

## **BAB III LANDASAN TEORI**

### **3.1 Aplikasi Mobile**

Pengertian aplikasi *mobile* ialah suatu aplikasi yang digunakan dalam piranti *smartphone*. Nama lain aplikasi *mobile* adalah *mobile apps*, dimana user biasanya mengakses informasi melalui hp atau *smartphone* mereka yang terkoneksi dengan layanan internet. Sehingga dapat mempermudah user dalam mendapat informasi tanpa harus melalui PC. Selain itu aplikasi *mobile* juga dapat diakses di mana saja tergantung dengan layanan koneksi internetnya (Turban, 2012).

Aplikasi *mobile* merupakan salah satu perangkat lunak yang difungsikan untuk melakukan tugas-tugas tertentu pada perangkat *mobile*, seperti *handphone*, *smartphone*, dan PDA. *Mobile application* tidak hanya mampu untuk melakukan proses layanan dasar seperti layanan telepon maupun layanan pesan, tetapi sudah mampu untuk melakukan tugas-tugas yang rumit seperti melakukan pencarian posisi pengguna, menampilkan dan memproses informasi dalam peta digital (Lee, V & Schneider, H & Schell, R. 2004)

Aplikasi *mobile* dapat digunakan untuk mengakses suatu aplikasi web secara nirkabel. Aplikasi *mobile* tidak membutuhkan bandwidth yang terlalu besar karena data yang ditampilkan hanya berupa teks. Penggunaan aplikasi *mobile* hanya memerlukan telepon seluler yang sudah dilengkapi dengan fasilitas *General Packet Radio Service (GPRS)* dan koneksi internetnya. Perangkat keras merupakan fitur penting dalam membangun sebuah aplikasi *mobile*. Selain itu

infrastruktur jaringan yang baik sangat mendukung dalam aplikasi mobile. Kondisi jaringan sudah memungkinkan untuk mendapatkan *bandwidth* yang cukup besar untuk jaringan seluler (Li Ma, Lei Gu & Jin Wang, 2014).

Selain itu, pertimbangan terhadap keterbatasan piranti mobile pun harus diperhatikan (Anindito, 2012), yaitu:

1. Keterbatasan kecepatan prosesor dalam mengeksekusi proses.
2. Keterbatasan RAM.
3. Ukuran layar yang berbeda beda..
4. Keterbatasan input pada masing-masing piranti *mobile*.
5. Ketahanan baterai yang berbeda pada setiap piranti *mobile*

### **3.2 Pariwisata**

Pariwisata merupakan kegiatan untuk bersantai atau juga kegiatan untuk mengisi waktu luang. Biasanya perjalanan pariwisata dilakukan seseorang dalam keadaan mereka cuti atau berlibur dimana orang tersebut bebas dari pekerjaan yang biasanya dilakukan setiap hari. Selain untuk menghilangkan kejenuhan dan kebosanan dengan aktifitas sehari-hari (Parroco et. all., 2012).

Pariwisata sebagai "jumlah dari ... perjalanan dari non-penduduk (turis) ke daerah tujuan, asalkan mereka tinggal tidak menjadi tempat tinggal permanen. Ini adalah kombinasi dari rekreasi dan bisnis ". Meskipun demikian, definisi pariwisata atau turis adalah yang paling umum digunakan oleh literatur tentang pariwisata. "Setiap orang yang berada dalam suatu negara, terlepas dari kebangsaan, bepergian ke tempat di dalam negeri ini selain tempat tinggalnya

untuk jangka waktu tidak kurang dari 24 jam atau satu malam untuk tujuan lain selain latihan dari pekerjaan yang dibayar di tempat yang dikunjungi. Motif wisata tersebut mungkin (1) rekreasi (rekreasi, liburan, kesehatan, studi, agama, olahraga); (2) bisnis, keluarga, misi, pertemuan (Awang, K.,H., Wan Hassan, W.,M. & Zahari, M. 2009).

### **3.3 Pemandu Wisata**

Pemandu wisata merupakan seseorang yang bertugas untuk memandu orang-orang di dalam kunjungan wisata dengan durasi yang terbatas. Definisi lain untuk pemandu wisata adalah individu yang mengarahkan jalan; yang memimpin orang lain di perjalanan atau tour; dan orang yang mengarahkan atau berfungsi sebagai model untuk orang lain dalam tindakannya. Definisi ini menunjukkan bahwa pemandu wisata bertanggung jawab untuk mengarahkan dan memimpin wisatawan di dalam tur atau jalan-jalan (Chilembwe, J., M. & Mweiwa, V., 2014).

Prakash dan Chowdhary pada tahun 2010 mendefinisikan pemandu wisata sebagai orang yang memandu pengunjung di bahasa pilihan mereka dan menafsirkan warisan budaya dan alam suatu daerah yang orang biasanya memiliki sebuah daerah khusus kualifikasi biasanya dikeluarkan dan atau diakui oleh otoritas yang tepat. Oleh karena itu, pemandu wisata harus terampil dan cekatan dalam memandu pengunjung karena dalam banyak cara, pemandu wisata berada di garis depan bisnis. Pemandu wisata akan berinteraksi sesering mungkin kepada pengunjung sehingga mereka adalah aset berharga bagi perusahaan pariwisata.

Pemandu wisata sebagai pribadi; Yang membimbing kelompok atau individu pengunjung dari luar negeri atau dari negara asal di sekitar monumen, situs dan museum dari kota atau wilayah; menafsirkan secara inspiratif dan menghibur, dalam bahasa pilihan pengunjung, termasuk warisan budaya dan alam dan lingkungan hidup (Chilembwe, J., M. & Mweiwa, V., 2014).

### **3.4 Surakarta**

Surakarta adalah salah satu kotamadya yang berada di provinsi Jawa Tengah yang pada umumnya orang mengetahuinya dengan nama Solo. Letak Surakarta antara  $110^{\circ} 45' 15'' - 110^{\circ} 45' 35''$  Bujur Timur dan  $70^{\circ} 36' - 70^{\circ} 56''$  Lintang Selatan dan dialiri sungai bengawan solo. Surakarta terletak pada cekungan lereng gunung lawu dan gunung merapi. Iklim di kota surakarta adalah tropis dengan suhu berkisar antara  $27^{\circ} - 35^{\circ}$  C.

Kota Surakarta merupakan daerah otonom dengan yang terdiri dari 5 kecamatan dengan jumlah penduduk kota Surakarta pada akhir 2015 adalah 585.486 jiwa dengan kepadatan  $13.294/km^2$ . Kota Surakarta memiliki luas sekitar  $44 km^2$  dan berbatasan dengan beberapa kabupaten yaitu :

- Bagian utara berbatasan dengan Kabupaten Karanganyar dan Kabupaten Boyolali.
- Bagian timur dan barat berbatasan dengan Kabupaten Karanganyar dan Kabupaten Sukoharjo.
- Bagian selatan dengan Kabupaten Sukoharjo.

Kota Surakarta memiliki banyak tempat wisata, mulai dari wisata alam, wisata budaya atau wisata sejarah. Wisata alamnya meliputi grojogan sewu, taman balekambang dan air terjun jumog. Sedangkan wisata budayanya meliputi museum batik danar hadi, keraton surakarta, museum keris dan lain lain. Selain itu juga ada beberapa candi-candi peninggalan kebudayaan Hindu-Buddha, seperti Candi Suku, Candi Cetho, Candi Monyet, dan lain-lain.

Disamping tempat wisata, kota surakarta juga terkenal dengan pasar pasar tradisionalnya. Salah satu pasar tradisional yang terkenal adalah Pasar Klewer. Pasar Klewer menjual beraneka ragam barang khas setempat dengan harga yang relatif murah. Banyak wisatawan yang tertarik untuk membeli oleh-oleh atau hanya sekedar untuk cuci mata / jalan jalan saja. Barang yang sering diburu wisatawan salah satunya kain batik.

Kota surakarta juga terkenal dengan kulinernya. Salah satu kuliner khas kota surakarta adalah serabi notosuman. Selain enak, harga serabi itu juga relatif sangat murah bekisar 2400 per buah. Sehingga banyak wisatawan membeli serabi tersebut untuk dijadikan oleh-oleh buat keluarganya. Kuliner yang terkenal lainnya adalah nasi liwet, pecel ndeso , timlo, sate kere dan sate buntel serta selat khas solo.

### **3.5 Android**

Android merupakan sistem operasi linux yang digunakan pada smartphone atau hp. Para pengembang dapat membuat aplikasi baru sesuai dengan keinginan mereka sendiri karena adanya platform yang disediakan oleh android (Bharathi et

al., 2010). Hal itu yang menyebabkan android banyak digunakan oleh orang dalam membantu kegiatannya (N. Safaat Harahap, 2012). Karakteristik utama dalam android meliputi terbuka, semua aplikasi dibuat sama, dapat menyelesaikan masalah dalam aplikasi dan aplikasinya mudah dan cepat untuk dikembangkan (Holla & Katti, 2012). Logo android seperti pada gambar 3.1.



Gambar 3. 1 Logo Android

Pada saat perilisan perdana *Android*, 5 November 2007, *Android* bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, *Google* merilis kode-kode *Android* di bawah lisensi apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler (Bharathi et al., 2010).

Android memiliki dua distributor, yaitu Google Mail Service (GMS) dan Open Handset Distributor (OHD). GMS adalah distributor Android yang mendapatkan dukungan penuh dari Google, sedangkan OHD adalah distributor Android tanpa dukungan langsung dari Google. Saat ini sudah banyak bermunculan vendor. Vendor untuk smartphone, yaitu diantaranya HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexus, SciPhone, WayteQ, Sony Ericsson, Acer, Philips, T Mobile, Nexian, IMO, Asus dan lainnya vendor yang memproduksi smartphone Android. Munculnya

vendor smartphone yang telah dikarenakan sistem operasi Android bersifat open source (Bharathi et al., 2010).

Android menjadi pesaing utama dari produk smartphone lainnya seperti Apple dan BlackBerry. Pesatnya pertumbuhan Android karena platform Android sangatlah lengkap baik dari segi sistem operasinya, aplikasi dan tools pengembangannya, market yang menyimpan berbagai aplikasi serta ditambah dengan berbagai dukungan oleh komunitas open source di dunia, sehingga Android berkembang pesat hingga saat ini, baik dari segi teknologi maupun dari segi jumlah. Pada bulan Juni 2011, Google mengumumkan bahwa rata - rata aktivasi *Android device* perhari mencapai setengah juta unit.

Terdapat beberapa versi pada sistem operasi *Android*, mulai dari versi 1.5 (Cup Cake), versi 1.6 (Donut), versi 2.1 (Eclair), versi 2.2 (Froyo), versi 2.3 (GingerBread), versi 3.0 (HoneyCome), hingga versi yang terbaru yaitu versi 4.0 Ice Cream Sandwich (Holla & Katti, 2012).

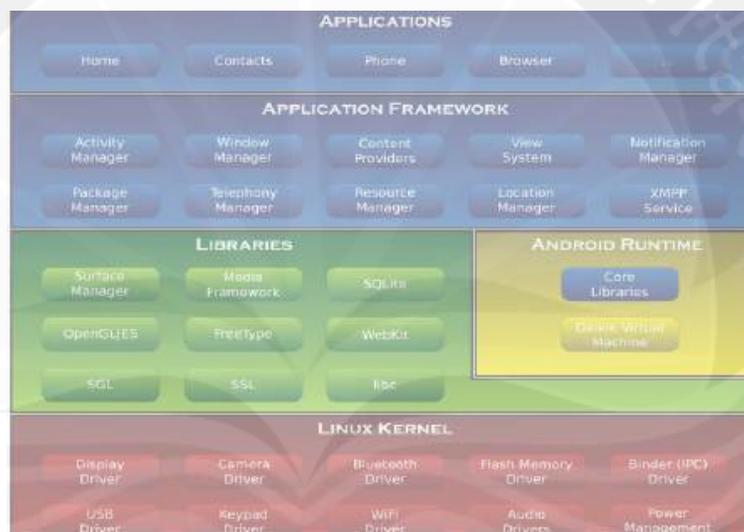
Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan *Android* bukanlah linux, karena dalam android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada android hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan driver. Kernel linux menyediakan driver *Layar*, kamera, keypad, WiFi, Flash Memory, audio, dan IPC (Interprocess Communication) untuk mengatur aplikasi dan lubang keamanan.

Android menggunakan beberapa paket *Library* yang terdapat pada C/C++ dengan standar Berkeley Software Distribution (BSD) hanya setengah dari yang aslinya untuk tertanam pada kernel Linux. Beberapa *library* diantaranya Media *Library* untuk memutar dan merekam berbagai macam format audio dan video. *Surface manager* digunakan untuk mengatur hak akses layer dari berbagai aplikasi. Graphic *Library* termasuk didalamnya SGL dan OpenGL, untuk tampilan 2D dan 3D. SQLite untuk mengatur relasi database yang digunakan pada aplikasi. SSL dan WebKit untuk browser dan keamanan internet.

*Library* tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi Android 1.5, pengembang dapat membuat dan menggunakan pustaka sendiri menggunakan Native Development Toolkit (NDK). Pada android tertanam paket pustaka inti yang menyediakan sebagian besar fungsi android. Inilah yang membedakan Android dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. Android Runtime merupakan mesin virtual yang membuat aplikasi android menjadi lebih tangguh dengan paket pustaka yang telah ada. Dalam Android Runtime terdapat 2 bagian utama, diantaranya Pustaka Inti dan Mesin Virtual Dalvik.

Fitur penting dalam android adalah bahwa satu aplikasi dapat menggunakan elemen dari aplikasi lain. Sebagai contoh, sebuah aplikasi memerlukan fitur scroller dan aplikasi lain telah mengembangkan fitur scroller yang baik dan memungkinkan aplikasi lain menggunakannya. Maka pengembang tidak perlu lagi mengembangkan hal serupa untuk aplikasinya, cukup

menggunakan scroller yang telah ada. Agar fitur tersebut dapat bekerja, sistem harus dapat menjalankan aplikasi ketika setiap bagian aplikasi itu dibutuhkan, dan pemanggilan objek java untuk bagian itu. Oleh karenanya android berbeda dari sistem-sistem lain, Android tidak memiliki satu tampilan utama program seperti fungsi main pada aplikasi lain. Sebaliknya, aplikasi memiliki komponen penting yang memungkinkan sistem untuk memanggil dan menjalankan ketika dibutuhkan. Arsitektur android dapat dilihat pada gambar 3.2.



Gambar 3. 2 Arsitektur Android

### 3.6 LBS

Layanan Berbasis Lokasi (LBS) adalah layanan informasi yang memanfaatkan kemampuan perangkat bergerak untuk menggunakan informasi lokasi dan dapat diakses dengan melalui jaringan telekomunikasi bergerak (Steiniger, Neun, & Edwardes, 2006). LBS terdiri dari beberapa komponen, antara lain.

#### *a. Mobile Devices*

Perangkat yang digunakan pengguna untuk mengakses informasi yang dibutuhkan. Hasil kalkulasi tersebut bisa berupa suara, gambar, teks, dan lainnya.

*b. Communication Network*

Komponen ini berupa jaringan telekomunikasi bergerak yang memindahkan data pengguna dan permintaan terhadap layanan dari perangkat bergerak ke penyedia layanan dan kemudian informasi yang diminta ke pengguna.

*c. Positioning Component*

Dalam pemrosesan layanan, posisi pengguna harus ditentukan. Posisi pengguna bisa didapatkan dengan menggunakan jaringan telekomunikasi bergerak, jaringan LAN nirkabel, atau dengan GPS.

*d. Service and Content Provider*

Penyedia layanan menyediakan sejumlah layanan berbeda kepada pengguna dan bertanggung jawab terhadap pemrosesan permintaan layanan. Contoh layanannya adalah kalkulasi posisi, pencarian rute, dan lainnya.

### **3.7 Algoritma kNN**

*k-nearest neighbour (KNN)* adalah salah satu algoritma mesin pembelajaran yang sederhana. Hal ini hanya didasarkan pada gagasan bahwa "benda-benda yang 'dekat' satu sama lain juga akan memiliki karakteristik yang mirip. Jadi jika Anda tahu ciri-ciri dari salah satu objek, Anda juga dapat memprediksi untuk tetangga terdekat "k -. NN adalah improvisasi lebih teknik

tetangga terdekat. Hal ini didasarkan pada gagasan bahwa setiap contoh baru dapat diklasifikasikan oleh suara mayoritas dari yang 'k' tetangga, di mana k adalah bilangan bulat positif, biasanya nomor kecil (Khamis, Cheruiyo and Kimani, 2014).

Sebagai contoh pelatihan perlu berada di memori pada saat waktu run maka algoritma kNN disebut klasifikasi berdasarkan memori. Ketika berhadapan dengan atribut yang berkelanjutan perbedaan antara atribut dihitung menggunakan jarak Euclidean. Masalah utama ketika berhadapan dengan rumus jarak Euclidean adalah bahwa nilai-nilai frekuensi yang besar rawa yang lebih kecil (Alpaydin, E., 1997).

Pada fase pembelajaran, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi dari data pembelajaran. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk data test (yang klasifikasinya tidak diketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor data pembelajaran dihitung, dan sejumlah k buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut.

Nilai k yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan cross-validation. Kasus khusus di mana klasifikasi diprediksikan berdasarkan data

pembelajaran yang paling dekat (dengan kata lain,  $k = 1$ ) disebut algoritma k-nearest neighbor.

Ketepatan algoritma k-NN ini sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur agar performa klasifikasi menjadi lebih baik.

Fungsi utama kNN adalah untuk mendapatkan jarak paling dekat antara data yang akan dikerjakan dengan k yang terdekat dengan data pelatihannya. Untuk mendapatkan jarak tersebut, maka rumus yang digunakan dalam mencari jarak terdekat pada kNN adalah

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2} \dots\dots\dots (1)$$

di mana :

- $x_1$  : sampel data
- $x_2$  : data uji
- $i$  : variable data
- $d$  : jarak
- $p$  : dimensi data

Algoritma KNN sangat sederhana. Algoritma ini bekerja berdasarkan jarak minimum dari data baru terhadap K tetangga terdekat yang telah ditetapkan. Setelah diperoleh K tetangga terdekat, prediksi kelas dari data baru akan ditentukan berdasarkan mayoritas K tetangga terdekat.

KNN dapat dibagi menjadi 2 jenis berdasarkan tetangga yang digunakan sebagai acuan perhitungan yaitu :

- a. 1-NN yaitu pengklasifikasikan dilakukan terhadap 1 data tetangga yang memiliki label terdekat
- b. k-NN yaitu pengklasifikasikan dilakukan terhadap k data tetangga yang memiliki label terdekat dengan K harus lebih besar 1 dan ganjil

Langkah langkah dalam kNN yaitu :

1. Menentukan parameter k dimana k= jumlah tetangga paling dekat.
2. Menghitung kuadrat jarak euclidean dari setiap objek terhadap data sampel yang diberikan.
3. Mengurutkan masing masing objek kedalam kelompok yang mempunyai jarak euclidean terkecil.
4. Mengumpulkan kategori klasifikasi nearest neighbour (Y).
5. Menggunakan kategori mayoritas untuk mendapatkan data hasil klasifikasi.

### **3.8 Algoritma K-Means**

K-means merupakan salah satu metode clustering yang paling populer. dalam setiap cluster pada k-means diwakili oleh nilai rata rata dari setiap objek yang terdapat didalam cluster. Dalam partisi satu set n objek ke k cluster sehingga antar cluster memiliki kesamaan yang rendah sedangkan intra cluster memiliki kesamaan yang tinggi. Kesamaan cluster ini diukur dalam hal nilai rata rata dari setiap objek dalam sebuah cluster (Yadav, J. And Sharma, M., 2013).

Pada algoritma k-means, partisi dataset diberikan ke c (atau k) kluster. Dalam hal ini membutuhkan parameter c yang mewakili jumlah cluster yang harus

diketahui atau ditentukan sebagai nilai apriori tetap sebelum masuk ke cluster analisis. K-means merupakan algoritma yang cepat dan sederhana, selain itu juga relatif efisien dalam kompleksitas waktu komputasinya. Sehingga k-means banyak digunakan dalam penelitian orang untuk menyelesaikan masalah (Cebec, Z. And Yildiz, F., 2015).

K - Means berarti pengelompokan pada dasarnya adalah metode partisi diterapkan untuk menganalisis data dan memperlakukan pengamatan data sebagai objek berdasarkan lokasi dan jarak antara berbagai titik input data. Partisi benda ke dalam kelompok saling eksklusif (K) dilakukan oleh dalam mode sehingga objek dalam setiap cluster tetap sedekat mungkin satu sama lain, tetapi sejauh mungkin dari objek dalam cluster lainnya (Ghosh, S. & Dubey, S. K., 2013).

Setiap cluster ditandai dengan titik pusatnya yaitu massa. Jarak yang digunakan dalam pengelompokan sebagian besar tidak benar-benar mewakili jarak spasial. Secara umum, satu-satunya solusi untuk masalah menemukan minimum global pilihan lengkap dari titik awal. Tapi penggunaan beberapa ulangan dengan titik awal acak mengarah ke solusi yaitu solusi global. Dalam dataset, sejumlah diinginkan cluster K dan satu set k titik awal awal, algoritma clustering K-Means menemukan nomor yang dikehendaki dari cluster yang berbeda dan centroid mereka. Sebuah pusat massa adalah titik yang koordinat diperoleh dengan cara menghitung rata-rata dari masing-masing koordinat titik-titik sampel ditugaskan untuk cluster (Ghosh, S. & Dubey, S. K., 2013).

Untuk mencari jarak antara data dan centroid diperlukan *distance space*. Salah satu rumus yang dipakai adalah *Euclidean Distance Space*. Dengan

menggunakan *euclidean distance space* akan diperoleh hasil jarak paling pendek antara dua titik yang diperhitungkan. Adapun persamaannya adalah sebagai berikut :

$$d_{ij} = \sqrt{\sum_{k=1}^p \{x_{ik} - x_{jk}\}^2} \dots\dots\dots(2)$$

Dimana :

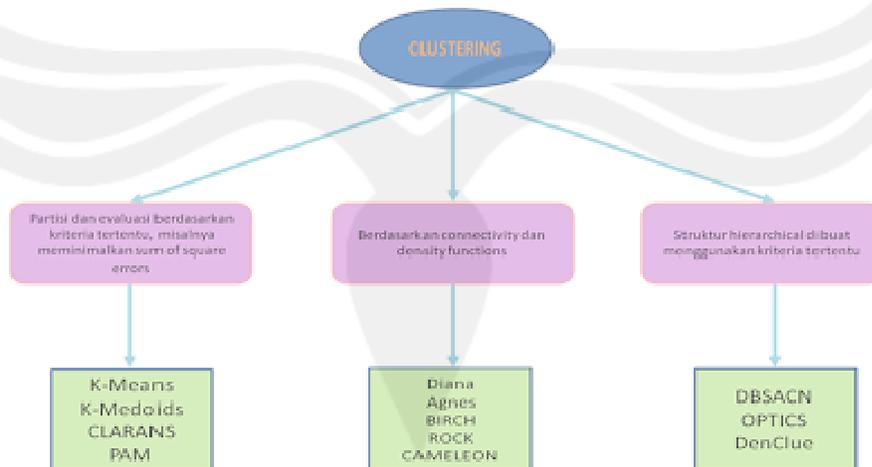
$d_{ij}$  = Jarak objek antara objek i dan j

P = Dimensi data

$X_{ik}$  = Koordinat dari obyek i pada dimensi k

$X_{jk}$  = Koordinat dari obyek j pada dimensi k

Fungsi k-means ialah mengelompokkan sejumlah obyek yang memiliki nilai parameter yang mendekati sama. Beberapa pendekatan clustering digambarkan dengan flowchart seperti pada gambar 3.2



Gambar 3. 3 Flowchart Pendekatan Clustering

Langkah dalam algoritma untuk K - Means Clustering :

1. Tentukan K

Digunakan untuk memilih nomor cluster yang diinginkan.

2. Inisialisasi

Berfungsi untuk memilih titik k awal yang digunakan sebagai perkiraan awal dari centroid cluster. Mereka diambil sebagai nilai awal.

3. Klasifikasi

Digunakan untuk memeriksa setiap titik dalam dataset dan menetapkan ke cluster yang centroid terdekat.

4. Perhitungan Centroid

Ketika setiap titik dalam kumpulan data ditugaskan untuk cluster, diperlukan untuk menghitung ulang centroid k baru.

5. Kriteria Konvergensi

Langkah-langkah (3) dan (4) perlu diulang sampai tidak ada titik perubahan tugas kelompok atau sampai centroid tidak lagi bergerak.

Sampel data harus dikumpulkan sebelum menerapkan algoritma k-means. Prioritas harus diberikan kepada fitur yang menggambarkan setiap sampel data dalam database. Seperti dalam algoritma pengelompokan lain, k-means mengharuskan jarak metrik antara titik yang akan ditentukan. metrik jarak ini digunakan dalam langkah yang disebutkan di atas (3) dari algoritma. Sebuah metrik jarak yang umum adalah Jarak Euclidean. Dalam kasus ini, fitur yang berbeda yang digunakan dalam vektor fitur memiliki nilai relatif berbeda dan rentang maka perhitungan jarak mungkin terdistorsi dan dapat ditingkatkan.

Parameter input dari algoritma clustering adalah jumlah cluster yang akan ditemukan bersama dengan nilai titik awal. Ketika nilai-nilai awal diberikan, jarak dari setiap titik sampel data ke masing-masing nilai awal awal ditemukan menggunakan persamaan. Kemudian setiap titik data ditempatkan di cluster yang terkait dengan titik awal terdekat. Setelah semua titik data ditugaskan untuk cluster, centroid cluster baru dihitung. Untuk setiap faktor dalam setiap cluster, nilai centroid baru kemudian dihitung. Centroid baru kemudian dianggap sebagai nilai awal awal baru dan langkah-langkah (3) dan (4) dari algoritma diulang. Proses ini berlanjut sampai tidak ada lagi perubahan data titik atau sampai centroid tidak lagi bergerak.

Dengan algoritma k-means, kita mampu mengelompokkan objek besar dan pencilan obyek dengan sangat cepat sehingga mempercepat proses pengelompokan. Akan tetapi walaupun algoritma ini bekerja dengan proses yang cepat, tetapi keakuratannya tidak dijamin. Seringkali algoritma ini mengalami konvergensi prematur. Pada algoritma ini juga tidak dijamin jarak antara masing-masing centroid tidak merentang. Misalnya, jika ada dua atau lebih kelompok dengan titik pusat massa yang berdekatan atau terdapat kelompok yang tidak memiliki anggota, maka hasilnya akan tidak memuaskan. Dimana algoritma ini sangat sensitive pada pembangkitan titik pusat awal secara random

### **3.9 Collaborative Filtering**

Collaborative Filtering adalah proses penyaringan atau mengevaluasi item menggunakan pendapat dari orang lain. Penyaringan ini dilakukan dengan

menggunakan profil. Teknik penyaringan kolaboratif mengumpulkan dan membangun profil, dan menentukan hubungan antar data sesuai dengan model kesamaan. Kategori berasal dari data di profil termasuk preferensi pengguna, pola perilaku pengguna, atau sifat barang (Mohd Abdul Hameed et al, 2013).

Collaborative Filtering (CF) adalah metode untuk mengidentifikasi klien yang sama dan merekomendasikan apa yang klien umum sukai. Sistem ini merekomendasikan item ke pengguna aktif atau target pengguna yang sama dengan pengguna lain dengan preferensi serupa di masa lalu. Kesamaan dalam preferensi dari dua pengguna dihitung / dievaluasi berdasarkan kesamaan dalam sejarah rating dari pengguna yang berbeda. Proses yang paling umum dari CF melakukan kesamaan perhitungan pada koleksi preferensi pengguna bahwa situs-situs E-Commerce biasanya dikumpulkan dari wisatawan terhadap produk. Perhitungan yang digunakan adalah teknik yang sama dalam rekomendasi dalam konten tapi fokus pada pendapat rekan. CF ini dianggap yang paling sukses, populer dan banyak diterapkan pada sistem rekomendasi (Kadam, N. & Kumar, S., 2016).

Secara umum proses pemberian rekomendasi terdiri atas tiga langkah, yaitu: penemuan *similar user*, pembuatan ketetanggan (*neighborhood*) dan perhitungan prediksi berdasarkan tetangga yang dipilih. *Collaborative filtering* menghasilkan prediksi atau rekomendasi bagi pengguna atau pelanggan yang dituju terhadap satu item atau lebih. Item dapat terdiri atas apa saja yang dapat disediakan manusia seperti misalnya buku, film, seni, artikel atau tujuan wisata. Rating dalam *collaborative filtering* dapat berbentuk, (a) model rating

skalar yang terdiri atas rating numerik seperti 1 sampai 5; (b) model rating biner dengan memilih antara setuju atau tidak setuju atau dapat pula baik atau buruk; (c) rating unary dapat mengindikasikan bahwa pengguna telah mengobservasi atau membeli item atau merating item dengan positif.

Tidak tersedianya rating mengindikasikan tidak terdapat informasi yang menghubungkan pengguna dengan item. Rating dapat dikumpulkan secara eksplisit, implisit, ataupun gabungan antara eksplisit dan implisit. Rating eksplisit yaitu rating yang didapatkan pada saat pelanggan/pengguna diminta menyediakan opini terhadap item tertentu. Rating implisit yaitu rating yang didapatkan melalui aksi yang dilakukan pelanggan

Collaborative Filtering (CF) dapat diklasifikasikan ke dalam kategori berikut sebagai berikut :

1. CF berbasis memori

Metode berbasis memori menggunakan pengguna ke pengguna dan konten untuk korelasi item berdasarkan perilaku wisatawan untuk memprediksi peringkat dan merekomendasikan item untuk pengguna di masa depan. Metode ini juga disebut sebagai collaborative filtering berbasis sekitar. Mekanisme ini menggunakan data dari wisatawan pengguna untuk menghitung kesamaan antara pengguna dan / atau barang yang digunakan untuk membuat rekomendasi. Mekanisme CF berbasis memori digunakan dalam banyak sistem komersial karena mudah diterapkan dan efektif.

2. CF berbasis model

Collaborative filtering berbasis model menggunakan informasi RS untuk membuat model yang menghasilkan rekomendasi. Tidak seperti CF berbasis memori, CF berbasis model ini tidak menggunakan seluruh dataset untuk menghitung prediksi untuk data real. Ada berbagai algoritma CF berbasis model termasuk Bayesian Networks, Model Clustering, dan Latent Model Semantic seperti Dekomposisi Nilai Singular (SVD), Principal Component Analysis (PCA) dan Probabilistik Matrix Faktorisasi untuk pengurangan dimensi dari matriks. Tujuan dari pendekatan ini adalah untuk mengungkap faktor laten yang menjelaskan penilaian yang diamati.

### 3. CF Hybrid

Untuk mengatasi kelemahan dari CF berbasis memori dan berbasis model seperti sparsity dan domba abu-abu ditangani oleh algoritma tersebut. CF hybrid adalah kombinasi pendekatan CF berbasis memori dan CF berbasis model. Hal ini ditujukan untuk meningkatkan kinerja prediksi algoritma CF. Metode Hybrid dapat memberikan lebih banyak rekomendasi akurat daripada pendekatan murni. Metode ini juga dapat digunakan untuk mengatasi beberapa masalah umum dalam sistem recommender seperti mulai dingin dan masalah sparsity.

Berikut adalah persamaan *adjusted cosine similarity* yang digunakan untuk menghitung nilai kemiripan diantara item.

Persamaan *adjusted cosine* :

$$sim(i,j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}} \dots \dots \dots (3)$$

Keterangan :

$sim(i,j)$  = Nilai kemiripan antara item i dan item j.

$u \in U$  = Himpunan *user* u yang merating *item* i dan *item* j.

$R(u,i)$  = Rating *user* u pada *item* i.

$R(u,j)$  = Rating *user* u pada *item* j.

$R_u$  = Nilai rata-rata rating user u.

Dalam menghitung nilai kemiripan, nilai yang akan dihasilkan oleh persamaan *adjusted-cosine similarity* adalah berkisar antara +1.0 dengan -1.0, sedangkan informasi korelasi antara dua item diketahui jika:

- a. Nilai kemiripan 0 : Kedua item tidak berkorelasi (independen).

Nilai kemiripan mendekati +1.0 : Kedua item cenderung akan mirip antara satu dengan yang lainnya, jadi apabila rating suatu item telah diketahui maka rating item yang lainnya dapat diketahui dan disimpulkan dengan probabilitas yang tinggi.

- b. Nilai kemiripan mendekati -1.0

Kedua item saling bertolak belakang dan dalam kasus ini juga rating suatu item bisa ditentukan berdasarkan rating dari item lainnya, tapi keadaannya sekarang apabila rating item pertama meningkat maka rating item kedua justru akan sebaliknya, yaitu menurun.

Tahapan selanjutnya yang paling penting dalam proses collaborative filtering adalah membuat prediksi. Setelah mendapatkan sekumpulan item yang

sangat mirip berdasarkan perhitungan kemiripan, dilakukan proses prediksi yang nantinya akan memperkirakan nilai rating dari user bagi suatu item yang belum pernah dirating sebelumnya oleh user tersebut.

Teknik yang digunakan untuk mendapatkan nilai prediksi adalah dengan persamaan *weighted sum*, teknik ini memprediksi item j untuk user u dengan menghitung jumlah nilai rating yang diberikan oleh user terhadap item yang berkorelasi dengan item j. Setiap rating yang diberikan pada item yang berkorelasi, akan dikalikan dengan nilai kemiripannya. Kemudian dibagi dengan jumlah nilai absolut kemiripan seluruh item yang berkorelasi.

Persamaan *weighted sum* :

$$P_{(u,j)} = \frac{\sum_{i \in I} (R_{u,i} * S_{i,j})}{\sum_{i \in I} |S_{i,j}|} \dots\dots\dots (4)$$

Keterangan :

$P(u,j)$  = Prediksi untuk user u pada item j.

$i \in I$  = Himpunan item yang mirip dengan item j.

$R(u,i)$  = Rating user u pada item i.

$S(i,j)$  = Nilai kemiripan antara item i dan item j.