

# Implementasi Algoritma K-Nearest Neighbor (KNN) pada Sistem Presensi Berbasis RFID

**Muhammad Zikrillah**

Jurusan Sistem Komputer  
Universitas Sriwijaya  
Indralaya, Indonesia  
emzikrillah54@gmail.com

**Ahmad Fali Oklilas**

Jurusan Sistem Komputer  
Universitas Sriwijaya  
Indralaya, Indonesia  
fali@ilkom.unsri.com

**Abstrak**—Sistem Presensi adalah memberikan suatu informasi tentang status kehadiran seseorang baik itu karyawan, pegawai, dan mahasiswa pada suatu institusi atau perusahaan. Algoritma *K-Nearest Neighbor* (KNN) dapat diimplementasikan untuk melakukan sistem presensi berbasis RFID. Pada alat RFID terdapat *tag*, *reader* dan PC yang digunakan untuk mengambil nilai RSSI guna mengetahui posisi dari objek yang telah dipasang *tag*. Setiap *tag* memiliki id unik untuk memberikan informasi pada *reader* agar dapat mengenali lokasinya dengan mudah. Hasil dari penelitian paper ini menghasilkan nilai rata-rata *error* sebesar 4,67126 cm dengan nilai minimum *error* 0,4320 cm dan nilai maksimum *error* 8,6655 cm.

**Kata Kunci** :—*rfid, k-nearest neighbor, rssi*

## I. PENDAHULUAN

*Radio Frequency Identification* (RFID) yang telah menjadi minat beberapa tahun terakhir untuk digunakan pada industri secara luas, jika di bandingkan dengan teknik- teknik lainnya untuk dipelajari secara ekstensif untuk melokalisasikan [1]. Pada lingkungan yang berada dalam ruangan informasi akan lebih spesifik dan lebih dominan menggunakan *Wireless Local Area Network* (WLAN) dari RFID, *Ultrasound*, *Bluetooth* dan kombinasi teknologi lainnya. Alat RFID terdapat *tag* RFID, *reader* RFID dan PC. Setiap *tag* yang ada memiliki ID unik sehingga menghubungkan berbagai informasi.

Pada ID unik memberikan informasi kepada pengguna dapat mengenali lokasi *tag* RFID dengan mudah. Lingkungan yang ada didalam ruangan terdapat sistem yang terdiri dari fungsi perkiraan posisi *tag* RFID dan *reader*, serta fungsi kuatnya sinyal. Sehingga menggunakan metode lokalisasi yang dapat dioperasikan menggunakan *Received Signal Strength Indicator* (RSSI) yang dapat mengurangi biaya dan bahkan dapat diukur. Pada keberadaan posisi suatu objek orang atau benda terdapat estimasi atau perkiraan posisi *tag* target salah satunya dengan menggunakan algoritma *K-Nearest Neighbor* (KNN) [2].

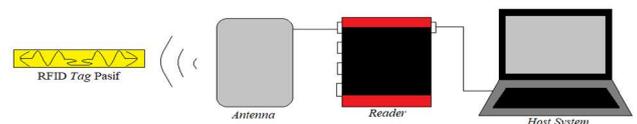
## II. TINJAUAN PUSTAKA

### A. Sistem Lokalisasi

Lokalisasi adalah perkiraan menentukan suatu posisi dengan melalui komunikasi antara titik satu dengan titik lainnya seperti *localized node* (*node* yang diketahui) dan *unlocalized node* (*node* yang tidak diketahui) [3].

### B. Radio Frequency Identification (RFID)

*Radio Frequency Identification* (RFID) adalah sensor yang mampu mengidentifikasi suatu objek seperti orang ataupun benda dengan menggunakan frekuensi radio [4]. Identifikasi dengan menggunakan RFID adalah teknologi identifikasi dan penangkapan data otomatis yang telah mendapatkan perhatian yang signifikan dalam beberapa tahun terakhir sebagai sarana meningkatkan ketertelusuran barang pada suatu rantai pasokan. Selanjutnya merupakan gambar sistem RFID yang digunakan pada gambar 1 sebagai berikut,



Gambar 1. Sistem RFID

### C. Sistem Presensi

Sistem Presensi adalah memberikan suatu informasi tentang status kehadiran seseorang baik itu karyawan, pegawai, dan mahasiswa pada suatu institusi atau perusahaan. Hal ini telah dikembangkan sebagai solusi hadir atau tidak nya seseorang dalam suatu lembaga tersebut yang menggunakan *presence system*. [5].

### D. Received Signal Strength Indicator (RSSI)

*Received Signal Strength Indicator* (RSSI) adalah suatu teknologi yang dapat digunakan sebagai pengukur indikator kekuatan sinyal yang diterima oleh perangkat wireless [6].

### E. Algoritma K-Nearest Neighbor (KNN)

Algoritma K-Nearest Neighbor (KNN) adalah metode klarifikasi yang mengelompokkan sebuah data baru berdasarkan dengan suatu jarak data baru tersebut kepada beberapa data tetangga (*neighbor*) yang terdekat [7]. Persamaan *k* Nearest Neighbour dapat dilihat pada persamaan (1), (2) dan (3) berikut:

$$E_i = |RSSI_{target} - R_i| \quad (1)$$

$$W_i = \frac{\frac{1}{E_i^2}}{\sum_{j=1}^k \frac{1}{E_j^2}} \quad (2)$$

$$(x_e, y_e) = \sum_{i=1}^k W_i(x_i, y_i) \quad (3)$$

Dengan menggunakan posisi fisik dari semua *tag* referensi ( $x_i, y_i$ ) untuk memperkirakan posisi target ( $x_e, y_e$ ). Untuk mencari kesalahan estimasi antara posisi aktual *tag* target ( $x, y$ ) dan posisi *tag* perkiraan ( $x_e, y_e$ ) dapat dihitung menggunakan persamaan (4) berikut:

$$error = \sqrt{(x - x_e)^2 + (y - y_e)^2} \quad (4)$$

### III. METODE PENELITIAN

Secara garis besar ada tiga tahapan proses utama dalam melakukan penelitian kerja ini yaitu: pertama pengambilan data nilai RSSI dari *tag* referensi dan *tag* target menggunakan alat perangkat RFID *reader*; kedua mencari jarak euclidian dari masing-masing *tag* referensi ke *tag* target serta mencari bobot nilai dari masing-masing jarak Euclidian; dan ketiga mencari posisi perkiraan koordinat ( $x, y$ ) juga mencari rata-rata error dari hasil pencarian posisi perkiraan tersebut.

Rancangan penelitian akan dilakukan melalui beberapa tahapan, tahapan yang dilakukan mengikuti kerangka kerja (*frame work*) yang telah dirancang sebelumnya oleh peneliti sehingga pengerjaan penelitian kerja ini terstruktur dan mengikuti sesuai alur. Adapun bentuk dari kerangka kerja tersebut dapat dilihat pada gambar 2.

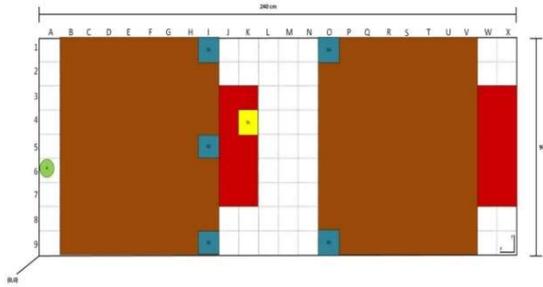


Gambar 2. Flowchart Penelitian

#### A. Inisialisasi Lingkungan Kerja

Tahapan ini akan menjelaskan lingkungan kerja yang akan digunakan dalam Penelitian ini yaitu ruangan Lab Jaringan Komputer Fasilkom Unsri Bukit. Ruangan ini memiliki ukuran 8.3m x 5m. Namun dalam penelitian kali ini hanya menggunakan ukuran 240 cm x 90 cm karena disesuaikan dengan batas stabil kemampuan jangkauan antenna *reader*. Pada meja yang berada di depan akan diletakan 3 buah *tag* referensi dan pada meja yang berada dibelakang akan diletakan 2 buah *tag* referensi. Sedangkan *tag* target diletakan di bagian kursi yang sejajar dengan meja.

Pada ruang kerja ini akan dibagi menjadi 24 kolom dan 9 baris, dimana pada bagian kolom akan diberi simbol huruf A sampai X sedangkan pada bagian baris akan diberi simbol angka 1 sampai 9. Pada ruang kerja ini akan terbentuk 216 bagian persegi yang masing-masing bagian persegi memiliki sisi ( $r$ ) sepanjang 10 cm. Untuk lebih jelasnya dapat dilihat pada gambar 3 berikut.



Gambar 3. Penempatan Tag-Tag

Untuk mengetahui lokasi koordinat pada antena, tag referensi dan tag target diuraikan pada tabel 1.

Tabel 1

Letak Antena Reader dan Tag Referensi

| No | Label           | Simbol | Lokasi | Koordinat  |
|----|-----------------|--------|--------|------------|
| 1  | Antena          | A      | A6     | (35 , 0)   |
| 2  | Tag Target      | Rt     | 4K     | (54 , 99)  |
| 3  | Tag Referensi 1 | R1     | 1I     | (84 , 79)  |
| 4  | Tag Referensi 2 | R2     | 5I     | (45 , 79)  |
| 5  | Tag Referensi 3 | R3     | 9I     | (6 , 79)   |
| 6  | Tag Referensi 4 | R4     | 1O     | (84 , 131) |
| 7  | Tag Referensi 5 | R5     | 9O     | (6 , 131)  |

### B. Pengambilan Data RSSI

Untuk dapat mengambil data nilai RSSI pada suatu tag, yang pertama harus dilakukan adalah masuk ke menu Tag Programmer yang terdapat pada interface pertama Alien RFID. Kemudian untuk dapat menggunakan menu ini pengguna diwajibkan untuk masuk/log in ke dalam sistem terlebih dahulu menggunakan username dan password.

### C. Perhitungan K Nearest Neighbor

Untuk mencari posisi perkiraan koordinat (x,y) pada tag target peneliti menggunakan persamaan *K Nearest Neighbor* (KNN). Setelah mendapatkan data dari nilai RSSI masing-masing tag menggunakan perangkat RFID reader selanjutnya melakukan proses pengolahan data untuk mencari posisi perkiraan koordinat (x,y) dan rata-rata error dari masing-masing pengambilan data nilai RSSI. Pada perhitungan K Nearest Neighbor ini nilai RSSI tag target dan tag referensi digunakan sebagai acuan untuk mencari jarak euclidian bobot nilai untuk mendapatkan hasil posisi perkiraan dan error yang dihasilkan.

### D. Pencarian Nilai Kesalahan (Error) dan Rata-Rata

Setelah melakukan proses perhitungan *K Nearest Neighbor* (KNN) dan mendapatkan posisi perkiraan koordinat (x,y) tag target terbaik maka selanjutnya

adalah melakukan pencarian nilai kesalahan (*error*) dari masing-masing posisi perkiraan koordinat tag target dengan posisi koordinat tag target sebenarnya dan menghitung rata-rata *error* yang dihasilkan.

## IV. HASIL DAN DISKUSI

Dalam percobaan penelitian paper kali ini pengambilan data nilai RSSI dari RFID reader pada 5 buah tag referensi dan 1 buah tag target akan diambil sebanyak 20 kali secara beruntun dan bersamaan. Maksudnya data nilai RSSI pada tag referensi 1, tag referensi 2, tag referensi 3, tag referensi 4, tag referensi 5 dan tag target diambil secara bersamaan bukan satu target persatu dan data tersebut diambil secara beruntun. Pada tabel dibawah ini akan diperlihatkan hasil pengambilan data nilai RSSI sebanyak 20 kali.

Tabel 2

Nilai RSSI pada Tag Target dan Tag Referensi

| No | Rt   | R1  | R2  | R3  | R4  | R5  |
|----|------|-----|-----|-----|-----|-----|
| 1  | 1329 | 897 | 620 | 397 | 625 | 767 |
| 2  | 1412 | 839 | 721 | 435 | 565 | 708 |
| 3  | 1397 | 878 | 667 | 389 | 521 | 674 |
| 4  | 1338 | 767 | 679 | 459 | 565 | 604 |
| 5  | 1334 | 897 | 676 | 416 | 617 | 638 |
| 6  | 1314 | 813 | 719 | 400 | 555 | 706 |
| 7  | 1449 | 776 | 657 | 456 | 528 | 740 |
| 8  | 1444 | 870 | 656 | 404 | 594 | 626 |
| 9  | 1359 | 820 | 631 | 467 | 590 | 759 |
| 10 | 1299 | 761 | 754 | 480 | 634 | 741 |
| 11 | 1461 | 798 | 760 | 433 | 551 | 774 |
| 12 | 1431 | 726 | 625 | 422 | 558 | 597 |
| 13 | 1365 | 777 | 787 | 407 | 539 | 691 |
| 14 | 1501 | 749 | 732 | 368 | 482 | 619 |
| 15 | 1366 | 782 | 713 | 400 | 594 | 690 |
| 16 | 1308 | 740 | 699 | 434 | 549 | 763 |
| 17 | 1348 | 812 | 728 | 444 | 613 | 681 |
| 18 | 1315 | 813 | 731 | 443 | 508 | 726 |
| 19 | 1341 | 703 | 602 | 420 | 563 | 652 |
| 20 | 1427 | 872 | 806 | 498 | 547 | 713 |

### A. Perhitungan K Nearest Neighbor

#### • Jarak Euclidian (Ei)

Jarak euclidian merupakan proses awal dalam perhitungan *k nearest neighbor* dengan menggunakan persamaan (1). Berikut hasil jarak euclidian pada tabel 3.

Tabel 3

Data nilai Jarak Euclidian (Ei)

| No | E1  | E2  | E3   | E4  | E5  |
|----|-----|-----|------|-----|-----|
| 1  | 432 | 709 | 932  | 704 | 562 |
| 2  | 573 | 691 | 977  | 847 | 704 |
| 3  | 519 | 730 | 1008 | 876 | 723 |
| 4  | 571 | 659 | 879  | 773 | 734 |
| 5  | 437 | 658 | 918  | 717 | 696 |
| 6  | 501 | 595 | 914  | 759 | 608 |
| 7  | 673 | 792 | 993  | 921 | 709 |
| 8  | 574 | 788 | 1040 | 850 | 818 |

|    |     |     |      |      |     |
|----|-----|-----|------|------|-----|
| 9  | 539 | 728 | 892  | 769  | 600 |
| 10 | 538 | 545 | 819  | 665  | 558 |
| 11 | 663 | 701 | 1028 | 910  | 687 |
| 12 | 705 | 806 | 1009 | 873  | 834 |
| 13 | 588 | 578 | 958  | 826  | 674 |
| 14 | 752 | 769 | 1133 | 1019 | 882 |
| 15 | 584 | 653 | 966  | 772  | 676 |
| 16 | 568 | 609 | 874  | 759  | 545 |
| 17 | 536 | 620 | 904  | 735  | 667 |
| 18 | 502 | 584 | 872  | 807  | 589 |
| 19 | 638 | 739 | 921  | 778  | 689 |
| 20 | 555 | 621 | 929  | 880  | 714 |

• **Faktor Bobot (Wi)**

Setelah hasil jarak euclidian didapat, maka selanjutnya mencari nilai faktor bobot dengan menggunakan persamaan (2). Berikut hasil faktor bobot pada tabel 4.

Tabel 4  
Data nilai Faktor Bobot (Wi)

| No | W <sub>1</sub> | W <sub>2</sub> | W <sub>3</sub> | W <sub>4</sub> | W <sub>5</sub> |
|----|----------------|----------------|----------------|----------------|----------------|
| 1  | 0,3916         | 0,1453         | 0,0841         | 0,1474         | 0,2314         |
| 2  | 0,3172         | 0,2181         | 0,1091         | 0,1452         | 0,2101         |
| 3  | 0,3792         | 0,1916         | 0,1005         | 0,1331         | 0,1954         |
| 4  | 0,3008         | 0,2258         | 0,1269         | 0,1641         | 0,1820         |
| 5  | 0,4109         | 0,1812         | 0,0931         | 0,1526         | 0,1620         |
| 6  | 0,3200         | 0,2269         | 0,0961         | 0,1394         | 0,2173         |
| 7  | 0,2765         | 0,1996         | 0,1270         | 0,1476         | 0,2491         |
| 8  | 0,3592         | 0,1906         | 0,1094         | 0,1638         | 0,1768         |
| 9  | 0,3113         | 0,1706         | 0,1136         | 0,1529         | 0,2512         |
| 10 | 0,2506         | 0,2442         | 0,1081         | 0,1640         | 0,2329         |
| 11 | 0,2650         | 0,2371         | 0,1102         | 0,1407         | 0,2468         |
| 12 | 0,2762         | 0,2113         | 0,1348         | 0,1801         | 0,1973         |
| 13 | 0,2717         | 0,2812         | 0,1023         | 0,1377         | 0,2068         |
| 14 | 0,2726         | 0,2606         | 0,1200         | 0,1484         | 0,1981         |
| 15 | 0,2870         | 0,2295         | 0,1049         | 0,1642         | 0,2142         |
| 16 | 0,2539         | 0,2208         | 0,1072         | 0,1421         | 0,2757         |
| 17 | 0,3052         | 0,2281         | 0,1072         | 0,1623         | 0,1970         |
| 18 | 0,3141         | 0,2320         | 0,1040         | 0,1215         | 0,2281         |
| 19 | 0,2663         | 0,1984         | 0,1277         | 0,1790         | 0,2283         |
| 20 | 0,3166         | 0,2529         | 0,1130         | 0,1259         | 0,1913         |

• **Pencarian Posisi Perkiraan (xe,ye)**

Proses terakhir dari perhitungan *k nearest neighbor* (KNN) adalah pencarian posisi posisi perkiraan. Berikut hasil posisi perkiraan pada tabel 5.

Tabel 5

Posisi perkiraan koordinat Tag Target

| No | Xe      | Ye       |
|----|---------|----------|
| 1  | 53,7075 | 98,6818  |
| 2  | 50,5835 | 97,4808  |
| 3  | 53,4395 | 96,0839  |
| 4  | 51,0841 | 97,0055  |
| 5  | 57,0305 | 95,3625  |
| 6  | 60,6955 | 97,5536  |
| 7  | 46,8721 | 99,6336  |
| 8  | 54,2329 | 96,7170  |
| 9  | 48,8755 | 100,0209 |
| 10 | 47,8677 | 99,6445  |
| 11 | 46,8971 | 99,1537  |
| 12 | 49,8414 | 98,6324  |
| 13 | 48,9105 | 96,9177  |
| 14 | 49,0095 | 97,0246  |
| 15 | 50,1542 | 98,6809  |
| 16 | 45,5099 | 100,7352 |
| 17 | 51,3618 | 97,6887  |
| 18 | 49,0319 | 97,1848  |
| 19 | 48,4810 | 100,1859 |
| 20 | 50,3929 | 95,5006  |

B. *Pencarian Nilai Rata-Rata Error*

Proses perhitungan *k nearest neighbor* (KNN) memiliki nilai kesalahan (*error*) dari posisi perkiraan terhadap posisi aktual tag target. Berikut hasil nilai *error* dan rata-rata *error* pada tabel 6.

Tabel 6

Posisi perkiraan, nilai kesalahan (*error*) dan rata-rata

| No              | xe      | Ye       | error   |
|-----------------|---------|----------|---------|
| 1               | 53,7075 | 98,6818  | 0,4320  |
| 2               | 50,5835 | 97,4808  | 3,7390  |
| 3               | 53,4395 | 96,0839  | 2,9694  |
| 4               | 51,0841 | 97,0055  | 3,5326  |
| 5               | 57,0305 | 95,3625  | 4,7344  |
| 6               | 60,6955 | 97,5536  | 3,6071  |
| 7               | 46,8721 | 99,6336  | 7,1559  |
| 8               | 54,2329 | 96,7170  | 2,2948  |
| 9               | 48,8755 | 100,0209 | 5,2251  |
| 10              | 47,8677 | 99,6445  | 6,1660  |
| 11              | 46,8971 | 99,1537  | 7,1045  |
| 12              | 49,8414 | 98,6324  | 4,1747  |
| 13              | 48,9105 | 96,9177  | 5,4989  |
| 14              | 49,0095 | 97,0246  | 5,3671  |
| 15              | 50,1542 | 98,6809  | 3,8589  |
| 16              | 45,5099 | 100,7352 | 8,6655  |
| 17              | 51,3618 | 97,6887  | 2,9460  |
| 18              | 49,0319 | 97,1848  | 5,2829  |
| 19              | 48,4810 | 100,1859 | 5,6449  |
| 20              | 50,3929 | 95,5006  | 5,0255  |
| Rata-rata error |         |          | 4,67126 |

V. ANALISA

Penggunaan perhitungan algoritma *k nearest neighbor* (KNN) dalam mencari posisi koordinat perkiraan objek (tag target) dapat menghasilkan posisi koordinat perkiraan terbaik karena posisinya yang

mendekati posisi aktual (posisi sebenarnya) dan memiliki nilai rata-rata kesalahan (*error*) yang tidak terlalu besar (dalam paper kali ini saya menggunakan satuan centimeter (cm)).

Nilai RSSI yang merupakan data awal memiliki peranan penting dalam pencarian posisi koordinat perkiraan objek yang dihasilkan menggunakan perhitungan *K Nearest Neighbor* (KNN). Sebab jika nilai RSSI yang dihasilkan oleh tag pasif terlalu kecil maka posisi perkiraan yang dihasilkan akan lebih jauh dari posisi aktual dan membuat nilai kesalahan (*error*) menjadi lebih besar. Oleh karena itu untuk pengambilan data nilai RSSI tag pasif oleh antena RFID reader sangat perlu diperhatikan dalam prosesnya. Faktor-faktor seperti posisi penempatan antena RFID dalam memancarkan sinyal, lokasi pengambilan data yang terlalu banyak noise, dan lain sebagainya.

Alangkah baiknya jika posisi aktual dari objek (*tag target*) berada pada sekitaran ruang lingkup posisi aktual *tag referensi* sehingga posisi perkiraan yang dihasilkan menjadi sangat baik dibandingkan dengan jika posisi aktual dari objek (*tag target*) ditempatkan di luar ruang lingkup posisi aktual *tag referensi*.

#### VI. KESIMPULAN

*K Nearest Neighbor* dapat dijadikan salah satu metode terbaik dalam mencari posisi suatu objek berdasarkan posisi perkiraan yang dihasilkan nilai RSSI *tag pasif*. Seperti pada paper kali ini posisi aktual dari objek yaitu (54,99) dan menghasilkan nilai kesalahan (*error*) yang tidak terlalu besar yaitu 4,67126 cm.

Nilai RSSI memegang peranan penting dalam mencari posisi perkiraan objek seperti perbandingan pada posisi koordinat perkiraan (53,7075), (98,6818) yang memiliki nilai *error* 0,4320 cm dan posisi koordinat

(45,5099), (100,7352) yang memiliki nilai *error* 8,6655 cm.

Merujuk pada kesimpulan nomor 2 maka semakin besar nilai RSSI yang dihasilkan maka semakin akurat posisi perkiraan yang didapat begitu juga sebaliknya semakin kecil nilai RSSI yang dihasilkan maka semakin jelek posisi perkiraan yang didapat.

Untuk mencari nilai kesalahan (*error*), nilai posisi perkiraan yang tidak boleh dibulatkan karena apabila hasil nilai posisi perkiraan dibulatkan maka nilai *error* yang didapatkan akan berbeda sangat jauh dengan nilai *error* yang didapatkan dengan yang tidak dibulatkan.

#### REFERENCES

- [1] Zheng Jiali, Tuanfa Qin, Jieming Wu and Li Wau, "RFID Indoor Localization Based on Relational Aggregation", Chang Mai, Thailand, pp.41-44, 2016
- [2] Adege Belay Abebe, Hsin-piao Lin, Yirga Yeayeh, Lei yen, Getaneh Berie, and Yun Ruei Li, "Indoor Localization Using K-nearest Neighbor and Artificial Neural Network Back Propagation Algorithms"WOCC, 2018
- [3] Putra Adam Surya, Prima Kristalina, dan Amang Sudarsono, "Aplikasi Indoor Secured-Localization System Menggunakan Jaringan Sensor Nirkabel untuk Koordinasi Pasukan PMK pada Kondisi Darurat Kebakaran di dalam Gedung", pp.C22-C30,2016
- [4] Rerungan Juprianto, Deny Wira Nugraha, dan Yusuf Anshori, "SISTEM PENGAMAN PINTU OTOMATIS MENGGUNAKAN RADIO FREQUENCY IDENTIFICATION (RFID) TAG CARD DAN PERSONAL IDENTIFICATION NUMBER (PIN) BERBASIS MIKROKONTROLER AVR ATMEGA128", Universitas Tadulako, 2014
- [5] Isoda Yodhinori, Hirobumi Nakayama, Shin Oguri and Naoharu Yamada "Indoor Presence System Using Wireless LAN," NTT DoCoMo Technical Journal Vol.9, No.2.
- [6] Zafari Faheem, and Athanasios, "A Survey of Indoor Localization Systems and Technologies", IEEE, 2018].
- [7] Rohman Abdul, "MODEL ALGORITMA K-NEAREST NEIGHBOR (K-NN) UNTUK PREDIKSI KELULUSAN MAHASISWA", Universitas Pandanaran, Semarang.