BAB I
PENDAHULUAN

1.1. Latar Belakang Masalah

Listrik sangat dibutuhkan dalam kehidupan sehari-hari namun tanpa disadari sebagian besar pengguna tidak mengetahui kapan penggunaan listrik tertinggi yang telah digunakannya, tagihan listrik yang keluar setiap bulan juga tidak memberi rincian penggunaan listrik yang dipakai dalam periode tersebut[1].

Seiring perkembangan dan pertumbuhan di bidang teknologi, kebutuhan listrik semakin meningkat. Ada banyak sektor yang mengkonsumsi energi listrik terbesar salah satunya adalah sektor komersial yang sub sektornya adalah gedung atau bangunan[2]. Instansi perguruan tinggi seperti halnya Telkom *University*, pasti memiliki banyak gedung dalam satu wilayah dan pasti penggunaan listriknya sangat besar dan bisa menyebabkan mahalnya pembayaran penggunaan listrik. Pemborosan akan terjadi jika penggunaan listrik setiap gedung tidak dipantau dan akan menyebabkan pembengkakan pada tagihan pembayaran listrik setiap periodenya[3].

Sistem pemantauan penggunaan listrik dibutuhkan untuk memantau penggunaan listrik suatu gedung. Jika penggunaan listrik setiap gedung tidak dipantau maka bisa menyebabkan naiknya biaya penggunaan listrik karena tidak terkendalinya penggunaan listrik yang digunakan disaat yang tidak perlu. Pengelompokan untuk data penggunaan listrik juga dibutuhkan agar dapat lebih mudah mengetahui penggunaan listrik berlebih.

Sistem pemantauan listrik sudah banyak diteliti karena masalah dari penggunaan listrik yang berlebih adalah masalah umum dimasyarakat, antara lain adalah pembangunan *monitoring* data penggunaan listrik berbasis IoT[4].

Pengelompokan data penggunaan listrik juga dibutuhkan agar pemantauan penggunaan listrik lebih mudah dilakukan karena data yang sangat banyak akan dikelompokan berdasarkan pembagian kelompok yang ditentukan. Dipenelitian ini mengusulkan menggunakan algoritma K-Means++ karena algoritma ini adalah

pengembangan dari algoritma K-Means yang sudah dipakai pada penelitian Tugas Akhir sebelumnya[5].

Pada Tugas Akhir ini hasilnya adalah berupa aplikasi berbasis *website* yang akan menampilkan informasi dan pengelompokan penggunaan listrik di Gedung Telkom *University* dengan mengusulkan algoritma K-Means++ *clustering* untuk pengelompokan data penggunaan listrik untuk satu gedung dan mengelompokan penggunaan listrik antar gedung.

1.2. Rumusan Masalah

Dari banyaknya masalah yang sudah disebutkan sebelumnya, maka didapat rumusan masalah pada Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana membantu tim logistic Telkom university untuk memonitoring penggunaan listrik Gedung?
2. Bagaimana mengetahui pengelompokan kenaikan penggunaan listrik di Gedung Telkom university?

1.3. Tujuan dan Manfaat

Dikembangkannya sistem *monitoring* penggunaan listrik di gedung Telkom *University* ini memiliki tujuan dan manfaat sebagai berikut:

1. Membangun sistem *monitoring* berbasis aplikasi *website* yang menampilkan informasi penggunaan listrik di gedung Telkom University agar dapat memudahkan tim logistik Telkom University untuk mengetahui detail penggunaan listrik di gedung Telkom University.
2. Membuat sistem pengelompokan data penggunaan listrik di gedung Telkom University menggunakan algoritma *clustering* K-Means++ dengan indikator yang ditentukan sehingga dapat membantu tim logistik.

1.4. Batasan Masalah

Adapun Batasan masalah pada Tugas Akhir ini antara lain:

1. Data yang digunakan merupakan data *real* dari gedung Deli Telkom University dan data yang dikirim dari *device virtual*.

2. Pengelompokan data menggunakan algoritma K-Means++.
3. Pengembangan aplikasi *website* yang dibangun secara *local.*
4. Menggunakan framework Flask dari Bahasa python.
5. Menggunakan MySQL untuk skema relasi *database*.
6. Menggunakan *library* ScikitLearn untuk menganalisa hasil pengelompokan algoritma K-Means++ dan menghitung Silhouette Score.

1.5. Sistematika Penulisan

Sistematika penulisan Tugas Akhir ini terdiri dari beberapa bab dan masing masing bab terbagi menjadi beberapa sub bab. Setiap bab memberikan gambaran secara keseluruhan mengenai isi dari tugas akhir ini.

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang penelitian, rumusan masalah, tujuan, dan batasan masalah dari tugas akhir yang dibangun, serta hipotesa, metodologi penelitian dan sistematika penulisan yang digunakan dalam tugas akhir ini.

BAB II LANDASAN TEORI

Bab ini berisi tentang penjelasan mengenai teori yang digunakan guna menunjang penyusunan tugas akhir ini, yaitu menjelaskan cara kerja sistem dan masing masing komponen yang digunakan pada tugas akhir ini.

BAB III PERANCANGAN SISTEM

Bab ini berisi tentang penjelasan rinci mengenai perancangan sistem tugas akhir yang dibangun, meliputi perancangan sistem, kebutuhan sistem dan implementasi *K-Means++ Clustering* dalam mengelompokan besaran penggunaan listrik Gedung.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini berisi tentang implementasi tugas akhir dan hasil pengujian tugas akhir dan dianalisis apakah sistem yang dibangun sudah sesuai dengan tujuan yang diharapkan.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan akhir dari perancangan sistem, pengujian, dan analisis yang diperoleh serta saran dan harapan untuk pengembangan sistem kedepannya.

BAB II
TINJAUAN PUSTAKA

2.1.Sistem Pemantauan Penggunaan Listrik

Adanya sistem pemantauan penggunaan listrik agar memudahkan petugas logistik untuk memantau penggunaan listrik tanpa harus mendatangi langsung lokasi gedung yang ingin dilihat penggunaan listriknya. Agar bisa melakukan *monitoring* penggunaan listrik dari jauh tanpa harus mendatangi ke tempatnya maka dibutuhkan adanya sistem untuk memantau penggunaan listrik.

Sistem pemantauan saat ini sudah banyak dilakukan dengan banyak metode dan fitur berbeda dari setiap sistem. Salah satunya adalah penggunaan aplikasi *website* dengan menggunakan *framework* angular.js dan node.js[5].

Pada Tugas Akhir ini berfokus pada metode algoritma clustering K-Means++ untuk mengelompokan data penggunaan listrik dan aplikasi berbasis *website* untuk menampilkan informasi dan pengelompokan penggunaan listrik.

2.2. *Clustering*

Clustering merupakan salah satu teknik yang biasa digunakan dalam bidang data *mining* dan analisis data statistik. Disebut juga sebagai proses *unsupervised learning*. Tujuan dari *clustering* adalah untuk mengelompokkan kumpulan data ke dalam *cluster* berdasarkan hubungan jarak diantara mereka[6].

Dengan menggunakan *clustering* ini memiliki banyak manfaat yang dapat diimplementasikan dalam kehidupan sehari-hari contoh penggunaan *clustering* adalah pada retail *marketing* untuk identifikasi pola pembelian pelanggan dan merekomendasikan barang kepada pelanggan, pada kedokteran seperti karakteristik perilaku pasien, pada pendidikan biologi seperti pengelompokan genetik untuk identifikasi ikatan keluarga[7].

2.3. K-Means++

Algoritma K-Means++ merupakan pengembangan dari algoritma K-Means yang bertujuan untuk memperbaiki kekurangan dan menyempurnakan

algoritma K-Means. Dengan menggunakan metode yang dinamakan *Weighting* yang berfungsi untuk menentukan *probabilitas* dari terpilihnya sebuah *centroid*[8].

$$K = \frac{D(x)^2}{\sum_{x \in X} D(x)^2} \quad (2.1)$$

Rumus *weighting* dimana:
K merupakan Probabilitas Nilai *Centroid.*
$D(x)^2$ adalah *Squared Distance.*
$\sum_{x \in X} D(x)^2$ merupakan Jumlah dari semua nilai *Squared Distance.*

Terbukti algoritma K-Means++ tidak memerlukan waktu yang lama dan akurasi yang dihasilkan lebih baik dari K-Means[9]. Langkah-langkah pengerjaan pada algoritma K-Means++ yaitu:

1. Pilih satu data untuk menjadi *centroid* acak.
2. Hitung nilai D(x), dimana D(x) adalah jarak antar titik x dengan *centroid* terdekat.
3. Hitung centroid selanjutnya menggunakan metode D^2 *weighting* untuk menentukan *probabilitas* dari terpilihnya sebuah *centroid*.
4. Ulangi poin 2 dan 3 hingga mendapatkan *centroid* sebanyak K.
   Jika sudah mendapat *centroid* sebanyak K, lanjutkan pengerjaan menggunakan algoritma K-Means.

Langkah-langkah pengelompokan data menggunakan algoritma K-Means adalah sebagai berikut[10]**:**

1. Letakkan titik K ke dalam ruang yang direpresentasikan oleh data yang sedang dikelompokkan. Titik-titik ini mewakili *centroid* kelompok awal.
2. Tetapkan setiap objek ke grup yang memiliki pusat massa terdekat.
3. Hitung ulang posisi sentroid K menggunakan *euclidean distance*, setelah ditetapkannya semua objek.
4. Ulangi poin 2 dan 3 hingga *centroid* tidak lagi bergerak.

$$D_e = \sqrt{(x_i - s_i)^2 + (y_i - t_i)^2} \quad (2.2)$$

Dimana :

D_e adalah *Euclidean Distance*

i adalah banyaknya objek

(x,y) merupakan koordinat object

(s,t) merupakan koordinat centroid.

2.4. *Silhouette Coefficient*

Silhouette Coefficient adalah metode validasi konsistensi dalam pengelompokan data. Teknik ini memberikan representasi tentang seberapa baik setiap objek dan data yang telah dikelompokan[11]. Metode ini merupakan salah satu metode yang digunakan untuk mengevaluasi hasil *clustering* serta menentukan berapa jumlah *cluster* yang cocok untuk sebuah objek dan data[12]. Nilai dari *Silhouette Coefficient* juga dapat dimaksimalkan pada jumlah *cluster* dengan menggunakan bobot fitur yang spesifik[13]. Tahapan untuk pengerjaan menghitung *silhouette coefficient* ini adalah sebagai berikut:

1. Hitung jarak rata-rata dari data ke-i ke semua data dalam satu *cluster*.
2. Hitung nilai dari rata-rata jarak data ke-i ke semua data pada *cluster* yang berbeda.
3. Selanjutnya baru hitung nilai *Silhouette Coefficient*nya.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i)b(i)\}} \quad (2.3)$$

Dimana:

s(i): *Silhouette Index* data ke-i

a(i): Rata rata jarak data ke-I semua data lainya dalam cluster *index* data ke-i

b(i): Nilai minimum dari rata rata jarak data ke-i terhadap semua data dari cluster yang tidak dalam satu cluster dengan *index* data ke-i.

Nilai yang dihasilkan dari Silhouette Coefficient ini adalah antara -1 sampai dengan 1, dimana jika mendekati 1 maka hasil pengelompokan adalah sangat baik.

Tabel 2. 1 Struktur Silhouette

Nilai	Interpretasi Silhouette Coefisien
0,71 – 1,00	Struktur Kuat
0,51 – 0,70	Struktur Baik
0,26 – 0,50	Struktur Lemah
≤ 0,25	Struktur Buruk

2.5. Bahasa Pemrograman Python

Bahasa pemrograman tingkat tinggi yang banyak digunakan di berbagai bidang. Bahasa Python membaca kode dengan penggunaan spasi putih yang signifikan. Python sudah menggunakan pendekatan orientasi pada objek yang sangat membantu untuk proyek skala kecil maupun besar[14].

2.6. *Framework* Flask

Flask merupakan framework dari Bahasa pemrograman Python, Flask juga sering disebut sebuah *microframework*. Flask memiliki dua dependensi utama. Perutean, *debugging*, dan subsistem *Web Server Gateway Interface (WSGI)* berasal dari Werkzeug, dan untuk template menggunakan Jinja2.

Banyak *library* yang disediakan Flask untuk membantu mengembangkan web. karena file yang dibuat sangat ringan dan tidak bergantung banyak pada eksternal *library* membuat Flask ini sangat sederhana dan mudah di pakai[15].

2.7. Pengujian Beta

Pengujian yang dilakukan secara objektif mengukur ketersediaan sistem, dimana sistem akan digunakan dan berinteraksi secara langsung dengan pengguna. *Usability testing* menggunakan USE (*Usefulness, Satisfaction, Ease*

of Use) kuesioner untuk mengukur semua aspek kegunaan, kepuasan, dan kenyamanan pengguna saat menggunakan aplikasi [16].

2.7.1 Uji Validitas

Uji validitas dilakukan untuk mengukur pertanyaan dari Kuesioner yang suda dibuat sudah valid atau tidak validnya pertanyaan maka dilakukan uji Kuesioner ini. Uji validitas ini menggunakan teknik korelasi *bivariate pearson* dengan menggunakan rumus sebagai berikut:

$$\frac{N\sum XY-(\sum X)(\sum Y)}{\sqrt{N\sum X^2-(\sum X^2))(N\sum Y^2-(\sum Y^2))}} \quad (2.4)$$

Keterangan:

rHitung: Koefisien Korelasi

N: Jumlah Responden

X: Skor Butir Soal per Responden

Y: Skor total soal per Responden

2.7.2 Uji Reliabilitas

Pengujian reliabilitas adalah pengujian yang menggunakan teknik *Alpha cronbach*. *Alpha Cronbach* digunakan untuk instrumen atau pertanyaan yang memiliki banyak jawaban. Perhitungan matematis alpha Cronbach dilakukan menggunakan rumus berikut:

$$r11 = \left[\frac{k}{k-1}\right]\left[1 - \frac{\sum ab^2}{a1^2}\right] \quad (2.5)$$

Keterangan:

r11 = Instrumen Reliabilitas

K = Jumlah butir soal

$\sum ab^2$ = Jumlah Varian

$a1^2$ = Varian Butir Soal

2.8.Penelitian Sebelumnya

Penelitian sebelum ini yang dijadikan sebagai referensi untuk membuat penelitian ini adalah jurnal penelitian oleh Ressy Aryani yang berjudul

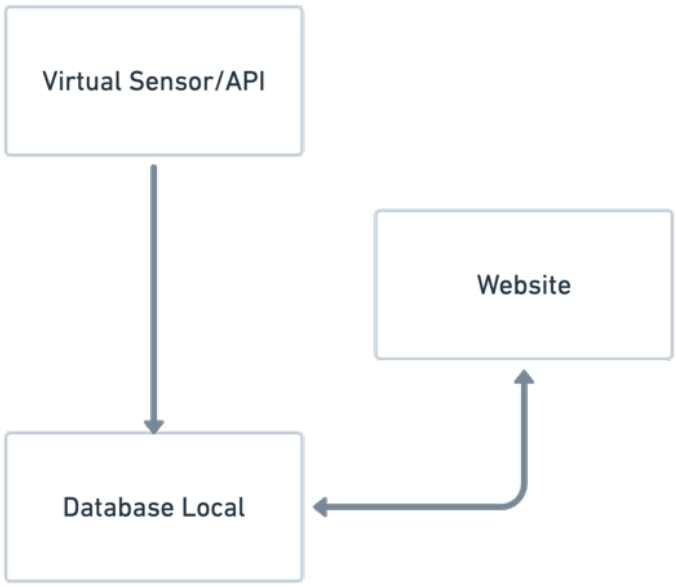
“Pembangunan Purwarupa Sistem Pemantauan dan Pengelompokan Data menggunakan K-Means *Clustering* pada Sistem Pemetaan Kwh Meter Berbasis Iot”. Pada penelitian yang dilakukan oleh Ressy Aryani ini menggunakan hanya satu algoritma K-Means. Pada penelitian ini menggunakan pengembangan algoritma dari K-Means yaitu K-Means++, ditambah dua algoritma lain yaitu *Agglomerrative Hierarchical Clustering* dan DBSCAN.

BAB III
PERANCANGAN SISTEM

3.1. Desain Sistem

Sistem yang dibuat merupakan sebuah aplikasi berbasis *website*, aplikasi akan dibuat untuk menyelesaikan masalah yang sudah dirumuskan dengan cara membangun sistem untuk mengetahui kapan dan gedung mana yang penggunaan listriknya berlebih pada gedung Telkom *University*. Sistem akan menggunakan pengelompokan dengan menggunakan metode K-Means++ *clustering*.

Gambaran umum dari sistem yang dibuat menggunakan 3 bagian yaitu sensor *virtual* atau API yang bertugas untuk mengirim data penggunaan listrik ke *database* dan *web app* yang fungsinya adalah untuk menampilkan informasi dari data-data yang sudah diproses menggunakan K-Means++ *clustering*.



Gambar 3. 1 Desain Umum Sistem

3.2. Desain Perangkat Lunak

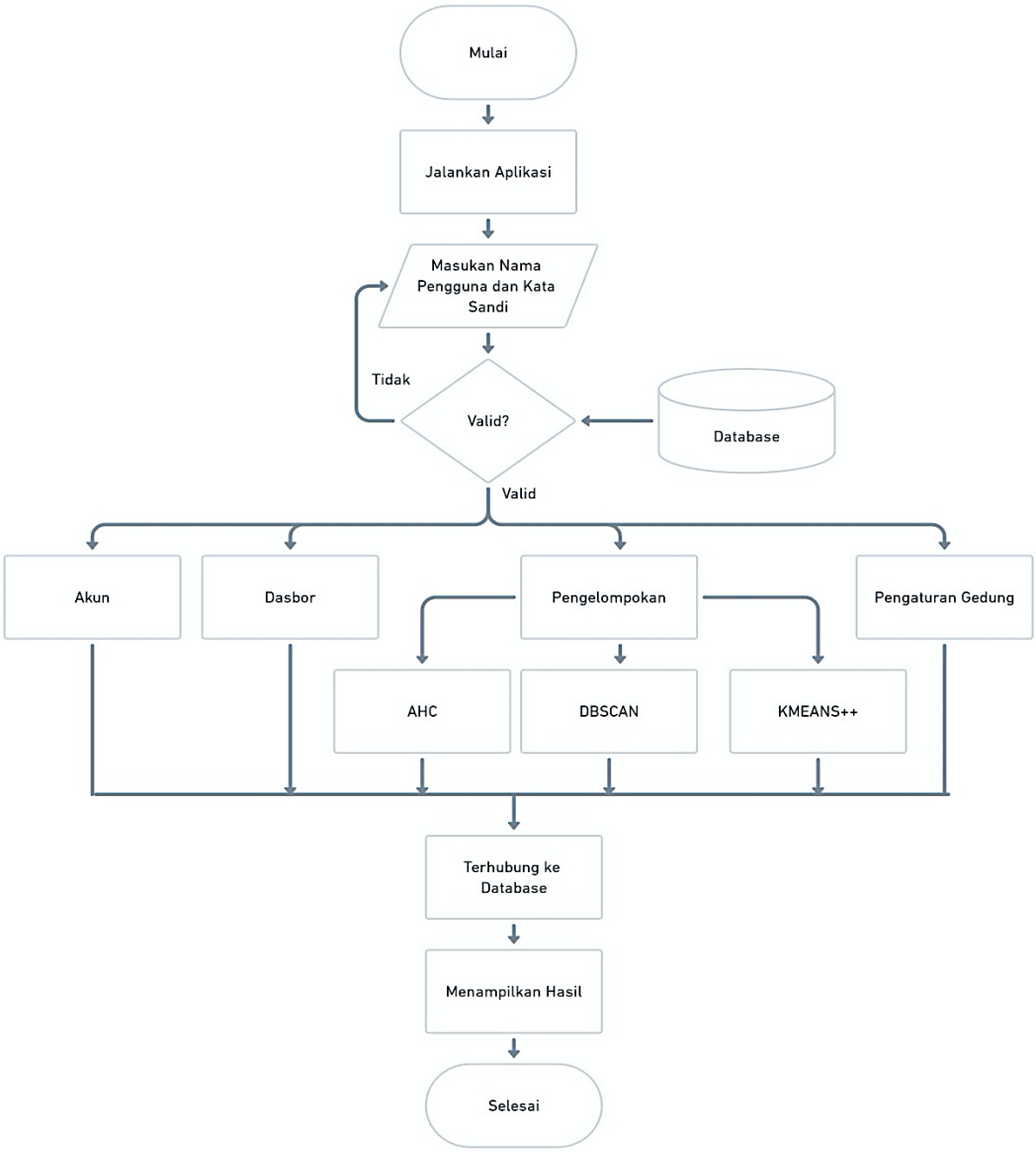
3.2.1. Fitur dan Fungsi

Pada aplikasi berbasis *web app* ini menampilkan informasi penggunaan listrik dan mengelompokan penggunaan listrik di Gedung Telkom *University* untuk membantu tim logistik Telkom *University* memantau penggunaan listrik. Adapun aplikasi berbasis *web app* ini memiliki fitur dan fungsi sebagai berikut :

1. Fitur *login* untuk *admin* dan *user* yang sudah dibuat oleh *admin*.
2. Fitur menambah akun oleh *admin.*
3. Fitur *dashboard* dengan menampilkan peta gedung dengan indikator hasil pengelompokan data penggunaan listrik antar gedung.
4. Fitur yang menampilkan informasi hasil dari pengelompokan dari data penggunaan listrik setiap gedung.
5. Fitur untuk mengedit gedung untuk memantau penggunaan listriknya.

3.2.2. Diagram Alir Sistem

Sistem *monitoring* penggunaan listrik di gedung Telkom *University* yang berbasis aplikasi *website* memiliki *flowchart* sebagai berikut :



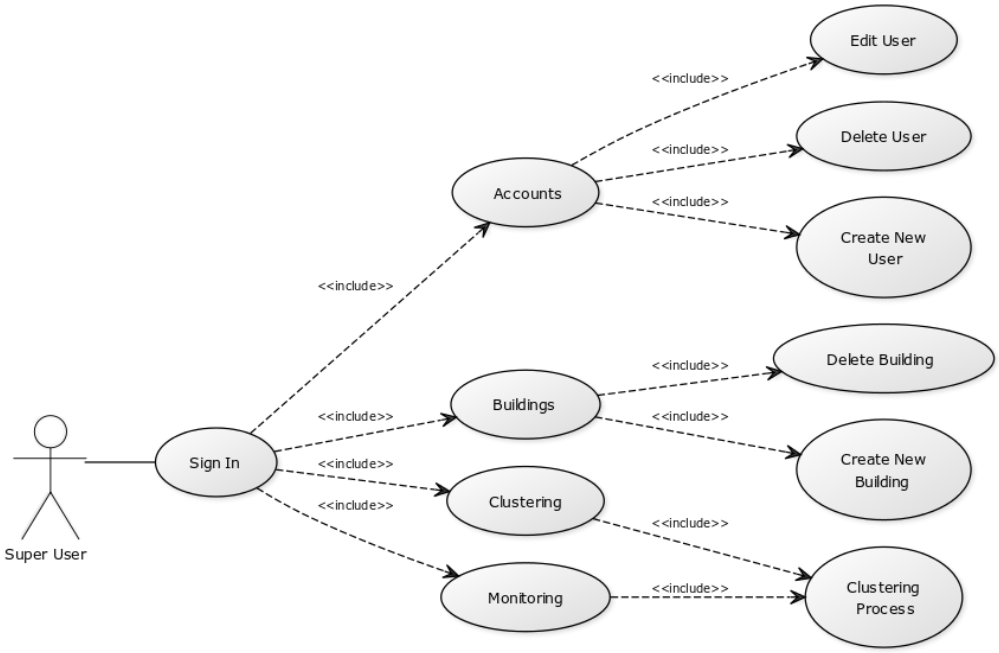
Gambar 3. 2 Flow Chart System

Dari gambar diagram alur yang sudah dibuat dalam 1 aplikasi berbasis *website* ini terdapat fitur *clustering* yang memiliki 3 pilihan fitur, yaitu *Agglomerative Hierarchial*, K-Means++, dan DBSCAN. Pada Tugas Akhir ini berfokus pada penggunaan algoritma K-Means++. *Flow Chart* dimulai dari halaman *login* yang akunnya dibuat oleh *admin*, jika sesuai dengan *database* lanjut ke *menu dashboard* yang memiliki fitur-fitur yang bisa dipilih oleh *user*. Ketika *login* berhasil maka akan masuk ke halaman *dashboard*. Halaman dashboard berisi beberapa *card* yang masing-masing isi dari *card* tersebut melambangkan setiap gedung yang ada, setiap card memiliki warna sesuai besar penggunaan kwh yang datanya diambil dari satu

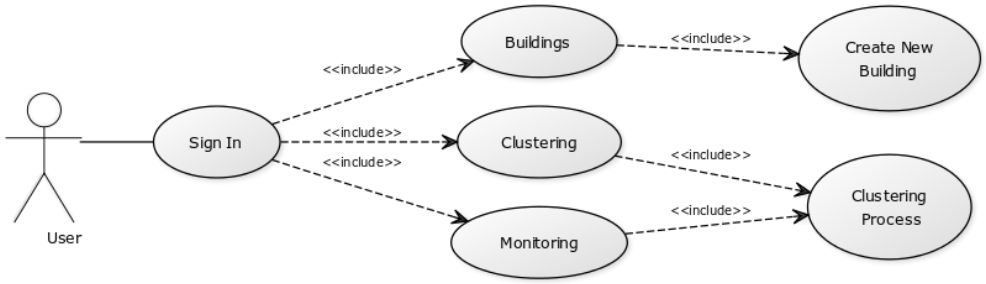
jam sebelum jam saat ini dan sudah dilakukan proses pengelompokan menggunakan algoritma K-Means++.

3.2.3 Rancangan *Use Case Diagram*

Dalam pembangunan web ini mempunyai rancangan yang mempermudah *user* untuk menggunakan fitur-fitur yang ada. Perancangan web ini menggunakan *Use Case Diagram* yang menunjukan interaksi *user* pada sistem sebagai berikut :



Gambar 3. 3 Use Case Super User



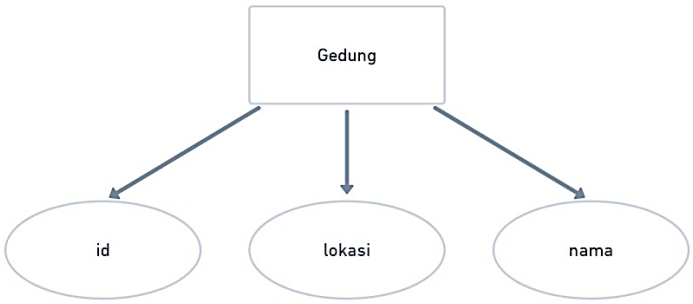
Gambar 3. 4 Use Case User

Sebagai Super user bisa mengakses halaman akun dimana *super user* bisa menambah, menghapus, dan mengganti pengaturan *user*. *Super user* juga bisa

mengakses fitur hapus gedung pada halaman gedung yang tidak dimiliki oleh user biasa.

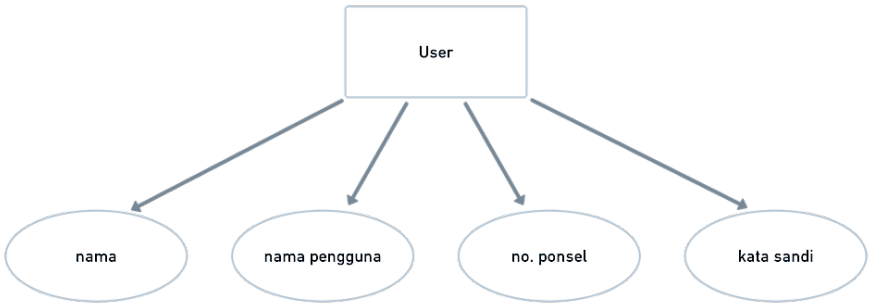
3.2.4 Rancangan *Entity Relationship Diagram*

Pada sistem ini dibuat rancangan fitur *entity relationship diagram* untuk basis data, berikut adalah rangkaian yang dibuat.



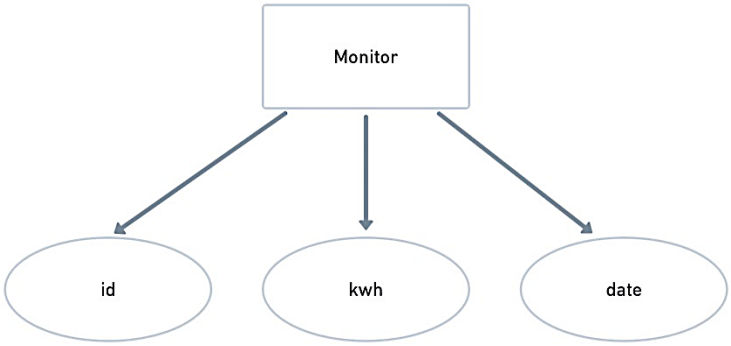
Gambar 3. 5 Entitas Gedung

Entitas Gedung berisi semua data gedung di dalam Telkom University yang memiliki attribute id, lokasi, dan nama.



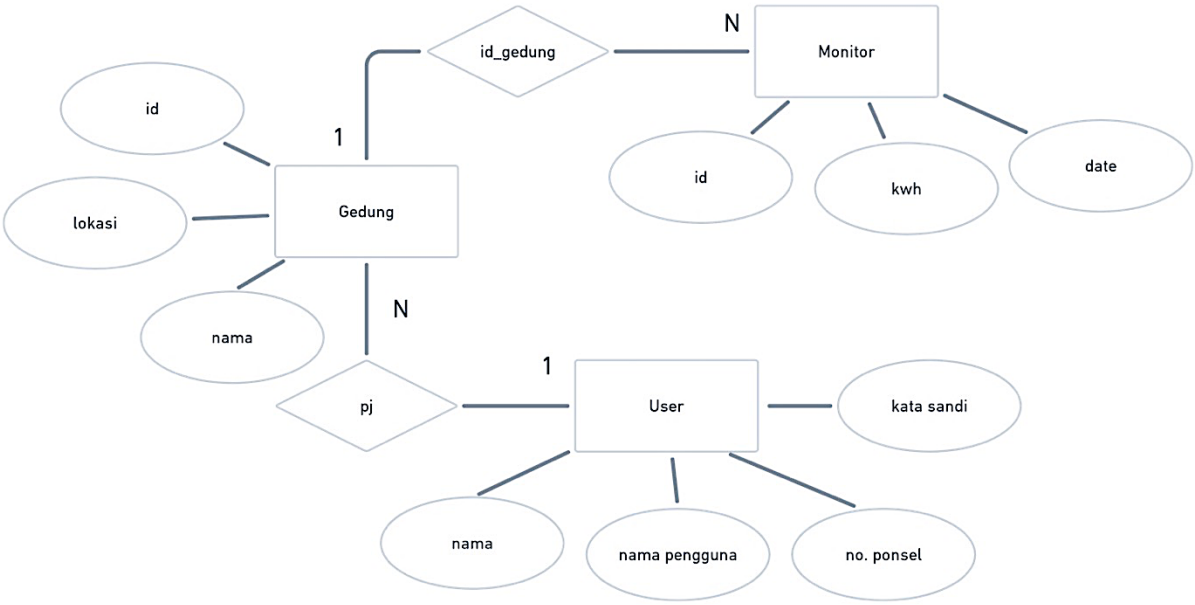
Gambar 3. 6 Entitas User

Entitas dari user yang berisi data pengguna yang memiliki attribute nama, nama pengguna, no.ponsel dan kata sandi.



Gambar 3. 7 Entitas Monitor

Entitas dari monitor yang berisi data penggunaan listrik dari setiap Gedung yang memiliki attribute id, kwh, dan *date*.



Gambar 3. 8 Relation Entity Diagram

ERD diatas merupakan keseluruhan sistem yang dibuat. *User* berelasi dengan gedung menggunakan PJ, jadi *user* akan memiliki banyak gedung dan gedung hanya memiliki satu *user*. Gedung akan berelasi dengan monitor, gedung memiliki banyak data monitor tetapi monitor hanya dimiliki satu gedung.

3.2.5 Pembuatan Skema Relasi

Skema relasi ini berisi tabel-tabel yang mempunyai informasi dan relasi berdasarkan sistem yang diimplementasikan. Skema relasi ini juga dibuat berdasarkan implementasi dari ERD yang telah dibuat. Gambar skema relasi terdapat pada lampiran C.

3.3 Data Collecting

Data yang akan digunakan pada sistem diambil dari server API dan dari *virtual device*. API yang digunakan berisi data listrik *device* 5 gedung Deli Telkom University yang diambil dari bulan Februari sampai bulan juli dan mengirim data terus dalam *interval* tertentu. Data tersebut adalah data *raw* yang belum dilakukan proses apapun. Berikut contoh yang di kirim dari server API berupa *format* JSON.


```
{
    "code": 200,
    "message": "Berhasil memuat data perangkat",
    "data": [
        {
            "id": "58624",
            "id_m_devices": "5",
            "va": "239.1",
            "vb": "240.6",
            "vc": "239.6",
            "vab": "0",
            "vbc": "0",
            "vca": "0",
            "ia": "1.6",
            "ib": "1.9",
            "ic": "2.3",
            "pa": "0.2",
            "pb": "0.09",
            "pc": "0.52",
            "pt": "0.23",
            "qa": "0.29",
            "qb": "0.4",
            "qc": "0.05",
            "qt": "0.06",
            "sa": "0.34",
            "sb": "0.41",
            "sc": "0.27",
            "st": "0.06",
            "pfa": "0.57",
            "pfb": "0.23",
            "pfc": "0.06",
            "freq": "49.88",
            "ep": "405",
            "eq": "0",
            "date": "2021-07-29",
            "time": "01:27:22"
        },
```


Gambar 3. 9 Data Raw JSON

3.4 Preprocessing Data

Preprocessing data dibutuhkan untuk merapikan data yang belum sesuai formatnya, menghilangkan nilai yang kosong, menghilangkan anomali pada data, dan menyesuaikan data dengan kebutuhan sistem yang dibuat agar mendapatkan hasil yang baik, *preprocess* data yang dilakukan pada sistem yang ini adalah sebagai berikut:

1. Memilih *key* dari *object* yang dipakai seperti *Date, Time, Device*, dan Kwh pada JSON yang akan dipakai dan diubah ke bentuk *dataframe* Bersama dengan *value* dari masing-masing *object*.

Tabel 3. 1 Data berupa dataframe

	Unnamed: 0	Date	Time	Device	Kwh
0	0	2021-02-16	21:12:43	5	0.0
1	1	2021-02-16	21:13:08	5	0.0
2	2	2021-02-16	21:13:33	5	0.0
3	3	2021-02-16	21:13:57	5	0.0
4	4	2021-02-16	21:14:22	5	0.0
...	...	...	...	...	...
32464	32464	2021-08-03	09:09:23	5	405.0
32465	32465	2021-08-03	09:14:24	5	405.0
32466	32466	2021-08-03	09:19:25	5	405.0
32467	32467	2021-08-03	09:24:26	5	405.0
32468	32468	2021-08-03	09:29:27	5	405.0

2. Menggabungkan kolom *Date* dengan *Time* dan mengubah formatnya menjadi bentuk *Datetime* dan menghapus kolom *Unnamed*: 0.

Tabel 3. 2 Dataframe tanpa kolom unnamed:0

	Device	Date	Kwh
0	5	2021-02-16 21:12:43	0.0
1	5	2021-02-16 21:13:08	0.0
2	5	2021-02-16 21:13:33	0.0
3	5	2021-02-16 21:13:57	0.0
4	5	2021-02-16 21:14:22	0.0
...	...	...	...
32464	5	2021-08-03 09:09:23	405.0
32465	5	2021-08-03 09:14:24	405.0
32466	5	2021-08-03 09:19:25	405.0
32467	5	2021-08-03 09:24:26	405.0
32468	5	2021-08-03 09:29:27	405.0

3. Membersihkan data dengan cara menghilangkan semua data yang tidak memiliki format *datetime* yang sesuai seperti 'Y-m-d H:M:S' dan menghapus semua Kwh yang bernilai 0.

Tabel 3. 3 Dataframe setelah kwh bernilai 0 dihapus

	Device	Date	Kwh
314	5	2021-03-13 20:49:13	20.0
315	5	2021-03-13 20:49:37	20.0
316	5	2021-03-13 20:50:02	20.0
317	5	2021-03-13 20:50:27	20.0
318	5	2021-03-13 20:51:35	20.0
...	...	...	...
32464	5	2021-08-03 09:09:23	405.0
32465	5	2021-08-03 09:14:24	405.0
32466	5	2021-08-03 09:19:25	405.0
32467	5	2021-08-03 09:24:26	405.0
32468	5	2021-08-03 09:29:27	405.0

4. Menghapus kolom device karena tidak dibutuhkan, dan mengubah *datetime* menjadi nilai *index* agar memudahkan visualisasi data.

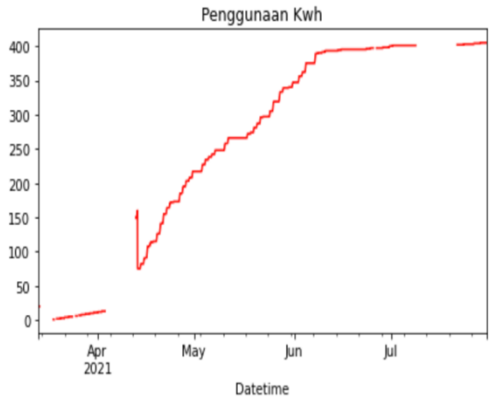
Tabel 3. 4 Dataframe datetime sebagai indeks

Datetime	Kwh
2021-03-13 20:49:13	20.0
2021-03-13 20:49:37	20.0
2021-03-13 20:50:02	20.0
2021-03-13 20:50:27	20.0
2021-03-13 20:51:35	20.0
...	...
2021-08-03 09:09:23	405.0
2021-08-03 09:14:24	405.0
2021-08-03 09:19:25	405.0
2021-08-03 09:24:26	405.0
2021-08-03 09:29:27	405.0

5. Melakukan *resample* data sehingga memiliki pencatatan data per jam, dengan cara mengambil sampel data dalam 30 menit lalu diambil nilai maksimalnya, setelah itu diambil lagi sampel datanya dalam 1 jam lalu mengambil nilai minimalnya.

Tabel 3. 5 Setelah Resample

	Kwh
Datetime	
2021-03-13 20:00:00	20.0
2021-03-13 21:00:00	20.0
2021-03-13 22:00:00	20.0
2021-03-13 23:00:00	20.0
2021-03-14 00:00:00	20.0
...	...
2021-08-03 05:00:00	405.0
2021-08-03 06:00:00	405.0
2021-08-03 07:00:00	405.0
2021-08-03 08:00:00	405.0
2021-08-03 09:00:00	405.0



Gambar 3. 10 Visualisasi setelah resample

6. Melakukan interpolasi data untuk mengisi data yang kosong di antara 2 nilai, berdasarkan persamaan linier dengan rumus berikut:

$$Interpolasi\ Linier : \frac{(x-x1)}{(x2-x1)} = \frac{(y-y1)}{(y2-y1} \qquad (3.1)$$

$$Y = Y1 + \frac{(X-x1)}{(x2-x1)}(Y2 - Y1)$$

Dimana:

x2: sumbu x setelah nilai yang dicari.

y2: sumbu y diatas nilai yang dicari.

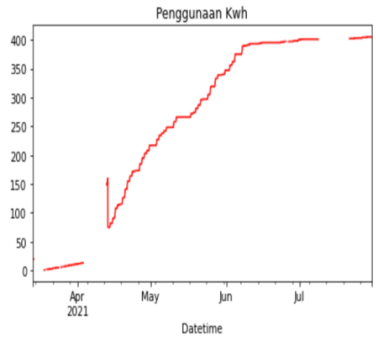
x1: sumbu x sebelum nilai yang dicari.

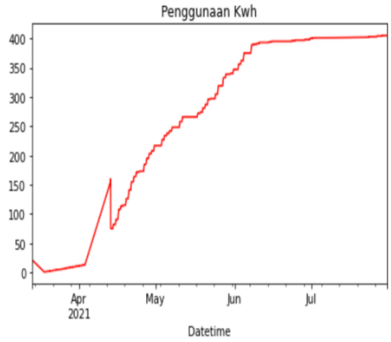
y1: sumbu y sebelum nilai yang dicari.

x: sumbu x yang akan dicari.

y: sumbu y yang akan dicari.

Dengan menjalankan rumus interpolasi linier akan didapatkan nilai di titik kosong dan nilai terlengkapi.



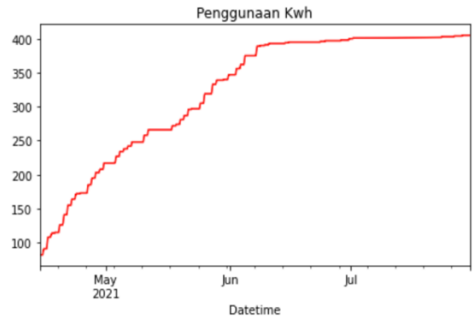


Gambar 3. 11 Sebelum Interpolasi Gambar 3. 12 Setelah Interpolasi

Pemotong data sebesar 30% yang bertujuan agar data listrik (kwh) tidak memiliki *anomaly* data.

Tabel 3. 6 Dataframe setelah dipotong 30%

	Kwh
Datetime	
2021-04-15 16:00:00	91.0
2021-04-15 17:00:00	91.0
2021-04-15 18:00:00	91.0
2021-04-15 19:00:00	91.0
2021-04-15 20:00:00	91.0
...	...
2021-08-03 05:00:00	405.0
2021-08-03 06:00:00	405.0
2021-08-03 07:00:00	405.0
2021-08-03 08:00:00	405.0
2021-08-03 09:00:00	405.0



Gambar 3. 13 Visualisasi setelah dipotong 30%

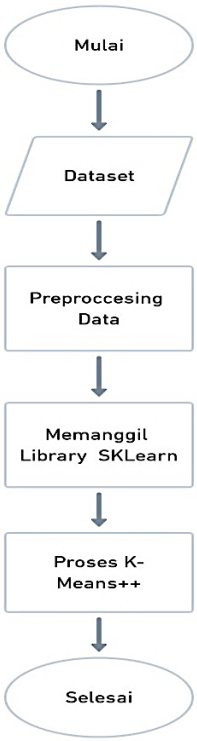
7. Terakhir adalah pencarian nilai kenaikan penggunaan listrik. Dengan mengurangi nilai kwh sekarang dengan nilai kwh satu jam sebelum sekarang dan menetapkan nilai *default* 0 untuk *index* pertama.

Tabel 3. 7 Dataframe dengan kolom delta Kwh

	Kwh	old_kwh	delta_kwh
Datetime			
2021-04-24 22:00:00	173.0	0.0	0.0
2021-04-24 23:00:00	173.0	173.0	0.0
2021-04-25 00:00:00	173.0	173.0	0.0
2021-04-25 01:00:00	173.0	173.0	0.0
2021-04-25 02:00:00	173.0	173.0	0.0
...	...	...	...
2021-08-03 05:00:00	405.0	405.0	0.0
2021-08-03 06:00:00	405.0	405.0	0.0
2021-08-03 07:00:00	405.0	405.0	0.0
2021-08-03 08:00:00	405.0	405.0	0.0
2021-08-03 09:00:00	405.0	405.0	0.0

3.5 Rancangan K-Means++ Clustering

Perancangan algoritma K-Means++ pada sistem dimulai dari pengambilan dataset yang diambil dari *database* yang sudah dibuat, lalu data melalui proses *preprocessing* untuk mencari nilai kenaikan dari penggunaan listriknya, jika sudah dapat nilainya, proses selanjutnya adalah memanggil *library* SKlearn dengan algoritma K-Means++ untuk pengelompokan data berdasarkan letak *centroid*. *variable* yang dimasukan ke dalam model KMeans++ adalah *variable* nilai kenaikan listrik yang sudah diambil dari hasil *preprocessing*.



Gambar 3. 14 Alur Kerja KMeans++ pada sistem

Pada dataset ini dibutuhkan *variable* kenaikan kwh. Hasil akhir dari klasterisasi adalah visualisasi berupa *doughnut chart* untuk memudahkan pengguna mengetahui kondisi penggunaan listrik dan terdapat pada tabel yang berisi data yang memiliki atribut *Date* yang berisi tanggal data, *Time* yang berisi waktu dari data, Kwh yang berisi penggunaan listrik dari data, Delta Kwh yang berisi nilai kenaikan penggunaan listrik, cluster yang berisi label hasil dari pengelompokan, dan terakhir adalah kategori yang berisi penjelasan dari label yang sudah terbentuk dari setiap data. Untuk perhitungan K-Means++ menggunakan 10 sampel data real pada tanggal 3 Mei 2021.

Tabel 3. 8 Sampel data tanggal 3 Mei 2021 jam 06:00 sampai 15:00.

Time	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**
Delta Kwh	0	0	1	0	1	2	2	1	2	1

Setelah mempunyai data sampel, pilih centroid awal secara acak, centroid awal secara acak yang dipilih adalah titik (0,0). Lalu hitung jarak antara delta kwh dan centroid awal. Karena titik centroid sudah sesuai dengan salah satu titik data

tidak perlu dicari nilai $D(x)^2$nya. Hitungan untuk centroid 2 dan 3 terdapat pada lampiran A.

$$D(x)^2=(\sqrt{(0-0)^2+(0-0)^2})^2=0$$

Tabel 3. 9 Hasil nilai Squared Distance dimana c1 = (0,0)

Time	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**
X	(0,0)	(0,0)	(1,1)	(0,0)	(1,1)	(2,2)	(2,2)	(1,1)	(2,2)	(1,1)
$(D(x,c1)^2)$	-	-	2	-	2	8	8	2	8	2

Kemudian cari nilai probabilitasnya menggunakan $D(x)^2$ weighting untuk menentukan *centroid* ke dua, pilih nilai yang paling sebanding nilai probabilitasnya.

$$K=\frac{D(x)^2=(\sqrt{(2-0)^2+(2-0)^2})^2}{Sum(D(x)^2)}=\frac{8}{32}$$

Tabel 3. 10 Probability of squared distance

Time	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**
x	(0,0)	(0,0)	(1,1)	(0,0)	(1,1)	(2,2)	(2,2)	(1,1)	(2,2)	(1,1)
Probability	-	-	$\frac{2}{32}$	-	$\frac{2}{32}$	$\frac{8}{32}$	$\frac{8}{32}$	$\frac{2}{32}$	$\frac{8}{32}$	$\frac{2}{32}$

Dilihat dari tabel nilai 8/32 adalah nilai terbesar yang berarti titik ini menjadi titik *centroid* kedua yaitu pada titk (2, 2).

$$D(x)^2=(\sqrt{(1-2)^2+(1-2)^2})^2=2$$

Tabel 3. 11 Table probability untuk centroid 2 yaitu 4/20

Time	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**
x	(0,0)	(0,0)	(1,1)	(0,0)	(1,1)	(2,2)	(2,2)	(1,1)	(2,2)	(1,1)
$(D(x,c1)^2)$	-	-	2	-	2	-	-	2	-	2

Untuk mencari *centroid* ketiga lakukan hal yang sama, ulangi kembali mencari nilai sebanding dari probability untuk menjadi centroid selanjutnya.

Tabel 3. 12 Tabel probability mencari centroid selanjutnya

Time	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**
x	(0,0)	(0,0)	(1,1)	(0,0)	(1,1)	(2,2)	(2,2)	(1,1)	(2,2)	(1,1)
Probability	-	-	2/32	-	2/32	-	-	2/32	-	2/32

Didapatkan hasil *centroid* dari hitungan *probability* menggunakan *weighting*. Dalam kasus ini sudah di dapatkan nilai masing-masing *cluster* dimana C1=(0,0), C2=(2,2), dan C3=(1,1). Setelah itu lakukan perhitungan K-Means biasa menggunakan rumus *Euclidean distance*.

$$D(x1)C1 = \sqrt{(0-0)^2 + (0-0)^2} = 0$$
$$D(x1)C2 = \sqrt{(2-0)^2 + (2-0)^2} = 2{,}8$$
$$D(x1)C3 = \sqrt{(1-0)^2 + (1-0)^2} = 1{,}4$$

Tabel 3. 13 Hasil Euclidean Distance

Time	**x**	**DC1**	**DC2**	**DC3**	**C1**	**C2**	**C3**
06:00	(0,0)	0	2,8	1,4	V		
07:00	(0,0)	0	2,8	1,4	V		
08:00	(1,1)	1,4	1,4	0			V
09:00	(0,0)	0	2,8	1,4	V		
10:00	(1,1)	1,4	1,4	0			V
11:00	(2,2)	2,8	0	1,4		V	
12:00	(2,2)	2,8	0	1,4		V	
13:00	(1,1)	1,4	1,4	0			V
14:00	(2,2)	2,8	0	1,4		V	
15:00	(1,1)	1,4	1,4	0			V

Dari perhitungan kmeans iterasi pertama dipilih nilai *Euclidean* paling terkecil diantara titik *centroid* yang lain untuk menentukan data tersebut masuk ke *cluster* mana. Didapatkan hasil 3 cluster dari iterasi pertama dimana jarak terjauh adalah 2,8 dan jarak terdekat adalah 0. Hitungan rumus sampel selanjutnya dapat dilihat pada lampiran A.

$$\text{Rata rata } centroid = \frac{sum(x\ value\ of\ c1)}{sum\ c1} = \frac{0}{3} = 0$$

$$\text{Rata rata } centroid = \frac{6}{3} = 2$$

$$\text{Rata rata } centroid = \frac{4}{4} = 1$$

Tabel 3. 14 Rata-rata nilai centroid

	x
C1	(0,0)
C2	(2,2)
C3	(1,1)

Langkah terakhir pada iterasi pertama adalah menghitung nilai rata-rata *centroid*, jika berubah, hitung kembali untuk iterasi kedua menggunakan nilai rata-rata *centroid* yang baru. Pada kasus ini nilai rata-rata *centroid* tidak berubah maka sudah selesai di iterasi pertama, disebabkan titik *centroid* awal sudah ditentukan menggunakan probabilitas *weighting*.

Setelah melakukan proses *clustering*, Perhitungan *Silhouette Coefficient* dilakukan untuk melakukan pengujian clustering menggunakan rumus (2.4) mencari rata rata jarak dalam 1 *cluster* dimana:

a= 0

b: min ($\frac{5,6}{4}, \frac{8,4}{3}$)

b: (1,4)

$s(i) = \frac{1,4-0}{1,4} = 1$

Didapatkan hasil dari *silhouette* nilainya 1 yang artinya memiliki struktur kuat. Dengan kata lain berarti hasil pengelompokan menggunakan KMeans++ pada data sampel, setiap datanya sudah berada di kelompok yang tepat.

3.5.1 Implementasi K-Means++ Clustering pada web app

***1. Clustering* antar Gedung**

Setelah berhasil masuk halaman *login, user* berada pada *dashboard* dimana terdapat *card* yang bertuliskan nama gedung, kenaikan kwh Gedung, dan warna dari setiap card yang berbeda-beda.

***2. Clustering* Gedung**

a. *Clustering* perhari

Pada halaman clustering, telah tersedia form kedua dimana user bisa memilih pada gedung, tanggal, bulan, dan tahun berapa data yang ingin dilihat hasil dari pengelompokannya. Setelah proses pengelompokan berhasil user bisa melihat apakah penggunaan listrik pada satu hari itu rendah, normal, atau tinggi.

b. Clustering perbulan

Pada form pertama halaman clustering tersedia form dimana user bisa memilih Gedung, bulan, dan tahun berapa data yang ingin dilihat hasil dari pengelompokannya agar mengetahui dalam satu bulan ini apakah penggunaannya listriknya normal, rendah, atau tinggi.

3.6. Kebutuhan Perangkat

3.6.1. Perangkat Lunak

Perangkat lunak yang digunakan untuk mengembangkan aplikasi *website* ini adalah sebagai berikut:

1. *Operating System* : MacOs Catalina 10.15.3
2. *Code Editor* : *Visual Studio Code*
3. *Search Engine* : Google Chrome
4. *Programing Language* : Python 3.6.5

3.6.2. Perangkat Keras

Perangkat keras yang digunakan untuk mengembangkan aplikasi *website* ini adalah sebagai berikut:

1. Processor : 1,6 GHz Dual-Core Intel Core i5
2. RAM : 8 GB 2133 MHz LPDDR3
3. Memory : SSD 128 GB
4. GPU : Intel UHD Graphics 617 1536 MB