

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Metode Penelitian**

Berdasarkan pada penjelasan penelitian sebelumnya, maka metode penelitian yang akan dipilih oleh penulis adalah K-Nearest Neighbor (K-NN) adalah sebuah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumnya. Termasuk dalam supervised learning, dimana hasil query instance yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori yang ada dalam K-NN.

##### **3.1.1 Metode KNN**

Algoritme k-nearest neighbor (k-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data pembelajaran. Sebuah titik pada ruang ini ditandai kelas c jika kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan jarak Euclidean.

Pada fase pembelajaran, algoritme ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi dari data pembelajaran. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk data test (yang klasifikasinya tidak diketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor data pembelajaran dihitung, dan sejumlah k buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut.

Nilai k yang terbaik untuk algoritme ini tergantung pada data; secara umumnya, nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Nilai k yang bagus dapat

dipilih dengan optimasi parameter, misalnya dengan menggunakan cross-validation. Kasus khusus di mana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat (dengan kata lain,  $k = 1$ ) disebut algoritme nearest neighbor.

Ketepatan algoritme k-NN ini sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritme ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur, agar performa klasifikasi menjadi lebih baik.

Terdapat beberapa jenis algoritme pencarian tetangga terdekat, diantaranya:

- a. Linear scan
- b. Pohon kd
- c. Pohon Balltree
- d. Pohon metrik
- e. Locally-sensitive hashing (LSH)

Algoritme k-NN ini memiliki konsistensi yang kuat. Ketika jumlah data mendekati tak hingga, algoritme ini menjamin error rate yang tidak lebih dari dua kali Bayes error rate (error rate minimum untuk distribusi data tertentu).

### **3.1.2 Kelebihan dan Kekurangan Laravel**

#### **1. Kelebihan Laravel**

- a. Waktu yang dibutuhkan untuk mengembangkan projek website dengan menggunakan framework ini menjadi lebih cepat.
- b. Dapat meningkatkan pengunjung webiste karena teknologi framework ini dapat digunakan di segala browser dan berbagai perangkat dengan baik.
- c. Laravel dilengkapi dengan utilitas pemrograman untuk membantu proses pengembangan aplikasi web dan juga moderasi dengan cara terbaik. Ini dikemas dengan Modular Packaging System (MPS) dengan pengaturan ketergantungan yang lengkap.

- d. Untuk menggunakan Laravel itu sendiri pengguna dibutuhkan untuk melakukan penginstalan sebuah composer. Composer adalah alat manajemen dependency pada PHP seperti npm (Node.js) dan Bundler (Ruby). Composer memungkinkan untuk membuat library pada project anda dan composer sendiri akan menginstall atau mengupdate secara otomatis tanpa anda harus menginstall manual.

## 2. Kekurangan Laravel

- a. Tidak Mudah memahami untuk pertama kali

Buat para beginner yang memahami laravel relatif akan kesulitan. sebab laravel tidak semudah codeigniter untuk dipahami. untuk masuk ke pintu gerbang nya diperlukan fokus agar bisa memahami laravel. Terlebih buat yang baru belajar Framework. Akan terbantu bila pernah belajar Framework sebelumnya seperti Codeigniter. jangankan untuk memahami alurnya, untuk install pertama kali pun kalau para beginner bisa dibuat bingung. karena cara install yang modern

- b. Update versi terlalu cepat

Update Laravel cepat bahkan menurut saya terlalu cepat. Sehingga bagi yang tidak mengikuti perkembangan laravel bisa ketinggalan jauh. Dan perubahan fitur nya pun terbilang signifikan. Terlebih yang dari versi 4 ke versi 5, sangat signifikan perubahannya baik fitur maupun struktur folder nya. Versi 5 pun cepat update versinya, sekarang saat saya menulis ini sudah sampai versi 5.5

- c. Install di Server tidak semudah Codeigniter

Bila Install codeigniter di ftp tinggal upload dan ubah konfigurasi database, maka install Laravel tidak semudah itu. Apabila kita sewa hosting di tempat lain, kita harus punya akses putty ke server agar bisa install composer. Untuk apa composer? Untuk kita menjalankan perintah composer update Kenapa kita jalankan perintah itu, karena beda lingkungan OS beda sistem

laravel. Laravel di windows beda dengan Laravel di Linux untuk itu jika kita membuat aplikasi di windows maka saat di upload di server Linux kita perlu jalankan perintah composer update di root aplikasi agar laravel bisa menyesuaikan lingkungan OS yang baru.

d. Relatif lebih berat

Laravel relatif lebih berat bila dibanding dengan framework Codeigniter. Hal ini wajar karena laravel meload banyak file dan asset untuk menjalankan aplikasinya. Tapi hal ini sudah ditanggulangi oleh Laravel 5 yang sudah menyederhanakan dari laravel versi 4 yang berat.

### **3.1.3 Kelebihan dan kekurangan Codeigniter**

#### **1. Kelebihan CodeIgniter**

- a. Berukuran sangat kecil. File download nya hanya sekitar 2MB, itupun sudah includedokumentasinya yang sangat lengkap.
- b. Dokumentasi yang bagus. Saat anda mendownloadnya, telah disertakan dengan dokumentasi yang berisi pengantar, tutorial, bagaimana panduan penggunaan, serta referensi dokumentasi untuk komponen-komponennya.
- c. Kompatibilitas dengan Hosting. CodeIgniter mampu berjalan dengan baik pada hampir semua platform hosting. CodeIgniter juga mendukung database-database paling umum, termasuk MySQL.
- d. Tidak ada aturan coding yang ketat. Terserah anda jika anda hanya ingin menggunakan Controller, tanpa View, atau tidak menggunakan Model, atau tidak salah satu keduanya. Namun dengan menggunakan ketiga komponennya adalah pilihan lebih bijak.
- e. Kinerja yang baik. Codeigniter sangat cepat bahkan mungkin bisa dibilang merupakan framework yang paling cepat yang ada saat ini.

- f. Sangat mudah diintegrasikan. CodeIgniter sangat mengerti tentang pengembangan berbagai library saat ini. Karenanya CodeIgniter memberikan kemudahan untuk diintegrasikan dengan library-library yang tersedia saat ini.
- g. Sedikit Konfigurasi. Konfigurasi CodeIgniter terletak di folder application/config. CodeIgniter tidak membutuhkan konfigurasi yang rumit, bahkan untuk mencoba menjalankannya, tanpa melakukan konfigurasi sedikitpun ia sudah bisa berjalan.
- h. Mudah dipelajari. Disamping dokumentasi yang lengkap, ia juga memiliki berbagai forum diskusi.

## 2. Kekurangan CodeIgniter

- a. CodeIgniter tidak ditujukan untuk pembuatan web dengan skala besar.
- b. Library yang sangat terbatas. Hal ini dikarenakan sangat sulit mencari plugin tambahan yang terverifikasi secara resmi, karena pada situsnya CodeIgniter tidak menyediakan plugin-plugin tambahan untuk mendukung pengembangan aplikasi dengan CI.
- c. Belum adanya editor khusus CodeIgniter, sehingga dalam melakukan create project dan modul-modulnya harus berpindah-pindah folder.

Terdapat beberapa data yang berasal dari questioner tentang klasifikasi framework codeigniter dan laravel.

Tabel 3.1 Data training

X1 = Kecepatan framework	X2 = Keamanan framework	Y = Klasifikasi
8	6	Baik
9	7	Baik
8	4	Buruk
7	7	Baik

8	5	Buruk
7	8	Baik

Tabel 3.2 Mencari Distance

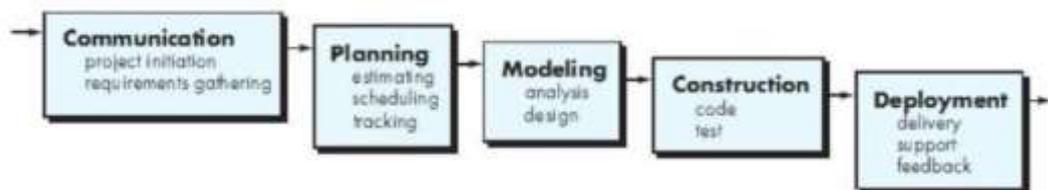
X1 = Kecepatan framework	X2 = Keamanan framework	Distance (7,4)
8	6	$(8-7)^2 + (6-4)^2 = 5$
9	7	$(9-7)^2 + (7-4)^2 = 13$
8	4	$(8-7)^2 + (4-4)^2 = 1$
7	7	$(7-7)^2 + (7-4)^2 = 9$
8	5	$(8-7)^2 + (6-4)^2 = 2$
7	8	$(7-7)^2 + (8-4)^2 = 16$

X1 = Kecepatan framework	X2 = Keamanan framework	Distance (7,4)	Jarak Terkecil	Apakah termasuk Nearest neighbor (K = 3)	Y = Kategori Nearest neighbor
8	6	$(8-7)^2 + (6-4)^2 = 5$	5	Ya	Baik
9	7	$(9-7)^2 + (7-4)^2 = 13$	13	Tidak	Buruk
8	4	$(8-7)^2 + (4-4)^2 = 1$	1	Ya	Baik
7	7	$(7-7)^2 + (7-4)^2 = 9$	9	Tidak	Buruk
8	5	$(8-7)^2 + (6-4)^2 = 2$	2	Ya	Baik
7	8	$(7-7)^2 + (8-4)^2 = 16$	16	Tidak	Buruk

## 3.2 Pengembangan Perangkat Lunak

### 3.2.1 Pengembangan Perangkat Lunak Menggunakan Metode *Waterfall*

Pada metode penelitian ini dilakukan rekayasa perangkat lunak yang digunakan adalah model *Waterfall* seperti pada gambar berikut, Fase-fase dalam *Waterfall Model* menurut referensi Pressman :



Gambar 3.1 Waterfall Pressman (Pressman, 2015)

Keterangan:

1. *Communication (Project Initiation & Requirements Gathering)*

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, dan internet.

2. *Planning (Estimating, Scheduling, Tracking)*

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan tracking proses pengerjaan sistem.

3. *Modeling (perancangan)*

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan

*interface*, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

4. *Construction (Code & Test)*

Tahapan *Construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

5. *Deployment (Delivery, Support, Feedback)*

Tahapan *Deployment* merupakan tahapan implementasi *software* ke *customer*, pemeliharaan *software* secara berkala, perbaikan *software*, evaluasi *software*, dan pengembangan *software* berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman,2015)

### 3.2.2 Rancangan Struktur Database

Struktur *database* yang di gunakan dalam system ini adalah sebagai berikut:

1. Struktur Tabel Login

Tabel ini di gunakan untuk login ke aplikasi

#	Nama	Jenis	Penyortiran
1	id 🗄️	int(20)	
2	username	varchar(30)	latin1_swedish_ci
3	password	varchar(100)	latin1_swedish_ci
4	nama_user	varchar(40)	latin1_swedish_ci
5	level	varchar(30)	latin1_swedish_ci

Gambar 3.2 Struktur Tabel Login

Tabel 3.3 Struktur Tabel Login

Field Name	Tipe Data	Size	
Id*	int	20	Id User Login
username	varchar	30	Username
password	varchar	100	Password
nama_user	varchar	40	Nama User Login
level	varchar	30	Level Login

## 2. Struktur Tabel Data Training

Tabel ini di gunakan untuk data perbandingan dengan data baru

#	Nama	Jenis	Penyortiran
1	id 🗄️	int(11)	
2	kcp_ci	char(2)	utf8mb4_general_ci
3	kcp_rl	char(2)	utf8mb4_general_ci
4	klasifikasi	varchar(5)	utf8mb4_general_ci

Gambar 3.3 Struktur Tabel Data Training

Tabel 3.4 Struktur Tabel Data Training

Field Name	Tipe Data	Size	
Id*	int	11	Id Data
kcp_ci	char	2	Data Kecepatan
Kcp_rl	char	2	Data Keamanan
Klasifikasi	varchar	5	Klasifikasi