pengembangan Sistem

Pengembangan sistem dilakukan dengan membangun sistem berdasarkan algoritma yang disusun. Tahapan dari deteksi meliputi ekstraksi embedded font dari berkas PDF. Kemudian verifikasi font menggunakan optical character recognition. Setelah itu dilakukan penandaan dokumen pada karakter yang bermasalah. Penandaan ini yang menjadi luaran dari sistem deteksi.

## Evaluasi Sistem

Evaluasi sistem meliputi analisis kesesuaian antara nilai prediksi dengan nilai sebenarnya menggunakan confusion matrix. Nilai metrik seperti accuracy, precision, recall, dan f-measure digunakan untuk mengukur efektivitas metode.

## Implementasi Sistem

Implementasi sistem meliputi proses deployment ke dalam container docker. Program ini nantinya menerima input dari message queue.

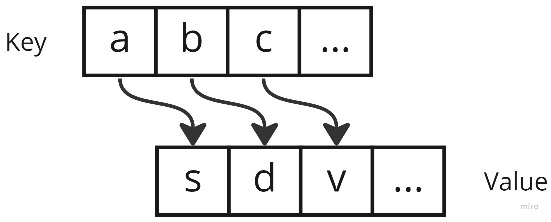
# BAB IV HASIL DAN PEMBAHASAN

## Hasil Pembuatan Data Pengujian

Skenario pengujian telah dilakukan dengan menggunakan dokumen yang dibuat manipulasinya secara acak. Dengan memanfaatkan pseudorandom, diharapkan dapat memberikan gambaran data yang mirip dengan kondisi aslinya.

### Proses Pembuatan Font

Tahapan dimulai dengan mendefinisikan list 62 karakter alfanumerik dan acak list tersebut. Selanjutnya menentukan subset karakter yang akan ditukar dengan melakukan slicing elemen pertama dimana . Subset digandakan, diacak lagi, dan dipasangkan sehingga pemetaan dibentuk menggunakan hashmap. Tahapan terakhir adalah dengan menghapus elemen pair yang memiliki key/domain dan value/range yang sama sehingga tidak ada karakter yang dipetakan ke karakter itu sendiri.



Gambar . Visualisasi Hashmap

Tahapan pembuatan font didasarkan pada hashmap. Misalnya karakter key yaitu huruf “a” memiliki data karakter value yaitu huruf “s”. Proses tersebut diimplementasikan terhadap seluruh informasi di dalam hashmap. Data yang ditukar oleh library Fonttools meliputi data glyf (bentuk glyph karakter) dan hmtx (informasi margin dan lebar karakter). Kemudian informasi font serta hashmap disimpan ke dalam MongoDB dengan pemodelan data sebagai berikut.



Gambar . Pemodelan Data Font

1. https://turnitin.com [↑](#footnote-ref-1)
2. https://pymupdf.readthedocs.io [↑](#footnote-ref-2)
3. https://github.com/JaidedAI/EasyOCR [↑](#footnote-ref-3)