Digital Repository Universitas Jember

IMPLEMENTASI *INDOOR POSITIONING SYSTEM (IPS)* MENGGUNAKAN ALGORITMA *WEIGHTED K-NEAREST NEIGHBOR* DI GEDUNG A FAKULTAS TEKNIK UNIVERSITAS JEMBER

SKRIPSI

Oleh:

Yudan Aries Maulana

NIM 121910201106

PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2019

Digital Repository Universitas Jember

IMPLEMENTASI *INDOOR POSITIONING SYSTEM (IPS)* MENGGUNAKAN ALGORITMA *WEIGHTED K-NEAREST NEIGHBOR* DI GEDUNG A FAKULTAS TEKNIK UNIVERSITAS JEMBER

SKRIPSI

Diajukan guna memenuhi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Teknik Elektro (S1)
dan mencapai gelar Sarjana Teknik

Oleh:

Yudan Aries Maulana

NIM 121910201106

PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2019

Digital Repository Universitas Jember

Persembahan

Skripsi ini saya persembahkan untuk:

1. Kedua orang tua saya, Mohammad Imam Wahyudi, Siti Bardiana Noor, Adik-adik, serta teman-teman seperjuangan. Terimakasih atas dukungan, doa, dan telah membantu menemani hingga penelitian ini dapat diselesaikan.
2. Guru - guru SD Negeri Badean 1 Bondowoso, SMP Negeri 2 Bondowoso, SMK Negeri 1 Bondowoso dan semua Dosen - dosen program studi Teknik Elektro Universitas Jember.
3. Almamater yang saya banggakan, Program Studi Teknik Elektro Fakultas Teknik Universitas Jember.

Digital Repository Universitas Jember

Motto

Everything is OK in the end.
If it's not OK, then it's not the end.

(Anonymous)

Digital Repository Universitas Jember

PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Yudan Aries Maulana

NIM : 121910201106

menyatakan dengan sesungguhnya bahwa karya tulis ilmiah yang berjudul "IMPLEMENTASI *INDOOR POSITIONING SYSTEM (IPS)* MENGGUNAKAN ALGORITMA *WEIGHTED K-NEAREST NEIGHBOR* DI GEDUNG A FAKULTAS TEKNIK UNIVERSITAS JEMBER" adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Jember, 27 Desember 2019
Yang menyatakan,

Yudan Aries Maulana
NIM. 121910201106

Digital Repository Universitas Jember

SKRIPSI

IMPLEMENTASI *INDOOR POSITIONING SYSTEM (IPS)* MENGGUNAKAN ALGORITMA *WEIGHTED K-NEAREST NEIGHBOR* DI GEDUNG A FAKULTAS TEKNIK UNIVERSITAS JEMBER

Oleh:

Yudan Aries Maulana

NIM 121910201106

Pembimbing:

Dosen Pembimbing Utama : Dodi Setiabudi, S.T., M.T.
NIP. 198405312008121004

Dosen Pembimbing Anggota : Widya Cahyadi, S.T., M.T.
NIP. 198511102014041001

Digital Repository Universitas Jember

PENGESAHAN

Skripsi berjudul "**IMPLEMENTASI *INDOOR POSITIONING SYSTEM (IPS)* MENGGUNAKAN ALGORITMA *WEIGHTED K-NEAREST NEIGHBOR* DI GEDUNG A FAKULTAS TEKNIK UNIVERSITAS JEMBER**" telah diuji dan disahkan pada:

Hari, tanggal : Jum'at, 27 Desember 2019

Tempat : Ruang Ujian 1 Fakultas Teknik Universitas Jember

Dosen Pembimbing Utama,	Dosen Pembimbing Anggota,
Dodi Setiabudi, S.T., M.T.	**Widya Cahyadi, S.T., M.T.**
NIP. 198405312008121004	**NIP. 198511102014041001**

Penguji 1,	Penguji 2,
Andrita Ceriana Eska, S.T., M.T.	**Wahyu Muldayani, S.T., M.T.**
NRP. 760014640	**NRP. 760016799**

Mengesahkan
Dekan,

Dr. Ir. Entin Hidayah, M.U.M.
NIP. 196612151995032001

Digital Repository Universitas Jember

Implementasi *Indoor Positioning System (IPS)* Menggunakan Algoritma *Weighted k-Nearest Neighbor* di Gedung A Fakultas Teknik Universitas Jember

Yudan Aries Maulana

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

ABSTRAK

Indoor Positioning System (IPS) merupakan solusi dari tidak dapat digunakannya *Global Positioning System (GPS)* dalam penentuan posisi lokasi dalam gedung. Salah satu teknologi yang dapat digunakan dalam penentuaan dalam gedung yaitu memanfaatkan *Received Signal Strength (RSS)* yang dipancarkan oleh *acces point*. Pada penelitian ini menggunakan 6 buah *accces point* yang telah terpasang di gedung. Algoritma *Weighted k-Nearest Neighbor(Weighted k-NN)* digunakan sebagai optimasi penentuan lokasi, algoritma ini merupakan pengembangan dari *k-Nearest Neighbor(k-NN)* dengan penambahan pembobotan. Pengujian sistem ini dilakukan dengan membandingkan performa algoritma *Weighted k-NN* dengan *k-NN*. Kemudian membandingkan pengaruh parameter *reference point* dan nilai *k* (jumlah tetangga terdekat) pada algoritma *Weighted k-NN*. Dari hasil pengujian performa algoritma *Weighted k-NN* terhadap *k-NN* didapatkan penyimpangan rata – rata setelah pembobotan sebesar 1,49 meter dan sebelum pembobotan sebesar 2,7 meter. Pada hasil uji algoritma *Weighted k-NN* terhadap jumlah *reference point* didapatkan hasil sebesar 3,84 meter saat *reference point* berjumlah 20 dan 1,49 meter saat *reference point* berjumlah 40. Kemudian hasil uji dengan perubahan nilai *k*, didapatkan hasil penyimpangan yang semakin meningkat terhadap banyaknya nilai *k*, saat *k* bernilai 1 penyimpangan rata – rata sebesar 0,26 meter dan meningkat menjadi 5,64 meter saat *k* bernilai 6.

Kata Kunci: *Indoor Positioning, Weighted k-Nearest Neighbor, k-Nearest Neighbor, Access Point.*

Digital Repository Universitas Jember

Implementation of Indoor Positioning System (IPS) Using the Weighted k-Nearest Neighbor Algorithm in Building A Faculty of Engineering University of Jember

Yudan Aries Maulana

Electrical Engineering, Engineering Faculty, Jember University

ABSTRACT

Indoor Positioning System (IPS) is a solution to the inability to use the Global Positioning System (GPS) in determining the position of locations in buildings. One technology that can be used in building determination is to utilize Received Signal Strength (RSS) emitted by access points. In this study, using 6 access points that have been installed in the building. Weighted k-Nearest Neighbor (Weighted k-NN) algorithm is used as an optimization of location determination, this algorithm is the development of k-Nearest Neighbor (k-NN) with the addition of weighting. The testing of this system is done by comparing the performance of the Weighted k-NN algorithm with k-NN. Then compare the effect of the reference point parameter and the value of k (number of nearest neighbors) on the Weighted k-NN algorithm. From the results of testing the performance of the Weighted k-NN algorithm against k-NN, it is found that the average deviation after weighting is 1.49 meters and before weighting is 2.7 meters. In the Weighted k-NN algorithm test results on the number of reference points obtained results of 3.84 meters when the reference point numbered 20 and 1.49 meters when the reference point numbered 40. Then the test results with a change in the value of k, obtained the results of increasing deviations towards the number of k values, when k is worth 1 deviation, is 0.26 meters and increases to 5.64 meters when k is 6.

Keywords: *Indoor Positioning, Weighted k-Nearest Neighbor, k-Nearest Neighbor, Access Point.*

Digital Repository Universitas Jember

RINGKASAN

Implementasi *Indoor Positioning System (IPS)* Menggunakan Algoritma *Weighted k-Nearest Neighbor* di Gedung A Fakultas Teknik Universitas Jember; Yudan Aries Maulana, 121910201106; 2019; 73 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Perkembangan *smartphone* saat ini semakin pesat, ditandai dengan hadirnya berbagai merek dan tipe dengan harga yang terjangkau masyarakat. Dengan meningkatnya pengguna *smartphone*, maka kebutuhan akan akses informasi melaui *internet* semakin bertambah. Sehingga tempat - tempat umum seperti rumah sakit, pusat perbelanjaan, lingkungan kampus, dan lain sebagainya menyediakan *Wireless Local Area Network (WLAN)* yang dapat digunakan *smartphone* mengakses *internet*. Dengan tersedianya jaringan tersebut, maka dapat dimanfaatkan untuk menentukan posisi dalam gedung dengan memanfaatkan sinyal yang dipancarkan.

Dari hasil pengujian performa algoritma *Weighted k-NN* terhadap *k-NN* didapatkan penyimpangan rata – rata setelah pembobotan sebesar 1,49 meter dan sebelum pembobotan sebesar 2,7 meter, terbukti bahwa algoritma *Weighted k-NN* lebih akurat terhadap algoritma *k-NN*. Kemudian pada hasil uji algoritma *Weighted k-NN* terhadap jumlah *reference point* didapatkan hasil sebesar 3,84 meter saat *reference point* berjumlah 20 dan 1,49 meter saat *reference point* berjumlah 40, jumlah *reference point* mempengaruhi hasil uji lokasi, semakin banyak *reference point* maka tingkat akurasi semakin meningkat. Kemudian hasil uji dengan perubahan nilai *k*, didapatkan hasil penyimpangan yang semakin meningkat terhadap banyaknya nilai *k*, saat *k* bernilai 1 penyimpangan rata – rata sebesar 0,26 meter dan meningkat menjadi 5,64 meter saat *k* bernilai 6, hal ini membuktikan nilai *k* dapat mempengaruhi hasil uji lokasi, semakin besar nilai *k* penyimpangannya semakin besar.

Digital Repository Universitas Jember

PRAKATA

Puji syukur kehadirat Allah SWT yang maha kuasa atas segalanya, karena dengan ridho, hidayah dan petunjuk-Nya, penulis dapat menyelesaikan skripsi ini. Selama penyusunan skripsi ini penulis mendapat bantuan berbagai pihak yang turut memberikan bantuan berupa motivasi, inspirasi, bimbingan, doa, fasilitas dan dukungan lainnya yang membantu memperlancar pengerjaan skripsi ini. untuk itu penulis mengucapkan terimakasih kepada.

1. Ibu Dr. Ir. Entin Hidayah, M.U.M., Selaku Dekan Fakultas Teknik Universitas Jember;
2. Bapak Dr. Bambang Sri Kaloko, S.T., M.T., Selaku Ketua Jurusan Teknik Elektro Universitas Jember;
3. Bapak Dodi Setia Budi, S.T., M.T. dan bapak Widya Cahyadi, S.T., M.T., selaku dosen pembimbing yang telah membimbing tugas akhir ini;
4. Bapak Andrita Ceriana Eska, S.T., M.T. dan bapak Wahyu Muldayani, S.T., M.T., selaku dosen penguji yang sudah memberikan saran untuk memperbaiki tugas akhir ini;
5. Kedua Orang tua Bapak Mohammad Imam Wahyudi dan Ibu Siti Bardiana Noor, yang telah membesarkan, mendidik, mendoakan tiada henti, memberi motivasi semangat, menitikkan air mata dan memberi kasih sayang yang tak pernah habis serta pengorbanannya selama ini;
6. Saudara-saudaraku Sagita Dwi Ambarwati, Mohammad Iqbal Agung Nugroho, dan Mohammad Andi Yasin;
7. Keluarga besar teknik eletro 2012 SATE UJ Serta semua pihak yang tidak dapat disebutkan satu per satu, yang telah mendukung dalam penyelesaian skripsi ini.

Jember, 27 Desember 2019

Penulis

Digital Repository Universitas Jember

DAFTAR ISI

Digital Repository Universitas Jember

Digital Repository Universitas Jember

Digital Repository Universitas Jember

DAFTAR TABEL

Digital Repository Universitas Jember

DAFTAR GAMBAR

Digital Repository Universitas Jember

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Teknologi merupakan suatu sarana yang digunakan oleh manusia dalam memudahkan suatu pekerjaan dalam kehidupan sehari - hari, salah satunya dalam bidang navigasi. Navigasi adalah penentuaan posisi dan arah tujuan perjalanan baik di medan sebenarnya maupun di peta mengunakan petunjuk arah. Kompas merupakan alat untuk menunjukan arah dengan memanfaatkan sifat kemagnetan bumi yang selalu menunjukan arah utara dan selatan. Sebelum teknologi kompas ditemukan, manusia menggunakan benda - benda langit seperti matahari, bulan, dan bintang. Dengan menggunakan peta dan kompas kita dapat menentukan posisi dalam peta.

Global Positioning System (GPS) adalah sistem navigasi berbasis satelit yang dikembangkan oleh Departemen Pertahanan Amerika. Sistem ini menggunakan 24 satelit yang mengirimkan gelombang sinyal mikro, kemudian sinyal ini diterima oleh alat penerima yang ada di permukaan bumi dan dapat digunakan sebagai penentuan posisi, kecepatan, arah, dan waktu (Suryanto, 2012). Penentuan posisi menggunakan GPS dapat dihandalkan dan akurat khususnya bila digunakan di ruangan terbuka *(outdoor)*. Namun sistem ini tidak dapat dihandalkan bila digunakan di dalam gedung *(indoor)* dikarenakan sinyal GPS terhalang bangunan (Ginardi, 2013).

Indoor Positioning System (IPS) merupakan teknologi untuk menentukan posisi suatu objek di dalam gedung, sistem ini didasari dari tidak dapat digunakannya GPS untuk menentukan posisi di dalam gedung. Banyak sekali teknologi yang dapat digunakan diantaranya adalah *Wireless Local Area Network (WLAN), bluetooth*, jaringan seluler, *Ultra Wideband (UWB),* dan *Near Field Electromagnetic Ranging (NFER)* (Bakal, 2016).

Perkembangan *smartphone* saat ini semakin pesat, ditandai dengan hadirnya berbagai merek dan tipe dengan harga yang terjangkau masyarakat. Dengan meningkatnya pengguna *smartphone,* maka kebutuhan akan akses informasi melaui *internet* semakin bertambah. Sehingga tempat - tempat umum seperti rumah sakit,

Digital Repository Universitas Jember

pusat perbelanjaan, lingkungan kampus, dan lain sebagainya menyediakan *Wireless Local Area Network (WLAN)* yang dapat digunakan *smartphone* mengakses *internet*. Dengan tersedianya jaringan tersebut, maka dapat dimanfaatkan untuk menentukan posisi dalam gedung dengan memanfaatkan sinyal yang dipancarkan. *Received Signal Strength (RSS)* merupakan kekuatan sinyal pada penerima yang berkurang secara proposional berdasar jarak penerima dengan pemancar. Apabila hubungan kekuatan sinyal dan jarak diketahui, maka posisi dapat ditentukan (Sutarti, 2015).

Beberapa penelitian mengenai *Indoor Positioning System* telah banyak dikembangkan dengan berbagai macam metode guna meningkatkan akurasi dalam menentukan posisi, seperti penelitian yang dilakukan oleh Mochamad Susantok, Ary Kurniawan, dan Hamid Azwar (2013) yang bejudul "*Wifi Positioning System (WPS)* Menggunakan Algoritma *Neural Network Backpropagation* di Area Kampus Politeknik Caltex Riau". Pada penelitian tersebut menggunakan tiga buah *acces point*, 18 *reference point* dan algoritma *Neural Network Backpropagation* untuk meningkatkan keakuratan penentuan posisi dan hasil yang didapatkan tingkat akurasi mencapai 44% dengan toleransi 2 meter serta tingkat akurasi 64% dengan toleransi 4 meter dalam kondisi diam dan dalam kondisi berjalan tingkat akurasi meningkat menjadi 50% pada toleransi 2 meter dan 83,3% pada toleransi 4 meter.

Penelitian selanjutnya dilakukan oleh Rindi Anggita Sari, Mochamad Susantok, dan Muhammad Diono (2014) yang berjudul "*Wireless Positioning System (WPS)* Menggunakan Algoritma *k-Nearest Neighbor* (k-NN) di area Kampus Politeknik Caltex Riau". Pada penelitian ini tetap menggunakan 3 buah *access point* dan algoritma yang digunakan adalah *k-Nearest Neighbor* (k-NN), hasil penelitian ini menunjukkan tingkat akurasi 80% dengan toleransi 2 meter serta 100% dengan toleransi 4 meter dalam kondisi aktif. Kemudian pada kondisi nonaktif didapat tingkat akurasi 90% pada toleransi 2 meter dan akurasi 100% pada toleransi 6 meter.

Kemudian penelitian yang dilakukan oleh Fananda Herda Perdana, R.V. Hari Ginardi, dan F.X. Arunanto (2016) yang berjudul "Implementasi *Indoor Positioning System* Berbasis *Smartphone* dengan Penambahan *Access Point* untuk

Digital Repository Universitas Jember

Studi Kasus Gedung Teknik Informatika ITS". Pada penelitian ini melakukan penambahan *access point* guna menambah tingkat akurasi serta menggunakan algoritma *Clustering Filtered K-Nearest Neighbors (CFK)*, hasil yang didapatkan tingakat akurasi meningkat hingga 16,67% untuk seluruh titik uji yang dilakukan. Hal ini membuktikan penambahan *access point* mampu meningkatkan tingkat akurasi

Berdasarkan beberapa penelitian yang telah dilakukan tersebut penulis mencoba melakukan implementasi *Indoor Positioning Sysytem (IPS)* mengunakan algoritma yang berbeda yaitu *Weighted k-Nearest Neighbor (Weighted k-NN)*, algoritma ini merupakan pengembangan dari *k-Nearest Neighbor*. *Weighted k-Nearest Neighbor* menggunakan pembobotan pada kelas data, bobot yang akan diberikan pada kelas yang minoritas lebih besar dan bobot lebih sedikit pada tetangga yang berasal dari kelas yang mayoritas (Indriati & Ridok, 2016).

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan diatas dapat dirumuskan beberapa masalah, diantaranya :

1. Bagaimana performa algoritma *Weighted k-Nearest Neighbor* terhadap tingkat akurasi penetuaan posisi dalam gedung.
2. Bagaimana pengaruh parameter berupa jumlah *reference point* dan nilai *k* terhadap tingkat akurasi.

1.3 Batasan Masalah

Berdasarkan rumusan masalah yang telah dijabarkan, maka penelitian ini dibatasi pada hal - hal sebagai berikut :

1. Penelitian dilakukan di lantai 3 gedung A Fakultas Teknik Universitas Jember.
2. Menggunakan protokol *Wireless* LAN IEEE 802.11.
3. Menggunakan *Android* dengan sistem operasi minimal *Lolipop* 5.0.
4. Menggunakan *Acces Point* yang telah terpasang di gedung.

Digital Repository Universitas Jember

1.4 Tujuan Penelitian

Adapun tujuan penelitian yang ingin dicapai adalah sebagai berikut :

1. Mengetahui performa algoritma *Weighted k-Nearest Neighbor* terhadap tingkat akurasi penetuaan posisi dalam gedung.
2. Mengetahui pengaruh parameter berupa jumlah *reference point* dan nilai *k* terhadap tingkat akurasi.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut :

1. Mampu mengetahui teknik untuk menentukan posisi objek dalam gedung menggunakan *Wireless Local Area Network (WLAN)* secara akurat.
2. Mampu memberikan informasi lokasi pengguna dalam gedung secara langsung dan akurat menggunakan *smartphone*.
3. Sebagai referensi penelitian selanjutnya.

1.6 Sistematika Penulisan

Secara garis besar penyusunan proposal skripsi ini adalah sebagai berikut :

BAB 1. PENDAHULUAN

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan pembahasan, manfaat pembahasan, dan sistematika pembahasan.

BAB 2. TINJAUAN PUSTAKA

Berisi penjelasan tentang teori yang berhubungan dengan penelitian.

BAB 3. METODOLOGI PENELITIAN

Berisi tentang metode dan langkah - langkah menyelesaikan penelitian.

BAB 4. HASIL DAN PEMBAHASAN

Berisi tentang hasil penelitian dan analisa hasil penelitian.

BAB 5. PENUTUP

Berisi tentang kesimpulan dan saran dari penulis.

DAFTAR PUSTAKA

LAMPIRAN

Digital Repository Universitas Jember

BAB 2. TINJAUAN PUSTAKA

Tinjauan pustaka sebagai daftar acuan dalam melakukan penelitian tugas akhir yang saat ini dikerjakan. Tinjauan pustaka berguna dapat memperdalam wawasan dan mengembangkan ilmu pengetahuan yang telah ada, sehingga bidang keilmuan penelitian yang diteliti akan mengalami perkembangan dan peningkatan.

2.1 Hasil Penelitian Sebelumnya

Matriks Permasalahan ini dapat memberikan wacana bagi peneliti agar dapat mengulas lebih dalam tentang topik yang diteliti serta dapat memberikan masukan lebih berarti bagi permasalahan yang diteliti. Informasi selengkapnya mengenai pustaka penelitian peneliti lain serta pembahasan terkait solusi, saran terhadap kesamaan konsep dengan penelitian ini dapat dilihat pada Tabel 2.1.

Tabel 2.1 Hasil Penelitian Sebelumnya

No	Pustaka	Solusi	Metode	Hasil	Saran
1.	*Wifi Positioning System (WPS)* Menggunakan Algoritma *Neural Network Backpropagation* di Area Kampus Politeknik Caltex Riau (Susantok Mochamad, dkk 2013).	Melakukan perancangan *Wifi Positioning System (WPS)* menggunakan 3 buah *acces point.*	Proses pengujian terhadap akurasi posisi, menggunakan algoritma *Neural Network Backpropagation.*	Menghasilkan prosentase 44 % dengan akurasi 2 meter dan 64% untuk akurasi 4 meter.	Perlunya penambahan *acces point.*

Digital Repository Universitas Jember

2.	Implementasi *Indoor Positioning System* Berbasis *Smartphone* dengan Penambahan *Access Point* untuk Studi Kasus Gedung Teknik Informatika ITS (Ginardi Hari, dkk 2016).	Melakukan penambahan *acces point menggunakan* untuk meningkatkan akurasi.	Menggunakan Algoritma *k-Nearest Neighbor (k-NN).*	Setelah dilakukan penambahan *access point* di beberapa titik lokasi yang memiliki cakupan sinyal *Wi-Fi* lemah, didapatkan hasil akurasi yang meningkat hingga 16,67%, dari akurasi semula 78,70% menjadi 95,36% untuk seluruh titik uji coba pada gedung kampus.	Sebagian sudut ruangan memiliki cakupan sinyal lemah.

Digital Repository Universitas Jember

3.	*Wireless Positioning System (WPS)* Menggunakan Algoritma *k-Nearest Neighbor (k-NN)* di area Kampus Politeknik Caltex Riau (Sari Rindi, dkk 2014)	Melakukan perancangan *Wireless Positioning System (WPS)* dengan spesifikasi 4 arah mata angin	Menggunakan Algoritma *k-Nearest Neighbor (k-NN)*	k-NN dapat dijadikan suatu metode untuk penentuan letak posisi *client*. Tingkat keakurasian untuk penentuan letak posisi *client* mencapai ≤ 100%.	Kedepannya dapat digunakan *acces point* yang lebih menunjang yakni dari segi daya pancar dan kestabilan.

2.2 *Indoor Positioning System (IPS)*

Indoor Positioning System merupakan sistem untuk menemukan objek atau orang didalam bangunan yang menggunakan gelombang radio, *magnetic fields*, *acoustic signal*, maupun sensor lainnya yang dapat mengirim informasi melalui perangkat bergerak (Setyawan, 2015). Sistem ini didasari dari tidak dapat digunakannya *Global Positioning System (GPS)* untuk menentukan posisi di dalam gedung. Karena pelemahan sinyal yang disebabkan oleh bahan - bahan konstruksi, *Global Positioning System (GPS)* yang menggunakan satelit kehilangan daya yang signifikan di dalam ruangan, yang mempengaruhi cakupan yang diperlukan untuk penerima. Selain itu, beberapa refleksi di permukaan menyebabkan propagasi multi jalur yang mengakibatkan kesalahan penentuan lokasi. Efek yang sama ini menurunkan semua solusi yang dikenal untuk penempatan di dalam ruangan yang menggunakan gelombang elektromagnetik dari pemancar dalam ruangan ke penerima dalam ruangan.

Digital Repository Universitas Jember

2.3 *Received Signal Strength Indicator (RSSI)*

Received Signal Strength Indicator merupakan teknologi yang digunakan untuk mengukur indikator kekuatan sinyal yang diterima oleh sebuah perangkat *wireless*. Namun, pemetaan langsung dari nilai RSSI yang berdasarkan jarak memiliki banyak keterbatasan, karena pada dasarnya, RSSI rentan terhadap *noise*, *multi-path fading*, gangguan, dan lain sebagainya yang mengakibatkan fluktuasi besar dalam kekuatan yang diterima (Sahu, dkk 2013).

Daya yang diterima oleh antena (P_r) ditempatkan pada jarak d dari antena pemancar dengan jumlah yang diketahui ditansmisikan daya (P_t) dan diberikan oleh persamaan Friis pada Persamaan 2.1.

$$P_r = P_t G_r G_t \left(\frac{\lambda}{4\pi d}\right) \tag{2.1}$$

Dimana G_t merupakan *Gain* dari antena pemancar, G_r adalah *Gain* dari antena penerima dan λ adalah panjang gelombang.

Kebalikan dari faktor yang berada dalam tanda kurung disebut sebagai *free space path loss*. Meskipun persamaan ini tidak dapat diterapkan di lingkungan *indoor* terestrial biasa atau pada komunikasi RF *outdoor*, perlu diketahui bahwa kekuatan sinyal yang ditransmisikan dapat melemahkan sesuai dengan jarak. Cara yang lebih realistis untuk mengkorelasikan RSSI jarak adalah dengan menggunakan *log* jarak *path loss models* yang memprediksi pertemuan sinyal *path loss* dengan jarak dalam lingkungan *indoor* (Alawi, 2011). Daya yang diterima dapat dinyatakan sebagai Persamaan 2.2.

$$P_r(d)(dBm) \sim N\ (\overline{\overline{P_r(d)}}(dBm), {\sigma^2}_{dB}) \tag{2.2}$$

Dimana $(\overline{\overline{P_r(d)}}$ = rata-rata daya yang diterima dan ${\sigma^2}_{dB}$ adalah varian yang berhubungan dengan efek *random shadowing*, oleh karena itu daya yang diterima dapat diberikan pada Persamaan 2.3.

Digital Repository Universitas Jember

$$\overline{\overline{P_r(d)}}(dBm) = P_r(d_0)(dBm) - 10n_p log_{10}\left(\frac{d}{d_0}\right) + X_\sigma \quad (2.3)$$

Dimana $P_r(d_0)$ adalah kekuatan sinyal dalam dBm terhadap referensi jarak n_p adalah *path loss* eksponen tergantung lingkungan media transmisi dan X_σ adalah *variable random* dengan distribusi normal dengan *mean* 0 dan standar deviasi. Secara realistis (Sahu dkk, 2013) menyatakan bahwa model *channel* seperti *log normal shadowing* memberikan nilai RSSI terhadap jarak d dari pemancar yang diberikan pada Persamaan 2.4.

$$RSSI(d) = P_t(d_0) - P_L(d_0) - 10n_p log_{10}\left(\frac{d}{d_0}\right) + X_\sigma \quad (2.4)$$

Dimana P_t adalah daya transmisi, $P_L(d_0)$ adalah *path loss* terhadap referensi jarak dan_pn adalah *path loss* eksponen tergantung lingkungan media transmisi. Variasi *random* terhadap RSSI dimodelkan sebagai *variable random Gaussian* dimana $X_\sigma = N(0, \sigma^2)$. Nilai dari n_p dan σ dapat diatur tergantung pada lingkungan propagasi.

2.4 *Wireless Local Area Network (WLAN)*

Wireless Local Area Network merupakan suatu jaringan yang menggunakan gelombang radio sebagai media transmisi data. Informasi ditransfer dari satu perangkat ke perangkat yang lainnya melalui gelombang radio, sepeti ditunjukan pada Gambar 2.1. WLAN disebut dengan jaringan nirkabel atau *wireless*. Jaringan WLAN sangat efektif digunakan didalam sebuah kawasan atau gedung. Dengan performa dan keamanan yang dapat diandalkan, pengembangan jaringan WLAN menjadi tren baru pengembangan jaringan menggantikan jaringan kabel. Solusi dari pengembangan WLAN dapat mencakup sebuah kawasan rumah, kantor kecil, perusahaan hingga ke area - area publik (Mulyanta, 2005).

Digital Repository Universitas Jember

Gambar 2.1 *Wireless* LAN

Komponen - komponen WLAN, pada umumnya terdiri dari:

1. *Mobile* atau *Desktop* PC : Merupakan perangkat akses bagi *user*.
2. *Access Point* : Perangkat yang menjadi pusat koneksi dari *user* ke jaringan *internet*. Dan memiliki fungsi untuk mengkonversikan sinyal frekuensi radio (RF) menjadi sinyal digital yang akan disalurkan melalui media kabel, ataupun disalurkan ke perangkat WLAN yang lainnya dengan dikonversikan ulang menjadi sinyal frekuensi radio.
3. WLAN *Interface* : Merupakan alat yang dipasangkan pada *mobile* atau *desktop* PC, yang dikembangkan secara massal dalam bentuk *Personal Computer Memory Card International Association (PCMCIA) card*, PCI *card* maupun *port* USB.
4. Antena *external* : Merupakan antena yang dipakai untuk memperkuat daya pancar. Antena tersebut dapat dirakit sendiri oleh *user*.

2.5 *Access Point (AP)*

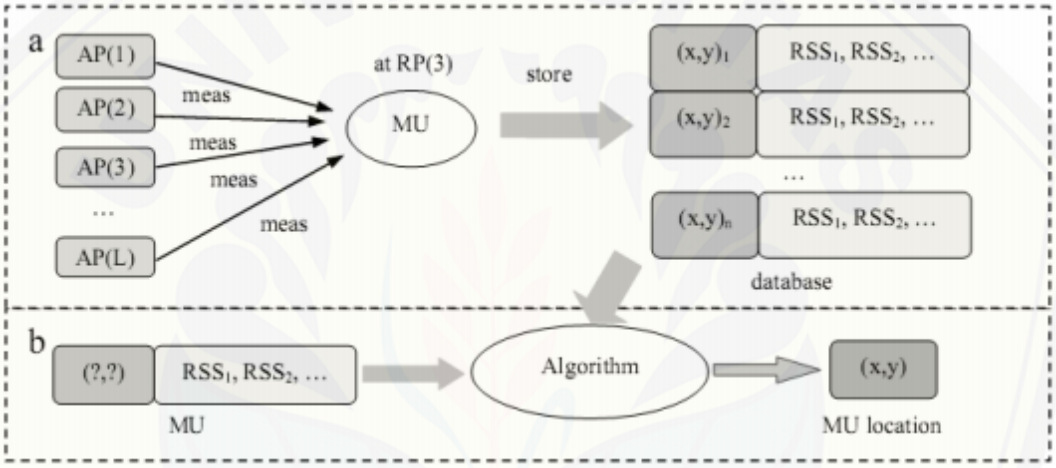
Acces point adalah sebuah *device half duplex* yang memiliki kepintaran, seperti *device switch*, fungsinya adalah mengirim dan menerima data, sebagai *buffer* data antara WLAN, serta berfungsi mengkonversi sinyal frekuensi radio (RF) menjadi sinyal digital yang akan disalurkan melalui kabel, atau disalurkan ke

Digital Repository Universitas Jember

perangkat WLAN yang lain dengan dikonversikan ulang menjadi sinyal frekuensi radio (Zaenal Arifin, 2002).

2.6 *Fingerprinting*

Fingerprinting merupakan metode untuk menentukan posisi menggunakan WLAN. Metode ini terdiri dari tahap pelatihan (*training*) dan tahap penentuan lokasi (*positioning*) (Li, 2007). Proses secara keseluruhan dapat dilihat pada Gambar 2.2.



Gambar 2.2 *Fingerprinting*

(Sumber : Li, 2007)

2.6.1 Tahap Pelatihan

Tahap pelatihan ini digunakan untuk mengumpulkan dan menyusun sebuah *fingerprint database*. Untuk mendapatkan *database* yang optimal, pemilihan *Reference Point (RP)* perlu dilakukan secara cermat. Setelah menentukan sejumlah RP, perangkat *client*, diletakkan pada sebuah RP dan dilakukan pengukuran *Received Signal Strength* dari masing - masing *Access Point (AP)* yang terdeteksi oleh *user*. Data karakteristik yang didapat dari RP tersebut, yaitu RSS dari tiap AP disimpan ke dalam *database* beserta koordinat posisi RP. Proses ini diulang untuk semua RP yang telah ditentukan.

Digital Repository Universitas Jember

2.6.2 Tahap Penentuan Lokasi

Pada tahap ini, *client* masih belum diketahui posisinya, akan mengukur sinyal yang diterima dari tiap AP yang terdeteksi. Hasil pengukuran sinyal lalu dibandingkan dengan *database* yang didapat dari tahap pelatihan, sehingga posisi dapat diketahui. Pada proses ini menggunakan teori pengukuran jarak atau *Euclidean Distance*.

2.7 *Euclidean Distance*

Euclidean Distance merupakan perhitungan jarak dari 2 buah titik dalam *Euclidean Space*. *Euclidean Space* diperkenalkan oleh Euclid, matematikawan asal Yunani untuk mempelajari hubungan antara sudut dan jarak. Teori ini berkaitan dengan *Teorema Phytagoras* yang biasanya diterapkan pada 1, 2 dan 3 dimensi. Perhitungan *Euclidean Distance* dapat dilihat pada Persamaan 2.5.

$$d = \sqrt{(x_1 - x_2)2 \quad + (y_1 - \ y_2)2} \tag{2.5}$$

Dengan :

d = Jarak antara 2 titik

x_1 = Sumbu koordinat x pada titik 1

x_2 = Sumbu koordinat x pada titik 2

y_1 = Sumbu koordinat y pada titik 1

y_{21} = Sumbu koordinat y pada titik 2

2.8 *k-Nearest Neighbor (k-NN)*

Algoritma *k-Nearest Neighbor (k-NN)* merupakan algoritma yang digunakan untuk melakukan klasifikasi terhadap suatu objek, berdasarkan *k* data latih yang jaraknya paling dekat dengan objek tersebut. Syarat nilai *k* adalah tidak boleh lebih besar dari jumlah data latih, dan nilai *k* harus ganjil dan lebih dari satu. Dekat atau jauhnya jarak data latih yang paling dekat dengan objek yang akan diklasifikasi dapat dihitung dengan menggunakan metode *cosine similiarity* (J.han, 2013).

Digital Repository Universitas Jember

Cosine similiarity merupakan salah satu metode yang dapat digunakan untuk melihat sejauh mana kemiripan isi antar dokumen. Dalam hal ini *cosine similiarity* berfungsi untuk menguji ukuran yang dapat digunakan sebagai interpretasi kedekatan jarak berdasarkan kemiripan dokumen. Rumus untuk menghitung jarak pada algoritma k-NN dengan metode *cosine similiarity* pada Persamaan 2.6.

$$Cos(\theta_{QD}) = \frac{\sum_{i=1}^{n} Q_i D_i}{\sqrt{\sum_{i=1}^{n}(Q_i)^2} \cdot \sqrt{\sum_{i=1}^{n}(D_i)^2}} \quad (2.6)$$

Dimana:

$Cos(\theta_{QD})$ = Kemiripan Q terhadap dokumen D

Q = Data uji

D = Data latih

n = Banyaknya data

2.9 *Weighted k-Nearest Neighbor (Weighted k-NN)*

Weighted K-Nearest Neighbor adalah algoritma perkembangan dari *k-Nearest Neighbor (k-NN). k-Nearest Neighbor (k-NN)* bertujuan untuk mengklasifikasikan sebuah subjek baru yang bersumber pada data pelatihan sampel serta atribut. Nantinya hasil dari sampel uji baru digolongkan sesuai dengan mayoritas dari kategori pada *k-NN*. Algoritma *k-NN* menggunakan klasifikasi untuk ketetanggaan yang digunakan sebagai nilai prediksi dari sampel uji yang baru (Krisandi, Helmi, & Prihandoso, 2013).

Algoritma *Weighted* k-*Nearest Neighbor (Weighted k-NN)* menggunakan pembobotan pada kelas data. Bobot yang akan diberikan pada kelas yang minoritas lebih besar dan bobot lebih sedikit pada tetangga yang berasal dari kelas yang mayoritas (Indriati & Ridok, 2016). Perhitungan bobot dapat digunakan pada Persamaan 2.7.

Digital Repository Universitas Jember

$$Weight = \frac{1}{\left(\frac{Num\left(c_i^d\right)}{Min\left\{Num\left(c_j^d\right)=1,\ldots,K*\right\}}\right)1/exp} \quad (2.7)$$

Dimana:

$Num\left(c_i^d\right)$ = banyaknya data latih d pada kelas i

$Num\left(c_j^d\right)$ = banyaknya data latih d pada kelas j,

dimana j terdapat dalam himpunan k tetangga terdekat

Exp = eksponen (nilai exp lebih dari 1)

Pada data setiap jenis yang sudah mendapatkan bobot, dipakai untuk perhitungan skor data uji terhadap setiap kelas. Nantinya hasil perhitungan pembobotan dikalikan menggunakan persamaan hasil skor. Perhitungan tersebut akan diterapkan pada rumus hasil skor dengan Persamaan 2.8.

$$score(d, c_i) = \sum_{d_j \in KNN(d)} weight_i \, Sim\left(d, d_j\right)\delta\left(d_j, c_i\right) \quad (2.8)$$

Dimana:

$Weighti$ = bobot pada jenis atau kelas i

$dj \in KNN(d)$ = data latih dj pada kumpulan tetangga terdekat dari data uji d

$\delta(dj, Ci)$ = akan bernilai 1 jika nilai jarak $\in Ci$ dan bernilai 0 jika jarak $\notin Ci$.

$Sim(d, dj)$ = nilai *Cosine Similiarity* antaradata uji dan data latih

Ci = jenis atau kelas i

2.10 *Android Studio*

Android Studio adalah *Integrated Development Enviroment (IDE)* untuk sistem operasi *Android*, yang dibangung diatas perangkat lunak *JetBrains* IntelliJ IDEA dan didesain khusus untuk pengembangan *Android*. IDE ini merupakan pengganti dari *Eclipse Android Development Tools (ADT)* yang sebelumnya merupakan IDE utama untuk pengembangan aplikasi *android*. IDE Ini tersedia untuk digunakan pada sistem operasi *Windows*, Mac OS X dan Linux.

Digital Repository Universitas Jember

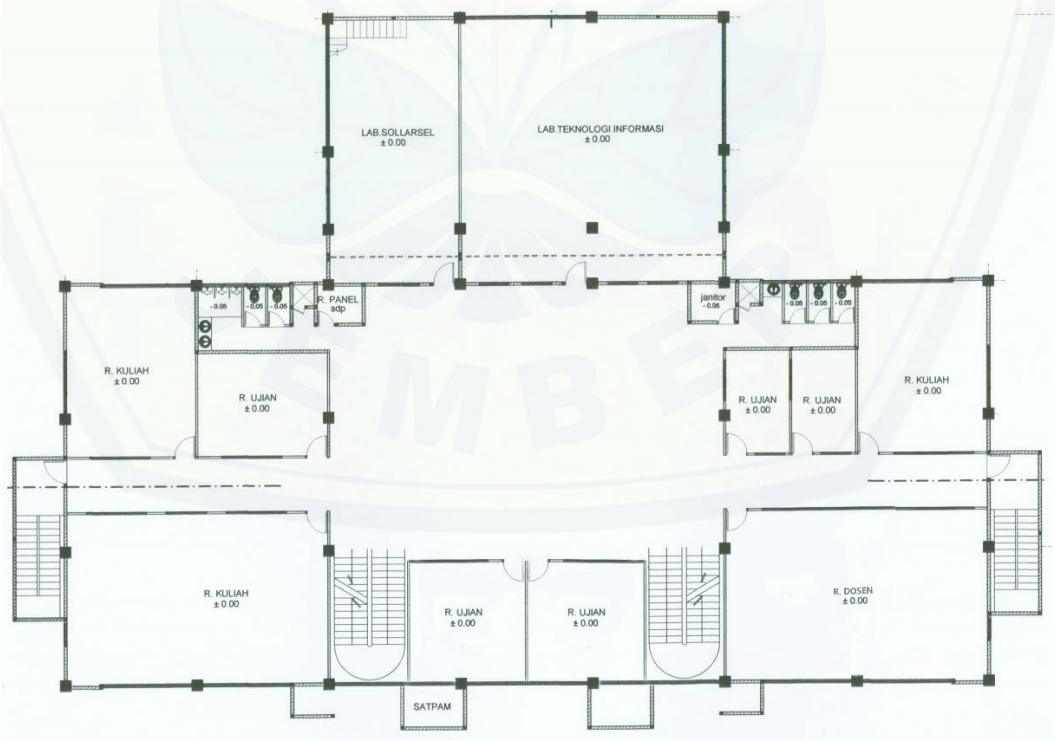
BAB 3. METODELOGI PENELITIAN

Pada metodologi penelitian ini akan dijelaskan mengenai hal utama yang akan dikaji dalam bentuk diagram alur (*flowchart*) yaitu objek penelitian, tahap penelitian, tempat dan waktu penelitian, alat dan bahan, pengambilan data, sampai dengan pengolahan data. Perancangan aplikasi dilakukan mengunakan *software android studio*. Kemudian aplikasi ini akan di *instal* ke *smartphone*, untuk dilakukan pengambilan data berdasarkan parameter yang akan diuji.

3.1 Tempat dan Waktu Penelitian

3.1.1 Tempat Penelitian

Proses perancangan aplikasi *Indoor Positioning System*, dilakukan di kediaman peneliti. Sedangkan proses pengambilan data dan pengujian aplikasi dilakukan di Lantai 3 Gedung A, Fakultas Teknik, Universitas Jember, seperti pada Gambar 3.1.



Gambar 3.1 Denah Lantai 3 Gedung A

Digital Repository Universitas Jember

3.1.2 Waktu Penelitian

Waktu pelaksanaan penelitian dimulai dari bulan September 2019 - Desember 2019, dengan jadwal pelaksanaan seperti pada Tabel 3.1 sebagai berikut:

Tabel 3.1 Jadwal Pelaksanaan Penelitian

No	Kegiatan	Bulan			
		1	2	3	4
1	Studi literatur				
2	Perancangan dan pembuatan aplikasi				
3	Pengujian aplikasi dan Pengambilan Data				
4	Analisa data				
5	Pembuatan laporan				

3.2 Alat dan Bahan

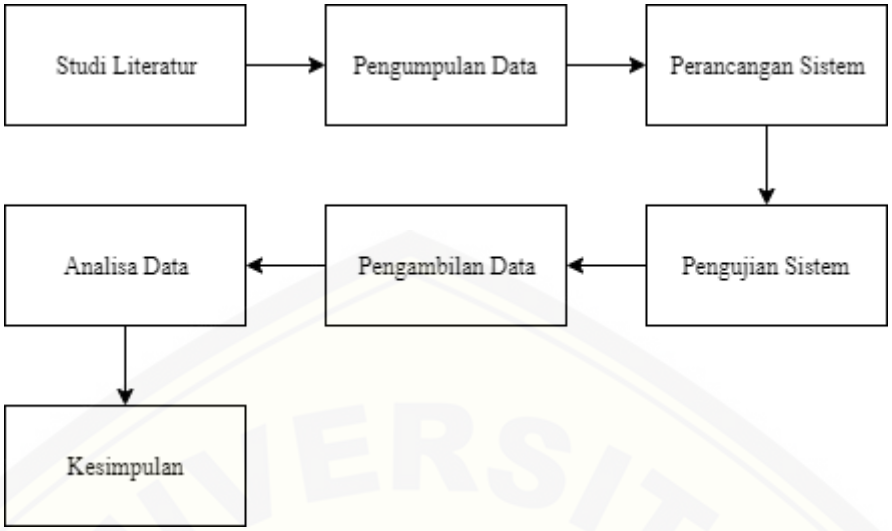
3.2.1 Alat

1. PC/Laptop
2. *Access Point*
3. *Smartphone*
4. *Android Studio*

3.3 Tahap Penelitian

Penyusunan laporan ini memiliki beberapa tahap untuk memperoleh informasi yang dibutuhkan, adapun tahap pengambilan data yang digunakan dalam penelitian ini dijelaskan pada Gambar 3.2 berikut:

Digital Repository Universitas Jember



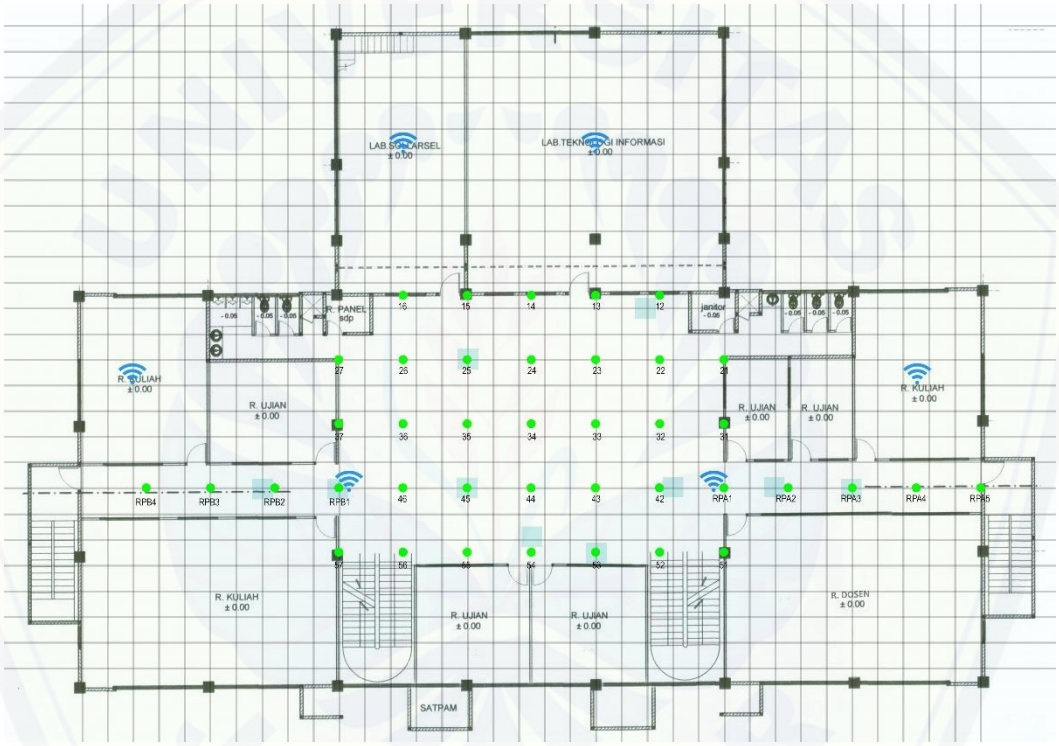
Gambar 3.2 Tahapan Penelitian

Pada alur tahapan penelitian ini, pertama dilakukan studi literatur tentang topik yang akan diteliti dan juga mencari kajian pustaka yang berhubungan dengan penelitian. Kemudian pada tahap pengumpulan data melakukan pemetaan lokasi yaitu di lantai 3 Gedung A Fakultas Teknik Universitas Jember untuk menentukan *access point* yang diguanakan, titik pelatihan (*reference point*) dan titik lokasi yang akan diuji. Selanjutnya merancang aplikasi mengunakan *software android studio*. Setelah aplikasi selesai, kemudian di *instal* pada *smartphone* untuk lanjut tahap pengujian. Pada tahap pengujian ini dilakukan tes dasar aplikasi untuk mengecek aplikasi telah berjalan seperti yang dirancang dan diprogram. Pada tahap pengambilan data, data yang diambil berupa *access point* yang digunakan, data *fingerprint* sesuai *reference point*, dan data uji lokasi berdasarkan skenario yang telah ditentukan yaitu menggunakan algoritma *k-Nearest Neighbor* dan algoritma *Weighted k-Nearest Neighbor*. Lalu memvariasi parameter jumlah *reference point*, nilai *k* pada algoritma *Weighted k-Nearest Neighbor*. Kemudian data ini dianalisa hingga diperoleh kesimpulan.

Digital Repository Universitas Jember

3.4 Pemetaan Lokasi

Sebelum proses perancangan aplikasi, dilakukan pemetaan lokasi terhadap posisi *acccess point* dan posisi pengambilan data *fingerprinting (reference point)*, seperti ditunjukkan pada Gambar 3.3. Pada tahap ini denah diukur kemudian dibagi kedalam sel - sel dengan ukuran 1 x 1 meter. Pemetaan ini mempengaruhi data pembelajaran yang dihasilkan dalam *database fingerprinting. Access point* yang digunakan berjumlah 6 buah dan total data *fingerprinting* yang diambil yaitu 40 *reference point.*

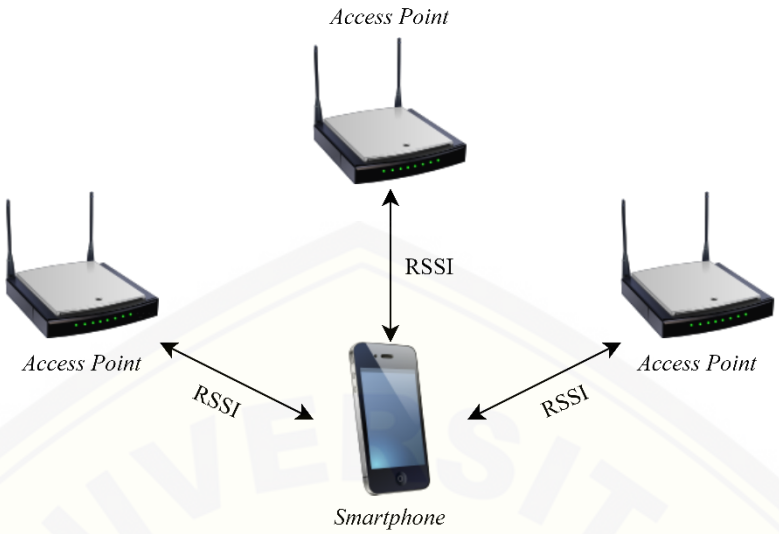


Gambar 3.3 Pemetaan lokasi

3.5 Desain Jaringan

Implementasi sistem ini menggunakan 6 buah *access point*. *Access point* ini digunakan untuk mengirimkan data RSS ke *smartphone*, kemudian data RSS beserta koordinat disimpan kedalam *database.* Topologi yang digunakan saat proses pengambilan data *fingerprinting* dan pengujian lokasi dapat dilihat pada Gambar 3.4.

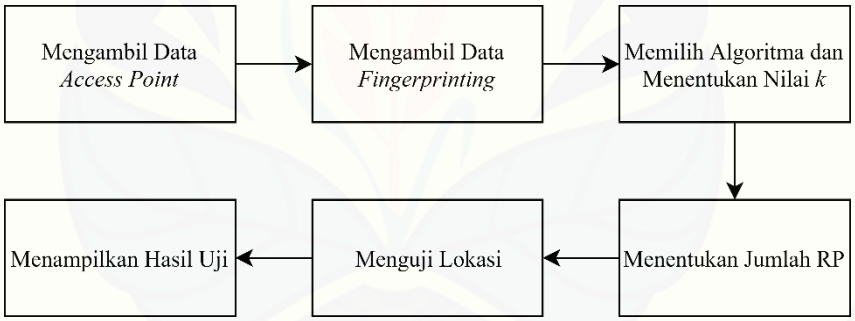
Digital Repository Universitas Jember



Gambar 3.4 Topologi Jaringan

3.6 Perancangan Sistem

3.6.1 Blok Sistem



Gambar 3.5 Blok Diagram Sistem

Dari blok diagram sistem pada Gambar 3.5, dapat diketahui bahwa sistem ini terdiri beberapa menu yang harus dilakukan sebelum dapat melakukan uji lokasi.

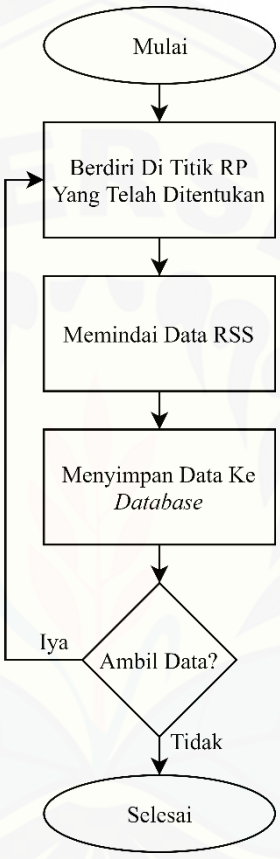
3.7 Proses Penentuan Lokasi

Sistem ini terdiri dari 2 tahapan yaitu pengambilan data pelatihan dan penentuan lokasi.

3.7.1 Pengambilan Data Pelatihan

Digital Repository Universitas Jember

Pertama aplikasi pada *smartphone* mengambil informasi RSS yang didapat dari AP berupa *mac address*, kekutan sinyal, dan koordinat posisi pengambilan data. Kemudian mengirim ke *database* dan menyimpannya. Seperti ditunjukkan pada Gambar 3.6.

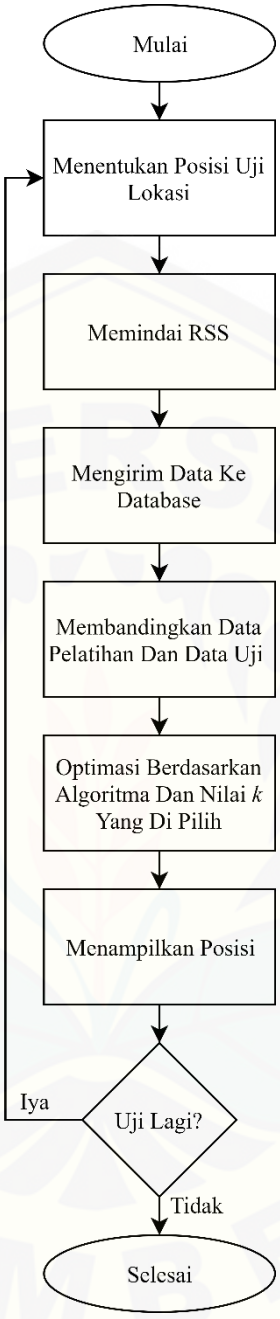


Gambar 3.6 *Flowchart* Pengambilan Data Pelatihan

3.7.2 Penentuan Posisi

Pada tahapan ini aplikasi pada *smartphone* mengambil informasi RSS yang didapat dari AP kemudian mengirimkannya ke *database*. Kemudian data tersebut dibandingkan dengan data pelatihan dengan menggunakan teori perhitungan jarak ditambah dengan algoritma yang dipilih yaitu *Weighted k-NN* dan *k-NN* beserta varisai parameter nilai *k* dan jumlah RP yang digunakan guna meningkatkan tingkat akurasi. Semua proses penentuan lokasi dan optimasi akurasi dilakukan di *smartphone*. Alur penentuan lokasi dapat dilihat pada Gambar 3.7.

Digital Repository Universitas Jember



Gambar 3.7 *Flowchart* Penentuan Posisi

Digital Repository Universitas Jember

3.8 Pengujian dan Analisa *Indoor Positioning System*

Pengujian pada sistem *Indoor Positioning* dilakukan dengan 3 tahapan yaitu pengujian performa algoritma *Weighted k-Nearest Neighbor* terhadap algoritma *k-Nearest Neighbor*, serta pengujian akurasi algoritma *Weighted k-Nearest Neighbor* terhdap jumlah *reference point* dan terhadap nilai *k* (jumlah tetangga terdekat).

3.8.1 Pengujian Algoritma *Weighted k-NN* Terhadap Algoritma *k-NN*

Pengujian algoritma *Weighted k-NN* terhadap algoritma *k-NN* dilakukan untuk menguji perbandingan tingkat akurasi antara kedua algoritma sebelum dan setelah pembobotan. Data yang dibandingkan berupa koordinat uji dengan koordinat hasil uji.

Tabel 3.2 Parameter Uji Algoritma *Weighted k-NN* Terhadap *k-NN*

Koordinat Uji (x,y)	*Reference Point*	Nilai *k*
(30,7)	40	1
(7,7)		
(22,14)		
(15,12)		
(10,7)		

3.8.2 Pengujian Terhadap Jumlah *Reference Point*

Pengujian terhadap jumlah *reference point* di lakukan untuk menguji tingkat akurasi algoritma *Weighted k-NN* berdasarkan jumlah data *fingerprint*. Hal ini dilakukan dengan mengubah jumlah *reference point* yang semula 40 *reference point* menjadi 20 *reference point*. Data yang dibandingkan berupa koordinat uji dengan koordinat hasil uji.

Digital Repository Universitas Jember

Tabel 3.3 Parameter Uji Terhadap Jumlah *Reference Point*

Koordinat Uji (x,y)	*Reference Point*		Nilai *k*
	Uji 1	Uji 1	
(30,7)	40	20	1
(7,7)			
(22,14)			
(15,12)			
(10,7)			

3.8.3 Pengujian Terhadap Nilai *k*

Pengujian terhadap nilai *k* di lakukan untuk menguji tingkat akurasi algoritma *Weighted k-NN* berdasarkan jumlah tetangga terdekat yang diuji. Hal ini dilakukan dengan mengubah parameter *k* yaitu secara berturut - turut; 1, 2 ,3, 4, 5, dan 6. Data yang dibandingkan berupa koordinat uji dengan koordinat hasil uji.

Tabel 3.4 Parameter Uji Terhadap nilai *k*

Koordinat Uji (x,y)	Nilai *k*						*Reference Point*
	Uji 1	Uji 2	Uji 3	Uji 4	Uji 5	Uji 6	
(20,4.5)	1	2	3	4	5	6	40
(27.5,7)							
(17.5,5.1)							
(23,7)							
(15,7)							

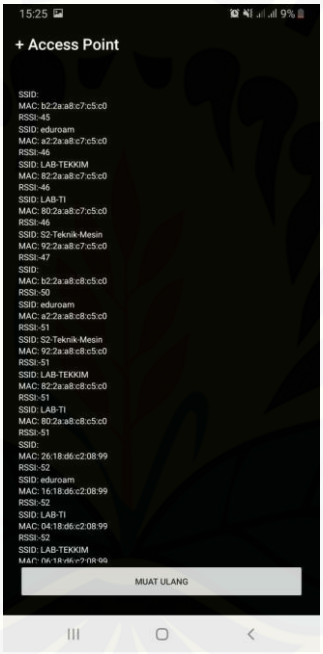
Digital Repository Universitas Jember

3.9 Proses Pengambilan Data

Proses pengambilan data dilakukan menggunakan aplikasi *Indoor Positioning System* yang telah dirancang sebelumnyaa menggunakan *software* android studio.

3.9.1 Pengambilan Data *Acces Point*

Pengambilan data dilakukan dengan memindai *access point* yang tersedia dan memilih *access point* yang akan digunakan dalam proses *fingerprint*. hasil pemindaaian *acces point* dapat dilihata pada gambar 3.8.

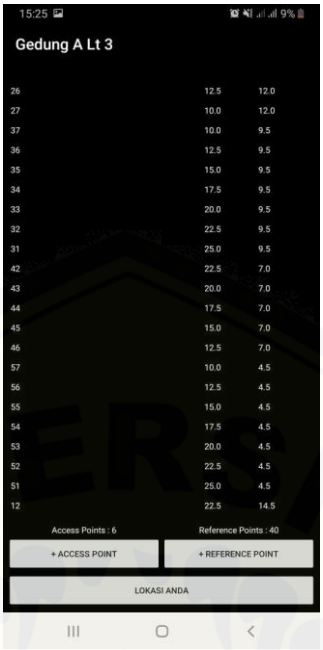


Gambar 3.8 Pengambilan Data *Access Point*

3.9.2 Pengambilan Data *Fingerprint*

Pengambilan data dilakukan dengan berdiri dititik *reference point* yang telah ditentukan, kemudian melakukan proses pemindaian RSSI dan mengatur titik koordinatnya. Hasil *fingerprint* dapat dilihat pada gambar 3.9.

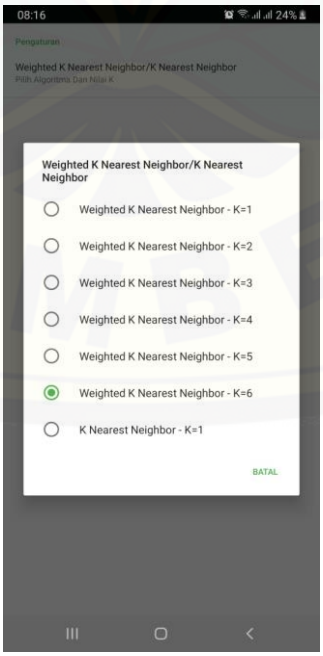
Digital Repository Universitas Jember



Gambar 3.9 Hasil Pengambilan Data *Fingerprint*

3.9.3 Pengambilan Data Uji Penentuan Lokasi

Sebelum melakukan uji lokasi, terlebih dahulu dilakukan pengaturan algoritma dan parameter seperti ditunjukkan pada gambar 3.10. Dan hasil penentuan lokasi dapat dilihat pada gambar 3.11



Gambar 3.11 Pengaturan Algoritma Dan Paramer *k*

Digital Repository Universitas Jember

Gambar 3.12 Hasil Penentuan Uji Lokasi

3.9.4 Perhitungan Penyimpangan

Perhitungan penyimpangan dilakukan untuk mengetahui besar penyimpangan yang didapatkan dari hasil percobaan berdasarkan data koordinat uji dan koordinat hasil uji. Diketahui koordinat uji (20,4.5) dan koordinat hasil uji (20,4.5), maka dapat kita tentukan penyimpangannya dengan rumus *euclidean distance* sebagai berikut:

$d = \sqrt{(x_1 - x_2)2 \quad + (y_1 - \ y_2)2}$

$d = \sqrt{(20 - 20)2 \quad + (4{,}5 - \ 4{,}5)2}$

$d = \sqrt{(0)2 \quad + (0)2}$

$d = \sqrt{0 + 0}$

$d = \sqrt{0}$

$d = \ 0$ Jadi didapatkan penyimpangan sebesar 0 meter.

Digital Repository Universitas Jember

BAB 5. PENUTUP

5.1 Kesimpulan

Dari hasil penelitian mengenai Implementasi *Indoor Positioning System* (*IPS*) Menggunakan Algoritma *Weighted k-Nearest Neighbor* di Gedung A Fakultas Teknik Universitas Jember yang telah dilakukan, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Performa algoritma *weighted k-nearest neighbor* cukup dapat di handalkan, terbukti dari hasil perbandingan uji lokasi menggunakan algoritma *weighted k-nearest neighbor* terhadap algoritma *k-nearest neighbor*. Pada hasil pengujian menggunakan algoritma *weighted k-nearest neighbor* didapatkan rata – rata penyimpangan dari 5 kali percobaan yaitu sebesar 1,49 meter, kemudian pada saat menggunakan algoritma *k-nearest neighbor* didapatkan rata – rata penyimpangan sebesar 2,7 meter.
2. Data *fingerprinting* sangat berpengaruh terhadap tingkat akurasi lokasi. Semakin banyak data *fingerprinting* berupa *reference point*, maka tingkat akurasi semakin akurat. Seperti pada percobaan perbandingan terhadap jumlah RP sebesar 40 dan jumlah RP sebesar 20. Pada percobaan RP sebesar 40, penyimpangan terbesar yang didapatkan sebesar 2,23 meter dan penyimpangan terkecil sebesar 0 meter. Sedangkan pada percobaan RP sebesar 20, penyimpangan terbesar yang didapatkan sebesar 5,83 meter dan penyimpangan terkecil sebesar 1 meter. Dari hasil data didapatkan tingkat akurasi pada saat RP berjumlah 40, lebih akurat dibandingkan pada saat RP berjumlah 20. Nilai *k* (jumlah tetangga terdekat) mempengaruhi tingkat akurasi lokasi yang didapatkan. Pada saat parameter *k* bernilai 1, penyimpangan rata - rata yang didaptkan pada 5 kali percobaan, didapatkan rata - rata penyimpanagn sebesar 0,26 meter. Sedangkan pada saat parameter *k* bernilai 6, didapatkan rata - rata penyimpanagn sebesar 5,64 meter.

Digital Repository Universitas Jember

5.2 Saran

Dari hasil pengujian Implementasi *Indoor Positioning System* (*IPS*) yang telah dilakukan, peneliti mempunyai saran agar penelitian ini dapat dikembangkan, yaitu menambah dan mengurangi jumlah *acces point.* Hal ini bertujuan untuk melihat performasi algoritma *weighted k-nearest neighbor.*

Digital Repository Universitas Jember

DAFTAR PUSTAKA

Suryanto, Agus. 2012. Aplikasi Teknologi Global Positioning System (GPS) Dan Telepon Selular (GSM) Untuk Monitoring Titik Akses kendaraan Dinas UNNES. *Jurnal Sain dan Teknologi Universitas Negeri Semarang*, 1(2): 23-32.

Ginardi, Hari. 2016. Impementasi Indoor Positioning System Berbasis Smartphone Dengan Penambahan Acces Point Untuk Studi Kasus Gedung Teknik Informatika ITS. Jurnal teknik ITS, 7(2): 119-224.

Bakal, J.W. 2016. Survey Of Indoor Positioning Measurements, Methods And Techniques. *International Jurnal of Computer Applications, 2(2): 22-33.*

Sutarti. 2015. Estimasi Lokasi Objek Berbasis Wifi Pada Gedung Bertingkat Menggunakan Metode Naïve Bayes. *Jurnal PROSISKO,* 2(2):41-50.

Susantok, Mochamad. 2013. Wifi Positioning System (WPS) Menggunakan Algoritma Neural Network Backpropagation di Area Kampus Politeknik Caltex Riau. *Jurnal Teknik Elektro dan Komputer,* 1(2):130-141.

J. Han, M. Kamber dan J. Pei, *Data Mining Concepts and Techniques*, Waltham: Elsevier, 2012.

Indriati, & Ridok, A. (2016). *Sentiment Analysis for Review Mobile Application Using Neighbor Weighted K-Nearest Neighbor(NWKNN). Journal of Enviromental Engineering & Sustainable Technology*, 23-32.

Sahu, P.K, Wu E.H dan Sahoo J, *Dual RSSI Trend Based Localization for Wireless Sensor Networks*. IEEE.

Li, B., Salter, J., Dempster, A., & Rizos, C. (2007). *Indoor positioning techniques based on wireless LAN*.

Digital Repository Universitas Jember

Krisandi, N., Helmi, & Prihandoso, B. (2013). Algoritma *K-Nearest Neighbor* Dalam Klasifikasi Data Hasil Produksi Kelapa Sawit Pada PT. MINAMAS Kecamatan Parindu. Buletin Ilmiah Math. Stat. dan Terapannya (Bimaster), 02, 33-38.

Indoor Positioning System – Wikipedia

https://en.wikipedia.org/wiki/Indoor_positioning_system

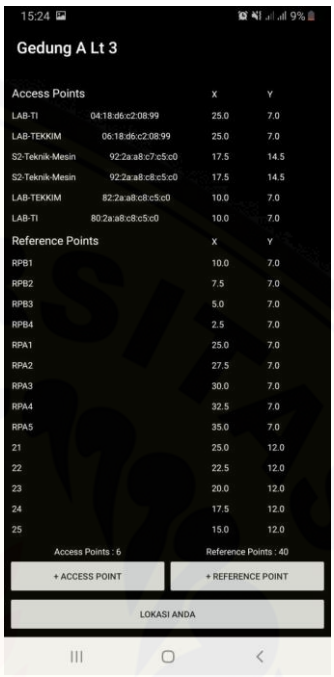
Weighted K-NN

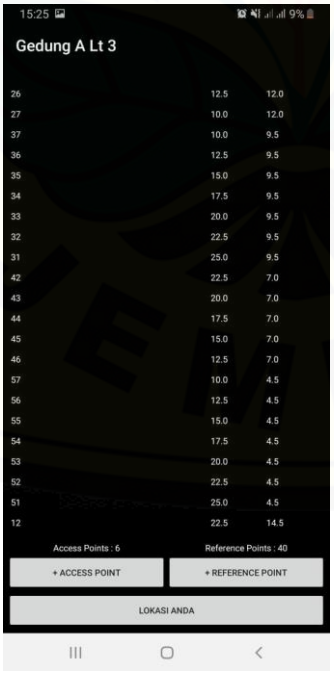
https://www.geeksforgeeks.org/weighted-k-nn/

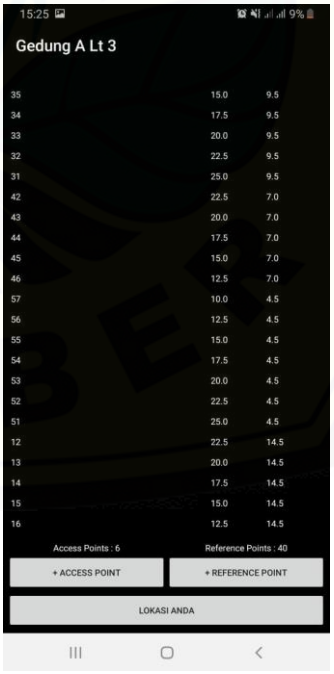
Digital Repository Universitas Jember

LAMPIRAN

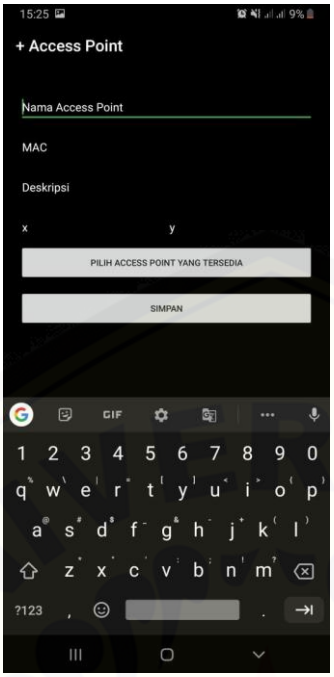
A. Tampilan Aplikasi *Indoor Positioning System*

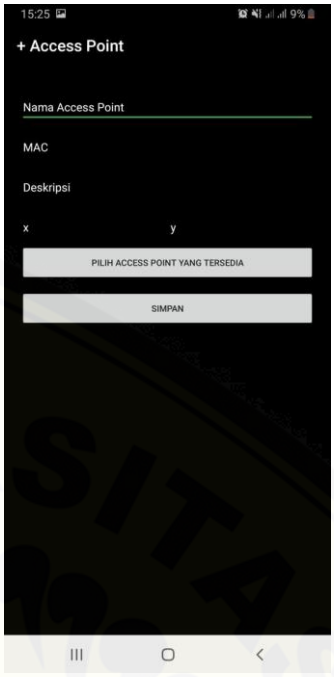


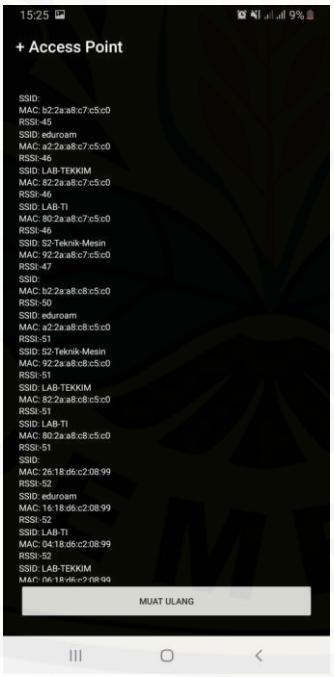




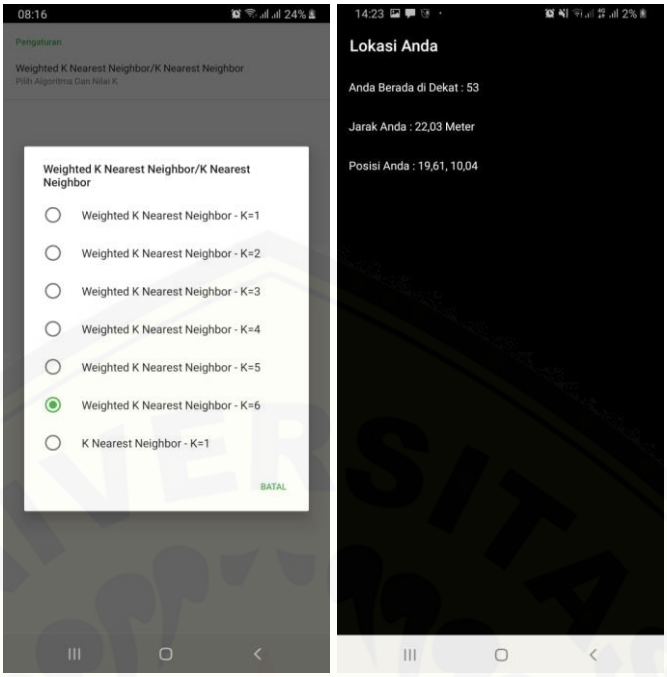
Digital Repository Universitas Jember







Digital Repository Universitas Jember



B. Data *Fingerprint*

RP	ACCESS POINT	MAC	RSSI	X	Y
RPA1	LAB-TI	80:2a:a8:c8:c5:c0	-53.0	25	7
RPA1	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-53.0	25	7
RPA1	LAB-TI	04:18:d6:c2:08:99	-34.2	25	7
RPA1	LAB-TEKKIM	06:18:d6:c2:08:99	-34.3	25	7
RPA1	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-53.0	25	7
RPA1	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-52.3	25	7
RPA2	LAB-TI	80:2a:a8:c8:c5:c0	-53.0	27,5	7
RPA2	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-53.0	27,5	7
RPA2	LAB-TI	04:18:d6:c2:08:99	-34.0	27,5	7
RPA2	LAB-TEKKIM	06:18:d6:c2:08:99	-34.0	27,5	7
RPA2	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-53.0	27,5	7
RPA2	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-53.0	27,5	7
RPA3	LAB-TI	80:2a:a8:c8:c5:c0	-53.6	30	7
RPA3	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-53.6	30	7
RPA3	LAB-TI	04:18:d6:c2:08:99	-35.2	30	7

Digital Repository Universitas Jember

RPA3	LAB-TEKKIM	06:18:d6:c2:08:99	-35.2	30	7
RPA3	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-53.6	30	7
RPA3	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-53.4	30	7
RPA4	LAB-TI	80:2a:a8:c8:c5:c0	-60.2	32,5	7
RPA4	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-59.3	32,5	7
RPA4	LAB-TI	04:18:d6:c2:08:99	-48.0	32,5	7
RPA4	LAB-TEKKIM	06:18:d6:c2:08:99	-45.0	32,5	7
RPA4	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-59.3	32,5	7
RPA4	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-58.5	32,5	7
RPA5	LAB-TI	80:2a:a8:c8:c5:c0	-60.6	35	7
RPA5	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-60.2	35	7
RPA5	LAB-TI	04:18:d6:c2:08:99	-47.4	35	7
RPA5	LAB-TEKKIM	06:18:d6:c2:08:99	-45.6	35	7
RPA5	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-60.2	35	7
RPA5	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-59.0	35	7
RPB1	LAB-TI	80:2a:a8:c8:c5:c0	-38.4	10	7
RPB1	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-38.4	10	7
RPB1	LAB-TI	04:18:d6:c2:08:99	-50.3	10	7
RPB1	LAB-TEKKIM	06:18:d6:c2:08:99	-51.1	10	7
RPB1	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-38.4	10	7
RPB1	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-35.3	10	7
RPB2	LAB-TI	80:2a:a8:c8:c5:c0	-39.0	7,5	7
RPB2	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-39.0	7,5	7
RPB2	LAB-TI	04:18:d6:c2:08:99	-49.0	7,5	7
RPB2	LAB-TEKKIM	06:18:d6:c2:08:99	-50.0	7,5	7
RPB2	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-39.0	7,5	7
RPB2	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-35.0	7,5	7
RPB3	LAB-TI	80:2a:a8:c8:c5:c0	-39.3	5	7
RPB3	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-39.3	5	7
RPB3	LAB-TI	04:18:d6:c2:08:99	-50.5	5	7
RPB3	LAB-TEKKIM	06:18:d6:c2:08:99	-51.3	5	7

Digital Repository Universitas Jember

RPB3	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-39.3	5	7
RPB3	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-36.3	5	7
RPB4	LAB-TI	80:2a:a8:c8:c5:c0	-38.0	2,5	7
RPB4	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-38.0	2,5	7
RPB4	LAB-TI	04:18:d6:c2:08:99	-53.0	2,5	7
RPB4	LAB-TEKKIM	06:18:d6:c2:08:99	-53.0	2,5	7
RPB4	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-38.0	2,5	7
RPB4	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-44.0	2,5	7
12	LAB-TI	80:2a:a8:c8:c5:c0	-57.0	22,5	14,5
12	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-57.9	22,5	14,5
12	LAB-TI	04:18:d6:c2:08:99	-62.6	22,5	14,5
12	LAB-TEKKIM	06:18:d6:c2:08:99	-62.6	22,5	14,5
12	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-57.9	22,5	14,5
12	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-59.5	22,5	14,5
13	LAB-TI	80:2a:a8:c8:c5:c0	-56.2	20	14,5
13	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-56.2	20	14,5
13	LAB-TI	04:18:d6:c2:08:99	-54.0	20	14,5
13	LAB-TEKKIM	06:18:d6:c2:08:99	-54.0	20	14,5
13	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-56.2	20	14,5
13	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-54.5	20	14,5
14	LAB-TI	80:2a:a8:c8:c5:c0	-55.3	17,5	14,5
14	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-55.3	17,5	14,5
14	LAB-TI	04:18:d6:c2:08:99	-57.1	17,5	14,5
14	LAB-TEKKIM	06:18:d6:c2:08:99	-57.1	17,5	14,5
14	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-55.3	17,5	14,5
14	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-57.2	17,5	14,5
15	LAB-TI	80:2a:a8:c8:c5:c0	-56.0	15	14,5
15	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-56.0	15	14,5
15	LAB-TI	04:18:d6:c2:08:99	-62.0	15	14,5
15	LAB-TEKKIM	06:18:d6:c2:08:99	-61.1	15	14,5
15	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-56.0	15	14,5

Digital Repository Universitas Jember

15	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-52.9	15	14,5
16	LAB-TI	80:2a:a8:c8:c5:c0	-55.5	12,5	14,5
16	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-55.5	12,5	14,5
16	LAB-TI	04:18:d6:c2:08:99	-61.6	12,5	14,5
16	LAB-TEKKIM	06:18:d6:c2:08:99	-60.6	12,5	14,5
16	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-55.5	12,5	14,5
16	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-52.5	12,5	14,5
21	LAB-TI	80:2a:a8:c8:c5:c0	-55.4	25	12
21	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-56.1	25	12
21	LAB-TI	04:18:d6:c2:08:99	-57.8	25	12
21	LAB-TEKKIM	06:18:d6:c2:08:99	-57.8	25	12
21	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-56.1	25	12
21	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-56.8	25	12
22	LAB-TI	80:2a:a8:c8:c5:c0	-55.0	22,5	12
22	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-56.0	22,5	12
22	LAB-TI	04:18:d6:c2:08:99	-58.0	22,5	12
22	LAB-TEKKIM	06:18:d6:c2:08:99	-58.0	22,5	12
22	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-56.0	22,5	12
22	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-57.0	22,5	12
23	LAB-TI	80:2a:a8:c8:c5:c0	-54.7	20	12
23	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-55.5	20	12
23	LAB-TI	04:18:d6:c2:08:99	-57.2	20	12
23	LAB-TEKKIM	06:18:d6:c2:08:99	-57.2	20	12
23	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-55.5	20	12
23	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-56.3	20	12
24	LAB-TI	80:2a:a8:c8:c5:c0	-55.0	17,5	12
24	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-55.0	17,5	12
24	LAB-TI	04:18:d6:c2:08:99	-54.0	17,5	12
24	LAB-TEKKIM	06:18:d6:c2:08:99	-54.0	17,5	12
24	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-55.0	17,5	12
24	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-52.0	17,5	12

Digital Repository Universitas Jember

25	LAB-TI	80:2a:a8:c8:c5:c0	-55.0	15	12
25	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-55.0	15	12
25	LAB-TI	04:18:d6:c2:08:99	-54.0	15	12
25	LAB-TEKKIM	06:18:d6:c2:08:99	-54.0	15	12
25	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-55.0	15	12
25	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-52.0	15	12
26	LAB-TI	80:2a:a8:c8:c5:c0	-50.7	12,5	12
26	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-50.7	12,5	12
26	LAB-TI	04:18:d6:c2:08:99	-53.0	12,5	12
26	LAB-TEKKIM	06:18:d6:c2:08:99	-53.0	12,5	12
26	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-50.7	12,5	12
26	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-46.7	12,5	12
27	LAB-TI	80:2a:a8:c8:c5:c0	-46.0	10	12
27	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-46.0	10	12
27	LAB-TI	04:18:d6:c2:08:99	-53.0	10	12
27	LAB-TEKKIM	06:18:d6:c2:08:99	-53.0	10	12
27	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-46.0	10	12
27	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-48.0	10	12
31	LAB-TI	80:2a:a8:c8:c5:c0	-54.9	25	9,5
31	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-55.0	25	9,5
31	LAB-TI	04:18:d6:c2:08:99	-41.7	25	9,5
31	LAB-TEKKIM	06:18:d6:c2:08:99	-41.5	25	9,5
31	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-55.0	25	9,5
31	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-52.1	25	9,5
32	LAB-TI	80:2a:a8:c8:c5:c0	-55.0	22,5	9,5
32	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-55.0	22,5	9,5
32	LAB-TI	04:18:d6:c2:08:99	-41.0	22,5	9,5
32	LAB-TEKKIM	06:18:d6:c2:08:99	-41.0	22,5	9,5
32	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-55.0	22,5	9,5
32	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-52.0	22,5	9,5
33	LAB-TI	80:2a:a8:c8:c5:c0	-54.0	20	9,5

Digital Repository Universitas Jember

33	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-54.0	20	9,5
33	LAB-TI	04:18:d6:c2:08:99	-42.8	20	9,5
33	LAB-TEKKIM	06:18:d6:c2:08:99	-42.7	20	9,5
33	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-53.8	20	9,5
33	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-51.9	20	9,5
34	LAB-TI	80:2a:a8:c8:c5:c0	-47.0	17,5	9,5
34	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-47.0	17,5	9,5
34	LAB-TI	04:18:d6:c2:08:99	-51.0	17,5	9,5
34	LAB-TEKKIM	06:18:d6:c2:08:99	-52.0	17,5	9,5
34	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-47.0	17,5	9,5
34	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-39.0	17,5	9,5
35	LAB-TI	80:2a:a8:c8:c5:c0	-47.3	15	9,5
35	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-47.3	15	9,5
35	LAB-TI	04:18:d6:c2:08:99	-52.1	15	9,5
35	LAB-TEKKIM	06:18:d6:c2:08:99	-52.5	15	9,5
35	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-47.3	15	9,5
35	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-39.2	15	9,5
36	LAB-TI	80:2a:a8:c8:c5:c0	-47.0	12,5	9,5
36	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-47.0	12,5	9,5
36	LAB-TI	04:18:d6:c2:08:99	-56.0	12,5	9,5
36	LAB-TEKKIM	06:18:d6:c2:08:99	-55.0	12,5	9,5
36	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-47.0	12,5	9,5
36	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-39.0	12,5	9,5
37	LAB-TI	80:2a:a8:c8:c5:c0	-46.1	10	9,5
37	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-46.1	10	9,5
37	LAB-TI	04:18:d6:c2:08:99	-55.3	10	9,5
37	LAB-TEKKIM	06:18:d6:c2:08:99	-54.6	10	9,5
37	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-46.1	10	9,5
37	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-39.0	10	9,5
42	LAB-TI	80:2a:a8:c8:c5:c0	-54.0	22,5	7
42	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-54.0	22,5	7

Digital Repository Universitas Jember

42	LAB-TI	04:18:d6:c2:08:99	-52.2	22,5	7
42	LAB-TEKKIM	06:18:d6:c2:08:99	-54.0	22,5	7
42	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-53.9	22,5	7
42	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-51.0	22,5	7
43	LAB-TI	80:2a:a8:c8:c5:c0	-54.1	20	7
43	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-54.1	20	7
43	LAB-TI	04:18:d6:c2:08:99	-48.9	20	7
43	LAB-TEKKIM	06:18:d6:c2:08:99	-48.0	20	7
43	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-53.2	20	7
43	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-51.3	20	7
44	LAB-TI	80:2a:a8:c8:c5:c0	-49.8	17,5	7
44	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-49.8	17,5	7
44	LAB-TI	04:18:d6:c2:08:99	-53.4	17,5	7
44	LAB-TEKKIM	06:18:d6:c2:08:99	-50.4	17,5	7
44	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-49.5	17,5	7
44	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-48.2	17,5	7
45	LAB-TI	80:2a:a8:c8:c5:c0	-48.0	15	7
45	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-48.0	15	7
45	LAB-TI	04:18:d6:c2:08:99	-55.0	15	7
45	LAB-TEKKIM	06:18:d6:c2:08:99	-52.0	15	7
45	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-48.0	15	7
45	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-47.0	15	7
46	LAB-TI	80:2a:a8:c8:c5:c0	-43.2	12,5	7
46	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-43.2	12,5	7
46	LAB-TI	04:18:d6:c2:08:99	-56.0	12,5	7
46	LAB-TEKKIM	06:18:d6:c2:08:99	-56.0	12,5	7
46	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-43.2	12,5	7
46	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-44.0	12,5	7
51	LAB-TI	80:2a:a8:c8:c5:c0	-56.1	25	4,5
51	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-56.1	25	4,5
51	LAB-TI	04:18:d6:c2:08:99	-54.3	25	4,5

Digital Repository Universitas Jember

51	LAB-TEKKIM	06:18:d6:c2:08:99	-54.1	25	4,5
51	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-56.1	25	4,5
51	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-58.6	25	4,5
52	LAB-TI	80:2a:a8:c8:c5:c0	-59.0	22,5	4,5
52	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-59.0	22,5	4,5
52	LAB-TI	04:18:d6:c2:08:99	-50.0	22,5	4,5
52	LAB-TEKKIM	06:18:d6:c2:08:99	-51.0	22,5	4,5
52	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-59.0	22,5	4,5
52	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-59.0	22,5	4,5
53	LAB-TI	80:2a:a8:c8:c5:c0	-51.5	20	4,5
53	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-51.5	20	4,5
53	LAB-TI	04:18:d6:c2:08:99	-53.7	20	4,5
53	LAB-TEKKIM	06:18:d6:c2:08:99	-54.0	20	4,5
53	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-51.5	20	4,5
53	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-54.6	20	4,5
54	LAB-TI	80:2a:a8:c8:c5:c0	-46.4	17,5	4,5
54	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-46.4	17,5	4,5
54	LAB-TI	04:18:d6:c2:08:99	-57.1	17,5	4,5
54	LAB-TEKKIM	06:18:d6:c2:08:99	-56.9	17,5	4,5
54	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-46.4	17,5	4,5
54	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-51.4	17,5	4,5
55	LAB-TI	80:2a:a8:c8:c5:c0	-45.0	15	4,5
55	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-45.0	15	4,5
55	LAB-TI	04:18:d6:c2:08:99	-59.0	15	4,5
55	LAB-TEKKIM	06:18:d6:c2:08:99	-58.3	15	4,5
55	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-45.0	15	4,5
55	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-45.9	15	4,5
56	LAB-TI	80:2a:a8:c8:c5:c0	-45.1	12,5	4,5
56	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-45.1	12,5	4,5
56	LAB-TI	04:18:d6:c2:08:99	-57.0	12,5	4,5
56	LAB-TEKKIM	06:18:d6:c2:08:99	-57.0	12,5	4,5

Digital Repository Universitas Jember

56	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-45.1	12,5	4,5
56	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-37.4	12,5	4,5
57	LAB-TI	80:2a:a8:c8:c5:c0	-43.9	10	4,5
57	S2-Teknik-Mesin	92:2a:a8:c8:c5:c0	-43.9	10	4,5
57	LAB-TI	04:18:d6:c2:08:99	-65.0	10	4,5
57	LAB-TEKKIM	06:18:d6:c2:08:99	-65.0	10	4,5
57	LAB-TEKKIM	82:2a:a8:c8:c5:c0	-44.9	10	4,5
57	S2-Teknik-Mesin	92:2a:a8:c7:c5:c0	-52.4	10	4,5

C. Program Yang Digunakan

Algoritma

```
package com.examples.indoorpositioningsystem.core;

import com.examples.indoorpositioningsystem.model.AccessPoint;
import com.examples.indoorpositioningsystem.model.IndoorProject;
import com.examples.indoorpositioningsystem.model.LocDistance;
import
com.examples.indoorpositioningsystem.model.LocationWithNearbyPlace
s;
import com.examples.indoorpositioningsystem.model.ReferencePoint;
import com.examples.indoorpositioningsystem.model.WifiDataNetwork;
import com.examples.indoorpositioningsystem.utils.AppContants;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;

import io.realm.RealmList;

public class Algorithms {


    /**
     *
     * @param latestScanList
     *            the current scan list of APs
     * @param proj
     *            the project details from db for current area
     *
     * @param algorithm_choice
     *            choice of several algorithms
```

Digital Repository Universitas Jember

```
     *
     * @return the location of user
     */
    public static LocationWithNearbyPlaces
processingAlgorithms(List<WifiDataNetwork> latestScanList,
IndoorProject proj, int algorithm_choice) {

        int i, j;
        RealmList<AccessPoint> aps = proj.getAps();
        ArrayList<Float> observedRSSValues = new ArrayList<Float>();
        WifiDataNetwork temp_LR;
        int notFoundCounter = 0;
        // Read parameter of algorithm
//      String NaNValue = readParameter(RM, 0);

        // Check which mac addresses of radio map, we are currently
listening.
        for (i = 0; i < aps.size(); ++i) {
            for (j = 0; j < latestScanList.size(); ++j) {
                temp_LR = latestScanList.get(j);
                // MAC Address Matched
                if
(aps.get(i).getMac_address().compareTo(temp_LR.getBssid()) == 0) {

observedRSSValues.add(Float.valueOf(temp_LR.getLevel()).floatValue
());
                    break;
                }
            }
            // A MAC Address is missing so we place a small value,
NaN value
            if (j == latestScanList.size()) {
                observedRSSValues.add(AppContants.NaN);
                ++notFoundCounter;
            }
        }

        if (notFoundCounter == aps.size())
            return null;

        // Read parameter of algorithm
        String parameter = readParameter(algorithm_choice);

        if (parameter == null)
            return null;

        switch (algorithm_choice) {

        case 1:
            return WKNN_Algorithm(proj, observedRSSValues, parameter,
true);
        case 2:
            return WKNN_Algorithm(proj, observedRSSValues, parameter,
true);
        case 3:
            return WKNN_Algorithm(proj, observedRSSValues, parameter,
```

Digital Repository Universitas Jember

```
true);
      case 4:
         return WKNN_Algorithm(proj, observedRSSValues, parameter,
true);
      case 5:
         return WKNN_Algorithm(proj, observedRSSValues, parameter,
true);
      case 6:
         return WKNN_Algorithm(proj, observedRSSValues, parameter,
true);
      case 7:
         return WKNN_Algorithm(proj, observedRSSValues, parameter,
false);
      }
      return null;

   }

   /**
    * Calculates user location based on Weighted/Not Weighted K
Nearest
    * Neighbor (KNN) Algorithm
    *
    * @param proj
    *            the project details from db for current area
    *
    * @param observedRSSValues
    *            RSS values currently observed
    * @param parameter
    *
    * @param isWeighted
    *            To be weighted or not
    *
    * @return The estimated user location
    */
   private static LocationWithNearbyPlaces
WKNN_Algorithm(IndoorProject proj, ArrayList<Float>
observedRSSValues, String parameter, boolean isWeighted) {
      RealmList<AccessPoint> rssValues;
      float curResult = 0;
      ArrayList<LocDistance> locDistanceResultsList = new
ArrayList<LocDistance>();
      String myLocation = null;
      int K;

      try {
         K = Integer.parseInt(parameter);
      } catch (Exception e) {
         return null;
      }

      // Construct a list with locations-distances pairs for
currently
      // observed RSS values
      for (ReferencePoint referencePoint : proj.getRps()) {
         rssValues = referencePoint.getReadings();
```

Digital Repository Universitas Jember

```
            curResult = calculateEuclideanDistance(rssValues,
observedRSSValues);

            if (curResult == Float.NEGATIVE_INFINITY)
               return null;

            locDistanceResultsList.add(0, new LocDistance(curResult,
referencePoint.getLocId(), referencePoint.getName()));
         }

         // Sort locations-distances pairs based on minimum distances
         Collections.sort(locDistanceResultsList, new
Comparator<LocDistance>() {
            public int compare(LocDistance gd1, LocDistance gd2) {
               return (gd1.getDistance() > gd2.getDistance() ? 1 :
(gd1.getDistance() == gd2.getDistance() ? 0 : -1));
            }
         });

         if (!isWeighted) {
            myLocation =
calculateAverageKDistanceLocations(locDistanceResultsList, K);
         } else {
            myLocation =
calculateWeightedAverageKDistanceLocations(locDistanceResultsList,
K);
         }
         LocationWithNearbyPlaces places = new
LocationWithNearbyPlaces(myLocation, locDistanceResultsList);
         return places;

      }


      /**
       * Calculates the Euclidean distance between the currently
observed RSS
       * values and the RSS values for a specific location.
       *
       * @param l1
       *            RSS values of a location in stored in AP obj of
locations
       * @param l2
       *            RSS values currently observed
       *
       * @return The Euclidean distance, or MIN_VALUE for error
       */
      private static float
calculateEuclideanDistance(RealmList<AccessPoint> l1,
ArrayList<Float> l2) {

         float finalResult = 0;
         float v1;
         float v2;
         float temp;
```

Digital Repository Universitas Jember

```
      for (int i = 0; i < l1.size(); ++i) {

         try {
            l1.get(i).getMeanRss();
            v1 = (float) l1.get(i).getMeanRss();
            v2 = l2.get(i);
         } catch (Exception e) {
            return Float.NEGATIVE_INFINITY;
         }

         // do the procedure
         temp = v1 - v2;
         temp *= temp;

         // do the procedure
         finalResult += temp;
      }
      return ((float) Math.sqrt(finalResult));
   }


   /**
    * Calculates the Average of the K locations that have the
shortest
    * distances D
    *
    * @param LocDistance_Results_List
    *           Locations-Distances pairs sorted by distance
    * @param K
    *           The number of locations used
    * @return The estimated user location, or null for error
    */
   private static String
calculateAverageKDistanceLocations(ArrayList<LocDistance>
LocDistance_Results_List, int K) {
      float sumX = 0.0f;
      float sumY = 0.0f;

      String[] LocationArray = new String[2];
      float x, y;

      int K_Min = K < LocDistance_Results_List.size() ? K :
LocDistance_Results_List.size();

      // Calculate the sum of X and Y
      for (int i = 0; i < K_Min; ++i) {
         LocationArray =
LocDistance_Results_List.get(i).getLocation().split(" ");

         try {
            x =
Float.valueOf(LocationArray[0].trim()).floatValue();
            y =
Float.valueOf(LocationArray[1].trim()).floatValue();
         } catch (Exception e) {
            return null;
```

Digital Repository Universitas Jember

```
            }

            sumX += x;
            sumY += y;
         }

         // Calculate the average
         sumX /= K_Min;
         sumY /= K_Min;

         return sumX + " " + sumY;
      }

      /**
       * Calculates the Weighted Average of the K locations that have
the shortest
       * distances D
       *
       * @param LocDistance_Results_List
       *            Locations-Distances pairs sorted by distance
       * @param K
       *            The number of locations used
       * @return The estimated user location, or null for error
       */
      private static String
calculateWeightedAverageKDistanceLocations(ArrayList<LocDistance>
LocDistance_Results_List, int K) {
         double LocationWeight = 0.0f;
         double sumWeights = 0.0f;
         double WeightedSumX = 0.0f;
         double WeightedSumY = 0.0f;

         String[] LocationArray = new String[2];
         float x, y;

         int K_Min = K < LocDistance_Results_List.size() ? K :
LocDistance_Results_List.size();

         // Calculate the weighted sum of X and Y
         for (int i = 0; i < K_Min; ++i) {
            if (LocDistance_Results_List.get(i).getDistance() != 0.0)
{
               LocationWeight = 1 /
LocDistance_Results_List.get(i).getDistance();
            } else {
               LocationWeight = 100;
            }
            LocationArray =
LocDistance_Results_List.get(i).getLocation().split(" ");

            try {
               x =
Float.valueOf(LocationArray[0].trim()).floatValue();
               y =
Float.valueOf(LocationArray[1].trim()).floatValue();
            } catch (Exception e) {
```

Digital Repository Universitas Jember

```
            return null;
         }

         sumWeights += LocationWeight;
         WeightedSumX += LocationWeight * x;
         WeightedSumY += LocationWeight * y;

      }

      WeightedSumX /= sumWeights;
      WeightedSumY /= sumWeights;

      return WeightedSumX + " " + WeightedSumY;
   }

   /**
    * Reads the parameters from the file
    *
    * @param file
    *            the file of radiomap, to read parameters
    *
    * @param algorithm_choice
    *            choice of several algorithms
    *
    * @return The parameter for the algorithm
    *
    */
   private static String readParameter(File file, int
algorithm_choice) {
      String line;
      BufferedReader reader = null;

      String parameter = null;

      try {
         FileReader fr = new
FileReader(file.getAbsolutePath().replace(".txt", "-
parameters2.txt"));

         reader = new BufferedReader(fr);

         while ((line = reader.readLine()) != null) {

            /* Ignore the labels */
            if (line.startsWith("#") || line.trim().equals("")) {
               continue;
            }

            /* Split fields */
            String[] temp = line.split(":");

            /* The file may be corrupted so ignore reading it */
            if (temp.length != 2) {
               return null;
            }
```

Digital Repository Universitas Jember

```
            if (algorithm_choice == 0 && temp[0].equals("NaN")) {
               parameter = temp[1];
               break;
            } else if (algorithm_choice == 1 &&
temp[0].equals("WKKN1")) {
               parameter = temp[1];
               break;
            } else if (algorithm_choice == 2 &&
temp[0].equals("WKKN2")) {
               parameter = temp[1];
               break;
            } else if (algorithm_choice == 3 &&
temp[0].equals("WKKN3")) {
               parameter = temp[1];
               break;
            } else if (algorithm_choice == 4 &&
temp[0].equals("WKKN4")) {
               parameter = temp[1];
               break;
            } else if (algorithm_choice == 5 &&
temp[0].equals("WKKN5")) {
               parameter = temp[1];
               break;
            } else if (algorithm_choice == 6 &&
temp[0].equals("WKKN6")) {
               parameter = temp[1];
               break;
            } else if (algorithm_choice == 7 &&
temp[0].equals("KKN1")) {
               parameter = temp[1];
               break;
            }

         }

      } catch (Exception e) {
         return null;
      } finally {
         if (reader != null)
            try {
               reader.close();
            } catch (IOException e) {
            }
      }

      return parameter;
   }

   private static String readParameter(int algorithm_choice) {
      String parameter = null;

      if (algorithm_choice == 1) {
         // && ("WKKN1")
         parameter = "1";
      } else if (algorithm_choice == 2) {
         // && ("WKKN2")
```

Digital Repository Universitas Jember

```
            parameter = "2";
        } else if (algorithm_choice == 3) {
            // && ("WKKN3")
            parameter = "3";
        } else if (algorithm_choice == 4) {
            // && ("WKKN4")
            parameter = "4";
        } else if (algorithm_choice == 5) {
            // && ("WKKN5")
            parameter = "5";
        } else if (algorithm_choice == 6) {
            // && ("WKKN6")
            parameter = "6";
        } else if (algorithm_choice == 7) {
            // && ("KKN1")
            parameter = "1";
        }
        return parameter;
    }

}
```