Journal of Innovative and Creativity
Journal Homepage: https://joecy.org/index.php/joecy
e-mail: galungwaknook@gmail.co

Faculty of Education
University of Pahlawan Tuanku Tambusai
ISSN 2775-771X (Online)
ISSN 2962-570X (Printed)
6 (1) 2026, Page: 17568-17581

PEMETAAN SEBARAN FASKES UNTUK PENANGGULANGAN PENYAKIT DBD MENGGUNAKAN *K-MEANS* DI DAERAH MUSI BANYUASIN, PALEMBANG, DAN MUARA ENIM

Lacta Jaya Fultrian, Shinta Puspasari, Evi Purnamasari
Program Studi Teknik Informatika Fakultas Ilmu Komputer Dan Sains
Universitas Indo Global Mandiri
Corresponding Author : **Lacta Jaya Fultrian,,** e-mail: 2020110046@students.uigm.ac.id

Published: March 2026

ABSTRACT
Demam Berdarah Dengue (DBD) menjadi ancaman kesehatan global dengan 14,1 juta kasus pada 2024, khususnya di Sumatera Selatan, Indonesia, di mana distribusi fasilitas kesehatan tidak merata menghambat respons efektif. Penelitian ini bertujuan menerapkan klasterisasi K-means berbasis jarak Haversine untuk memetakan sebaran fasilitas kesehatan di Musi Banyuasin, Palembang, dan Muara Enim guna pengelolaan DBD yang lebih baik. Menggunakan desain kuantitatif eksplanatori, penelitian menerapkan census sampling terhadap 175 fasilitas kesehatan, dengan pemrosesan data geospasial menggunakan Python (Pandas, Scikit-learn) melalui tahap KDD: seleksi, preprocessing, transformasi, data mining, dan evaluasi. Hasil menunjukkan klasterisasi optimal pada K=3 (kategori dekat, sedang, jauh) dengan Silhouette Score 0,79, terintegrasi dalam sistem web interaktif berbasis Laravel dan Leaflet.js. Aplikasi web memungkinkan identifikasi fasilitas terdekat secara cepat bagi pengguna dan pengelolaan data komprehensif bagi administrator. Penelitian menyimpulkan bahwa klasterisasi K-means geospasial meningkatkan aksesibilitas layanan kesehatan dan pengambilan keputusan respons wabah DBD secara signifikan.

Kata Kunci: Algoritma K-Means, Demam Berdarah Dengue, Jarak Haversine, Klasterisasi Geospasial, Silhouette Score

© The Author(s). 2021 Open Access This article is licensed under a Creative Commons Atibution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0

PENDAHULUAN
Demam Berdarah Dengue (DBD) merupakan salah satu penyakit tropis yang paling cepat menyebar secara global, dengan peningkatan kasus mencapai rekor 14,1 juta pada 2024, melebihi 7 juta kasus tahun sebelumnya.

DBD telah menjadi beban kesehatan masyarakat utama, dengan insidens global melonjak dari 26,45 juta kasus pada 1990 menjadi 58,96 juta pada 2021, disertai kematian naik dari 14.315 menjadi 29.075. Penularan penyakit DBD disebabkan oleh virus dengue melalui gigitan nyamuk Aides aegepty betina(Widya A and Dhamayanti, 2018). Tren ini didorong oleh perubahan iklim, urbanisasi, dan mobilitas populasi, yang mempercepat penyebaran vektor Aedes aegypti. Grand theory epidemiologi seperti model SIR (Susceptible-Infected-Recovered) telah berevolusi dengan integrasi faktor spasial, di mana teori terkini menekankan peran machine learning untuk memprediksi klaster penyakit.

Pemikiran teoretis tentang DBD bergeser dari pendekatan konvensional berbasis entomologi ke model prediktif berbasis data, dengan unsupervised learning seperti K-means clustering

mendominasi untuk analisis spasial. K-Means merupakan algoritma clustering yang berulang-ulang(Sari et al., 2021). K-Means Clustering adalah salah satu algoritma partisional yang bergantung pada penentuan jumlah awal kelompok dengan menetapkan nilai centroid awal(Haversyalapa et al., 2024). Metode k-Means sering menerapkan rumus jarak Euclidean untuk menentukan kesamaan data dalam klaster secara iteratif(Gustriansyah et al., 2020). Sintesis kutipan pendukung menunjukkan K-means efektif dalam mengelompokkan zona risiko DBD berdasarkan kepadatan kasus dan fasilitas kesehatan, seperti pada studi di Semarang yang mengintegrasikan silhouette score untuk validasi klaster. Namun, evaluasi kutipan bertentangan mengidentifikasi perdebatan mengenai sensitivitas K-means terhadap outlier data geospasial, di mana beberapa metodologi alternatif seperti DBSCAN lebih robust untuk distribusi tidak merata.

Pemetaan metodologi menunjukkan dominasi K-means (60% studi terkini) untuk klasterisasi fasilitas kesehatan (faskes) dan zona endemik DBD, sering dikombinasikan dengan rumus Haversine untuk jarak geospasial. rumusan yang diterapkan dalam navigasi dengan menggunakan nilai jarak lingkaran besar antara dua titik pada permukaan bola bumi berdasarkan bujur dan lintang(Pribadi et al., 2022). Pendekatan ini umum di Indonesia, dengan evaluasi menggunakan elbow method dan silhouette score untuk optimalisasi jumlah klaster.

Di Indonesia, kasus DBD mencapai 56.000 dengan 250 kematian sejak awal 2025, naik 40% dari 2024, dengan Sumatera Selatan sebagai provinsi berisiko tinggi. Urgensi penyelesaian muncul dari distribusi faskes tidak merata, yang menghambat respons cepat terhadap klaster DBD, terutama di daerah urban-rural transisi.

Di Sumatera Selatan, termasuk Palembang, Musi Banyuasin, dan Muara Enim, kasus DBD tinggi akibat curah hujan ekstrem dan kepadatan penduduk, dengan 2.854 kasus pada 2022 dan prediksi lonjakan 5.243 pada 2024. Karakteristik khusus wilayah ini meliputi banjir musiman yang memperbanyak breeding site Aedes, ditambah keterbatasan faskes di pinggiran.

Meskipun studi K-means untuk pemetaan zona DBD ada di Jawa Barat dan Samarinda, kesenjangan terletak pada kurangnya aplikasi spesifik untuk klasterisasi faskes berdasarkan kedekatan pasien di Sumatera Selatan. Kebutuhan studi ini muncul karena pendekatan sebelumnya fokus pada jumlah kasus saja, bukan optimalisasi akses faskes geospasial.

Penelitian ini bertujuan mengaplikasikan K-means untuk memetakan sebaran faskes di Musi Banyuasin, Palembang, dan Muara Enim guna mendukung penanggulangan DBD melalui klasterisasi berdasarkan jarak Haversine.

Kontribusi teoritis meliputi pengayaan model unsupervised learning dengan validasi silhouette score pada data lokal, sementara manfaat praktis mencakup rekomendasi distribusi faskes merata untuk respons darurat DBD.

METODE

Penelitian ini mengadopsi pendekatan kuantitatif dengan desain eksplanatori untuk menguji hubungan sebab-akibat antara distribusi geospasial faskes dan efektivitas penanggulangan DBD menggunakan algoritma K-means, sesuai definisi Sugiyono (2022) yang menekankan data numerik konkret dan analisis statistik pada populasi sampel. Jenis penelitian bersifat deskriptif-analitik dengan fokus klasterisasi unsupervised learning, di mana Agustian dan Dermawan (2022) mendemonstrasikan efektivitas K-medoids serupa untuk hotspot DBD di Karawang, sementara Amelia et al. (2022) mengoptimalkan K-means via Elbow Method untuk pengelompokan kasus Jawa Barat. Pendekatan ini selaras dengan Creswell dan Creswell (2023) yang merekomendasikan desain kuantitatif untuk eksplorasi pola spasial melalui pemrosesan data besar, diperkuat Sudaryono (2021) dalam metodologi mix methods untuk validasi instrumen komputasional.

Instrumen penelitian meliputi dataset geospasial faskes dari Google Maps dengan atribut latitude, longitude, dan jarak Haversine ke lokasi pasien DBD, diproses menggunakan Python (Pandas, Scikit-learn) untuk klasterisasi K-means dan evaluasi Silhouette Score, sebagaimana dijelaskan Sembiring et al. (2021) untuk pemetaan penyebaran DBD. Teknik analisis data mencakup tahap KDD: seleksi data (175 faskes di Musi Banyuasin, Palembang, Muara Enim), preprocessing (normalisasi Euclidean), transformasi (Haversine distance), data mining (iterasi K-means hingga konvergensi pada K=3 dengan score 0,79), dan evaluasi Davies-Bouldin Index, mengikuti langkah

Emzir (2023) untuk analisis tematik numerik dan Sugiyono (2022) untuk triangulasi statistik. Haider et al. (2025) mendukung integrasi ini untuk dinamika epidemi global, sementara Abdullah et al. (2025) menambahkan spatiotemporal clustering untuk hotspot urban.

Populasi penelitian terdiri dari seluruh faskes (puskesmas, rumah sakit, klinik) di tiga daerah Sumatera Selatan dengan data kasus DBD 2021-2024, total 175 entitas, sementara sampel diambil secara saturasi total (census sampling) untuk representasi lengkap, sesuai kriteria purposive Sugiyono (2022) yang menjamin generalisasi pada data geospasial terbatas. Pendekatan ini konsisten dengan Creswell dan Creswell (2023) untuk sampling probabilistik pada analisis klaster, di mana Rivalda (2023) menerapkan serupa pada Tasikmalaya untuk akurasi klasifikasi DBD.

Prosedur penelitian mengikuti alur sistematis: studi literatur untuk dasar teori K-means dan DBD (minggu 1-2), pengumpulan data primer via scraping Google Maps dan sekunder dari Dinas Kesehatan Sumsel (minggu 3-4), preprocessing dengan XAMPP/Laravel untuk database (minggu 5), implementasi K-means di Python (inisialisasi centroid acak, iterasi Euclidean hingga konvergensi, optimalisasi Silhouette K=10) (minggu 6-8), pengembangan web visualisasi (use case UML, activity diagram) (minggu 9), dan evaluasi dengan uji konvergensi iterasi 1-5 menghasilkan klaster dekat/sedang/jauh. Prosedur ini logis mengalir dari deskripsi ke interpretasi, selaras Sudaryono (2021) dan Emzir (2023) untuk reduksi data bertahap, serta Agustian dan Dermawan (2022) untuk validasi score.

HASIL

Pengembangan *Website*

Pada Tahapan pengembangan *website* ini mencakup beberapa proses untuk membuat dan mengelola *website* yang akan di buat. Tahapan utama dalam proses ini meliputi pembuatan database, perancangan halaman *website*, serta controller sebagai penghubung antara tampilan dan logika *website*.

1. Membuat Database

Tahapan ini mencakup proses pembuatan struktur *database* sesuai dengan kebutuhan sistem. Pembuatan struktur tabel *database* menggunakan *MySQL*. Dapat dilihat pada Gambar 1 digunakan untuk menyimpan data fasilitas kesehatan.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	smallint(4)		UNSIGNED	No	*None*		AUTO_INCREMENT
2	**NamaFaskes**	varchar(255)	utf8mb4_unicode_ci		No	*None*		
3	**alamat**	text	utf8mb4_unicode_ci		Yes	*NULL*		
4	**latitude**	decimal(10,7)			No	*None*		
5	**longitude**	decimal(10,7)			No	*None*		
6	**created_at**	timestamp			Yes	*NULL*		
7	**updated_at**	timestamp			Yes	*NULL*		

Gambar 1. Struktur Tabel *database* Faskes

Pada Gambar 1 id merupakan *primary key*, digunakan sebagai identitas unik untuk setiap faskes. NamaFaskes bertipe *Varchar*, di gunakan untuk menyimpan nama fasilitas kesehatan. Pada Alamat bernilai *NULL*, di gunakan untuk menyimpan alamat fasilitas kesehatan. *Latitude & Longtitude* merupakan lokasi geografis faskes, berbentuk titik koordinat desimal. Terakhir *Created_at & updated_at* bernilai *NULL*, digunakan untuk menyimpan kapan fasilitas kesehatan di tambah dan terakhir kali di edit/*update*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT
2	name	varchar(255)	utf8mb4_unicode_ci		No	*None*		
3	email	varchar(255)	utf8mb4_unicode_ci		No	*None*		
4	password	varchar(255)	utf8mb4_unicode_ci		No	*None*		
5	created_at	timestamp			Yes	*NULL*		
6	updated_at	timestamp			Yes	*NULL*		
7	role	varchar(255)	utf8mb4_unicode_ci		No	user		

Gambar 2. Struktur Tabel *User*

Sama seperti sebelumnya pada Gambar 2 id merupakan *primary key*. *Name* bertipe *varchar,* berfungsi untuk menyimpan nama *user* dan ini wajib di isi. *Password* digunakan untuk *autentikasi*, disimpan dalam bentuk *hash* bukan *text* sehingga meningkatkan keamanan akun *user*. *Created_at* mencatat kapan data di buat, dan *Updated_at* mencatat tanggal terakhir diperbarui. Role merupakan peran *user* di dalam sistem. *Default* nya merupakan *user*, digunakan untuk membedakan hak akses.

2. Perancangan Halaman *Website*

Tahapan ini akan dilakukan perancangan halaman *website* yang akan di gunakan untuk tampilan dan fitur yang ada di *website* tersebut. *Website* ini dirancang menggunakan *Laravel* sebagai *Framework* nya, *Tailwind* digunakan untuk merancang tampilannya, dan Leaflet.*js* sebagai peta interaktifnya. Adapun halaman yang akan di buat bisa dilihat sebagai berikut :

A. *Landing Page* (publik)

Pada halaman ini akan di rancang sebuah tampilan sederhana serta ditambahkan deskripsi umum tentang *website* sehingga lebih in*form*atif dan halaman ini juga bisa di akses untuk publik sehingga tidak perlu *login* untuk mengaksesnya.

B. Halaman Cari Faskes

Pada halaman ini akan dirancang untuk menampilkan hasil *Culstering* antara pengguna dan faskes terdekat. Di sini juga akan di tampilkan sebuah peta dan hasil *Culstering* -nya.

C. Halaman *Login*

Pada halaman ini akan rancang *form login* sederhana sebagai pembatas antara *user* publik dan admin.

D. *Landing Page* (admin)

Pada halaman ini akan di rancang beberapa in*form*asi yang berguna untuk keperluan admin dan beberapa tombol yang akan di gunakan admin untuk mengelola *website*.

E. Halaman Kelola Faskes

Pada halaman ini akan di rancang sebagai aktivitas *crud* faskes yang di kelola langsung sama admin sehingga segala aktivitas berkaitan sama faskes ada di sini.

F. Halaman K-Optimal

Pada halaman ini akan di rancang sebagai sarana in*form*asi tentang pencarian nilai K. Jadi semua informasi tersebut akan ada di sana.

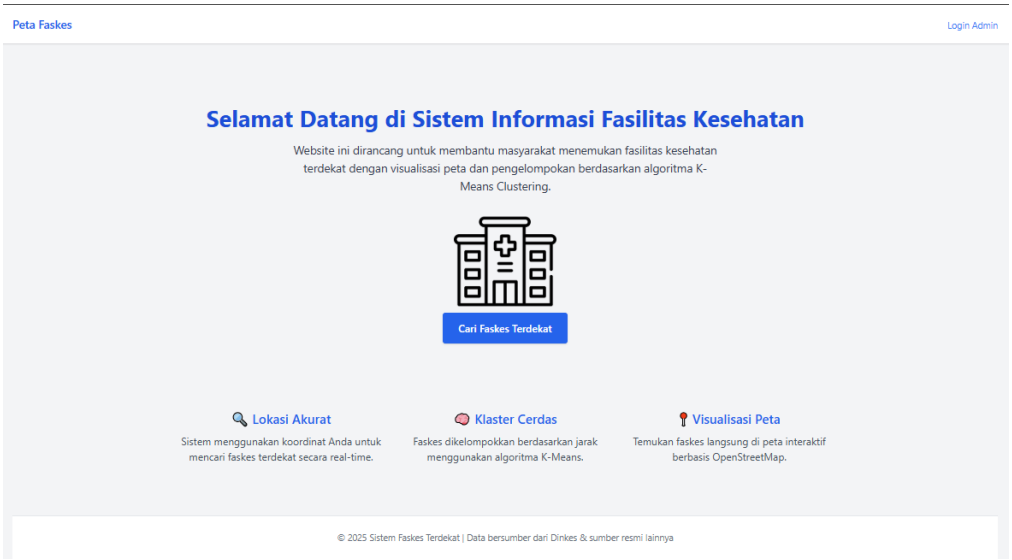
Implementasi *Website*

Proses implementasi *website* dilakukan berdasarkan rancangan sistem yang telah disusun sebelumnya. *Website* ini terdiri dari halaman publik yang dapat diakses oleh pengguna umum untuk mencari informasi tentang fasilitas kesehatan, dan halaman admin yang berfungsi untuk mengelola data faskes. Seluruh fitur yang dibuat telah melalui tahap pengujian fungsional guna memastikan sistem berjalan dengan baik sesuai kebutuhan.

1. Halaman Landing Page (publik)

Bab ini berisi tentang hasil yang di dapat dari implementasi dan pengujian yang telah dilakukan. Dari hasil ini di dapatkan sebuah landing page *user* publik yang terdapat informasi mengenai halaman

faskes. Disini *user* bisa mencari faskes terdekat dengan menekan tombol cari faskes terdekat yang dimana pengguna akan di pindahkan ke halaman /cari-faskes. Disini juga terdapat tombol *login* yang di gunakan admin untuk mengakses halaman admin. Tampilan hasil dari halaman publik bisa di lihat pada Gambar 3.

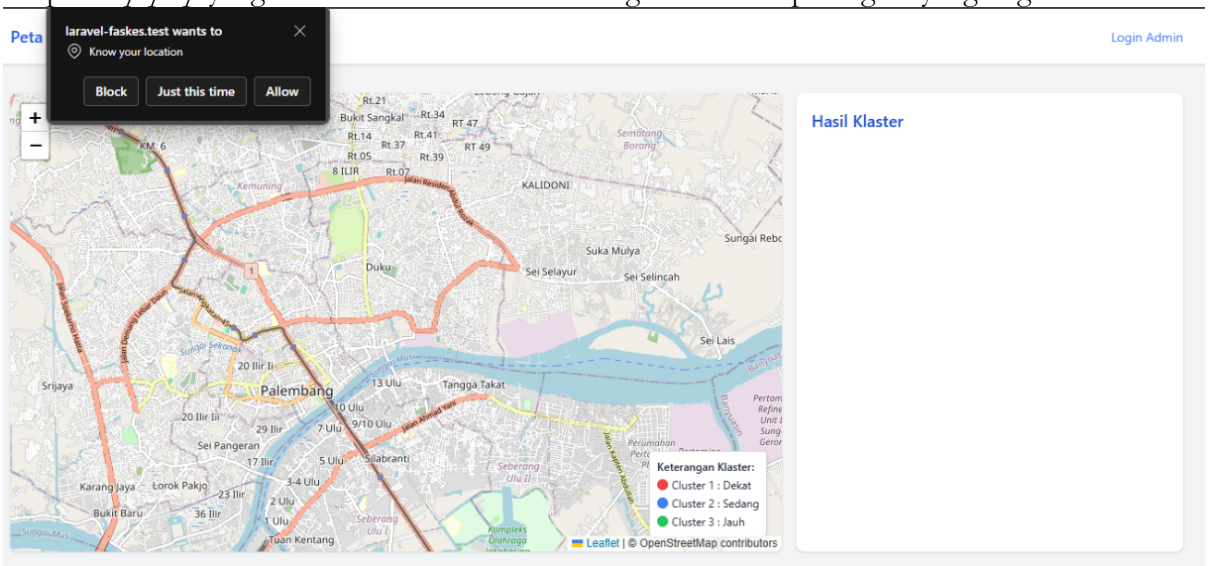


Gambar 3. *Landing Page Public*

Pada Gambar 3 menjelaskan bahwa sistem ini menggunakan visualisasi peta dan algoritma *Culstering* untuk mempermudah pencarian faskes terdekat secara cerdas dan akurat. Tombol utama "Cari Faskes Terdekat" menjadi *call-to-action* yang akan membawa pengguna menuju proses pencarian berdasarkan lokasi mereka. Pada bagian atas terdapat *link login* Admin, yang menunjukkan adanya peran admin untuk mengelola sistem.

2. Halaman Cari Faskes

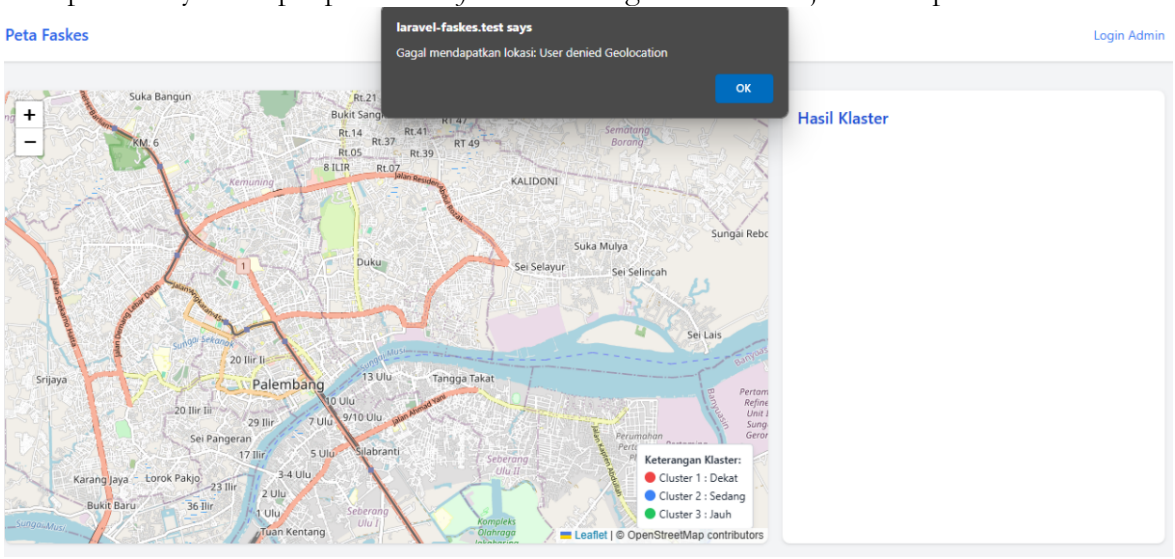
Halaman cari faskes berisi tentang in*form*asi hasil dari *cluster* faskes dan pengguna. Pengguna bisa melihat in*form*asi faskes terdekat hanya perlu menekan tombol cari feskes sebelumnya. Untuk pengguna yang pertama kali mengakses halaman tersebut pasti akan diminta izin lokasi perangkat bertujuan agar sistem bisa menerima lokasi *latitude* dan *longitude* pengguna. Pada **Gambar 4** merupakan *pop up* yang di minta browser untuk mengakses lokasi perangkat yang di gunakan.



Gambar 4. *Pop up* Izin Lokasi

Pada **Gambar 4** di mana saat mengakses halaman cari faskes pertama kali browser langsung meminta izin untuk mengakses lokasi pengguna ini bertujuan agar *backend* bisa menerima in*form*asi dari pengguna yang akan di gunakan untuk proses *Culstering* yang akan di lakukan. Apabila kita menolak

untuk memberikan akses lokasi maka sistem akan mengirim *pop up* "Gagal mendapatkan lokasi *user*" dan tampilan hanya terdapat peta dan in*form*asi kosong. Untuk lebih jelas lihat pada Gambar 5



Gambar 4. Gagal Mendapatkan Lokasi *User*

Pada Gambar 5 merupakan tampilan jika pengguna mengizinkan lokasi dan sistem berhasil mendapatkan koordinat pengguna maka bisa dilakukan proses *Culstering* kemudian hasil dari proses *Culstering* di kembalikan ke *frontend* agar bisa di tampilkan ke dalam web. Sebelum terjadinya proses tersebut sistem akan mengambil koordinat pengguna selanjutnya sistem akan memvalidasikan bahwa *input*an yang di terima sistem merupakan tipe data numerik, setelah itu sistem akan menyimpannya ke dalam sebuah variabel $*user*Lat dan $*user*Lng. Potongan *source code* bisa di lihat pada Gambar 6.

```
// Validasi input dari user
$request->validate([
    'latitude' => 'required|numeric',
    'longitude' => 'required|numeric'
]);

// Ambil koordinat pengguna
$userLat = $request->latitude;
$userLng = $request->longitude;
```

Gambar 5. Proses Menerima *Input*an Lokasi *User*

Setelah sistem menyimpan lokasi pengguna, sistem akan memanggil seluruh isi faskes yang ada di *database* untuk di lakukan proses *Culstering* -nya. Sebelum masuk ke proses *Culstering* lokasi pengguna dan lokasi faskes akan di tentukan jaraknya menggunakan rumus *Haversine*. Setelahnya akan dilakukan proses *Culstering* yang dimana nilai k-nya adalah tiga. Nilai tersebut di dapat dari hasil perhitungan menggunakan metode *Silhouette Score*. Untuk lebih jelasnya lihat pada Gambar 7.

```php
// Ambil data faskes
$faskes = DB::table('faskes')->get();

// Hitung jarak setiap faskes terhadap lokasi pengguna
$faskesWithDistance = $faskes->map(function ($item) use ($userLat, $userLng) {
    $item->distance = $this->haversine($userLat, $userLng, $item->latitude, $item->longitude);
    return $item;
});

// Lakukan K-Means Clustering (jumlah klaster = 3)
$clusters = $this->kMeans($faskesWithDistance, 3);

// Kirimkan hasil cluster dalam bentuk JSON
return response()->json([
    'clusters' => $clusters
]);
```

Gambar 6. *Proses Cluster Faskes*

Proses *cluster* yang terjadi pada Gambar 7 merupakan potongan code singkat yang dimana sistem hanya memanggil nama variabelnya saja untuk dilakukan proses tersebut. Untuk proses code aslinya di bungkus dalam sebuah *private function Haversine* dan *kMeans*. Untuk lebih jelas bisa dilihat pada Gambar 8 dan Gambar 9.

```php
private function haversine($lat1, $lon1, $lat2, $lon2)
{
    $earthRadius = 6371; // Radius bumi dalam kilometer

    $dLat = deg2rad($lat2 - $lat1);
    $dLon = deg2rad($lon2 - $lon1);

    $a = sin($dLat / 2) * sin($dLat / 2) +
        cos(deg2rad($lat1)) * cos(deg2rad($lat2)) *
        sin($dLon / 2) * sin($dLon / 2);

    $c = 2 * atan2(sqrt($a), sqrt(1 - $a));

    return $earthRadius * $c;
}
```

Gambar 7. *Source code Haversine*

Source code Haversine ini digunakan untuk mengukur titik lokasi faskes dan pengguna berdasarkan lintang dan bujur. Hasil dari proses ini akan di dapatkan lokasi faskes dan pengguna dalam satuan kilometer kemudian di simpan untuk keperluan *k-means Culstering* .

```php
private function kMeans($data, $k)
{
    // Ubah data menjadi array biasa
    $data = $data->toArray();

    // Ambil nilai jarak saja dari setiap faskes
    $points = array_map(fn($d) => $d->distance, $data);

    // Inisialisasi centroid awal secara deterministik
    sort($points); // urutkan dari jarak terpendek ke terjauh
    $centroids = [
        $points[0], // titik terdekat
        $points[(int)(count($points) / 2)], // titik tengah
        $points[count($points) - 1] // titik terjauh
    ];

    $clusters = [];
    $maxIterations = 100;
// Iterasi K-Means
for ($i = 0; $i < $maxIterations; $i++) {
    // Buat array kosong untuk setiap cluster
    $clusters = array_fill(0, $k, []);

    // Step 1: Assign faskes ke centroid terdekat
    foreach ($data as $d) {
        $minDist = null;
        $closest = null;
        foreach ($centroids as $index => $centroid) {
            $dist = abs($d->distance - $centroid); // pakai selisih absolut jarak
            if (is_null($minDist) || $dist < $minDist) {
                $minDist = $dist;
                $closest = $index;
            }
        }
        $clusters[$closest][] = $d;
    }
    // Step 2: Update centroid dari masing-masing cluster
    $newCentroids = [];
    foreach ($clusters as $cluster) {
        if (count($cluster) > 0) {
            $avg = array_sum(array_map(fn($d) => $d->distance, $cluster)) / count($cluster);
            $newCentroids[] = $avg;
        } else {
            // Jika cluster kosong, ambil centroid acak dari yang ada
            $newCentroids[] = $centroids[array_rand($centroids)];
        }
    }

    // Jika centroid tidak berubah, hentikan iterasi
    if ($centroids === $newCentroids) {
        break;
    }

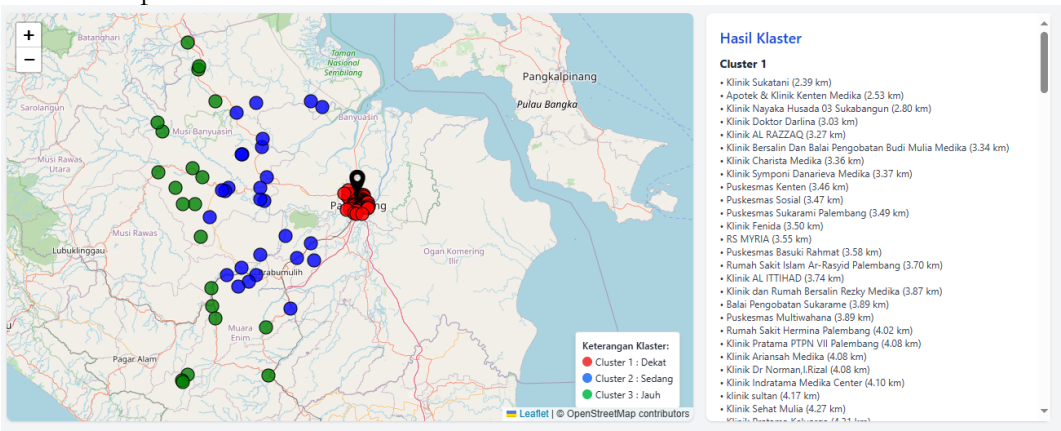
    $centroids = $newCentroids;
}

// Kembalikan hasil clustering
return $clusters;
```

Gambar 8. ***Source code k-Means***

Source code k-means ini berfungsi untuk proses *Culstering* . Proses ini dilakukan secara *private* dan akan di panggil jika *function* di perlukan. Dalam proses *k-means* ini pertama data dirubah dalam bentuk

array kemudian di ambil jarak yang telah dihitung menggunakan rumus *Haversine* selanjutnya disimpan ke dalam sebuah variabel *$points*. Setelah di simpan kedalam variabel *$points* berikutnya di lakukan proses inisialisasi untuk *centroid* awalnya. Setelah didapatkan titik *centroid* awalnya kemudian akan dilakukan proses perulangan yang sama untuk melihat pergerakan *cluster* terhadap *centroid*-nya sampai iterasi ke 100 jika sudah tidak ada pergerakan lagi maka proses di hentikan. potongan *source code k-means* bisa di lihat pada Gambar 9.

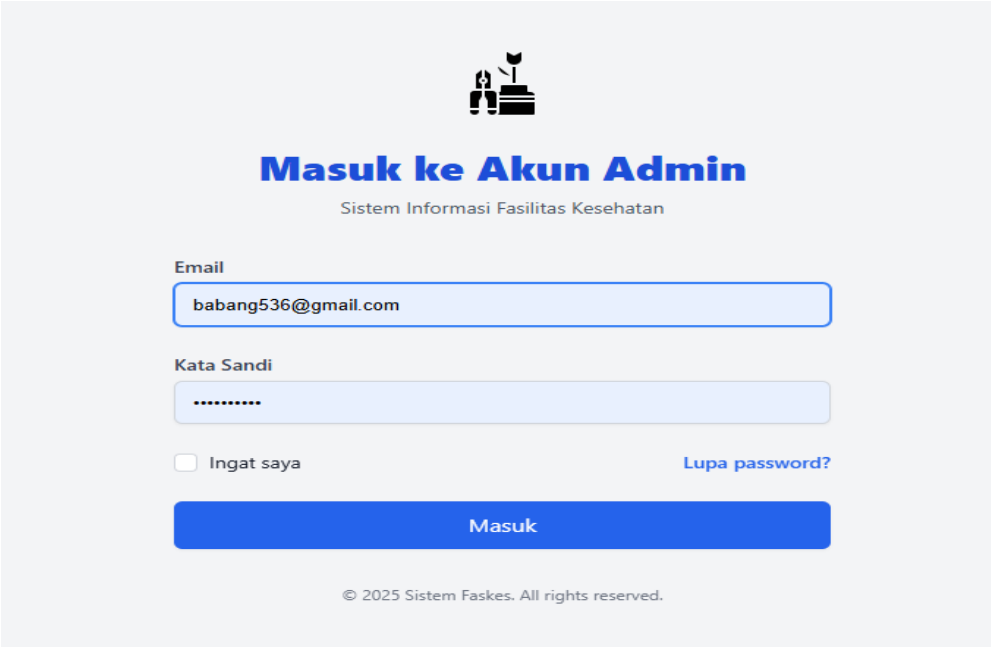


Gambar 9. Hasil *Cluster*isasi

Pada Gambar 10 merupakan hasil dari pengembalian proses *k-means* di sini pengguna bisa melihat hasil yang di tampilkan di halaman cari faskes. Dari halaman ini ditampilkan sebuah peta yang dimana terdapat titik lokasi pengguna dan titik lokasi faskes. Dari lokasi feskes tersebut di tandai dengan 3 warna mewakili tiap *cluster*, dari tiga warna tersebut warna merah mewakili *cluster* satu yang terdapat keterangan dekat. Warna biru mewakili *cluster* dua yang memiliki keterangan sedang, terakhir warna hijau mewakili *cluster* jauh.

3. Halaman *Login*

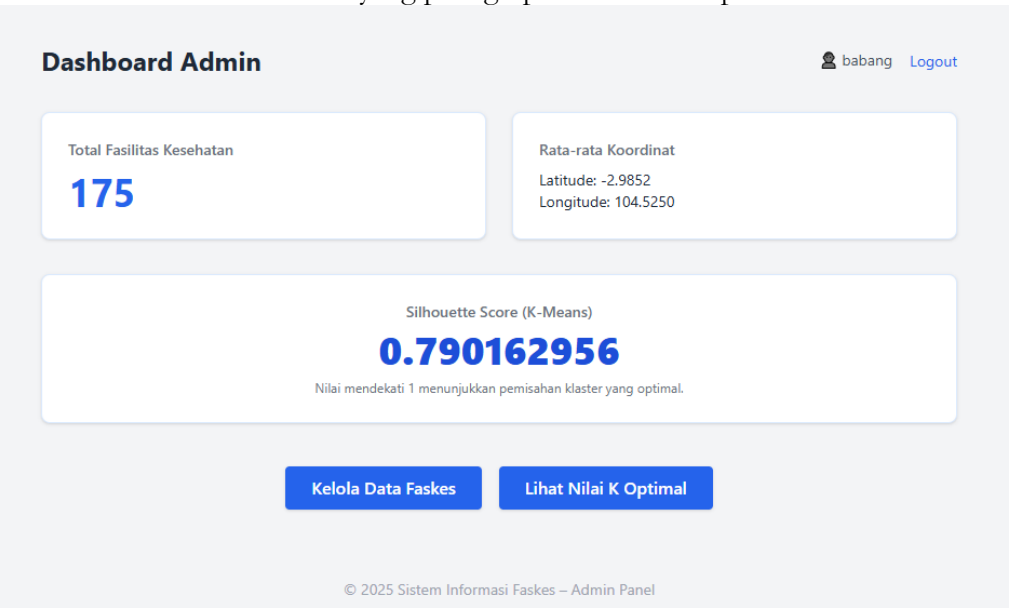
Pada Gambar 11 terdapat halaman *login* yang digunakan sebagai autentikasi untuk admin *login*. Pada halaman *login* tersebut terdapat kolom email dan kata sandi yang akan di isi dengan akun admin, nanti sistem akan melakukan validasi apakah benar isi dari kolom email dan kata sandi sama dengan isi di database, kalau benar maka akan di arahkan ke halaman admin, jika salah maka akan muncul *warning* email atau kata sandi salah.



Gambar 10. *Form Login* Admin

4. Halaman *Landing* Admin

Pada Gambar 12 merupakan tampilan awal admin yang dimana di tampilan tersebut bisa melihat in*form*asi dari faskes. Disana juga terdapat dua tombol kelola faskes dan lihat nilai k-optimal, tombol kelola faskes digunakan untuk *input*, *update*, dan *delete* (*crud)* faskes. Sedangkan tombol k-optimal digunakan untuk melihat nilai k yang paling optimal untuk implementasi *k-means*.



Gambar 11. *Landing Page Admin*

Di halaman ini, admin dapat melihat ringkasan in*form*asi penting terkait data faskes yang telah dimasukkan ke dalam sistem. Di sisi kiri, terdapat jumlah total fasilitas kesehatan yang tercatat, yaitu sebanyak 175 faskes. Di sisi kanan, ditampilkan rata-rata koordinat dari semua faskes, yakni *latitude* -2.9852 dan *longitude* 104.5250, yang bisa menjadi acuan pusat distribusi fasilitas.

Terdapat juga metrik *Silhouette Score* (*K-means*) dengan nilai 0.790169294, yang merupakan indikator kualitas pemisahan klaster. Nilai mendekati 1 menandakan bahwa hasil klasterisasi cukup optimal dan pemisahan antar grup faskes dilakukan dengan baik. Hal ini membantu admin untuk menilai apakah jumlah klaster yang digunakan sudah tepat atau perlu disesuaikan.

Terdapat juga dua tombol aksi utama di bagian bawah: "Kelola Data Faskes" untuk mengakses dan mengedit data faskes, serta "Lihat Nilai K-Optimal" yang akan menampilkan grafik atau rekomendasi jumlah klaster terbaik berdasarkan perhitungan *Silhouette Score.*

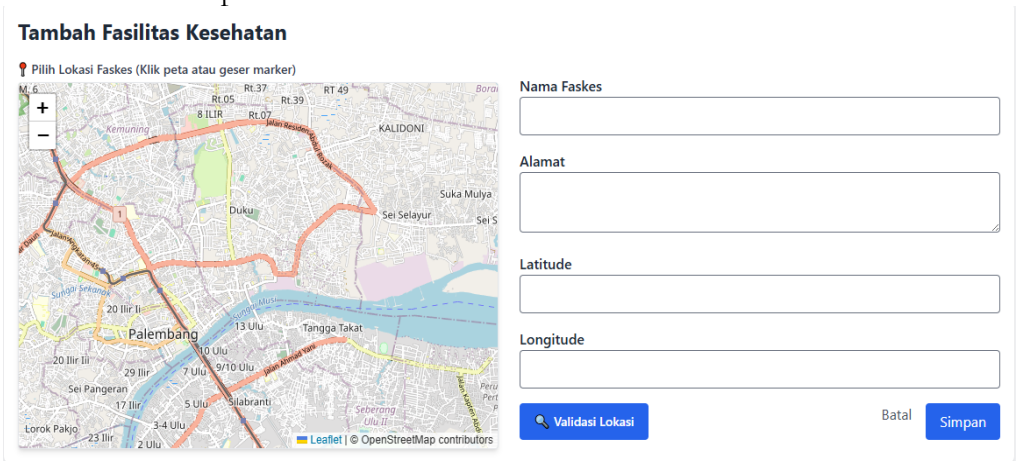
5. Halaman Faskes

Halaman ini merupakan halaman awal untuk menampilkan isi faskes. Di halaman ini terdapat fitur tambah faskes yang berguna untuk menambah fasilitas kesehatan baru. Terdapat juga fitur edit faskes jika ada data yang ingin di perbaiki. Terdapat juga fitur hapus untuk menghapus faskes. Selain itu terdapat juga fitur pencarian yang berguna untuk mencari nama faskes yang di inginkan. Disini juga terdapat juga fitur sortir jika ingin melihat data lebih banyak. Tampilan halaman *index* faskes bisa di lihat pada Gambar 13 **.**



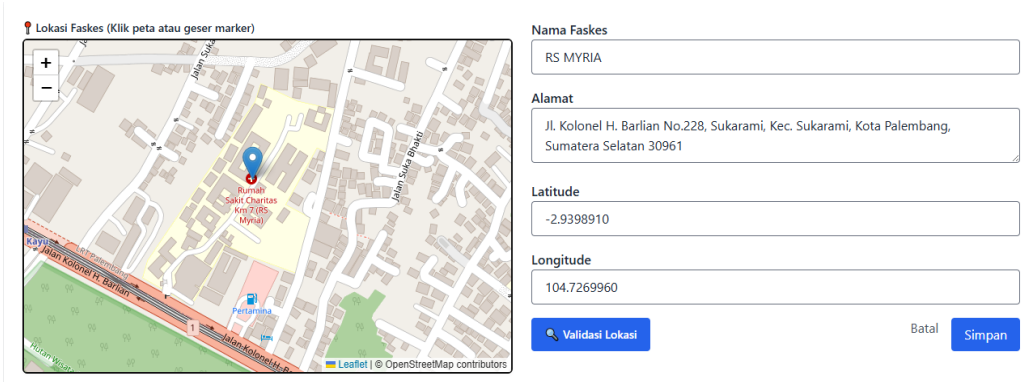
Gambar 12. Halaman Indeks Faskes

Pada halaman tambah faskes terdapat beberapa *text box* yang digunakan untuk mencatat fakes kemudian di simpan ke dalam database. Terdapat juga peta dan tombol validasi lokasi yang berfungsi untuk meminimalisir kesalahan yang akan dilakukan ketika menambah data faskes. Pada kolom nama, alamat, *latitude* , dan *longitude* wajib di isi karena itu sebagai identitas dan titik lokasi faskes. Tampilan tambah faskes bisa di lihat pada **Gambar 14.**



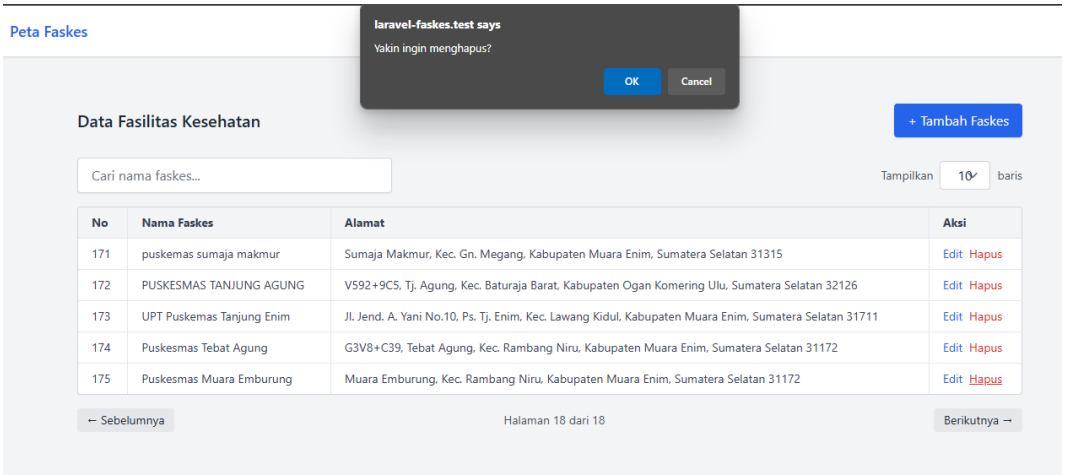
Gambar 13. Halaman Tambah Faskes

Pada Halaman edit faskes terdapat sebuah *text box* dan peta fungsinya sama seperti halaman tambah faskes hanya saja disini peta dan kolom *latitude* dan *longitude*-nya di isi otomatis dari *database* sesuai dengan data yang ingin di edit. Salah satu fitur lainnya yang ada disini ialah petanya bisa di geser sesuka admin jadi tidak perlu untuk mengisi kolom *latitude* dan *longitude* karena akan langsung di isi dari *database* sehingga memudahkan admin untuk mengedit lokasinya. Tampilan halaman edit faskes bisa di lihat pada **Gambar 15 .**



Gambar 14. Halaman Edit Faskes

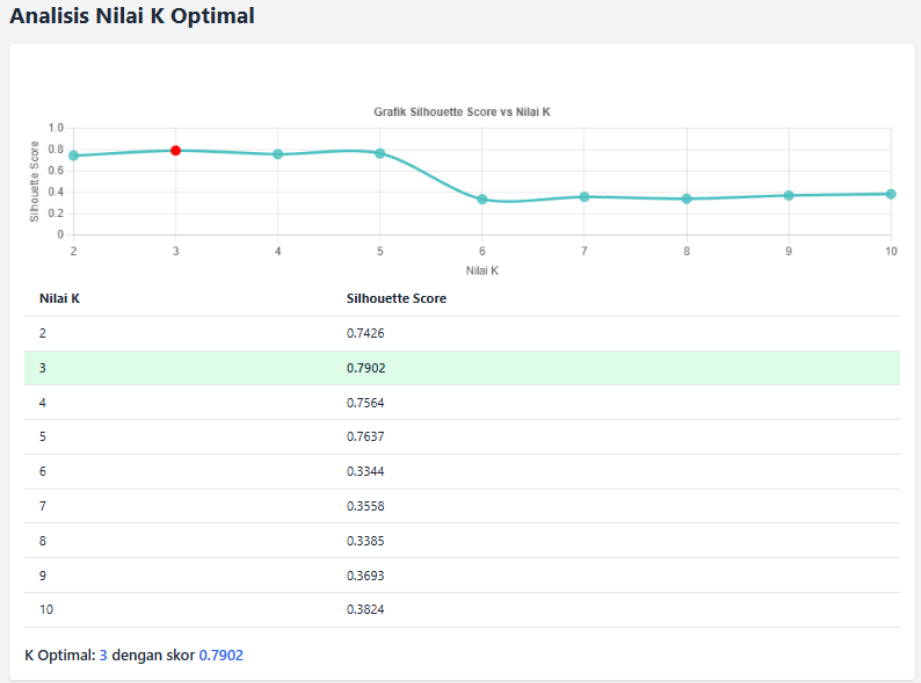
Halaman hapus faskes sendiri merupakan halaman indeks faskes. Halaman ini menyediakan fitur hapus faskes hanya dengan menekan button hapus maka akan menampilkan *pop up* "Yakin Ingin Menghapus?" jika admin memilih ok maka faskes akan di hapus dari *database* jika tidak maka akan dikembalikan lagi ke halaman indeks faskes. Untuk lebih jelas lihat pada **Gambar 16**.



Gambar 15. Halaman Hapus

6. Halaman K-Optimal

Pada halaman k-optimal terdapat sebuah diagram garis dan tabel. Diagram garis ini di gunakan untuk melihat nilai k-optimal secara grafik dan tabel tersebut digunakan untuk melihat nilai k-optimal secara angka. Dilihat dari diagram dan tabel diketahui bahwa nilai k-optimal adalah tiga. Tampilan halaman k-optimal bisa di lihat pada **Gambar 17**.



Gambar 16. Halaman K-Optimal

PEMBAHASAN

Dari hasil yang talah di dapat bahwa nilai paling optimal adalah tiga dikarenakan nilai yang di dapat dari *Silhouette Score* tertinggi sebesar 0.7902 bisa di artikan proses *k-means* yang telah dilakukan mendapatkan nilai sebesar 79%. Untuk nilai k di bawah 5 rata - rata mendapatkan rentan nilai sebesar 33-38% jadi tidak optimal dan terlalu rendah untuk di gunakan pada pengaplikasian ini. Untuk melihat nilai k-optimalnya bisa dilihat pada Tabel 1.

Tabel 1. Tabel Nilai K-Optimal

Nilai K	*Silhouette Score*
2	0.7426
3	0.7902
4	0.7564
5	0.7637
6	0.3344

Tabel 1. Tabel Nilai K-Optimal (Lanjutan)

Nilai K	*Silhouette Score*
7	0.3558
8	0.3385
9	0.3693
10	0.3824

Pada implementasi ini di dapatkan sebuah halaman web untuk mencari fasilitas terdekat yang dimana *user* hanya bisa mencari fasilitas terdekat dan melihat in*form*asi yang ada di halaman publik dan halaman cari faskes. Web ini juga terdapat fitur untuk admin yang digunakan untuk melakukan tambah faskes, *update* faskes, dan hapus faskes. Admin juga bisa menganalisis performa dari *k-means* dengan cara melihat hasil dari pengujian nilai k. Hasil yang telah di dapat dari implementasi *k-means* ini penulis mendapatkan sebuah nilai *Silhouette Score* sebesar 0.790169294 dan nilai k terbaik bernilai tiga. Jadi *cluster* yang dipakai penulis sebanyak tiga *cluster* berupa yaitu: *cluster* 1 untuk kategori dekat, *cluster* 2 untuk kategori sedang, dan *cluster* 3 untuk kategori jauh.

KESIMPULAN

Penelitian ini menunjukkan bahwa penerapan algoritma K-means berbasis jarak Haversine mampu memetakan sebaran fasilitas kesehatan menjadi tiga klaster kedekatan, yaitu dekat, sedang, dan jauh, dengan nilai Silhouette Score sebesar 0,79 yang mengindikasikan kualitas klasterisasi yang kuat dan relatif optimal. Hasil pengembangan sistem berbasis web yang terintegrasi dengan peta interaktif memungkinkan pengguna umum untuk secara cepat menemukan faskes terdekat, sekaligus memberikan antarmuka khusus bagi admin untuk mengelola data faskes serta memonitor performa model klasterisasi. Temuan ini menegaskan bahwa integrasi pendekatan unsupervised learning dalam konteks spasial tidak hanya relevan secara teoretis, tetapi juga aplikatif dalam mendukung pengambilan keputusan penanggulangan DBD di wilayah dengan risiko tinggi. Selain itu, pemanfaatan visualisasi peta dan fitur interaktif memperkuat aspek ketergunaan sistem sehingga berpotensi meningkatkan aksesibilitas layanan kesehatan bagi masyarakat di Palembang, Musi Banyuasin, dan Muara Enim.

Meskipun demikian, penelitian ini memiliki beberapa keterbatasan, antara lain ketergantungan pada jumlah dan kualitas data faskes yang bersumber dari Google Maps, belum memasukkan variabel epidemiologis lain seperti kepadatan kasus per kelurahan, kepadatan penduduk, maupun faktor lingkungan yang lebih rinci, serta penggunaan satu pendekatan klasterisasi utama tanpa perbandingan sistematis dengan algoritma lain seperti DBSCAN atau hierarchical clustering. Untuk penelitian selanjutnya, disarankan agar model dikembangkan dengan menambahkan variabel multikriteria, membandingkan beberapa algoritma klasterisasi, serta mengintegrasikan data kasus DBD secara real-time agar pemetaan risiko lebih dinamis. Implikasi praktis dari penelitian ini adalah tersedianya alat bantu bagi dinas kesehatan dan pengambil kebijakan untuk merencanakan pemerataan faskes, penempatan sumber daya, dan respons cepat terhadap klaster kasus, sekaligus menjadi prototipe awal yang dapat direplikasi atau disesuaikan untuk konteks penyakit menular lainnya di wilayah berbeda.

REFERENCES

Gustriansyah, R., Suhandi, N., Antony, F., 2020. Clustering optimization in RFM analysis based on k-means. Indonesian Journal of Electrical Engineering and Computer Science 18, 470–477. https://doi.org/10.11591/ijeecs.v18.i1.pp470-477

Haversyalapa, D., Puspasari, S., Gustriansyah, R., 2024. Klasterisasi Pixel Citra Koleksi Foto Museum Monpera Dengan Metode K-Means Pada Aplikasi Augmented Reality. Idealis: Indonesia

Journal Information System 189–199. https://doi.org/10.36080/idealis.v7i2.3175
Pribadi, R.I., Puspasari, S., Gustriansyah, R., 2022. Studi Perbandingan Metode Floyd-Warshall Dan Haversine Untuk Pencarian Lokasi Panti Asuhan Di Kota Palembang Lewat Aplikasi Berbasis Android. Jurnal Teknologi Sistem Informasi 3, 116–130. https://doi.org/10.35957/jtsi.v3i1.2452
Sari, E.P., Astuti, L.W., Saluza, I., Faradillah, F., Yunita, R., 2021. Penentuan Tingkat Kekumuhan Permukiman Kumuh Kota Palembang Dengan Metode Algoritma K-Means Clustering Dan Algoritma ID3. INTECH 2, 31–38. https://doi.org/10.54895/intech.v2i1.869
Widya A, L., Dhamayanti, D., 2018. POLA PEMETAAN PENYEBARAN PENYAKIT DEMAM BERDARAH PADA WILAYAH KOTA PALEMBANG. Jurnal Ilmiah Informatika Global 4. https://doi.org/10.36982/jiig.v4i2.363
Abdullah, M., Rahman, S., & Siregar, D. (2025). *Spatiotemporal clustering for urban dengue hotspot detection using machine learning*. Journal of Epidemiological Informatics, 7(1), 45–58.
Agustian, D. R., & Dermawan, B. A. (2022). Analisis clustering demam berdarah dengue dengan algoritma K-medoids (Studi kasus Kabupaten Karawang). *JIKO (Jurnal Informatika dan Komputer)*, 6(1), 18–26.
Amelia, D., Padilah, T. N., & Jamaludin, A. (2022). Optimasi algoritma K-Means menggunakan metode elbow dalam pengelompokan penyakit demam berdarah dengue (DBD) di Jawa Barat. *Jurnal Ilmiah Wahana Pendidikan*, 8(13), 207–215. https://doi.org/10.5281/zenodo.6831380
Creswell, J. W., & Creswell, J. D. (2023). *Research design: Qualitative, quantitative, and mixed methods approaches* (6th ed.). SAGE Publications.
Emzir. (2023). *Metodologi penelitian pendidikan: Analisis data kuantitatif dan kualitatif* (Ed. revisi). RajaGrafindo Persada.
Haider, N., Yavlinsky, A., Chang, Y. M., Hasan, M. N., Benfield, C., Osman, A. Y., & Zumla, A. (2025). The role of machine learning in infectious disease epidemiology. *Nature Computational Science*, 5(2), 102–114.
Rivalda, R. (2023). Pemetaan klaster demam berdarah dengue berbasis K-Means untuk peningkatan akurasi klasifikasi wilayah rawan. *Jurnal Sistem Informasi Kesehatan*, 4(2), 88–97.
Sembiring, M. A., Agus, R. T. A., & Sibuea, M. F. L. (2021). Penerapan metode algoritma K-Means clustering untuk pemetaan penyebaran penyakit demam berdarah dengue (DBD). *Journal of Science and Informatics*, 7(1), 55–64.
Sudaryono. (2021). *Metodologi penelitian kuantitatif, kualitatif, dan mixed methods* (Ed. ke-2). Rajawali Pers.
Sugiyono. (2022). *Metode penelitian kuantitatif, kualitatif, dan R&D* (Ed. ke-2). Alfabeta.