

Perbandingan Metode-Metode Klasifikasi Untuk *Indoor Positioning System*

Yuan Lukito^{#1}, Antonius R. Chrismanto^{#2}

Program Studi Teknik Informatika, Universitas Kristen Duta Wacana
Jl. Dr. Wahidin Sudiro Husodo 5-25 Yogyakarta

^{#1}yuanlukito@ti.ukdw.ac.id

^{#2}anton@ti.ukdw.ac.id

Abstract — **Indoor Positioning System** can provide position and navigation guidance inside a building. This paper discusses about systematic comparisons of classifiers over Wi-Fi-based Indoor Positioning System dataset. The dataset is collected using a custom Android application, which able to receive, measure, and record Wi-Fi signal strengths from the surrounding Wi-Fi access points in UKDW campus. The dataset consists of 11658 Received Signal Strength (RSS) data from 41 public locations, spread across 11 buildings in UKDW campus. We use 10-folds cross validation and T-Test with 0.05 significance level to compare K-Nearest Neighbors (KNN), Naïve Bayes, J48, and Support Vector Machine (SVM) classifiers. Based on the experiment results, we can conclude that KNN classifier produces highest accuracy rate, 83.58%, while the lowest accuracy rate, 24.89%, produced by SVM. All classifiers implemented with little or no tuning, so it can be treated as a baseline for other experiments.

Keywords— Classification, Indoor Positioning System, K-NN, Naïve Bayes, J48, SVM.

I. PENDAHULUAN

Global Positioning System (GPS) dapat digunakan untuk menentukan posisi seseorang di Bumi, khususnya di ruangan terbuka (*outdoor*). GPS memiliki beberapa kelemahan, terutama pada penggunaan di dalam ruangan/gedung (*indoor*). Sinyal GPS umumnya sulit diterima jika posisi perangkat penerima berada di dalam gedung bertingkat. Untuk menentukan posisi seseorang di dalam gedung dapat menggunakan beberapa pendekatan yang bermacam-macam, umumnya memanfaatkan berbagai peralatan tambahan seperti sensor *ultrasonic*, sensor *infrared*, sensor LED, sensor RFID maupun menggunakan sinyal Wi-Fi.

Saat ini banyak institusi menyediakan akses Internet berbasis Wi-Fi. Beberapa *access points* ditempatkan sedemikian rupa sehingga sinyal Wi-Fi dapat menjangkau ke sebagian besar area di dalam gedung. Kondisi tersebut dapat dimanfaatkan untuk keperluan penentuan posisi seseorang di dalam gedung (*Indoor Positioning System*). Kuat atau lemahnya sinyal Wi-Fi yang diterima pada umumnya berbanding lurus dengan jarak perangkat penerima dengan *access point*. Dengan mengumpulkan

kekuatan sinyal (*Received Signal Strength*) yang diterima dari beberapa *access points* yang berbeda, posisi seseorang di dalam gedung dapat dihitung dan ditentukan.

Kumpulan data mengenai kekuatan sinyal tersebut dapat digunakan untuk proses pembelajaran terbimbing (*supervised learning*) untuk kemudian diimplementasikan sebuah model yang dapat mempelajari pola-pola tersebut dan dapat menentukan kemiripan atau kedekatan dengan pola-pola berikutnya. Beberapa penelitian terkait *Indoor Positioning System* (IPS) telah banyak dilakukan, dengan berbagai metode yang berbeda-beda, namun tidak pernah dibandingkan secara langsung hasilnya.

Penelitian ini berupa percobaan membandingkan beberapa metode klasifikasi dengan pembelajaran terbimbing, yaitu K-Nearest Neighbors, Naïve Bayes, J48, dan Support Vector Machine. Data yang digunakan dalam percobaan dikumpulkan dari beberapa ruang publik yang ada di kampus UKDW Yogyakarta, menggunakan aplikasi Android yang dikembangkan sendiri. Melalui penelitian ini diharapkan dapat memberikan kontribusi mengenai informasi dan hasil analisis perbandingan tingkat akurasi metode-metode klasifikasi dalam masalah *Indoor Positioning System*.

A. Permasalahan

Penelitian ini difokuskan pada permasalahan utama bagaimana hasil perbandingan dan analisis tingkat akurasi metode klasifikasi K-Nearest Neighbors, Naïve Bayes, J48 dan *Support Vector Machine* untuk penentuan posisi di dalam gedung.

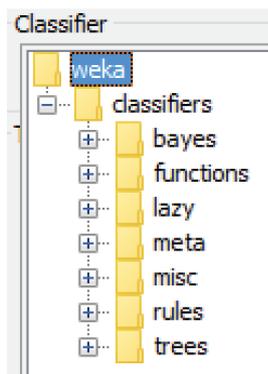
B. Tinjauan Pustaka

1) *Indoor Positioning System (IPS)*: *Global Positioning System* (GPS) merupakan sebuah sistem yang dapat menghitung dan menentukan posisi seseorang di Bumi secara akurat, umumnya direpresentasikan dalam bentuk *latitude* dan *longitude*. Walaupun demikian, sinyal GPS sulit diterima jika posisi perangkat penerimanya berada di dalam gedung, terutama gedung bertingkat. IPS adalah suatu sistem yang dapat menentukan posisi seseorang di dalam suatu ruangan tertutup/ gedung. Sistem ini selain dapat menentukan posisi, juga dapat menentukan orientasi dan

arah pergerakan seseorang [1]. Implementasi IPS dapat dilakukan menggunakan beberapa pendekatan yang bermacam-macam, umumnya menggunakan peralatan tambahan seperti sensor *ultrasonic*, sensor *infrared*, RFID dan beberapa peralatan lainnya. Kombinasi sensor *infrared* dan *ultrasonic* [2] memberikan akurasi penentuan posisi yang cukup akurat dengan tingkat kesalahan ± 2 cm, untuk ruangan dengan luas 4 m². Selain itu juga dapat dilakukan pendekatan optis menggunakan LED [3] maupun pemanfaatan sinyal Wi-Fi [4], [5] dan [6].

2) *Wi-Fi fingerprint*: Kumpulan kekuatan sinyal Wi-Fi pada suatu lokasi memiliki pola tertentu, berbeda dengan lokasi-lokasi lainnya. Pola kekuatan sinyal tersebut sering disebut sebagai *Wi-Fi fingerprint*. Beberapa metode yang dapat digunakan untuk menentukan tingkat kemiripan *Wi-Fi fingerprint* adalah *Euclidean distance* [5], pendekatan probabilitas [4] dan pendekatan metode *clustering* [6].

3) *Weka*: Merupakan paket perangkat lunak terbuka (*open source*) berisi implementasi algoritma-algoritma *machine learning* yang dapat digunakan untuk pemrosesan awal data, pembangunan model *machine learning* dan pengujian model [7]. Metode-metode yang dibandingkan pada penelitian ini sudah tersedia dalam *library* Weka. Untuk klasifikasi, Weka membagi metode-metode klasifikasi menjadi beberapa kategori, seperti terlihat pada Gambar 1.



Gambar 1. Kelompok-kelompok metode klasifikasi yang disediakan oleh Weka.

Penelitian ini membandingkan metode Naïve Bayes (kategori bayes), KNN (*lazy*), J48 (*trees*) dan SVM (*functions*).

4) *K-Nearest Neighbors (KNN)*: Merupakan salah satu metode klasifikasi *instance-based learning*, menggunakan pendekatan *supervised learning* sehingga membutuhkan data pelatihan yang sudah dilabeli.

Menurut Han [8], KNN membandingkan data pelatihan dan data pengujian, dimana data pelatihan dideskripsikan dalam sejumlah atribut berjumlah n . Setiap data pelatihan merepresentasikan sebuah titik dalam ruang berdimensi n

(atau vektor berukuran n). Klasifikasi data baru dilakukan dengan cara menghitung tingkat kemiripan atau kedekatan dari data baru tersebut terhadap seluruh data pelatihan yang ada. Tingkat kemiripan dapat dihitung menggunakan beberapa metode, salah satunya adalah *Euclidean distance*. Jarak dua vektor berukuran n , misalnya $X = (X_1, X_2, \dots, X_n)$ dan $Y = (Y_1, Y_2, \dots, Y_n)$ dapat dihitung dengan persamaan 1:

$$dist(X, Y) = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad \dots\dots (1)$$

Metode ini membutuhkan parameter nilai k , yaitu jumlah data pelatihan yang memiliki jarak terdekat dengan data baru, untuk menentukan hasil klasifikasi dari data baru tersebut. Nilai k dapat bervariasi, namun pada umumnya digunakan nilai ganjil sehingga tidak ada tetangga terdekat (*nearest neighbors*) berjumlah sama dalam kelas yang sama saat menentukan hasil akhir klasifikasi [9].

Hasil klasifikasi selain ditentukan oleh sejumlah (k) tetangga terdekat, juga dapat diimplementasikan *distance weighted* [10], dengan menghitung jarak sebagai fungsi pembobotan. Kelas hasil klasifikasi merupakan kelas yang memiliki bobot terbesar (jarak terkecil), yang umumnya dihitung dengan persamaan 2 berikut ini:

$$W_i = \frac{1}{dist_i} \quad \dots\dots (2)$$

dengan W_i adalah bobot jarak data ke- i , yang didapatkan dari 1 dibagi dengan jarak data ke- i dengan data baru yang akan diklasifikasikan. Hasil klasifikasi ditentukan berdasarkan kelas tetangga-tetangga terdekat yang memiliki jumlah bobot paling besar.

5) *Naïve Bayes*: Metode ini menggunakan pendekatan teorema Bayes, dengan asumsi setiap atribut bersifat saling bebas, tidak memiliki hubungan atau ketergantungan satu sama lain. Metode Naïve Bayes menggunakan prinsip teorema Bayes, yaitu menghitung probabilitas suatu kejadian berdasarkan suatu kondisi tertentu, dengan menggunakan persamaan 3.

$$P(A|B) = \frac{P(A) P(B|A)}{P(B)} \quad \dots\dots (3)$$

dengan $P(A|B)$ adalah peluang terjadinya A jika diketahui B, $P(A)$ adalah peluang terjadinya A, $P(B)$ adalah peluang terjadinya B dan $P(B|A)$ adalah peluang terjadinya B jika diketahui A.

Menurut Han [8], berikut ini adalah langkah-langkah metode Naïve Bayes:

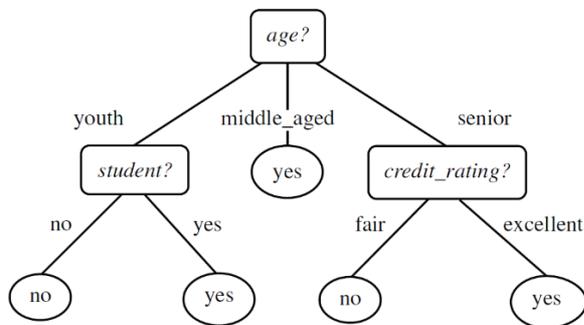
- a. Ditentukan D adalah data pelatihan yang sudah dilabeli sesuai dengan kelas atau kategorinya masing-masing. Setiap data pelatihan direpresentasikan dalam

vektor berdimensi n yang berisi seluruh atribut-atribut dari data tersebut, misalnya vektor $X = (X_1, X_2, \dots, X_n)$.

- b. Jumlah kelas sudah ditentukan dari awal, misalnya sejumlah m, dinyatakan dalam bentuk vektor $C = (C_1, C_2, \dots, C_m)$.
- c. Hasil klasifikasi data baru (C_d) dihitung berdasarkan persamaan 4:

$$C_d = \underset{k \in \{1, \dots, k\}}{\operatorname{argmax}} p(C_k) \prod_{i=1}^n p(X_i | C_k) \quad \dots\dots (4)$$

6) J48: merupakan implementasi algoritma C4.5 dalam bahasa pemrograman Java yang termasuk di dalam library Weka. Metode C4.5 dapat digunakan untuk klasifikasi dengan cara membentuk pohon keputusan (*decision tree*) dari data pelatihan yang diberikan [11]. Contoh sebuah decision tree dapat dilihat pada Gambar 2 [8].



Gambar 2. Contoh sebuah decision tree [8].

Menurut Kotsiantis [12], berikut ini adalah langkah-langkah yang diperlukan untuk pembentukan sebuah *decision tree*:

- a. Lakukan pengecekan untuk *base case*.
- b. Untuk setiap atribut yang ada di data pelatihan, hitung nilai *information gain* yang didapatkan jika membuat cabang pada atribut tersebut.
- c. Setelah didapatkan atribut yang memberikan nilai *information gain* tertinggi, lakukan percabangan pada pohon keputusan tersebut.
- d. Secara rekursif, lakukan langkah-langkah b dan c pada sub-atribut yang didapatkan, kemudian tambahkan ke dalam percabangan-percabangan berikutnya.

Nilai dari *information gain* didapatkan dengan menggunakan persamaan 5.

$$\operatorname{Gain}(A) = \operatorname{Info}(D) - \operatorname{Info}_A(D) \quad \dots\dots (5)$$

dengan $\operatorname{Info}(D)$ adalah jumlah informasi yang diperlukan untuk dapat menentukan kelas dari suatu data pelatihan yang diberikan. Nilai $\operatorname{Info}(D)$ dihitung menggunakan persamaan 6.

$$\operatorname{Info}(D) = - \sum_{i=1}^m p_i \log_2(p_i) \quad \dots\dots (6)$$

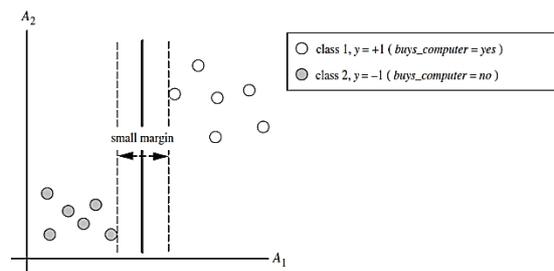
dengan m adalah jumlah data pelatihan, p_i adalah probabilitas data pelatihan ke-i masuk dalam kelas C. Nilai $\operatorname{Info}(D)$ ini juga dikenal dengan nama nilai *entropy*. Nilai $\operatorname{Info}_A(D)$ dihitung menggunakan persamaan 7.

$$\operatorname{Info}_A(D) = \sum_{j=1}^v \frac{D_j}{D} \times \operatorname{Info}(D_j) \quad \dots\dots (7)$$

$\operatorname{Info}_A(D)$ adalah jumlah informasi yang diperlukan untuk mengklasifikasikan sebuah data pelatihan jika dilakukan percabangan pada A. Secara umum nilai $\operatorname{Gain}(A)$ menunjukkan seberapa banyak informasi yang didapatkan jika dilakukan percabangan pada atribut A. Untuk setiap proses percabangan, selalu dipilih dan dilakukan percabangan pada atribut yang dapat memberikan nilai *Gain (information gain)* paling maksimal.

7) Support Vector Machine (SVM): merupakan salah satu metode klasifikasi dengan pembelajaran terbimbing (*supervised learning*). Pada sejumlah data pelatihan yang memiliki sejumlah p atribut (vektor berukuran p dimensi), metode SVM berusaha menemukan sebuah *hyperplane* berukuran (p-1) dimensi yang dapat memisahkan data pelatihan berdasarkan kelasnya.

Ilustrasi yang menggambarkan *hyperplane* (atau biasa disebut sebagai *decision boundary*) dari sejumlah data pelatihan dapat dilihat pada Gambar 3 [8].



Gambar 3. Contoh decision boundary untuk sejumlah data pelatihan dengan jarak/perbedaan kecil [8].

Jika diketahui data pelatihan merupakan data yang telah dilabeli dan memiliki sejumlah p atribut (atau biasa dinamakan sebagai *tuple*), (x_i, y_i) dengan $i = 1, 2, \dots, n$, dengan n adalah jumlah data pelatihan, sedangkan x_i adalah kumpulan atribut pada data pelatihan ke-i dan y_i adalah kelas dari data pelatihan ke-i tersebut, maka SVM secara

umum menghitung masalah optimisasi seperti pada persamaan 8 [8].

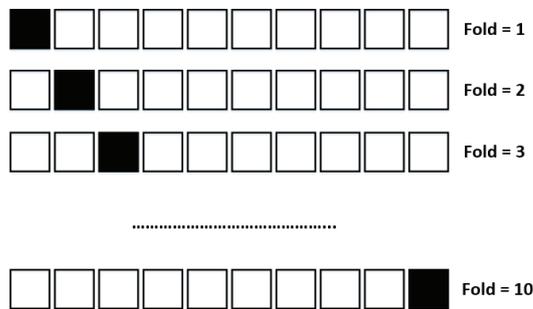
$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^p \xi_i \quad \dots (8)$$

dengan ketentuan [8]:

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \text{ dan } \xi_i > 0.$$

Metode SVM memiliki kelemahan pada proses perhitungan yang relatif lebih lama dibandingkan dengan metode-metode klasifikasi lainnya.

8) *Cross validation*: Merupakan salah satu metode validasi, biasa juga disebut *k-fold* atau *rotation estimation*. Ilustrasi dari *k-fold* atau *rotation estimation* dapat dilihat pada Gambar 4.



Gambar 4. Ilustrasi k-fold untuk k = 10.

Sebuah dataset D secara acak dibagi menjadi subset D_1, D_2, \dots, D_k yang disebut *fold*. Untuk $k = \text{jumlah fold}$ digunakan. Setiap *fold* (kotak hitam pada Gambar 4) kemudian digunakan sebagai data uji, sedangkan *fold* lainnya sebagai data pelatihan. Lakukan langkah tersebut sampai seluruh *fold* telah digunakan sebagai data uji [13]. Penggunaan *cross validation* diharapkan dapat mengurangi nilai bias yang mungkin muncul berdasarkan pemilihan data pelatihan dan pemilihan data uji yang kurang mewakili permasalahan yang dihadapi.

C. Tujuan Penelitian

Penelitian ini dilaksanakan dengan tujuan untuk mendapatkan hasil perbandingan metode-metode klasifikasi KNN, Naïve Bayes, J48 dan SVM. Tujuan lain dari penelitian ini adalah membentuk dataset *Indoor Positioning System* menggunakan data kekuatan sinyal Wi-Fi yang dikumpulkan dari beberapa lokasi publik di kampus UKDW.

D. Manfaat Penelitian

Hasil dari penelitian ini dapat dijadikan sebagai acuan dalam pemilihan metode yang terbaik untuk membuat implementasi *Indoor Positioning System* berbasis Wi-Fi.

Dataset yang dihasilkan diharapkan dapat digunakan pada penelitian-penelitian berikutnya terkait dengan *Indoor Positioning System* berbasis Wi-Fi.

II. METODOLOGI PENELITIAN

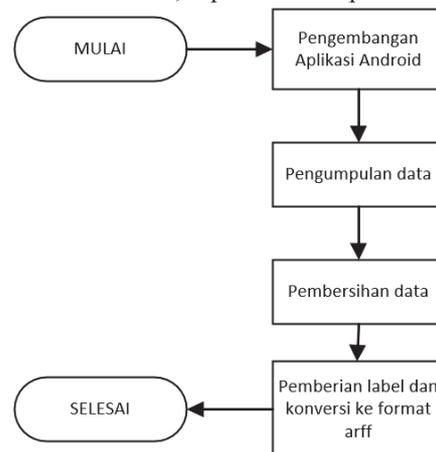
Secara umum penelitian ini dilaksanakan dalam tiga tahap, yaitu pembentukan *dataset*, pengujian dan analisis, seperti terlihat pada Gambar 5.



Gambar 5. Tahapan penelitian yang dilakukan.

A. Pembentukan Dataset

Tahap pembentukan *dataset* ini terdiri dari beberapa sub-tahap, yaitu: pengembangan aplikasi Android, pengumpulan data, pembersihan data dan pemberian label serta konversi dataset dalam format arff, seperti terlihat pada Gambar 6.



Gambar 6. Langkah-langkah pembentukan *dataset*.

Pengumpulan data menggunakan perangkat *smartphone* berbasis Android yang dilengkapi dengan Wi-Fi. Untuk penelitian ini, dikembangkan sebuah aplikasi Android yang

dapat melakukan pemindaian terhadap seluruh *access point* yang dapat dijangkau oleh *smartphone* tersebut. Tampilan aplikasi tersebut dapat dilihat pada Gambar 7.



Gambar 7. Tampilan aplikasi Wi-Fi Data Collector yang digunakan untuk pengumpulan data kekuatan sinyal Wi-Fi.

Pengumpulan data dilakukan dengan menggunakan aplikasi *Wi-Fi Data Collector* berbasis Android yang dikembangkan sendiri dengan pertimbangan kemudahan dalam mengumpulkan data kekuatan sinyal Wi-Fi. Perangkat bergerak seperti *tablet* dan *smartphone* yang berbasis Android pada umumnya memiliki konektivitas Wi-Fi, mudah dibawa berkeliling dan mudah dalam pengoperasiannya. Langkah-langkah penggunaan aplikasi *Wi-Fi Data Collector* untuk pengumpulan data adalah:

1. Ambil posisi sedekat mungkin dengan pintu masuk dari salah satu ruang publik yang ada di kampus UKDW.
2. Jalankan aplikasi *Wi-Fi Data Collector*. Secara otomatis aplikasi ini akan mengaktifkan Wi-Fi jika belum aktif.
3. Pilih ruangan yang sesuai, kemudian *tap* pada tombol Scan. Tunggu sebentar.
4. Aplikasi akan menampilkan daftar *access points* yang ada di sekitar ruang publik tersebut, beserta informasi nama *access point* dan kekuatan sinyal yang diterima.
5. *Tap* tombol Save untuk menyimpan data kekuatan sinyal yang diterima ke dalam *database*.

Pada penelitian ini dilakukan pengambilan data kekuatan sinyal Wi-Fi untuk 41 lokasi ruang publik yang ada di kampus UKDW, meliputi Fakultas, Biro Universitas, Unit Pembantu, Ruang Rektorat, Kantin, Koperasi dan beberapa ruang publik lainnya. Pada kampus UKDW terpasang 177 *access points* yang tersebar di 11 gedung, dimana beberapa gedung terdiri lebih dari 3 tingkat.

Skema gedung-gedung kampus UKDW dapat dilihat pada Gambar 8.



Gambar 8. Skema 11 gedung di kampus UKDW.

Pengumpulan data dilakukan dengan bantuan mahasiswa berjumlah 10 orang. Setiap orang mengunjungi 41 lokasi ruang publik yang sudah ditentukan, kemudian merekam dan menyimpan data kekuatan sinyal Wi-Fi sebanyak 10 kali, untuk tiga rentang waktu yang berbeda (pagi, siang dan sore). Pengumpul data mengunjungi lokasi ruang publik tersebut, kemudian menjalankan aplikasi *Wi-Fi Data Collector*. Memilih ruangan yang sesuai, kemudian melakukan scan terhadap sinyal-sinyal Wi-Fi yang dapat ditangkap pada lokasi tersebut. Pilihan lokasi ruang publik pada aplikasi *Wi-Fi Data Collector* dapat dilihat pada Gambar 9.



Gambar 9. Beberapa pilihan lokasi ruang publik pada aplikasi *Wi-Fi Data Collector*.

Beberapa kendala sempat ditemui dalam tahap pengumpulan data, misalnya pada ketidaktepatan mahasiswa dalam memilih ruang publik. Masalah ini dapat dipecahkan dengan mencatat ruangan dan waktu pengambilan untuk setiap pengambilan data yang tidak sesuai.

Setelah tahap pengumpulan data selesai, didapatkan secara keseluruhan 12300 data. Setelah dilakukan pengecekan, beberapa data dihapus karena ada kesalahan pemilihan ruangan, berdasarkan catatan yang dibuat oleh masing-masing mahasiswa yang membantu penelitian ini. Setelah proses pengecekan dan penghapusan dilakukan, didapatkan 11658 data.

Format dan contoh data kekuatan sinyal yang didapatkan pada tahap pengumpulan data dapat dilihat pada Tabel I.

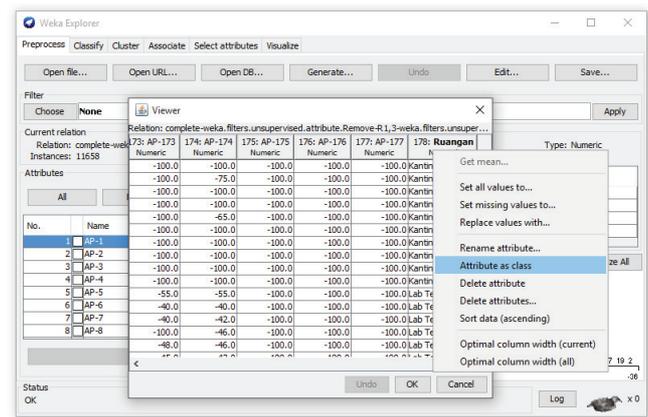
TABEL I
FORMAT DAN CONTOH DATA KEKUATAN SINYAL

Ruangan	AP1	AP2	AP3	AP4	...	AP177
Rektorat	-86	-70	-100	-100	...	-84
Biro II	-70	-89	-89	-89	...	-100
Koperasi	-100	-82	-70	-100	...	-65
...
Fak. TI	-100	-100	-60	-100	...	-80

Untuk setiap baris data terdiri dari 178 kolom, dengan kolom pertama merupakan nama ruangan dan kolom-kolom berikutnya merupakan kekuatan sinyal yang diterima dari seluruh *access points* yang ada di kampus UKDW.

Kekuatan sinyal dalam satuan dbm, semakin besar nilainya maka sinyal yang diterima lebih kuat. Untuk *access points* di gedung berbeda sebagian tidak dapat ditangkap sinyalnya untuk beberapa lokasi tertentu, kekuatan sinyalnya ditentukan dengan nilai -100, yang menunjukkan tidak ada sinyal yang diterima.

Tahap pengumpulan data ini menghasilkan sebuah *dataset* dalam format CSV. Setelah tahap pengumpulan data selesai, berikutnya adalah tahap pengujian menggunakan perangkat lunak Weka. Untuk memudahkan tahap pengujian, dataset dalam format CSV tersebut dikonversi terlebih dahulu ke format ARFF (*.arff, Attribute-Relation File Format) dengan menggunakan Weka Explorer, pada sub-bagian *Preprocess*, seperti terlihat pada Gambar 10. Pada tahap konversi ini, kolom Ruangan dijadikan sebagai *class/label*.



Gambar 10. Preprocessing dataset dan konversi ke format arff menggunakan Weka Explorer.

B. Pengujian

Tahap pengujian dilakukan menggunakan perangkat lunak Weka. Beberapa metode yang akan dibandingkan dapat dilihat pada Tabel II.

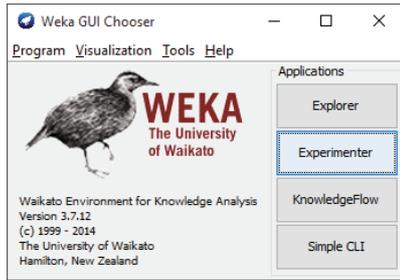
TABEL II
METODE-METODE YANG DIBANDINGKAN DALAM PENGUJIAN

Metode	Parameter
KNN	k = 1, distance weighting = false
	k = 3, distance weighting = false
	k = 1, distance weighting = true
	k = 3, distance weighting = true
Naïve Bayes	useKernelEstimator = false
	useKernelEstimator = true
J48	unpruned = false
	unpruned = true
SVM	parameter default

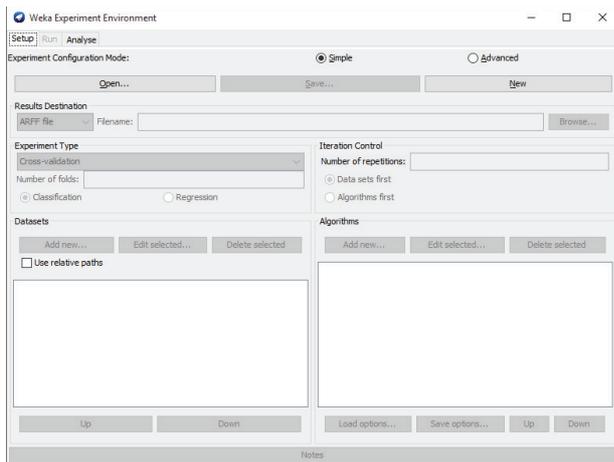
Untuk metode KNN terdiri dari 4 variasi untuk melihat pengaruh nilai k maupun penggunaan metode distance weighting terhadap tingkat akurasi. Pada metode Naïve Bayes, dilakukan dalam dua variasi, yaitu penggunaan kernel estimator atau tidak. Untuk metode J48, parameter yang divariasikan adalah apakah akan dilakukan proses pruning atau tidak. Sedangkan pada metode SVM dilakukan dengan konfigurasi parameter *default*, tanpa mengubah parameter.

Tahap ini terdiri dari beberapa langkah, yaitu:

1. Jalankan perangkat lunak Weka, pilih menu *Experimenter*, seperti pada Gambar 11 dan Gambar 12.

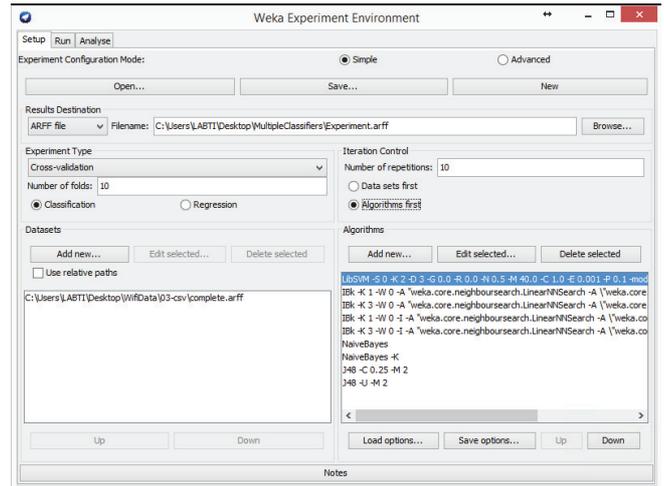


Gambar 11. Tampilan awal aplikasi Weka, pilih menu Experimenter.



Gambar 12. Window Experimenter pada Weka.

2. Tentukan jenis percobaan menjadi *Cross-validation*, dengan jumlah *folds* sebanyak 10 dan jumlah *repetitions* sebanyak 10.
3. Buka *dataset* yang dihasilkan dari tahap pengumpulan data.
4. Tentukan kolom nama ruangan sebagai kelas (class).
5. Pilih metode-metode yang akan dibandingkan, sesuaikan parameter-parameter yang dibutuhkan berdasarkan ketentuan pada Tabel II.
6. Setelah konfigurasi percobaan sudah lengkap seperti pada Gambar 13, percobaan dapat dijalankan.



Gambar 13. Seluruh parameter pengujian sudah dimasukkan, siap untuk tahap pengujian.

7. Pilih *tab Run* untuk menjalankan pengujian, tunggu sampai seluruh metode selesai dijalankan.

C. Analisis Hasil Pengujian

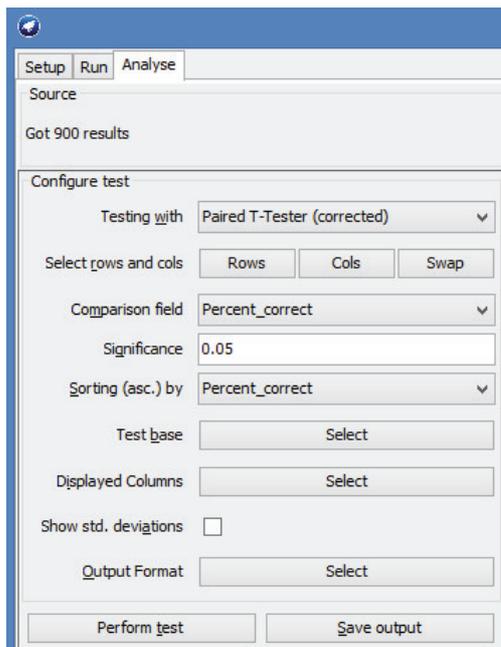
Setelah proses pengujian selesai, tahap berikutnya adalah tahap analisis hasil pengujian. Setiap metode yang diujikan akan menghasilkan 100 hasil pengujian karena menggunakan *k-fold* dengan nilai $k = 10$ dan jumlah *repetitions* sebanyak 10. Dengan demikian secara keseluruhan akan didapatkan 900 hasil pengujian.

Untuk menentukan metode yang menghasilkan tingkat akurasi klasifikasi tertinggi, digunakan metode uji statistik, seperti pada Tabel III.

TABEL III
PARAMETER YANG DIGUNAKAN PADA TAHAP ANALISIS

Parameter	Nilai
Testing with	<i>Paired T-Tester (corrected)</i>
Comparison field	<i>Percent correct</i>
Significance	0.05 (<i>two tailed</i>)

Tampilan konfigurasi parameter untuk tahap analisis menggunakan Weka dapat dilihat pada Gambar 14.



Gambar 14. Konfigurasi parameter untuk tahap analisis.

Pada penelitian ini, metode Naïve Bayes tanpa kernel estimator dijadikan sebagai pembanding (*test base*) bagi metode-metode lainnya. Setelah tahap analisis selesai dijalankan, hasilnya dapat dilihat pada Tabel IV.

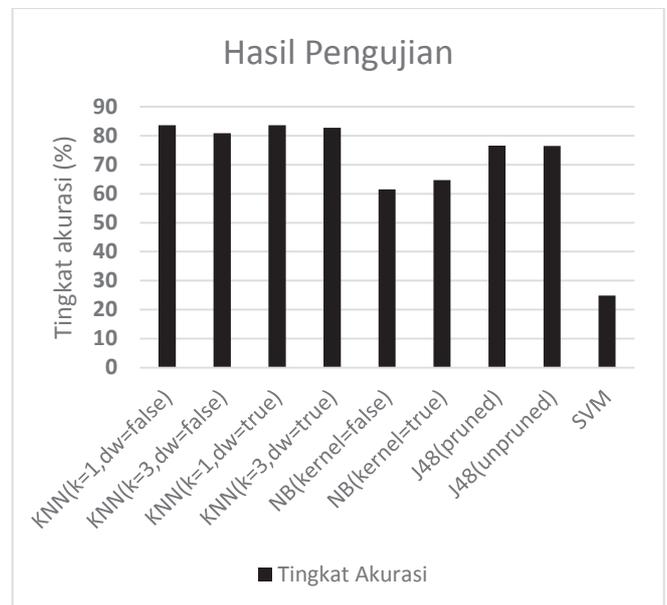
TABEL IV
HASIL ANALISIS PENGUJIAN CORRECTED PAIRED T-TEST

Metode	Akurasi (%)	Perbandingan
KNN (k = 1, dw = false)	83,58	(1/0/0)
KNN (k = 3, dw = false)	80,92	(1/0/0)
KNN (k = 1, dw = true)	83,58	(1/0/0)
KNN (k = 3, dw = true)	82,72	(1/0/0)
Naïve Bayes (kernel = false)	61,52	(v/ /*)
Naïve Bayes (kernel = true)	64,75	(1/0/0)
J48 (pruned)	76,60	(1/0/0)
J48 (unpruned)	76,48	(1/0/0)
SVM	24,89	(0/0/1)

Kolom perbandingan pada Tabel IV merupakan informasi hasil perbandingan masing-masing metode dengan metode yang dijadikan *test base*, yaitu Naïve Bayes tanpa *kernel estimator*. Nilai (v/ /*) menunjukkan metode tersebut dijadikan sebagai *test base*, sedangkan nilai (1/0/0) menunjukkan metode tersebut memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan *test base*. Nilai (0/0/1) menunjukkan metode tersebut menghasilkan tingkat akurasi yang lebih rendah dibandingkan dengan *test base*.

Metode Naïve Bayes tanpa *kernel estimator* menghasilkan tingkat akurasi sebesar 61,52%, yang dijadikan sebagai acuan/ pembanding (*test base*) bagi metode-metode lainnya. Secara umum sebagian besar

metode memiliki tingkat akurasi di atas *test base*, hanya metode SVM yang menghasilkan tingkat akurasi di bawah *test base*, yaitu hanya 24,89%. Hasil perbandingan tingkat akurasi dapat dilihat dalam bentuk grafik batang, seperti pada Gambar 15.



Gambar 15. Hasil pengujian yang didapatkan dari pengujian.

Metode SVM yang diimplementasikan dalam penelitian ini menggunakan konfigurasi parameter *default* sehingga hasilnya tidak baik. Dengan konfigurasi parameter yang tepat, masih ada kemungkinan bagi metode SVM untuk menghasilkan tingkat akurasi yang lebih baik lagi.

Metode Naïve Bayes dengan kernel estimator menghasilkan tingkat akurasi yang lebih baik (64,75%) jika dibandingkan dengan metode Naïve Bayes tanpa *kernel estimator* (61,52%). Penggunaan *kernel estimator* ternyata dapat meningkatkan tingkat akurasi metode Naïve Bayes.

Metode klasifikasi berbasis *decision tree*, J48, menghasilkan tingkat akurasi yang lebih baik dari Naïve Bayes, yaitu 76,60% dan 76,48%, masing-masing untuk J48 dengan *pruning* dan J48 tanpa *pruning*. Proses *pruning* secara umum dapat meningkatkan tingkat akurasi dan mempercepat proses klasifikasi, namun dalam penelitian ini tingkat akurasi yang didapatkan hanya berbeda sedikit.

Seluruh variasi metode KNN yang diimplementasikan dalam penelitian ini menghasilkan tingkat akurasi melebihi *test base* dan secara konsisten memberikan hasil yang lebih baik dibandingkan dengan metode-metode klasifikasi lainnya. Penggunaan nilai k=1 menghasilkan tingkat akurasi yang lebih baik dibandingkan dengan nilai k=3. Penggunaan *distance weighting* juga dapat meningkatkan akurasi dari KNN. Untuk nilai k=1, penggunaan *distance weighting* tidak berpengaruh, sehingga tingkat akurasi yang didapatkan sama. Penggunaan *distance weighting* berpengaruh untuk

nilai $k > 1$, meningkatkan tingkat akurasi dari 80,92% ($k=3$, $dw=false$) menjadi 82,72% ($k=3$, $dw=true$).

Penelitian ini melakukan pengujian tingkat akurasi klasifikasi beberapa metode dengan menggunakan *dataset* yang didapatkan dari tahap pengumpulan data. Ciri (*features*) yang digunakan untuk klasifikasi adalah seluruh kolom yang ada, sehingga ada kemungkinan terjadi *curse of dimensionality*, menyebabkan tingkat akurasi menurun karena dimensi data yang terlalu besar. Untuk menghindari masalah *curse of dimensionality* tersebut, dapat dilakukan langkah-langkah pemrosesan awal terhadap *dataset* yang digunakan, misalnya dengan menggunakan *Principal Component Analysis (PCA)*, *High Correlation Filter*, *Singular Value Decomposition (SVD)*, maupun beberapa metode-metode *feature/dimensionality reduction* lainnya.

Untuk meningkatkan tingkat akurasi dari metode KNN dapat mencoba pengujian menggunakan nilai k yang berbeda-beda, untuk mendapatkan hasil yang terbaik. Pengurangan dimensi ciri (*dimensionality reduction*) dari *dataset*, dapat menggunakan *Condensed Nearest Neighbors (CNN)* maupun *Generalised Condensed Nearest Neighbors (GCNN)*. Peningkatan akurasi juga bisa didapatkan dengan menggabungkan beberapa metode klasifikasi sekaligus, atau menggabungkan metode klusterisasi dengan metode klasifikasi.

III. SIMPULAN

Dari penelitian ini, dapat ditarik beberapa simpulan berikut ini:

1. Proses pengumpulan data membutuhkan langkah-langkah tambahan untuk pengecekan dan pembersihan *dataset* dari kesalahan yang umumnya dilakukan, yaitu ketidaktepatan saat memilih ruangan di aplikasi Android yang digunakan.
2. Metode K-Nearest Neighbors menghasilkan tingkat akurasi tertinggi, kemudian berikutnya metode J48, Naïve Bayes dan SVM di urutan terakhir.
3. Metode K-Nearest Neighbors menghasilkan tingkat akurasi yang tertinggi jika dibandingkan dengan metode-metode lainnya, yaitu dengan tingkat akurasi tertinggi 83,58% untuk nilai $k=1$.
4. Penggunaan *pruning* pada metode J48 dapat meningkatkan tingkat akurasi yang dihasilkan, yaitu dari 76,48% menjadi 76,60%.
5. Penggunaan *kernel estimator* pada metode Naïve Bayes dapat meningkatkan tingkat akurasi yang dihasilkan, yaitu dari 61,52% menjadi 64,75%.
6. Metode-metode yang dibandingkan pada penelitian ini diimplementasikan dengan tidak banyak mengubah konfigurasi awal, sehingga dimungkinkan untuk dapat meningkatkan tingkat akurasi dengan

mengubah konfigurasi dan melakukan tahap *preprocessing* pada *dataset* sebelum dilakukan tahap klasifikasi.

UCAPAN TERIMA KASIH

Penelitian ini diselenggarakan menggunakan dana penelitian Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, No. 134/D.02/FTI/UKDW/2015. Kami mengucapkan terima kasih atas dukungan dan bantuan dari pihak-pihak berikut ini:

1. Fakultas Teknologi Informasi UKDW selaku pemberi dana penelitian.
2. Mahasiswa Teknik Informatika UKDW yang telah membantu dalam tahap pengumpulan data.

DAFTAR PUSTAKA

- [1] A. Ghose, A. Pal, A. D. Choudhury, V. Chandel, C. Bhaumik & T. Chattopadhyay, "Indoor Positioning System", US Patent No. 14/450,890, Feb. 5, 2015.
- [2] H. Yucel, T. Ozkir, R. Edizkan & A. Yazici, "Development of Indoor Positioning System with Ultrasonic and Infrared Signals", *Prosiding INISTA*, 2012, p. 1-4.
- [3] S. Y. Jung, S. Hann, S. Park & C. S. Park, "Optical Wireless Indoor Positioning System Using Light Emitting Diode Ceiling Lights", *Microw. Opt. Technol. Lett.*, 54: 1622-1626. Doi: 10.1002/mop.26880.
- [4] N. Le-Dortz, N. Gain & P. Zetterberg, "Wi-Fi Fingerprint Indoor Positioning System Using Probability Distribution Comparison", *Prosiding ICASSP*, 2012, p. 2301-2304.
- [5] J. Y. Lee, C. H. Hyunjae & J. So, "Analysis of Location Estimation Algorithms for Wi-Fi Fingerprint-based Indoor Localization", *Prosiding The 2nd International Conference on Software Technology*, 2013, vol. 19 p. 89-92.
- [6] O. Dousse, J. Eberle & M. Mertens, "Place Learning via Direct Wi-Fi Fingerprint Clustering", *Prosiding Mobile Data Management*, 2013, 2012, p. 282-287.
- [7] (2015) Machine Learning Group at the University of Waikato. [Online]. Tersedia: <http://www.cs.waikato.ac.nz/ml/weka/>
- [8] J. Han, M. Kamber & J. Pei, *Data Mining Concepts and Techniques*, 3rd ed., Amsterdam, Elsevier, 2011.
- [9] C. D. Manning, P. Raghavan & H. Schütze, *Introduction to Information Retrieval*, New York, Cambridge University Press, 2008.
- [10] S. A. Dudhani, "The Distance-Weighted k-Nearest Neighbor Rule", *Prosiding IEEE Transactions on Systems, Man and Cybernetics*, 1976, vol. 6, p. 325-327.
- [11] I. H. Witten & E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed., San Francisco, Morgan Kaufman, 2005.
- [12] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques", *Prosiding of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real World AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, 2007, p. 3-24.
- [13] R. Kohavi, "A Study of Cross Validation and Bootstrap for Accuracy Estimation and Model Selection", *Prosiding of the 14th International Joint Conference on Artificial Intelligence*, 1995, vol. 2 p. 1137-1143.