

Indoor Positioning System dengan Algoritma K-Means dan KNN

Hizkia Juan Suryanto^{#1}, Antonius Rachmat C.^{#2}, Yuan Lukito^{#3}

[#]Program Studi Teknik Informatika, Universitas Kristen Duta Wacana
Jl. Dr. Wahidin Sudirohusodo 5-25 Yogyakarta

¹hizkia.juan@ti.ukdw.ac.id

²anton@ti.ukdw.ac.id

³yuanlukito@ti.ukdw.ac.id

Abstract — Indoor Positioning System (IPS) can determine someone's position inside a building. The common method used is implemented by WiFi signal strength analysing. This paper discusses about how to do IPS using K-Means and K-Nearest Neighbor (KNN) method, that also analyze the accuracy. K-Means is used to cluster dataset. Each data in certain cluster then classified using KNN method. The dataset consists of 11658 Received Signal Strength (RSS) from 177 Access Point (AP) in UKDW. Accuracy of system is analyzed using 10-fold Cross Validation method which is applied in a range of $k=2$ to $k=11$ for clusterization process, then $k=1$ to $k=5$ for classification process. Based on the experiment results, system can determine someone's position with 88.49% accuracy which k optimum is 10 for clusterization process, and $k=1$ for classification process.

Keywords— Classification, Clustering, Indoor Positioning System, KNN, K-Means.

I. PENDAHULUAN

Indoor Positioning System (IPS) merupakan sistem penentuan posisi seseorang dalam konteks *indoor* (dalam gedung/ruangan). Terdapat berbagai metode untuk mengimplementasi IPS, salah satu yang populer diterapkan adalah dengan memanfaatkan *wireless LAN* (WLAN). Pada WLAN tersebut, data kuat sinyal yang terdapat pada *Access Point* (AP) yang memancarkan sinyal *WiFi* dapat diolah lebih lanjut untuk menentukan posisi akhir seseorang di suatu gedung/ruangan.

Tahap pengumpulan data telah lebih dahulu dilakukan pada penelitian sebelumnya oleh dosen dari UKDW [1]. Hasil yang diperoleh adalah 11658 data kumpulan sinyal *WiFi* dari 177 AP yang tersebar di 11 gedung di UKDW. Dalam penelitian ini dilakukan proses klusterisasi dan klasifikasi data latih tersebut untuk menentukan posisi seseorang dalam konteks *indoor*. Konteks *indoor* yang dimaksud menggunakan studi kasus lingkup Universitas Kristen Duta Wacana (UKDW). Proses klusterisasi

menggunakan metode K-Means digunakan untuk membagi data latih menjadi beberapa kluster. Selanjutnya dilakukan proses klasifikasi menggunakan metode *K-Nearest Neighbor* (KNN) yang mengacu pada kluster tertentu untuk menentukan posisi akhir. Selain itu, penelitian ini juga akan menganalisa tingkat akurasi yang dihasilkan dari penggunaan dua metode tersebut.

Penelitian ini bertujuan mengembangkan sistem yang dapat mengidentifikasi posisi seseorang dalam konteks *indoor* (UKDW) dengan menggunakan gabungan algoritma K-Means dan KNN, serta menganalisis tingkat akurasi penentuan posisi yang didapat. Hasil penelitian dapat dimanfaatkan sebagai tolak ukur mengenai seberapa akurat penggunaan kombinasi beberapa metode dibandingkan dengan satu metode tertentu dalam menentukan posisi seseorang dalam konteks *indoor*. Berdasarkan tolak ukur yang diperoleh, penelitian lebih lanjut dapat dilakukan untuk dapat mengembangkan sistem yang lebih akurat dalam menentukan posisi seseorang dalam konteks *indoor*.

II. TINJAUAN PUSTAKA

A. Indoor Positioning System (IPS)

IPS adalah suatu sistem yang dapat menentukan posisi seseorang di dalam suatu ruangan tertutup/gedung [1]. Tidak hanya itu, IPS juga mampu menentukan orientasi dan arah pergerakan seseorang [2]. Telah banyak penelitian dilakukan, namun hingga saat ini implementasi IPS yang paling umum digunakan adalah menggunakan pemanfaatan WLAN untuk menentukan posisi seseorang [3]. Teknologi WLAN cukup membutuhkan infrastruktur *wireless Access Point* (AP). Setiap *Received Signal Strength* (RSS) yang diterima perangkat dapat diolah berdasarkan AP sekitar sehingga dapat menentukan keberadaan perangkat tersebut.

B. Wireless Local Area Network (WLAN)

WLAN merupakan jaringan komputer lokal yang menggunakan koneksi nirkabel (*wireless*) yang memancarkan gelombang radio frekuensi tinggi melalui perangkat *Access Point* (AP). Setiap AP tersebut akan memancarkan informasi salah satunya berupa *Received Signal Strength* (RSS). RSS tersebut dinyatakan dalam satuan *desibel* (dB), dan digolongkan menjadi 5 kategori kualitas sinyal [4], seperti terlihat pada Tabel I.

TABEL I
KATEGORI KEKUATAN SINYAL RADIO

No	Kekuatan Sinyal	Kualitas Sinyal
1	Kurang dari -40 dB	Perkecualian
2	-40 dB hingga -55 dB	Sangat Baik
3	-55 dB hingga -70 dB	Baik
4	-70 dB hingga -80 dB	Cukup Baik
5	Lebih dari -80 dB	Buruk

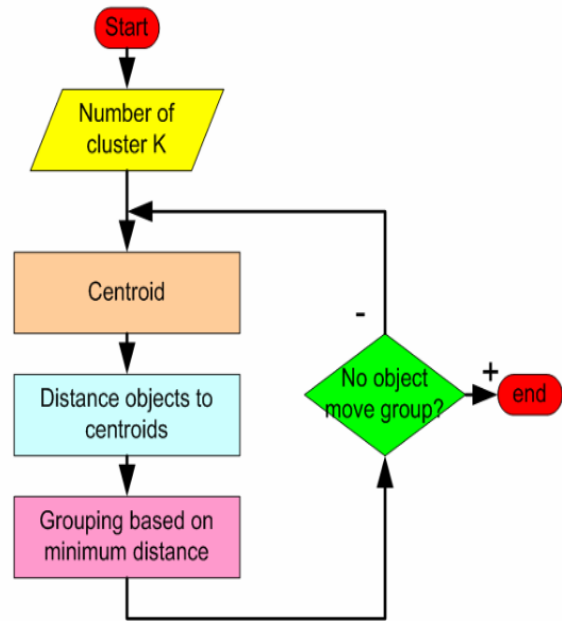
C. *K-Means*

K-Means merupakan metode klusterisasi yang membagi sekumpulan data latih menjadi beberapa kluster sesuai ciri tertentu. Prinsip utama metode ini yakni menentukan titik tengah dari kluster yang didapat, kemudian meminimalkan jarak antara titik tengah tersebut dengan data diluar kluster yang akan ditentukan klusternya [5]. Nilai *k* merepresentasikan jumlah kluster yang akan dibuat. Penentuan nilai *k* di awal iterasi proses sangat menentukan hasil klusterisasi yang didapat [6]. Langkah-langkah standar metode *K-Means* [7] seperti terlihat pada Tabel II.

TABEL III
LANGKAH - LANGKAH ALGORITMA K-MEANS

Dasar Algoritma K-Means	
1	Pilih titik sebanyak <i>k</i> sebagai <i>centroid</i> awal
2	Lakukan iterasi :
3	Bentuk kluster sebanyak <i>k</i> dengan mengambil titik terdekat <i>centroid</i>
4	Hitung ulang <i>centroid</i> setiap kluster
5	Hingga : <i>centroid</i> tidak ada perubahan

Alur kerja algoritma *K-Means* [8] seperti terlihat pada Gambar 1.



Gambar 1. Alur Kerja K-Means.

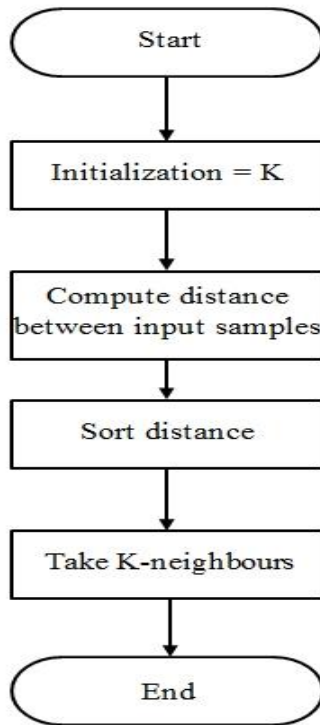
Berdasarkan alur kerja tersebut, penghitungan jarak sebuah objek ke *centroid* dapat dilakukan dengan menggunakan fungsi hitung jarak *Euclidean Distance*, seperti terlihat pada persamaan 1.

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

Dimana X dan Y merupakan sebuah vektor dengan panjang *n*, sehingga diperoleh $X = (X_1, X_2, \dots, X_n)$ dan $Y = (Y_1, Y_2, \dots, Y_n)$. Persamaan menghitung jarak kedua vektor X dan Y dari *i=1* hingga *i=n*.

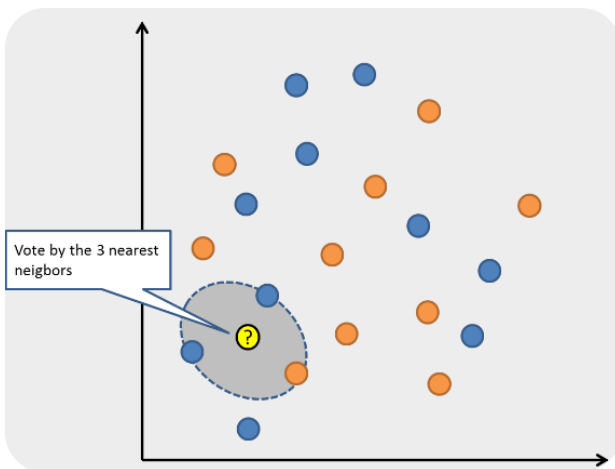
D. *K-Nearest Neighbor (KNN)*

KNN telah digunakan sejak tahun 1970 dalam perkiraan statistik dan pengenalan pola [9]. KNN merupakan salah satu metode klasifikasi *instance-based learning*, menggunakan pendekatan *supervised-learning* sehingga membutuhkan data pelatihan yang sudah dilabeli [1]. Berdasarkan data latih tersebut, satu titik dapat diklasifikasikan dengan cara menghitung jarak terdekat dengan *k*-tetangga terdekat di *class* sekitarnya. Alur kerja algoritma KNN [10] secara umum dapat dilihat pada Gambar 2.



Gambar 2. Alur Kerja KNN.

Berdasarkan Gambar 2, sebagai contoh KNN dengan nilai $k=3$ dapat dilihat pada Gambar 3.



Gambar 3. KNN dengan $k=3$.

Untuk menghitung jarak terdekat, fungsi jarak yang umum digunakan adalah *Euclidean Distance*.

E. K-Fold Cross Validation

K-Fold Cross Validation merupakan metode pengujian/validasi yang umum digunakan untuk memperkirakan performa/akurasi dari sebuah data uji [11]. Langkah-langkah validasi untuk satu kali iterasi seperti terlihat pada Tabel III.

TABEL III
LANGKAH - LANGKAH *K-FOLD CROSS VALIDATION* UNTUK SATU KALI PERULANGAN

1	Susun secara acak <i>dataset m</i>
2	Bagilah <i>dataset m</i> menjadi <i>k folds</i> (dengan demikian setiap <i>fold</i> terdapat <i>l</i> data pelatihan)
3	Lakukan perulangan, dimana $i = 1$
4	Tentukan satu data uji dari <i>l</i> data pelatihan yang ada dalam <i>fold i</i>
5	Latih data uji tersebut dengan semua data pelatihan diluar <i>fold i</i>
6	Uji data uji tersebut dengan semua data pelatihan di <i>fold i</i>
7	Hitung n_i , jumlah data pelatihan yang salah terklasifikasi di <i>fold i</i>
8	Hingga : nilai $i = k$
9	Hitung perkiraan kesalahan klasifikasi

Berdasarkan langkah tersebut, perkiraan kesalahan klasifikasi dapat dihitung dengan persamaan 2.

$$E = \frac{\sum_{i=1}^k n_i}{m} \quad \dots(2)$$

Dimana m merupakan jumlah keseluruhan *dataset*, yang kemudian dibagi menjadi sejumlah *k-folds*. Sedangkan n_i merupakan jumlah data pelatihan yang salah terklasifikasi di *fold* ke- i . Dengan demikian E yang merupakan nilai persentase kesalahan/*error percentage*, dapat diperoleh dengan cara membagi total jumlah data pelatihan yang salah terklasifikasi dari *fold* ke- i hingga k , dengan jumlah keseluruhan *dataset m*.

III. METODOLOGI PENELITIAN

A. Pengumpulan dan Persiapan Data

Tahap awal dari penelitian ini adalah tahap pengumpulan data terkait informasi yang dibutuhkan dalam *Indoor Positioning System*. Tahap pengumpulan data ini telah lebih dahulu dilakukan pada penelitian sebelumnya oleh dosen dari UKDW [1]. Hasil yang diperoleh adalah data kumpulan sinyal WiFi dari 177 AP yang tersebar di 11 gedung di UKDW.

Tahap selanjutnya adalah tahap persiapan data pelatihan. Pada tahap ini, peneliti menerima kumpulan data pelatihan

dalam format *file csv* (*Comma Separated Value*) yang berisikan detail data berupa: ID data, nama ruang publik, waktu pengambilan data, dan data kekuatan sinyal WiFi di setiap AP dalam satuan dB (*desibel*). Untuk AP yang terletak di gedung yang berbeda sebagian tidak dapat ditangkap sinyalnya untuk beberapa lokasi tertentu, kekuatan sinyalnya ditentukan dengan nilai -100, yang menunjukkan tidak adanya sinyal yang diterima.

B. Pengujian Data

Data pelatihan yang telah diperoleh nantinya diuji terlebih dahulu menggunakan metode *K-Fold Cross Validation* yang dibutuhkan pada aplikasi *clustering* pada *desktop*. Langkah pertama yakni melakukan klusterisasi data latih menggunakan algoritma K-Means dengan variabel *k* yang berbeda-beda. Setiap menentukan nilai *k*, dilanjutkan dengan menguji data pelatihan dengan algoritma KNN, pula dengan variabel *k* yang berbeda dan fungsi jarak yang berbeda. Selama proses uji dilakukan, sistem mencatat parameter yang memberikan hasil optimal pada keakuratan pengujian data pelatihan. Parameter optimal tersebut nantinya diimplementasi pada aplikasi *mobile*.

C. Implementasi

Tahap implementasi sistem dibagi menjadi dua bagian. Bagian pertama yaitu proses klusterisasi data pelatihan yang dilakukan di PC menggunakan aplikasi *desktop*. Proses klusterisasi terhadap data pelatihan menggunakan algoritma K-Means dengan parameter optimal yang telah didapat dari proses sebelumnya dalam tahap pengujian data. Hasil klusterisasi kemudian disajikan dalam format *file csv* untuk kemudian dibutuhkan dalam aplikasi *mobile*. Proses pengujian dan analisis data latih juga dilakukan dalam aplikasi *desktop* ini setelah proses klusterisasi. Dengan demikian nilai *k* optimal akan didapat secara langsung.

Bagian kedua adalah proses klasifikasi sesuai kluster yang telah didapatkan dari proses sebelumnya, dilakukan di perangkat *mobile end-user* menggunakan aplikasi *mobile*. Klasifikasi dilakukan dengan algoritma KNN dengan fungsi jarak yang digunakan adalah *Euclidean Distance*, dimana nilai *k* yang digunakan adalah sebagaimana yang telah didapat dalam tahap pengujian data. Setelah berhasil terklasifikasi, posisi akhir dari obyek dapat langsung ditentukan melalui aplikasi *mobile* tersebut.

D. Analisis

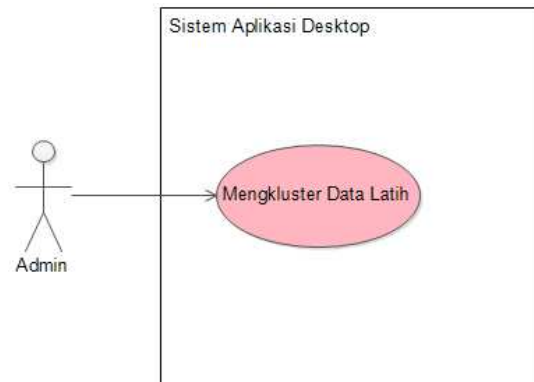
Berdasarkan pengujian yang dilakukan di aplikasi *desktop*, akurasi penentuan posisi akhir dapat dianalisis secara langsung. Proses analisis keakuratan dilakukan di PC menggunakan aplikasi *desktop* yang juga digunakan untuk proses klusterisasi. Metode pengujian yang digunakan adalah *K-Fold Cross Validation*, dengan parameter yang digunakan seperti terlihat pada Tabel IV.

TABEL IVV
PARAMETER YANG DIGUNAKAN DALAM TAHAP PENGUJIAN DAN ANALISIS

Parameter	Nilai
<i>Number of fold</i>	<i>10 folds</i>
<i>Number of iteration</i>	<i>1 times</i>
<i>Stratified Cross Validation</i>	<i>true</i>

IV. ANALISIS DAN RANCANGAN SISTEM

Pada penelitian ini akan dibahas arsitektur perancangan sistem, yaitu *Use Case Diagram*. *Use case* ini memaparkan *use case* dari masing-masing sistem (*desktop* dan *mobile*). *Use case* sistem *desktop* terdapat admin sebagai aktor, yang dapat melakukan klusterisasi data latih, seperti terlihat pada Gambar 4.



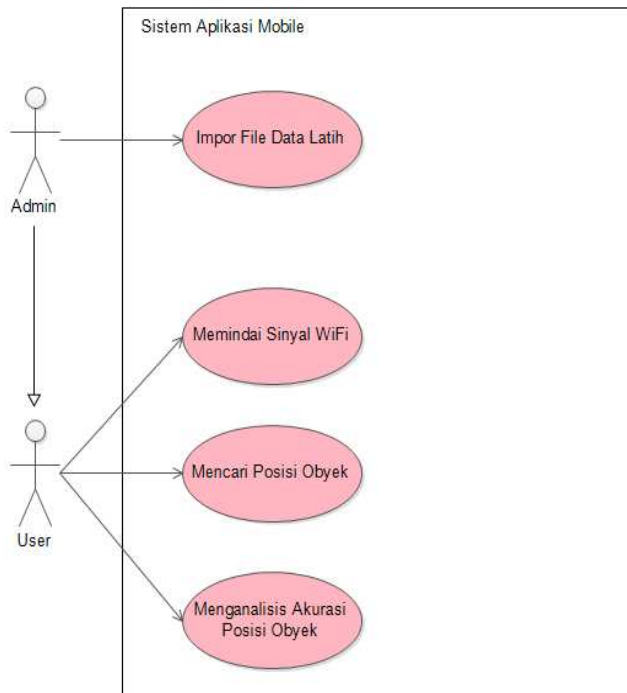
Gambar 4. *Use Case Diagram* Sistem Aplikasi *Desktop*.

Berdasarkan *use case diagram* pada Gambar 4, penjelasan skenario sukses dapat terlihat pada Tabel V.

TABEL V
USE CASE 01 - MENGLUSTER DATA LATIH

Sistem	:	Aplikasi <i>Desktop</i>
Use Case	:	UC01 - Mengkluster Data Latih
Actor	:	Admin
Precondition	:	-
Main Success Scenario	:	
1.		Admin mengimpor <i>file</i> data latih
2.		Admin menentukan jumlah atribut
3.		Sistem melakukan proses klusterisasi data latih
4.		Sistem melakukan proses uji validasi
5.		Sistem menghasilkan <i>file</i> data latih baru yang telah terkluster (optimal)

Use case sistem *mobile* terdapat admin dan pengguna/user sebagai aktor, seperti terlihat pada Gambar 5.



Gambar 5. Use Case Diagram Sistem Aplikasi Mobile.

Berdasarkan use case diagram pada Gambar 5, penjelasan skenario sukses dapat dilihat pada Tabel VI hingga Tabel IX.

TABEL VI
USE CASE 02 - IMPOR FILE DATA LATIH

Sistem	:	Aplikasi <i>Mobile</i>
Use Case	:	UC02 - Impor <i>File</i> Data Latih
Actor	:	Admin
Precondition	:	Data latih telah terkluster (UC01)
Main Success Scenario	:	
1.		Admin memilih untuk mengimpor <i>file</i> data latih
2.		Admin memilih <i>file</i> data latih yang telah terkluster

TABEL VII
USE CASE 03 - MEMINDAI SINYAL WiFi

Sistem	:	Aplikasi <i>Mobile</i>
Use Case	:	UC03 - Memindai Sinyal <i>WiFi</i>
Actor	:	<i>User</i>
Precondition	:	Data latih telah terimpor (UC02)
Main Success Scenario	:	
1.		User memilih untuk memindai sinyal <i>WiFi</i> pada posisi tertentu
2.		Sistem memindai sinyal <i>WiFi</i> disekitar obyek
3.		Sistem menampilkan daftar sinyal <i>WiFi</i> yang terdeteksi

TABEL VIII
USE CASE 04 - MENCARI POSISI OBYEK

Sistem	:	Aplikasi <i>Desktop</i>
Use Case	:	UC04 - Mencari Posisi Obyek
Actor	:	<i>User</i>
Precondition	:	Sistem telah memindai sinyal <i>WiFi</i> (UC03)
Main Success Scenario	:	
1.		<i>User</i> memilih untuk mencari posisi obyek
2.		Sistem mencocokkan hasil pindai sinyal <i>WiFi</i> dengan kuat sinyal tiap kluster pada data latih
3.		Sistem menentukan penggolongan kluster untuk hasil pindai sinyal <i>WiFi</i>
4.		Sistem melakukan proses klasifikasi pada kluster terpilih
5.		Sistem menampilkan informasi posisi akhir obyek

TABEL IX
USE CASE 05 - MENGANALISIS AKURASI POSISI OBYEK

Sistem	:	Aplikasi <i>Mobile</i>
Use Case	:	UC05 - Menganalisis Akurasi Posisi Obyek
Actor	:	<i>User</i>
Precondition	:	<i>User</i> telah memperoleh posisi obyek (UC04)
Main Success Scenario	:	
1.		<i>User</i> memilih untuk menganalisis akurasi posisi obyek
2.		Sistem melakukan analisis terhadap akurasi posisi akhir obyek
3.		Sistem menampilkan informasi akurasi posisi akhir obyek

V. HASIL PENELITIAN

Data latih yang digunakan dalam penelitian berjumlah 11658, terdiri dari 177 atribut yang merupakan kumpulan AP yang ada di UKDW. Pada tahap pengujian, data latih dibagi kedalam kluster tertentu sesuai hasil proses klusterisasi. Dalam tiap-tiap kluster tersebut, dilakukan proses uji validasi *10-Fold Cross Validation*. Nilai *k* yang diuji untuk proses klusterisasi adalah *k=2* hingga *k=11*. Sedangkan nilai *k* yang diuji untuk proses klasifikasi adalah *k=1* hingga *k=5*.

Proses implementasi sistem dilakukan dengan membuat aplikasi *desktop* dan aplikasi *mobile*. Aplikasi *desktop* digunakan untuk mengimplementasi proses klusterisasi data latih sekaligus proses pengujian. Oleh karenanya, hasil kluster yang diperoleh sudah merupakan hasil dengan pilihan *k* optimal. Aplikasi *mobile* digunakan untuk mengimplementasi proses klasifikasi data latih, untuk menentukan posisi akhir.

1. Implementasi Aplikasi Desktop

Sistem yang diimplementasikan di aplikasi *desktop* digunakan untuk proses klusterisasi data latih, dalam hal ini menggunakan metode K-Means. Pada tahap inisialisasi, pengguna harus meng-*import* terlebih dahulu *file* data latih yang akan dikluster, kemudian menentukan apakah terdapat informasi *header* pada baris pertama atau tidak. Selanjutnya pengguna menentukan jumlah atribut yang terdapat pada data latih tersebut.

Dalam kasus ini, *file* data latih yang digunakan adalah data latih berisi kumpulan sinyal *WiFi* dari 177 AP yang ada di UKDW. Dengan demikian, jumlah atribut sebanyak 177, dan jumlah kluster akan ditentukan berdasarkan pilihan *setting* yang dipilih pengguna. Jika pengguna memilih

Optimum Setting, jumlah kluster akan otomatis ditentukan oleh sistem dengan nilai *k* yang optimum. Jika *Custom Setting*, nilai *k* ditentukan secara manual oleh pengguna.

Pada tahap klusterisasi, pada iterasi pertama penentuan *centroid* dilakukan secara *random* dari data latih yang ada. Selanjutnya menghitung jarak tiap AP (total 177) ruangan tertentu dengan *centroid* menggunakan fungsi jarak *Euclidean Distance* sesuai persamaan 3.

$$dist(X, Y) = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad \dots(3)$$

Salah satu analisis perhitungan, yaitu antara ruangan Auditorium terhadap *centroid* awal *c1*, seperti ditunjukkan pada persamaan 4.

$$dist_{Auditorium, c1} = \sqrt{(-100 - (-100))^2 + (-80 - (-70))^2 + \dots + (-75 - (-85))^2} \quad \dots(4)$$

$$= 96.51795780318892$$

Setelah semua perhitungan ruangan terhadap setiap *centroid* selesai dilakukan, hasil perhitungan disimpan dalam *matrixDistance* seperti pada matrix persamaan 5.

$$D^0 = \begin{bmatrix} 0 & 100 & \dots & 67.78 \\ 98.77 & 66.72 & \dots & 0 \\ 88.98 & 76.77 & \dots & 100 \\ 78.25 & 55.32 & \dots & 100 \end{bmatrix} \quad \dots(5)$$

index : 0 1 ... 176

Setiap kolom menandakan tiap data ruangan tertentu, sedangkan baris pertama menandakan jarak ruang tertentu terhadap *centroid c1*, dan baris kedua menandakan jarak ruang tertentu terhadap *centroid c2*, dan seterusnya. Langkah selanjutnya adalah mengelompokkan data berdasarkan jarak minimum yang diperoleh berdasarkan *matrixDistance* tersebut. Dengan demikian, pada iterasi ke-0 didapat hasil seperti tertera pada matrix persamaan 6.

$$G^0 = \begin{bmatrix} grup1 & - & \dots & - \\ - & - & \dots & grup2 \\ - & grup4 & \dots & - \end{bmatrix} \quad \dots(6)$$

index : 0 1 ... 176

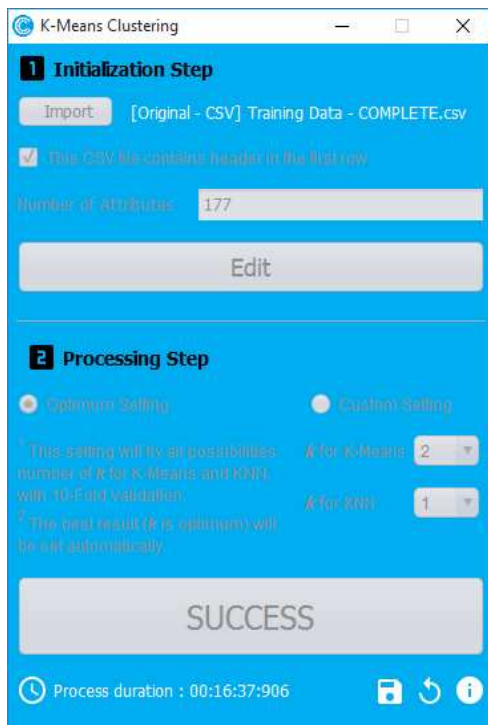
Dengan demikian iterasi pertama selesai. Tahap selanjutnya penentuan *centroid* lagi berdasarkan data terbaru

sesuai klusternya. Iterasi terus berlanjut hingga tidak ada perubahan penggolongan kluster data latih. Implementasi sistem pada aplikasi *desktop*, seperti terlihat pada Gambar 6



dan Gambar 7.

Gambar 6. Tampilan Awal Aplikasi Desktop.



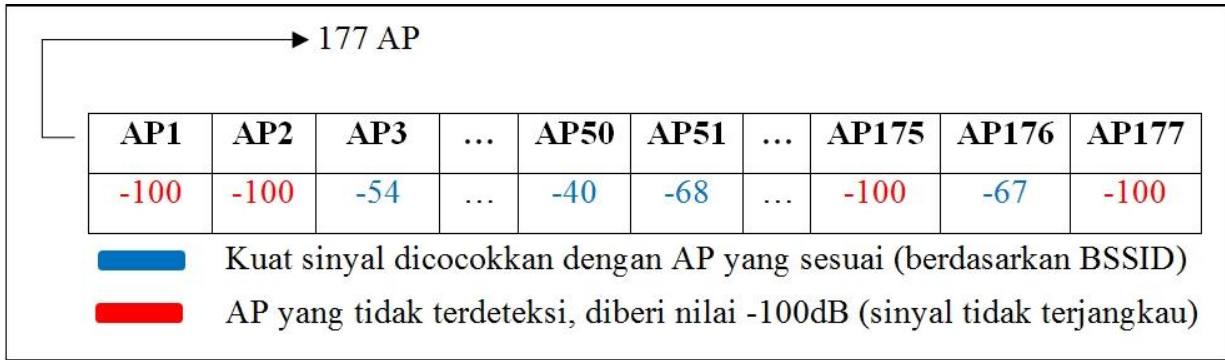
Gambar 7. Tampilan Akhir Aplikasi Desktop.

2. Implementasi Aplikasi Mobile

Sistem yang diimplementasikan di aplikasi *mobile* digunakan untuk proses klasifikasi data hasil *scan* terhadap data latih yang dibutuhkan dalam aplikasi. Proses klusterisasi untuk menentukan posisi akhir pengguna menggunakan metode KNN. Pada tahap awal menjalankan aplikasi, sistem akan mengaktifkan fitur *WiFi* perangkat *mobile*, kemudian memuat semua data yang dibutuhkan dalam proses.

Data yang dibutuhkan, antara lain: *centroid.csv*, *data_clustered.csv*, *data_room.csv*, *data_statistic.csv*, *data_bssid.csv*, dan *data_photo.csv*. Data *centroid.csv*, *data_clustered.csv*, *data_room.csv*, dan *data_statistic.csv* diperoleh dari hasil proses klusterisasi yang telah dilakukan sebelumnya menggunakan aplikasi *desktop*. Sedangkan *data_bssid.csv* digunakan untuk panduan penggolongan sinyal *WiFi* yang ditangkap perangkat supaya dicocokkan dengan AP yang sesuai. Sedangkan *data_photo.csv* berisi informasi *id* foto dan nama ruangan yang sesuai, sehingga pada tahap akhir sistem tidak hanya menampilkan nama ruangan, namun juga foto ruangan yang sesuai.

Setelah *WiFi* perangkat aktif dan sistem berhasil memuat semua data yang dibutuhkan, perangkat segera menangkap hasil *scan* sinyal *WiFi* kemudian ditampilkan dalam *List View*. Setiap kali sistem berhasil menangkap sinyal *WiFi*, maka sistem akan memproses kuat sinyal yang telah ditangkap tersebut. Kuat sinyal dalam satuan dB (*decibel*) akan dicocokkan dengan AP yang sesuai berdasarkan panduan dari *data_bssid.csv*. Jika jumlah kuat sinyal yang ditangkap kurang dari 177 (merupakan jumlah AP) maka sistem mengatur kuat sinyal lainnya sebesar -100 dB hingga jumlah kuat sinyal hasil *scan* tepat 177. Analisis proses pencocokkan kuat sinyal *WiFi* seperti terlihat pada Gambar 8.



Gambar 8. Proses Pencocokkan Kuat Sinyal.

Setelah proses *scan* dan pencocokkan kuat sinyal selesai, tombol “Where am I?” akan aktif. Saat pengguna melakukan *tap* pada tombol tersebut, maka *Activity* selanjutnya segera dijalankan, yakni untuk menentukan posisi akhir pengguna. Pada *Activity* ini, sistem akan melakukan dua kali proses KNN. Proses KNN pertama digunakan untuk menentukan penggolongan kluster, sedangkan proses KNN kedua digunakan untuk menentukan posisi akhir pengguna.

Kedua proses KNN menerapkan fungsi jarak *Euclidean Distance* untuk menghitung kedekatan data hasil *scan* dengan data latih, seperti terlihat pada persamaan 7.

$$dist(X, Y) = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad \dots\dots(7)$$

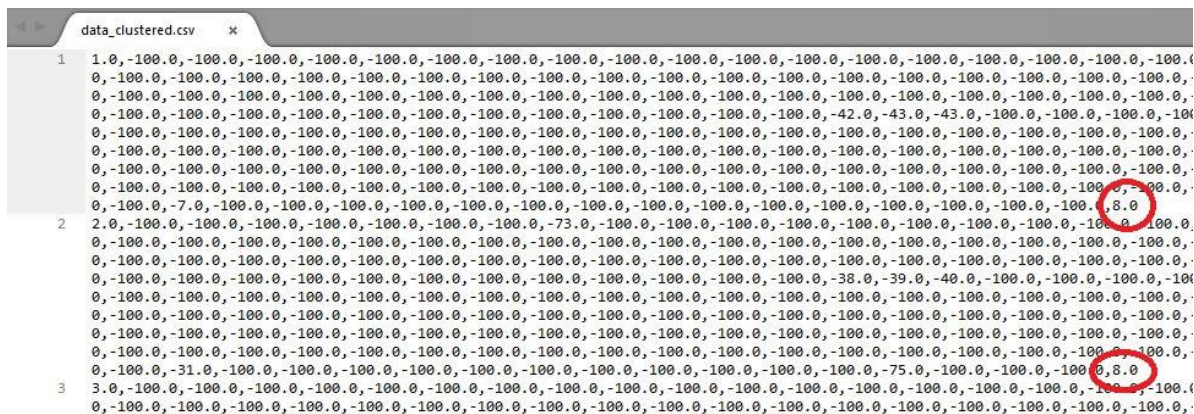
Salah satu analisis perhitungan antara data hasil *scan* terhadap data latih, seperti ditunjukkan pada persamaan 8.

$$\begin{aligned}
 & dist_{dataScanned, data_clustered} \\
 &= \sqrt{(-100 - (-55))^2 + (-48 - (-40))^2 + \dots + (-98 - (-100))^2} \quad \dots\dots(8) \\
 &= 63.77988645
 \end{aligned}$$

Tahap selanjutnya sistem akan mengambil hasil perhitungan *Euclidean Distance* yang terkecil (minimum), kemudian mengambil informasi kluster dari *data_clustered* yang digunakan sebagai data latih. Kluster terpilih inilah yang merupakan penggolongan kluster dari data hasil *scan*. Nomor kluster terletak pada bagian akhir tiap data dalam *data_clustered*, seperti terlihat pada Gambar 9.

Proses selanjutnya, sistem akan melakukan KNN yang kedua. Proses KNN kedua ini hanya dilakukan pada *data_clustered* dengan kluster terpilih saja. Kluster terpilih didapat dari proses KNN pertama yang telah dilakukan sebelumnya. Setelah proses perhitungan jarak dengan *Euclidean Distance* selesai, sistem akan mengurutkan nilai jarak yang didapat secara *ascending*, kemudian mengambil sebanyak *k* data yang memiliki nilai jarak terkecil. Nilai *k* sudah ditentukan pula dari awal sesuai data yang terdapat pada *data_statistic.csv*. Nilai *k* ini sudah merupakan nilai *k* yang optimum, yang didapat dari proses klusterisasi. Sistem akan menghasilkan klasifikasi akhir sesuai dengan nilai *majority voted* terbanyak.

Salah satu analisis penentuan klasifikasi akhir sesuai *majority voted* terbanyak dengan nilai *k* yang ditentukan adalah 5, dapat dilihat pada Tabel X.



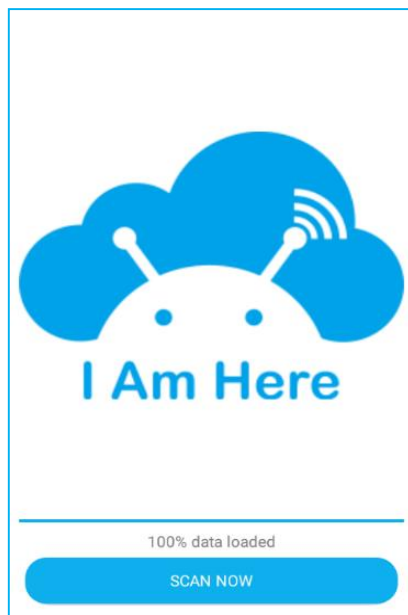
Gambar 9. Letak Nomor Kluster Tiap Data.

TABEL X
MAJORITY VOTED PADA PROSES KNN AKHIR

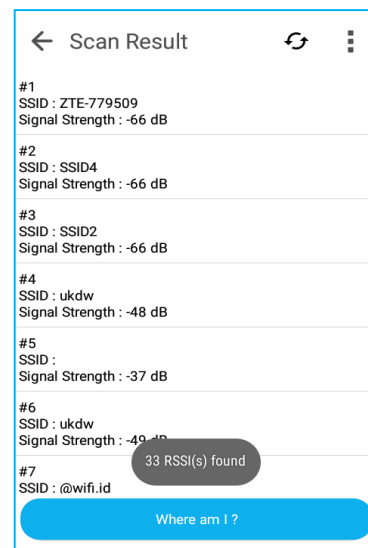
AP1	AP2	...	AP176	AP177	distance	Klasifikasi
-100	-100	...	-40	-43	68.75	Koperasi
-52	-39	...	-100	-50	74.85	Kantin
-48	-55	...	-80	-78	81.55	Kantin
-89	-80	...	-40	-38	109.48	Koperasi
-53	-51	...	-89	-100	113.78	Kantin

Dengan demikian posisi akhir sudah dapat diketahui berdasarkan hasil akhir proses KNN kedua tersebut. Tahap selanjutnya, sistem akan mencari dan menampilkan data foto yang sesuai dengan nama ruangan yang didapat. Data foto yang digunakan adalah *data_photo.csv* yang telah dimuat pada sistem. Setelah foto dan nama ruangan tertampil pada aplikasi, tombol “*See the rooms around you*” akan aktif. Pengguna dapat melakukan tap pada tombol tersebut untuk menampilkan 5 ruangan yang ada disekitar ruangan yang tertampil.

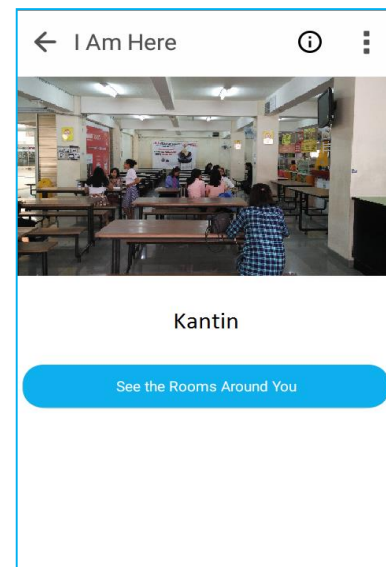
Implementasi sistem pada aplikasi *mobile* dapat dilihat pada Gambar 10 hingga Gambar 12.



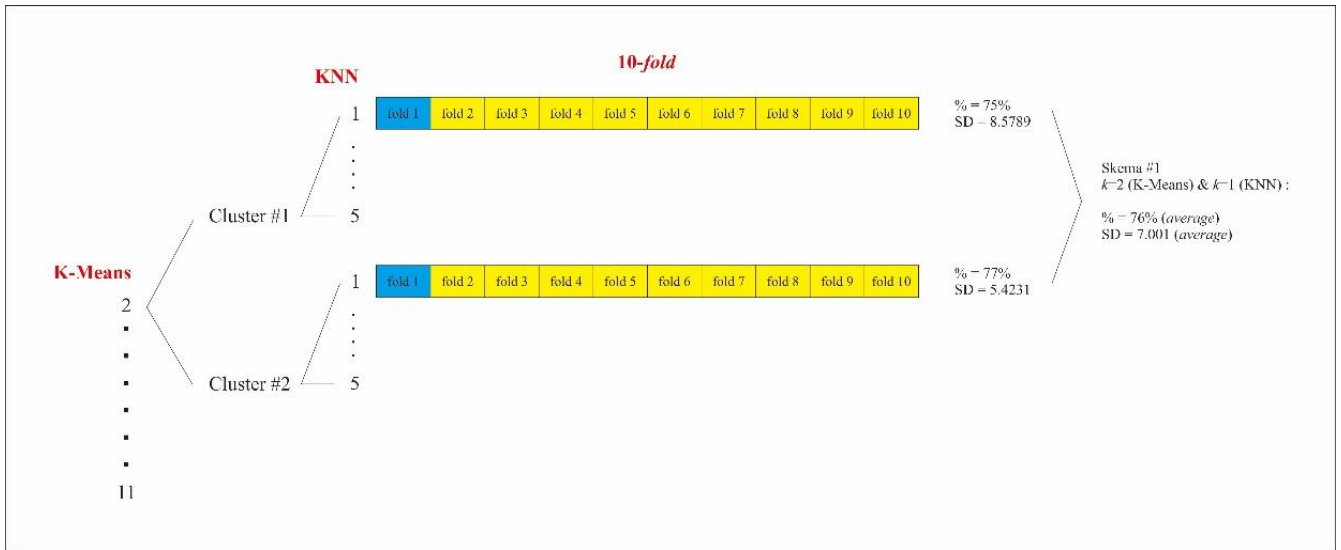
Gambar 10. Tampilan Awal Aplikasi *Mobile*.



Gambar 11. Hasil memindai sinyal *WiFi* pada Aplikasi *Mobile*.



Gambar 12. Hasil Akhir yang Diperoleh pada Aplikasi *Mobile*.



Gambar 13. Proses Uji Validasi

Proses analisis sistem juga dilakukan pada aplikasi *desktop*, sekaligus proses pengujian data latih. Dari setiap proses pengujian data, akan dicatat persentase benar (*correct percentage*) yang diperoleh. Persentase tersebut mengindikasikan seberapa akurat pilihan *k* untuk proses klusterisasi dan klasifikasi dalam menentukan posisi akhir. Persentase terbaik mengindikasikan pilihan *k* yang optimal untuk proses klusterisasi dan klasifikasi. Gambaran secara umum proses uji validasi dan analisis seperti terlihat pada Gambar 13.

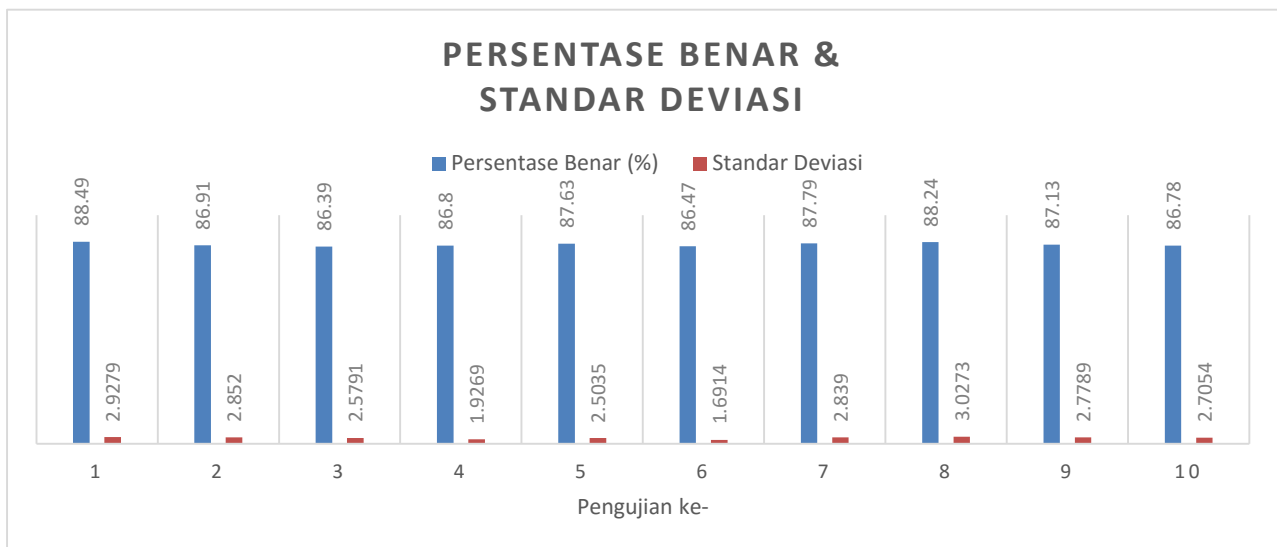
Parameter-parameter yang digunakan dalam proses pengujian, dapat dilihat pada Tabel XI.

TABEL XI

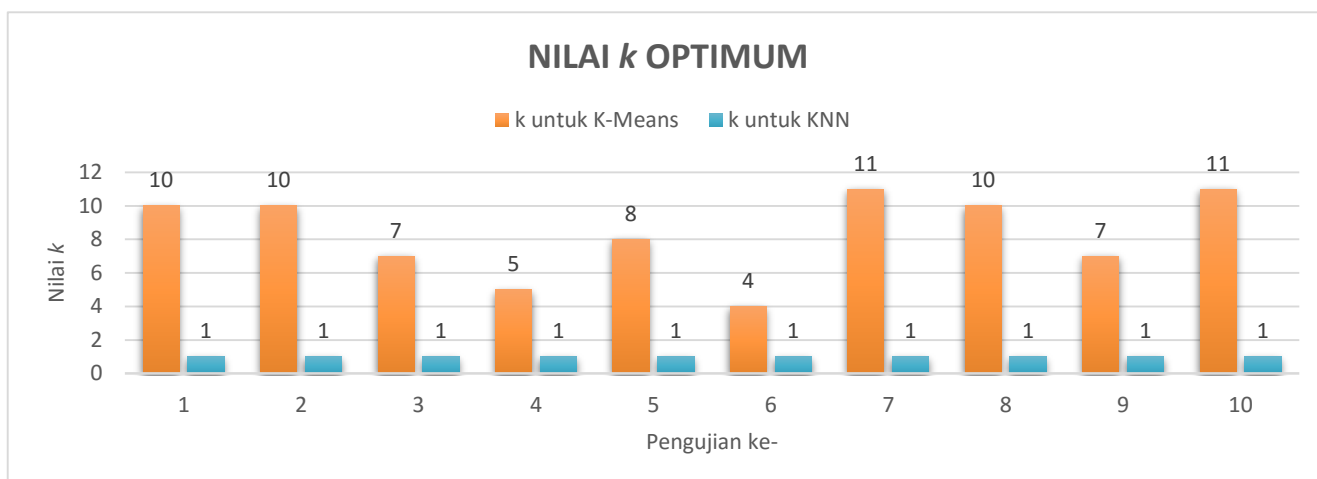
PARAMETER YANG DIGUNAKAN DALAM TAHAP PENGUJIAN DAN ANALISIS

Parameter	Nilai
Number of fold	10 folds
Number of iteration	1 times
Stratified Cross Validation	true

Dalam satu kali proses pengujian, dilakukan 1 kali iterasi proses 10-Fold Cross Validation. Hasil pengujian tersebut menghasilkan persentase benar (*correct percentage*) yang merupakan tingkat akurasi penentuan posisi. Proses pengujian dilakukan sebanyak 10 kali. Dari 10 kali proses pengujian tersebut, hanya akan diambil data hasil pengujian yang memiliki persentase benar terbesar. Hasil proses pengujian selengkapnya dapat dilihat pada Gambar 14.



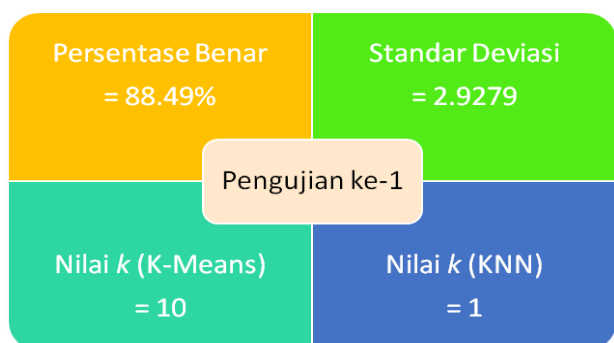
Gambar 14. Persentase Benar dan Standar Deviasi



Gambar 15. Nilai k Optimum

Berdasarkan Gambar 14, data hasil pengujian ke-1 merupakan yang paling optimum, dimana persentase benar mencapai 88.49%, dengan standar deviasi sebesar 2.9279. Nilai k optimum yang didapat dari setiap proses pengujian dapat dilihat pada Gambar 15.

Oleh karena persentase benar terbesar diperoleh saat proses pengujian ke-1, maka nilai k optimal untuk proses klusterisasi (K-Means) adalah 10, sedangkan nilai k optimal untuk proses klasifikasi (KNN) adalah 1. Data statistik hasil optimal dapat disimpulkan seperti terlihat pada Gambar 16.



Gambar 16. Hasil Optimal

VI. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan, berikut kesimpulan yang diperoleh:

1) Sistem berhasil mengimplementasi algoritma klusterisasi K-Means dan klasifikasi KNN untuk menentukan posisi seseorang dalam konteks *indoor*.

2) Berdasarkan hasil pengujian sistem menggunakan metode *K-Fold Cross Validation*, diperoleh tingkat akurasi sistem sebesar 88.49%.

3) Berdasarkan tingkat akurasi yang dihasilkan sistem, nilai k optimal yang didapat adalah $k=10$ untuk tahap klusterisasi, dan $k=1$ untuk tahap klasifikasi. Dengan menggunakan nilai k optimal tersebut, aplikasi akan lebih akurat dalam menentukan posisi

Berdasarkan kesimpulan yang didapat, berikut saran pengembangan yang dapat diterapkan pada sistem:

1) Pada tahap klusterisasi data latih, proses penentuan *centroid* awal yang dilakukan secara *random* dapat dioptimasi lebih lagi. Hal ini sangat berpengaruh pada kecepatan pemrosesan dan juga akurasi penggolongan kluster data uji.

2) Keakuratan data latih dapat dioptimasi lebih lagi dengan cara konsisten secara berkala diperbaharui menyesuaikan kondisi yang ada (penambahan/perubahan letak ruangan maupun *Access Point*).

DAFTAR PUSTAKA

- [1] Yuan Lukito and Antonius R Chrismanto, "Perbandingan Metode-Metode Klasifikasi Untuk Indoor Positioning System," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 1, no. 2, pp. 123-131, 2015.
- [2] Avik Ghose et al., "Indoor Positioning System," 14/450,890, February 5, 2015.
- [3] Aare Puusaar, *Indoor Positioning Using WLAN Fingerprinting with Post-Processing Scheme*. Tartu, 2014.
- [4] Veris Industries. (2013) Veris White Paper. [Online]. http://www.veris.com/docs/whitePaper/vwp18_RSSI_RevA.pdf
- [5] Tacha Serif and Bulut Altintas, "Improving RSS-Based Indoor Positioning Algorithm via K-Means Clustering," in *Wireless Conference 2011 - Sustainable Wireless Technologies (European Wireless)*, 11th European, Istanbul, 2011.
- [6] Anand M Baswade and Prakash S Nalwade, "Selection of Initial Centroids for k-Means," *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 7, pp. 161-164, July 2013.
- [7] Zijun Zhang. (2012, Mar.) K-means Algorithm Cluster Analysis in Data Mining.

- [8] Kardi Teknomo, "K-Means Clustering Tutorial," 2007.
- [9] Young-Seuk Park et al., Eds., *Advanced Modelling Techniques Studying Global Changes in Environmental Sciences*. Copenhagen, Denmark: Elsevier, 2015, vol. 27.
- [10] Divya Kadyan and Sanjay Singla, "Sentiment Analysis," *International Journal of Latest Trends in Engineering and Technology*, vol. 7, no. 3, pp. 238-244, September 2016.
- [11] National Taiwan University. (2008, Dec.) k-fold Cross Validation. [Online].
<http://www.csie.ntu.edu.tw/~b92109/course/Machine%20Learning/Cross-Validation.pdf>