**IMPLEMENTASI ALGORITMA *K-NEAREST NEIGHBOR* DALAM SISTEM *CASE BASED REASONING* UNTUK PEMBENTUKAN IDENTITAS JAWABAN**

**OTOMATIS DAN PENCARI**

**KEMIRIPAN JAWABAN**

**DARI SOAL-SOAL**

**ALGORITMA**

**TUGAS AKHIR**

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

pada jurusan Teknik Informatika Fakultas Sains dan Teknologi

Universitas Islam Negeri Sunan Gunung Djati Bandung

**Oleh**

**Nurida Ahsanti**

**1127050118**

****

**BANDUNG**

**2016M/1437H**

# BAB I

# PENDAHULUAN

## Latar Belakang

Di era digital ini cukup jelas membuktikan betapa pesatnya perkembangan teknologi dan informasi. Hal ini akan terus berlangsung karena semakin banyaknya minat orang dalam persaingan mengembangkan atau membuat inovasi baru tentang teknologi. Pada nyatanya, bukan hanya orang yang khusus terlibat di bidang IT saja yang berpotensi untuk bersaing tetapi juga orang-orang di luar IT pun ikut tertarik. Oleh karena itu, semakin banyak orang yang ingin mempelajari tentang bagaimana proses dalam pembuatan perangkat lunak melalui berbagai contoh program/*coding* yang sudah ada.

Cara yang paling efektif dalam belajar pemrograman adalah mempelajari *source code* yang dibuat orang lain. Karena memang sudah menjadi kenyataan, bahwa segala hal yang dikerjakan tidak ada yang murni 100% orisinil. Untuk pembelajaran hanya perlu melihat hasil karya orang lain dan melakukan modifikasi. Terkadang melakukan modifikasi saja sulit, apalagi membuat sebuah program. Maka, jika dihadapkan pada sebuah *source code* program, yang perlu dianalisis adalah inti dari *source code* tersebut.[1]

Untuk dapat membuat sebuah program, maka terlebih dahulu harus mengetahui bagaimana memecahkan suatu permasalahan yang ada. Karena belajar memprogram adalah belajar tentang metodologi pemecahan masalah, kemudian menuangkan algoritma pemecahan masalah dalam suatu notasi tertentu.[2] Namun*,* ketika menuangkan algoritma pemecahan masalah kedalam suatu notasi dirasakan cukup sulit bagi yang belum memahami *coding.* Sehingga mudah untuk mengatasinya yaitu dengan membandingkan antara soal yang akan dipecahkan dengan soal-soal dan jawaban yang sudah ada dan diperkirakan mirip atau tidaknya saja. Hal tersebut dilakukan dengan tanpa mengetahui apakah soal dan jawaban tersebut benar-benar mirip atau tidak, sehingga tidak jarang pula hasil jawaban yang diperoleh kurang memuaskan.

Dalam membuat notasi dari pemecahan permasalahan algoritma, diperlukan suatu identitas yang dapat mewakili pemecahan masalah untuk jawaban yang diperoleh dari soal algoritma. Dan yang perlu diketahui bahwa sebuah algoritma dapat dibangun dari tiga struktur dasar yaitu Runtunan (*Sequence*), Pemilihan (*Selection*), dan Pengulangan (*Repetition*). Sedangkan pada sebuah *coding* disusun oleh arsitektur *coding* yang terdiri dari tipe data, konstanta, variabel dan algoritma dalam bentuk notasi itu sendiri. Dalam hal ini, pembentukan identitas jawaban dapat diukur dari arsitektur *coding* dengan cara membandingkan identitas baru dengan identitas-identitas jawaban sebelumnya sehingga dapat diklasifikasikan berdasarkan struktur dasar tersebut agar mendapatkan hasil yang paling mirip menurut voting terbanyak.

Dalam hal ini sejalan dengan metode Case Based Reasoning (CBR), yaitu sebuah metode bagian dari sistem pendukung keputusan yang dapat memecahkan masalah/kasus baru. Pada prosesnya CBR melalui 4 tahap, yaitu: *Retrieve* (mengambil masalah/kasus yang paling serupa), *Reuse* (menggunakan kembali masalah/kasus untuk mencoba memecahkan masalah/kasus.), *Revise* (merevisi solusi yang diajukan jika perlukan) dan *Retain* (mempertahankan/menyimpan solusi baru sebagai bagian dari masalah/kasus baru).[3]

Sedangkan pada perhitungan tingkat kemiripan (jarak) antar identitas jawaban didasarkan pada penggunaan beberapa atribut yang terdefinisi sebelumnya, yaitu dari penjumlahan antara pembobotan dengan tingkat kemiripan (jarak) keseluruhan atribut. Pada proses ini dapat digunakan algoritma klasifikasi seperti *K-Nearest Neighbor,* *Naive bayes*, LVQ dan *K-Means.*

Pada hasil penelitian yang dilakukan sebelumnya bahwa algoritma *K-Nearest Neighbor* (KNN) merupakan suatu metode yang menggunakan algoritma *supervised* dimana hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori yang ada sebelumnya pada KNN. Tujuan dari algoritma ini adalah mengklasifikasikan obyek baru bedasarkan atribut dan *training sample*.[4]

Lain halnya dengan algoritma klasifikasi *Naive bayes* yangtermasuk *multinomial* dalam mengambil jumlah kata yang muncul pada sebuah dokumen. Pada model ini, sebuah dokumen terdiri atas beberapa kejadian kata dan diasumsikan panjang dokumen tidak bergantung pada kelasnya. Sehingga memungkinkan pada tiap kejadian kata dalam sebuah dokumen dapat bebas dan tidak terpengaruh dengan konteks kata atau posisi kata dalam dokumen.[4]

Selain itu, algorima klasifikasi *Learning Vector Quantization* (LVQ) merupakan suatu metode untuk melakukan pembelajaran pada lapisan kompetitif terawasi yang nantinya akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor input. Jika 2 vektor input mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor tersebut ke dalam kelas yang sama.[4]

Sedangkan pada algorima klasifikasi *K-Means*, setiap data harus masuk *cluster* (kelompok) tertentu. Prinsip utama dari teknik ini adalah menyusun *k*buah prototipe atau pusat massa (*centroid*) atau pun rata-rata (*mean*) dari sekumpulan data berdimensi *n*. Teknik ini mensyaratkan nilai *k* sudah diketahui sebelumnya.[4]

Dari keempat algoritma klasifikasi yang telah dijabarkan sebelumnya dapat diambil kesimpulan bahwa algoritma *Naive bayes* memang lebih unggul dalam mengambil jumlah kata yang muncul tetapi cenderung lebih bebas dan tidak bergantung pada kelasnya. Sedangkan *Learning Vector Quantization* (LVQ) mempunyai prinsip pengklasifikasian berdasarkan vektor-vektor input sehingga kurang efektif jika pembandingnya berupa data yang banyak. Dan *K-Means* berprinsip setiap data harus masuk *cluster* (kelompok) dahulu jika tidak maka algoritma ini tidak bisa dilakukan. Lain halnya dengan *K-Nearest Neighbor* (KNN) yang hasil dari pengklasifikasian obyek baru didasarkan pada atribut dan *training sample* atau singkatnya mengklasifikasikan dengan voting terbanyak yang diperoleh dari perbandingan data yang telah ada sebelumnya sehingga menjadikan algoritma ini dianggap lebih cocok dengan konsep penelitian yang akan dilakukan.

Dalam mengklasifikasikan pencarian kemiripan soal dan jawaban algoritma dengan yang paling mirip merupakan solusi untuk memecahkan permasalahan dari soal algoritma. Untuk itu, maka disusunlah penelitian “**Implementasi Algoritma *K-Nearest Neighbor* Dalam *Sistem Case Based Reasoning* Untuk Pembentukan Identitas Jawaban Otomatis dan Pencari Kemiripan Jawaban dari Soal-soal Algoritma**”.

## Rumusan Masalah

Rumusan masalah yang akan dijadikan objek pada penelitian adalah sebagai berikut:

1. Bagaimana membangun sistem *Case Based Reasoning* yang dapat membentuk identitas jawaban otomatis dari soal algoritma dan mencari kemiripan jawaban tersebut menggunakan algoritma *K-Nearest Neighbor*?
2. Bagaimana penerapan algoritma *K-Nearest Neighbor* dan metode *Case Based Reasoning* dalam tingkat akurasi hasil pembentukan identitas jawaban otomatis dan pencari kemiripan jawaban?

## Tujuan

Adapun tujuan yang ingin dicapai dari penelitian ini yakni sebagai berikut:

1. Membangun sistem *Case Based Reasoning* yang dapat membentuk identitas jawaban otomatis dari soal algoritma dan mencari kemiripan tersebut menggunakan algoritma *K-Nearest Neighbor.*
2. Mengetahui penerapan algoritma *K-Nearest Neighbor* dan metode *Case Based Reasoning* dalam tingkat akurasi hasil pembentukan identitas jawaban otomatis dan pencari kemiripan jawaban.

## Batasan Masalah

Adapun batasan permasalahan dari penelitian ini yakni sebagai berikut:

1. Implementasi dari sistem ini berbasis dekstop *offline*.
2. Pembentukan identitas jawaban otomatis dilakukan agar hasilnya dapat dibandingkan kemiripannya dengan identitas jawaban yang sebelumnya.
3. Kemiripan jawaban hanya sebagai gambaran untuk mengerjakan soal-soal algoritma yang jawabannya harus menggunakan *coding.*
4. Hasil tingkat akurasi di dapat dari metode *Case Based Reasoning* dan algoritma *K-Nearest Neighbor*.

## Metodologi Penelitian

### Teknik Pengumpulan Data

Metodologi Penelitian diperlukan sebagai alat bantu untuk memudahkan pekerjaan didalam melakukan perancangan aplikasi. Metodologi penelitian yang digunakan yaitu sebagai berikut:

1. **Studi Kepustakaan**

Studi pustaka merupakan langkah awal dalam metode pengumpulan data. Studi pustaka merupakan metode pengumpulan data yang diarahkan kepada pencarian data dan informasi melalui jurnal penelitian, internet, buku dan *e-book* yang dapat mendukung dalam proses penulisan.”Hasil penelitian juga akan semakin kredibel apabila didukung foto-foto atau karya tulis akademik dan seni yang telah ada.”[5] Maka dapat dikatakan bahwa studi pustaka dapat memengaruhi kredibilitas hasil penelitian yang dilakukan.

1. **Wawancara**

Wawancara atau *interview* merupakan teknik pengumpulan data dengan cara bertatap muka secara langsung dengan informan dan dosen yang bersangkutan. Wawancara dilakukan jika data yang diperoleh kurang mendalam. Wawancara digunakan sebagai teknik pengumpulandata apabila peniliti ingin mengetahui hal-hal dari informan yang lebih mendalam.[5]

### Model Proses Pengembangan Perangkat Lunak

Adapun untuk pembuatan aplikasi ini, menggunakan metode pengembangan perangkat lunak *Prototype,* karena metode ini lebih memudahkan dalam proses membangun sebuah perancangan aplikasi. Alur dari metode *Prototype* terdiri dari langkah-langkah sebagai berikut:

1. *Communication,* yaitu *Developer* dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diinginkan dan gambaran bagian-bagian yang akan dibutuhkan berikutnya.
2. *Quick Plan* atau Perancangan yang dilakukan dengan cepat dan mewakili semua aspek *software* yang diketahui, dan rancangan ini menjadi dasar pembuatan *Prototype*.
3. *Modelling Quick Design*, langkah ini berfokus pada representasi aspek *software* yang bisa dilihat *customer/User* dan cenderung ke pembuatan *prototype*.
4. *Construction of Prototype,* yaitu membangun kerangka atau rancangan *prototype* dari *software* yang akan dibangun.
5. *Deployment Delivery & Feedback* merupakan *Prototype* yang telah dibuat oleh *developer* akan disebarkan kepada *User*/klien untuk dievaluasi, kemudian klien akan memberikan *feedback* yang akan digunakan untuk merevisi kebutuhan *software* yang akan dibangun. Pengulangan proses ini terus berlangsung sampai semua kebutuhan terpenuhi.

## Sistematika Penulisan

Sistematika penulisan laporan tugas akhir disusun dalam beberapa bab yang masing-masing bab menguraikan beberapa pokok pembahasan. Adapun sistematika penulisan laporan ini adalah sebagai berikut:

### BAB 1 PENDAHULUAN

Bab ini terdiri dari enam subbab yaitu latar belakang menggambarkan hal-hal yang perlu dijadikan penelitian dari realita yang ada, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika penulisan. Subbab Latar belakang masalah menggambarkan hal-hal yang perlu dijadikan penelitian dari realita yang ada. Subbab Rumusan masalah dituliskan kedalam bentuk poin yang menjadi sasaran utama pada objek yang akan diteliti. Subbab Tujuan menggambarkan hal-hal yang ingin dicapai. Subbab Batasan masalah berisi batasan yang ditentukan dalam perancangan sistem berupa hal-hal terkait dengan sistem. Subbab Metodologi penelitian terbagi kedalam dua tahapan yaitu teknik pengumpulan data dan menggambarkan model proses pengembangan perangkat lunak yang dibuat. Dan bagian terakhir dari bab ini yakni dengan subbab sistematika penulisan yang menguraikan urutan penyajian yang digunakan dalam penyusunan skripsi.

### BAB II LANDASAN TEORI

Bab ini menjelaskan teori-teori yang digunakan dalam menganalisis permasalahan yang ada dan teori-teori yang akan digunakan untuk membangun perangkat lunak ini.

### BAB III ANALISIS DAN PERANCANGAN

Pada bagian ini akan menguraikan hasil analisis dan perancangan perangkat lunak yang akan dibagun.

### BAB IV IMPLEMENTASI

Pada bab ini menjelaskan tentang spesifikasi perangkat lunak, kebutuhan perangkat lunak, implementasi perangkat lunak, dan pengujian yang dilakukan terhadap perangkat lunak yang dibangun.

### BAB V PENUTUP

Berisi tentang pernyataan singkat berupa kesimpulan dari pembahasan perangkat lunak yang dibuat secara keseluruhan dan saran untuk mengembangkan perangkat lunak yang lebih baik.

**DAFTAR PUSTAKA**

1. Utami, E., & Sukrisno. (2005). *10 Langkah Belajar Logika dan Algoritma Menggunakan Bahasa C dan C++ di GNU/Linux.* Yogyakarta: ANDI.
2. Munir, R. (2011). *Algoritma dan Pemrograman dalam Bahasa Pascal dan C, Edisi ke-3, Buku 1.* Bandung: Informatika Bandung.
3. Luthfi, E. T. (2010). Penerapan Case Based Reasoning dalam Mendukung Penyelesaian Kasus. *JURNA DASI*, 10.
4. Santoso, D., Ratnawati, D. E., & Indriati. (2014). Perbandingan Kinerja Metode Naïve Bayes, K-Nearest Neighbor, Dan Metode Gabungan K-Means Dan Lvq Dalam Pengkategorian Buku Komputer Berbahasa Indonesia Berdasarkan Judul Dan Sinopsis. *Jurnal Universitas Brawijata*, 14.
5. Sugiyono. (2005). *Metode Penelitian Kualitatif.* Bandung: Alfabeta.
6. Pressman, R. S. (2002). *Rekayasa Perangkat Lunak.* Yogyakarta: Andi.