**BAB 2**

**TINJAUAN PUSTAKA**

* 1. **Plagiarisme**
     1. **Definisi**

Plagiarisme atau sering disebut plagiat adalah penjiplakan atau pengambilan karangan, pendapat, dan sebagainya dari orang lain dan menjadikannya seolah karangan dan pendapat sendiri. Plagiat dapat dianggap sebagai tindak pidana karena mencuri hak cipta orang lain. Di dunia pendidikan, pelaku plagiarisme dapat mendapat hukuman berat seperti dikeluarkan dari sekolah/universitas. Pelaku plagiat disebut sebagai plagiator. Singkat kata, plagiat adalah pencurian karangan milik orang lain.Dapat juga diartikan sebagai pengambilan karangan (pendapat dan sebagainya) orang lain yang kemudian dijadikan seolah-olah miliknya sendiri. Setiap karangan yang asli dianggap sebagai hak milik si pengarang dan tidak boleh dicetak ulang tanpa izin yang mempunyai hak atau penerbit karangan tersebut.Sesudah 2 × 24 jam berita surat kabar tersiar, maka seseorang dapat mengambil alih dengan syarat harus menyebutkan sumbernya.

**2.1.1.1 Undang Undang mengenai Plagiarisme**

Pasal 72 ayat (1) :

“Barangsiapa dengan sengaja dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam Pasal 2 ayat (1) atau Pasal 49 ayat (1) dan ayat (2) dipidana dengan pidana penjara masing-masing paling singkat 1 (satu) bulan dan/atau denda paling sedikit Rp 1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp 5.000.000.000,00 (lima miliar rupiah)”.

dimana Pasal 2 ayat (1) tersebut :

“Hak Cipta merupakan hak eksklusif bagi Pencipta atau Pemegang Hak Cipta untuk mengumumkan atau memperbanyak Ciptaannya, yang timbul secara otomatis setelah suatu ciptaan dilahirkan tanpa mengurangi pembatasan menurut peraturan perundang-undangan yang berlaku”.

* + - 1. **Jenis Jenis Plagiarisme**

1. Plagiarisme kata per kata.
2. Plagiarisme Parafrase.
3. Plagiarisme sumber sekunder.
4. Plagiarisme struktur sekunder.
5. Plagiarisme ide.
6. Plagiarisme *authorship.*
   * + 1. **Contoh**

Banyak kasus plagiarisme yang terjadi di dunia akademik Indonesia. Kasus yang pernah marak di tahun 2010 adalah plagiarisme yang dilakukan seorang profesor muda bidang Hubungan Internasional (HI) di salah satu PTS terkemuka Bandung. Tidak ada yang menyangka bahwa profesor tersebut akan melakukan plagiarisme terkait dengan strata sekolahnya yang cukup tinggi. Profesor tersebut melakukan plagiat dari jurnal ilmiah Australia. Tulisan plagiat tersebut sebelumnya dimuat dalam koran The Jakarta Post yang kemudian ditarik kembali oleh The Jakarta Post setelah menerima laporan adanya plagiarisme. Diketahui bahwa selain tulisan yang dimuat di The Jakarta Post, profesor tersebut disinyalir telah melakukan plagiat pada empat tulisan yang juga telah dimuat di majalah populer lainnya. Atas perbuatannya tersebut sanksi yang diterima cukup berat yaitu pemberhentian tidak hormat dari PTS tempatnya bekerja dan dicabutnya gelar profesor. Hal ini sejalan dengan peraturan menteri pendidikan nasional nomor 17 tahun 2010, mengenai pencegahan dan penanggulangan plagiat di perguruan tinggi, bab iv pasal 12 ayat 2 dan 3.

* + 1. **Preprocessing**

Pengertian Preprocessing adalah Metode pengumpulan seringkali dikendalikan secara longgar, menghasilkan nilai out-of-range, kombinasi data yang tidak tepat, nilai yang hilang, dll. Menganalisis data yang belum diskrining dengan hati-hati agar masalah tersebut dapat menghasilkan hasil yang menyesatkan. Dengan demikian, representasi dan kualitas data pertama dan terutama sebelum menjalankan analisis.

Jika ada banyak informasi yang tidak relevan dan berlebihan, data yang berisik dan tidak dapat diandalkan, maka penemuan pengetahuan selama fase pelatihan lebih sulit. Persiapan data dan langkah penyaringan dapat memakan banyak waktu pemrosesan. Data pra-pengolahan meliputi pembersihan, seleksi Instance, normalisasi, transformasi, ekstraksi fitur dan seleksi, dll. Produk data pre-processing adalah setelah pelatihan akhir. menyajikan algoritma yang terkenal untuk setiap langkah pra-pengolahan data.

* + - 1. Tahap 1

*Case Folding*

* + - 1. Tahap 2

*Tokenizing*

* + - 1. Tahap 3

*Filtering*

* + - 1. Tahap 4

*Stemming*

**2.1.3 Algoritma *Rabin Karp***

Algoritma *Rabin Karp* adalah algoritma pencarian kata yang mencari sebuah pola berupa *substring* dalam sebuah teks menggunakan *hashing*. Algoritma ini sangat efektif untuk pencocokan kata dengan pola banyak. Salah satu aplikasi praktis dari algoritma *Rabin Karp* adalah dalam pendeteksian plagiarisme.

Algoritma Rabin-Karp adalah salah satu algoritma yang dapat digunakan untuk mencari dimana sebuah *string* (dalam kasus ini dinamakan sebagai pola) apakah ditemukan di dalam kumpulan *string* lain dengan ukuran yang lebih besar. Contoh yang dibahas kali ini adalah mengenai pencarian kata dari sebuah input teks.

Untuk mempercepat pencarian pola dalam kalimat, *Rabin-Karp* menggunakan Metode *hash*. Metode ini mengubah setiap *string* menjadi angka, yang dinamakan nilai *hash*. Apabila 2 *string* adalah sama persis, maka nilai hashnya juga akan sama, sehingga pencarian *string* dapat diturunkan dengan cara menghitung nilai *hash* dari pola, dan kemudian melakukan pencarian pola dengan nilai *hash* yang sama pada *data* input.

* + 1. **Metode *Dice Similarity Coefficient***

Atau nama lain nya sorensen dice coefficient. Untuk menghitung kemiripan dapat digunakan *Dice Similarity Coeficients* dengan cara menghitung jumlah K-Gram yang digunakan pada kedua dokumen yang diuji. Nilai kemiripan tersebut dapat dihitung dengan rumus : dimana S adalah nilai kemiripan, C adalah jumlah K-Gram yang sama dan A serta B masing-masing adalah jumlah K-Gram dari masing-masing string yang diujikan. Hasil yang didapat adalah bilangan dari 0 sampai 100 dimana bilangan tersebut adalah persentase kemiripan 2 *string* yang diujikan.

Metode *Hash* merupakan Metode yang menerima masukan string yang panjangnya sembarang kemudian mentransformasikannya menjadi string keluaran yang panjangnya tetap *(fixed)* dan umumnya berukuran jauh lebih kecil daripada ukuran string semula.

* 1. **Perancangan**

1. *Hypertext Markup Language* (HTML)

adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah *web* Internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegerasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman *web* dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman *web*. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C). HTML dibuat oleh kolaborasi *Caillau* TIM dengan Berners-lee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa).

1. *Hypertext Preprocessor* (PHP)

adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML.PHP banyak dipakai untuk memprogram situs web dinamis. PHP dapat digunakan untuk membangun sebuah CMS. PHP saat ini adalah bahasa pemrograman interpreter yang paling banyak digunakan saat ini dikarenakan bersifat open source dan juga paling banyak didukung oleh banyak web server. PHP dapat digunakan oleh banyak sistem operasi dari Windows, Linux maupun BSD. PHP umumnya diintegrasikan dengan aplikasi database yang juga open source seperti MySQL maupun PostgreSql, tapi bisa juga di integrasikan dengan Microsoft SQL, *Access* maupun *Oracle*.

Menurut Didik Dwi Presetyo (2004 : 76), PHP merupakan bahasa *scripting* *server-side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server*lah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan.

1. *JAVASCRIPT*

*JavaScript* adalah bahasa pemrograman tingkat tinggi dan dinamis. *JavaScript* populer di internet dan dapat bekerja di sebagian besar penjelajah *web* populer seperti *Internet Explorer (IE), Mozilla Firefox, Netscape* dan *Opera*. Kode *JavaScript* dapat disisipkan dalam halaman *web* menggunakan tag *SCRIPT*.

1. *LARAVEL FRAMEWORK*

*Laravel* adalah *framework* web PHP open source yang gratis, dibuat oleh Taylor Otwell dan ditujukan untuk pengembangan aplikasi *web* mengikuti pola arsitektur *model-view-controller* (MVC). Beberapa fitur *Laravel* adalah sistem pengemasan modular dengan manajer ketergantungan khusus, berbagai cara untuk mengakses basis data relasional, utilitas yang membantu penerapan dan pemeliharaan aplikasi, dan orientasinya terhadap gula sintaksis.

Pada bulan Maret 2015, *Laravel* dianggap sebagai salah satu kerangka kerja PHP yang paling populer, bersama dengan *Symfony, CodeIgniter, Yii2* dan lainnya.

Kode sumber *Laravel* di-*host* di GitHub dan dilisensikan dengan lisensi MIT.

1. *COMPOSER*

Komposer adalah manajer paket tingkat aplikasi untuk bahasa pemrograman PHP yang menyediakan format standar untuk mengelola ketergantungan perangkat lunak PHP dan perpustakaan yang dibutuhkan. Ini dikembangkan oleh Nils Adermann dan Jordi Boggiano, yang terus mengelola proyek ini. Mereka memulai pengembangan pada bulan April 2011 dan pertama kali merilisnya pada tanggal 1 Maret 2012. Komposer sangat terinspirasi oleh Node.js's "npm" dan "bundler" Ruby. Algoritma pemecahan masalah proyek dimulai sebagai port berbasis *openSUSE* dari *libzypp satpolver*.

Komposer berjalan melalui *command line* dan menginstal *dependencies* untuk sebuah aplikasi. Ini juga memungkinkan pengguna untuk menginstal aplikasi PHP yang tersedia di "*Packagist*" yang merupakan *repositori* utamanya yang berisi paket yang tersedia. Ini juga menyediakan kemampuan *autoload* untuk perpustakaan yang menentukan informasi *autoload* untuk memudahkan penggunaan kode pihak ketiga.

1. XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache* HTTP *Server*, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis.

**2.3 METODOLOGI**

**2.3.1 *UNIFIED MODELLING LANGUAGE*** **(*UML)***

Adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan kebutuhan (*requirement*), membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa-Salahuddin, 2011:113). Berdasarkan pendapat Rosa di atas, dalam proses pengembangan perangkat lunak penggunaan *UML* dapat digunakan pada tahap analisis dan desain. Namun, untuk membedakan tahap analisis dan desain, diputuskan *UML* digunakan pada tahap desain walaupun sebenarnya dapat membantu dalam menganalisis kebutuhan. Dalam proses pembangunannya *UML* bersifat independen terhadap bahasa pemodelan visual.

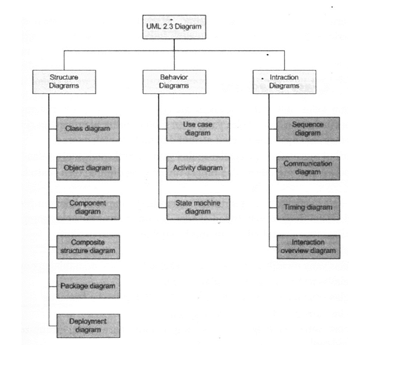
**2.3.1.1 *Class* Diagram**

***Class* diagram** adalah model statis yang menggambarkan struktur dan deskripsi **class** serta hubungannya antara ***class***. ***Class* diagram** mirip ER-**Diagram** pada perancangan *database*, bedanya pada ER-**diagram** tidak terdapat operasi/*methode* tapi hanya atribut. ***Class*** terdiri dari nama kelas, atribut dan operasi/*methode.*

*Class* diagram adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki 3 bagian utama yaitu *attribute, operation*, dan *name*. kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Sususan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut : Kelas *Main*. Yaitu kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan. Kelas *Interface*. Kelas yang mendefinisikan dan mengatur tampilan ke pemakai. Biasanya juga disebut kelas *boundaries*. Kelas yang diambil dari pendefinisian *usecase*. Merupakan kelas yang menangani fungsi-fungsi yang harus ada dan diambil dari pendefinisian *usecase*. Kelas Entitas. Merupakan kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Ada 13 Macam diagram UML yang dikelompokan ke dalam 3 kategori.

Lihat gambar berikut:



GAMBAR 2.1 DIAGRAM UML

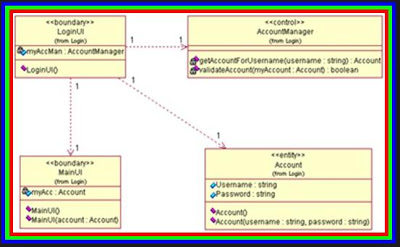
Tujuan utama dari *class* diagram adalah untuk menciptakan sebuah kosa kata yang digunakan oleh analis dan pengguna. Diagram kelas biasanya merupakan hal-hal, ide-ide atau konsep yang terkandung dalam aplikasi. Misalnya, jika anda sedang membangun sebuah aplikasi penggajian, diagram kelas mungkin akan berisi kelas yang mewakili hal-hal seperti karyawan, cek, dan pendaftaran gaji. Diagram kelas juga akan menggambarkan hubungan antara kelas.

*Class* memiliki 3 area pokok :

**1. *Name* (dan *stereotype*).**

**2. *Attribute*.**

**3. *Method.***



GAMBAR 2.5 *CLASS* DIAGRAM

**1. *Use Case Diagram***

*Use case* adalah abstraksi dari interaksi antara *system* dan *actor. Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah *system* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah *system* dipakai.

Diagram *Use Case* berguna dalam tiga hal :

1. Menjelaskan fasilitas yang ada *(requirement)*
2. Komunikasi dengan klien
3. Membuat test dari kasus-kasus secara umum

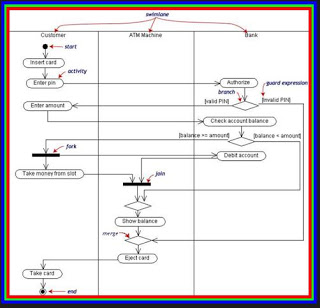


GAMBAR 2.2 *USE CASE* DIAGRAM

**2. *Activity* Diagram**

*Activity* diagram menyediakan analis dengan kemampuan untuk memodelkan proses dalam suatu *array* informasi. *Activity* diagram dapat digunakan untuk alur kerja model, *use case* individual, atau logika keputusan yang terkandung dalam metode individual. Activity diagram juga menyediakan pendekatan untuk proses pemodelan *parallel*. *Activity* diagram lebih lanjut .

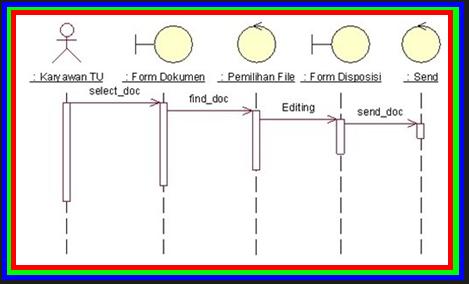
Pada dasarnya, diagram aktifitas canggih dan merupakan diagram aliran data yang terbaru. Secara teknis, diagram aktivitas menggabungkan ide-ide proses pemodelan dengan teknik yang berbeda termasuk model acara, *statecharts*, dan *Petri Nets*.



GAMBAR 2.3 *ACTIVITY DIAGRAM*

**3. *Sequence* Diagram**

*Sequence* diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram.



GAMBAR 2.4 *SEQUENCE* DIAGRAM

**2.3.2 *ENTITY RELATIONSHIP DIAGRAM* (ERD)**

adalah merupakan salah satu model yang digunakan untuk mendesain *database* dengan tujuan menggambarkan data yang berelasi pada sebuah database. Umumnya setelah perancangan **ERD** selesai berikutnya adalah mendesain *database* secara fisik yaitu pembuatan tabel, *index* dengan tetap mempertimbangkan *performance*.

Menurut Marlinda (2004:28) Entity Relationship Diagram (ERD) adalah diagram dari sistem yang menggambarkan hubungan antar entitas beserta relasinya yang saling terhubung. ERD menyediakan bentuk untuk menunjukkan struktur keseluruhan dari data user. Dalam ERD data-data tersebut digambarkan dengan menggunakan simbol entity. Dalam perancangan sistem ini terdapat beberapa entity yang saling terkait untuk menyediakan data-data yang dibutuhkan oleh system.

**2.3.3 *WATERFALL***

*Waterfall* atau AIR terjun adalah model yang dikembangkan untuk pengembangan perangkat lunak, membuat perangkat lunak. model berkembang secara sistematis dari satu tahap ke tahap lain dalam mode seperti air terjun.

Model ini mengusulkan sebuah pendekatan kepada pengembangan software yang sistematikdan sekuensial yang mulai dari tingkat kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan. Model ini melingkupi aktivitas-aktivitas sebagai berikut : rekayasa dan pemodelan sistem informasi, analisis kebutuhan, desain, koding, pengujian dan pemeliharaan.

Model pengembangan ini bersifat linear dari tahap awal pengembangan system yaitu tahap perencanaan sampai tahap akhir pengembangan system yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan dan tidak bisa kembali atau mengulang ke tahap sebelumnya.

* 1. **Penelitian yang Relevan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Peneliti | Judul | Tahun | Penerbit | Hasil |
| 1 | David Indra Lesmana, Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, | Pembuatan Sistem Penilaian Otomatis Pada Jawaban Ujian Berbentuk Esai Menggunakan Metode *Rabin Karp* | 2012 | Universitas Islam Negeri Maulana Malik Ibrahim Malang | Penelitian ini bertujuan untuk merancang dan membangun sistem penilaian otomatis pada jawaban ujian berbentuk esai menggunakan metode rabin karp sehingga dapat digunakan untuk membantu kinerja dosen dalam melakukan penilaian. |
| 2 | Eko Nugroho, Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, | Pembuatan Sistem Deteksi Plagiarisme Dokumen Teks dengan Menggunakan Algoritma *Rabin-Karp* | 2011 | Universitas Brawijaya Malang | Pada penelitian ini dilakukan modifikasi pada algoritma *Rabin-Karp* yaitu dengan menyisipkan metode stemming dengan menggunakan algoritma Arifin-Setiono pada tahap *preprocessing*-nya dan melakukan modifikasi pada saat proses *hashing* serta perubahan pada proses *string-matching*. |
| 3 | Pembuatan aplikasi anti Plagiarisme dengan Algoritma *Karp-Rabin* | Mutiara, A. Benny and Agustina, Sinta |  | Universitas Gunadarma | membuat sebuah aplikasi yang digunakan untuk mencari 5 jenis presentase kesamaan dari penulisan ilmiah yang satu dengan penulisan ilmiah yang lainnya pada bagian atau bab-bab tertentu. Presentase yang didapatkan yaitu 0%, dibawah 15%, antara 15-50%, diatas 50% dan 100%, Sehingga diharapkan hasilnya dapat berguna bagi dosen pembimbing maupun dosen penguji dari Mahasiswa yang bersangkutan pada Universitas Gunadarma. semua berdasarkan algoritma *Rabin Karp*. |
| 4 | Arsitektur Untuk Aplikasi Deteksi Kesamaan Dokumen Bahasa Indonesia | Anna Kurniawati,Kemal Ade Sekarwati, dan I Wayan Simri Wicaksana | 2012 | Universitas Gunadarma | Penelitian pengukuran kesamaan dokumen Bahasa Indonesia yang ada hanya mengukur kesamaan kataataupun kalimat, belum mempertimbangkan struktur kalimat, jumlah kalimat, posisi kalimat dan makna kata untuk membandingkan kalimat. Penelitian ini bertujuan untuk mengembangkan arsitektur aplikasi deteksi kesamaan dokumen teks Bahasa Indonesia dengan mempertimbangkan struktur kalimat, jumlah kalimat, posisi kalimat dan memperhitungkan faktor sinonim kata untuk melihat dari sisi makna kata. |