

BAB 2

LANDASAN TEORI

2.1 Rekomendasi

Rekomendasi adalah suatu bentuk komunikasi sekaligus promosi yang dilakukan secara tidak langsung oleh konsumen atau pelanggan yang telah pernah membeli dan menikmati produk atau jasa yang ditawarkan dan kemudian menceritakan berbagai pengalamannya terkait dengan produk atau jasa tersebut kepada orang lain, sehingga dapat menarik minat beli bagi konsumen yang lain setelah mendengarkan pembicaraan tersebut (Sumali, 2014). Salah satu bentuk proses rekomendasi yang paling umum dipakai adalah rekomendasi dari mulut ke mulut atau yang biasa dikenal dengan *Word of Mouth Communication* (WOM) (K. Kotler Philip. dan Keller, 2007).

2.2 Produk

Produk adalah elemen yang penting bagi pelanggan atau konsumen dalam melakukan transaksi pembelian terhadap suatu barang (Rachim dan Setiawan, 2014). Produk merupakan elemen penting dari kumpulan paduan pemasaran dan harus dipahami dan dikelola dengan baik agar dapat memenuhi kebutuhan pelanggan sehingga tercapainya tujuan organisasi (Akpoyomare, Adeosun, dan Ganiyu, 2013).

Segala sesuatu yang ditawarkan ke pasar untuk mendapatkan perhatian, yang kemudian dibeli, dipergunakan dan sehingga dapat memuaskan keinginan atau kebutuhan disebut dengan produk (P. Kotler dan Armstrong, 2001). Baik yang dapat diraba atau nyata maupun yang tidak dapat diraba atau jasa layanan (Tengor, 2016) termasuk didalamnya warna, kemasan, kualitas, harga dan merek beserta jasa dan reputasi penjualannya (Habibah dan Sumiati, 2016).

Perusahaan tidak akan dapat melakukan apapun dari usahanya tanpa adanya produk, begitulah artian penting produk bagi perusahaan. Produk akan dibeli oleh pembeli jika dirasa cocok, oleh karena itu produk yang dihasilkan harus disesuaikan dengan minat, keinginan ataupun kebutuhan pembeli agar pemasaran produk tepat pada sasaran. Dengan kata lain, produk lebih baik dibuat berorientasikan keinginan pasar atau selera konsumen (Riyono dan Budiharja, 2016).

Produk dianggap unsur yang paling penting, karena mencakup seluruh perencanaan yang mendahului produksi aktual, riset dan pengembangan, dan semua layanan yang menyertai produk seperti instalisasi dan pemeliharaan (Suatma, 2013).

Produk yang ditawarkan harus memiliki kualitas yang baik, unik dan penampilan yang menarik sehingga dapat menambah ketertarikan konsumen untuk membeli produk tersebut (Zulaicha dan Irawati, 2016). Produk merupakan tolak ukur keberhasilan pemasaran. Produk yang baik mempunyai peluang lebih besar untuk direspon dengan baik oleh konsumen.

Produk dibagi menjadi lima tingkatan yaitu (Indriani, 2006):

1. Produk inti, yang menawarkan manfaat atau kegunaan utama yang dibutuhkan pelanggan.
2. Produk generik, yang mencerminkan versi dasar (fungsional) dari suatu produk.
3. Produk yang diharapkan, yaitu sekumpulan atribut dan kondisi yang diharapkan pelanggan pada saat membeli.
4. Produk tambahan, yaitu memberikan jasa dan manfaat tambahan sehingga memberikan perbedaan terhadap produk yang ditawarkan oleh perusahaan pesaing.
5. Produk potensial, yaitu penambahan dan transformasi pada produk yang mungkin dilakukan di masa yang akan datang.

2.3 Pelanggan

Siapa saja yang melakukan pembelian dan keputusan untuk melakukan transaksi dalam bentuk barang ataupun jasa pada suatu toko atau perusahaan disebut dengan pelanggan (Shih, Yu, dan Tseng, 2015).

Perusahaan memang seharusnya berorientasi pada keinginan pelanggan, agar memiliki nilai kompetitif saat berhadapan dengan pesaing. Dengan memberikan kepuasan pada pelanggan akan membangun loyalitas pelanggan dan sehingga dapat menciptakan hubungan yang erat antara pelanggan dan perusahaan (Rabiqy, 2017).

Pelanggan dikelompokkan menjadi 2, yaitu pelanggan internal dan pelanggan eksternal. Pelanggan Internal adalah orang-orang, departemen ataupun organisasi yang dilayani oleh apa yang kita lakukan, contohnya karyawan. Beberapa perusahaan mengukur keefektifan pelayanan pelanggan dengan mengukur kepuasan karyawan. Jika karyawan puas dan terlibat dalam pekerjaan-pekerjaan yang positif maka secara otomatis dapat diindikasikan bahwa pelanggan dapat dilayani dengan baik. Satu-satunya yang tidak memiliki pelanggan internal adalah seseorang yang benar-benar bekerja sendirian dan untuk dirinya sendiri (Harnoto, 2014).

Para pelanggan eksternal adalah orang atau departemen yang merupakan pengguna akhir dari produk atau jasa yang kita tawarkan. Pelanggan eksternal memiliki istilah lainnya, seperti konsumen, pelanggan, klien, tamu, pasien, patron,

kasus, waralaba, penumpang, siswa dan lain sebagainya (Harnoto, 2014).

2.4 Analisis Keranjang Pasar

Analisis Keranjang Pasar (*Market Basket Analysis*), atau juga dikenal sebagai *association rule mining* atau analisis afinitas, adalah salah satu teknik dalam *data mining* (Berry dan Linoff, 2004) yang digunakan untuk mengidentifikasi hubungan antara kelompok produk, *item*, atau kategori. Analisis keranjang belanja awalnya berasal dari bidang pemasaran. Analisis Keranjang Pasar memudahkan peneliti untuk memahami perilaku pelanggan melalui produk mana yang dibeli secara bersamaan (Aguinis, Forcum, dan Joo, 2012).

Analisis keranjang pasar berfokus untuk menemukan pola pembelian dengan mengekstraksi asosiasi atau kejadian yang terjadi secara bersamaan dari data transaksi. Metode ini biasa digunakan untuk menentukan produk yang dibeli secara bersama dan untuk mengatur tata letak produk *supermarket*, serta untuk merancang strategi promosi sehingga dapat meningkatkan minat pelanggan dalam berbelanja (Annie dan Kumar, 2012) dan memperluas strategi pemasaran (Gupta dan Mamtara, 2014).

2.5 Data Mining

Data mining adalah proses untuk menemukan informasi yang bermanfaat yang tersembunyi dalam data dengan jumlah yang besar, serta membantu dalam pengambilan keputusan bagi *stakeholder* dalam membuat keputusan, guna mengembangkan bisnisnya. *Data mining* saat ini sudah banyak diterapkan di berbagai bidang, seperti penelitian, akademisi dan industri. Teknologi *data mining* menggabungkan kecerdasan buatan (*Artificial Intelligence*), mesin pembelajaran (*Machine Learning*), pengenalan pola (*Pattern Recognition*), teknik visualisasi, statistik (Liu, Zhang, Li, dan Sun, 2014) dan sistem manajemen basis data (Diwate dan Sahu, 2014).

Data mining merupakan teknologi yang relatif baru dan belum sepenuhnya matang. Namun demikian, beberapa bidang industri sudah menggunakan *data mining* secara teratur. Seperti toko ritel, rumah sakit, bank, dan perusahaan asuransi. Banyak dari organisasi ini menggabungkan *data mining* dengan hal-hal seperti statistik, pengenalan pola, dan *tools* lainnya. *Data mining* dapat digunakan untuk menemukan pola dan hubungan yang dapat dikatakan sulit untuk ditemukan. Teknologi ini populer karena dapat memungkinkan untuk mempelajari lebih banyak tentang pelanggan (Ramageri, 2010).

Data mining merupakan gabungan dari beberapa disiplin ilmu yang bertujuan untuk memperbaiki teknik yang bisa dikatakan jadul atau tradisional sehingga

dapat menangani (Fajrin dan Maulana, 2018):

1. Data dengan jumlah yang sangat besar
2. Data dengan dimensi yang tinggi
3. Data yang heterogen dan berbeda sifat

2.5.1 Pengelompokan *Data Mining*

Ada beberapa teknik utama dalam *Data Mining* yang berhasil dikembangkan dan digunakan dalam penelitian penelitian yang telah dilakukan, seperti klasifikasi, pengklasteran atau pengelompokan, regresi, asosiasi, jaringan syaraf, dll, untuk menemukan pengetahuan dari *database* (Ramageri, 2010).

1. Klasifikasi (*Classification*)

Klasifikasi adalah proses pencarian suatu himpunan fungsi (model) yang dapat mendeskripsi dan membedakan kelas-kelas data atau konsep-konsep, yang bertujuan untuk menggunakan model tersebut guna memprediksi kelas dari suatu objek yang mana kelasnya belum diketahui (L. N. Rani, 2016).

Klasifikasi adalah teknik *data mining* yang paling umum diterapkan, Pendekatan ini sering menggunakan pohon keputusan atau algoritma klasifikasi berdasarkan jaringan saraf. Proses klasifikasi data melibatkan pembelajaran dan klasifikasi. Dalam Pembelajaran data latih dianalisis dengan menggunakan algoritma klasifikasi. Dalam data uji algoritma klasifikasi digunakan untuk memperkirakan akurasi dari aturan atau *rule* klasifikasi. Jika akurasi dapat diterima, aturan dapat diterapkan ke tupel data yang baru.

Jenis model klasifikasi:

- (a) *Classification by decision tree induction.*
- (b) *Bayesian Classification.*
- (c) *Neural Networks.*
- (d) *Support Vector Machines (SVM).*
- (e) *Classification Based on Associations.*

Contoh penggunaan pendekatan klasifikasi ialah:

- (a) Penggolongan pendapatan.
- (b) Pendeteksian penipuan dan aplikasi resiko kredit.
- (c) Dan lain sebagainya.

2. Pengklasteran (*Clustering*)

Clustering atau pengklasteran dapat dikatakan sebagai identifikasi kelas objek yang serupa sehingga data dikelompokkan berdasarkan kemiripan antar data ditiap kelompok dan berbeda dengan kelompok lainnya. Dengan menggunakan teknik *clustering*, kita dapat mengidentifikasi lebih lanjut da-

erah padat dan jarang di ruang objek dan dapat menemukan pola distribusi secara keseluruhan dan korelasi antar atribut data. Pendekatan klasifikasi juga dapat digunakan untuk cara yang efektif membedakan kelompok atau kelas objek tetapi menjadi merugikan sehingga *clustering* dapat digunakan sebagai pendekatan *preprocessing* untuk pemilihan dan klasifikasi *subset* atribut.

Jenis metode pengelompokan:

- (a) Metode *Partitioning*.
- (b) Metode *Hierarchical Agglomerative (divisive)*.
- (c) Metode *Density based*.
- (d) Metode *Grid-based*.
- (e) Metode *Model-based*.

Contoh penggunaan teknik *clustering* ialah:

- (a) Membentuk kelompok pelanggan berdasarkan pola pembelian.
- (b) Mengelompokkan gen dengan fungsi serupa.
- (c) Dan lain sebagainya.

3. Regresi (*Regression*)

Teknik regresi dapat diadaptasi untuk prediksi. Analisis regresi dapat digunakan untuk memodelkan hubungan antara satu atau lebih variabel independen dan variabel dependen. Dalam *data mining*, variabel independen adalah atribut yang sudah diketahui dan variabel respon adalah apa yang ingin kita prediksi. Sayangnya, banyak masalah di dunia nyata bukan sekadar prediksi. Misalnya, volume penjualan, harga saham, dan tingkat kegagalan produk semuanya sangat sulit untuk diprediksi karena mungkin bergantung pada interaksi kompleks dari beberapa variabel prediktor. Oleh karena itu, teknik yang lebih rumit (misalnya, regresi logistik, pohon keputusan, atau jaringan saraf) mungkin diperlukan untuk meramalkan nilai masa depan. Tipe model yang sama sering dapat digunakan untuk regresi dan klasifikasi. Sebagai contoh, algoritma pohon keputusan (*classification and regression trees*) dapat digunakan untuk membangun kedua pohon klasifikasi (untuk mengklasifikasikan variabel respon kategori) dan pohon regresi (untuk memperkirakan variabel respon berkelanjutan). Jaringan syaraf juga dapat membuat model klasifikasi dan regresi.

Jenis metode regresi:

- (a) Regresi *Linear*.
- (b) Regresi *Linear Multivariat*.
- (c) Regresi *Nonlinear*.

(d) Regresi *Nonlinier Multivariat*.

4. Asosiasi (*Association Rule*)

Asosiasi biasanya digunakan untuk menemukan temuan *item set* yang sering muncul di antara *set* data besar. Jenis temuan ini membantu bisnis untuk membuat keputusan tertentu. Algoritma *Association Rule* harus mampu menghasilkan aturan dengan nilai-nilai *confidence* kurang dari satu.

Jenis aturan asosiasi:

- (a) Aturan asosiasi multilevel.
- (b) Aturan asosiasi multidimensi.
- (c) Aturan asosiasi kuantitatif.

Contoh penggunaan teknik asosiasi ialah:

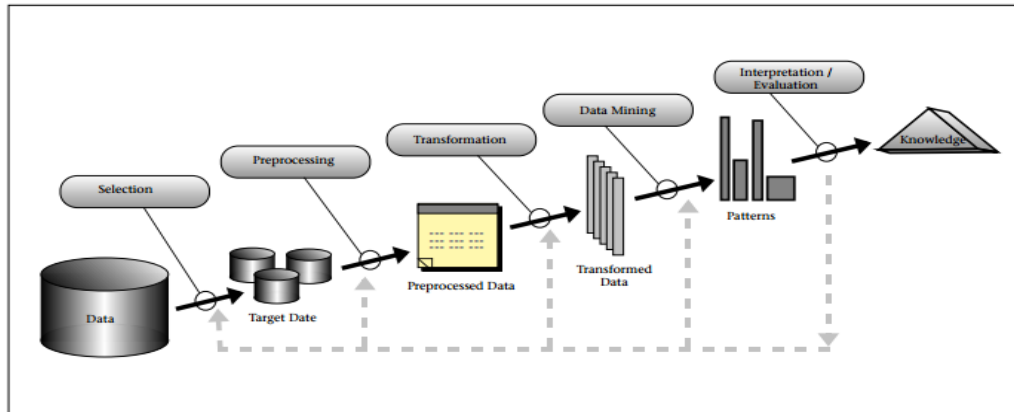
- (a) Desain katalog belanja.
- (b) Analisis pemasaran.
- (c) Perilaku belanja pelanggan.

5. Jaringan Syaraf (*Neural Networks*) Jaringan syaraf adalah sekumpulan unit *input/output* yang terhubung dan setiap sambungan memiliki bobot yang ada bersamanya. Selama fase pembelajaran, jaringan belajar dengan menyesuaikan bobot sehingga dapat memprediksi label kelas yang benar dari masukan tupel. Jaringan syaraf memiliki kemampuan luar biasa untuk memperoleh makna dari data yang rumit atau tidak tepat dan dapat digunakan untuk mengekstrak pola dan mendeteksi tren yang terlalu rumit untuk diperhatikan oleh manusia atau teknik komputer lainnya. Ini sangat cocok untuk *input* dan *output* bernilai berkelanjutan. Misalnya *re-organisasi* karakter tulisan tangan, untuk melatih komputer untuk mengucapkan teks bahasa Inggris dan banyak masalah bisnis dunia nyata dan telah berhasil diterapkan di banyak industri. Jaringan syaraf paling baik dalam mengidentifikasi pola-pola dalam data dan sangat cocok untuk prediksi atau perkiraan kebutuhan. Jenis jaringan saraf seperti *Back Propagation*.

2.5.2 Proses *Data Mining*

Banyak yang menggunakan istilah *Knowledge Discovery in Database* (KDD) dan *data mining* untuk menjelaskan proses penggalian informasi yang tersembunyi dalam suatu *database* yang besar. Padahal, kedua istilah tersebut memiliki konsep yang berbeda, tetapi saling berkaitan antara satu dan yang lainnya, dan *data mining* merupakan salah satu tahapan dalam keseluruhan proses KDD.

Proses dari KDD dapat dilihat pada Gambar 2.1.



Gambar 2.1. *Data mining* dan proses KDD (Fayyad dkk., 1996)

Proses KDD secara garis besar dapat dijelaskan sebagai berikut (Mardi, 2017):

1. *Data Selection.*

Pemilihan (seleksi) data dari sekumpulan data operasional dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data dari hasil seleksi akan digunakan dalam proses *data mining*, dan disimpan dalam suatu berkas yang terpisah dari *database* operasional.

2. *Pre-processing / Cleaning.*

Sebelum proses *data mining* dilakukan, ada satu tahap lagi yang perlu dilakukan yaitu proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* ini seperti membuang duplikasi data, memeriksa data yang tidak konsisten/*inkonsisten*, dan memperbaiki kesalahan yang ada pada data, seperti kesalahan cetak. Selain itu, proses *enrichment* juga dilakukan, yaitu proses “memperkaya” data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal lainnya yang dibutuhkan.

3. *Transformation .*

Coding adalah proses transformasi data yang telah dipilih, sehingga data tersebut sudah sesuai untuk proses *data mining*. Proses *coding* pada KDD merupakan sebuah proses yang sangat tergantung pada jenis atau pola informasi seperti apa yang akan dicari dalam *database*.

4. *Data Mining.*

Data mining adalah proses mencari pola atau informasi menarik dalam data yang telah terpilih dengan menggunakan teknik atau metode tertentu. Teknik-teknik, metode-metode, atau algoritma dalam *data mining* sangat banyak jenis dan fungsinya. Pemilihan metode atau algoritma sangat

bergantung pada tujuan dan proses KDD secara keseluruhan.

5. *Interpretation / Evaluation.*

Pola informasi yang dihasilkan dari proses *data mining* disajikan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan atau *stakeholder*. Tahap ini merupakan proses KDD yang disebut *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang sudah ada sebelumnya. Bila ternyata hasil yang diperoleh tidak sesuai hipotesa, ada beberapa alternatif yang dapat diambil seperti menjadikannya *feedback* untuk memperbaiki proses *data mining*, mencoba teknik *data mining* lain yang lebih sesuai atau menerima hasil ini sebagai suatu hasil yang di luar dugaan yang mungkin bermanfaat.

2.6 Clustering

Clustering adalah suatu proses pengelompokan kumpulan data ke dalam kelompok-kelompok sehingga objek-objek yang terdapat dalam suatu kelompok memiliki kesamaan yang tinggi dan berbeda dengan objek yang berada dalam kelompok lain (Ghuman, 2016).

Clustering adalah bagian penting dari analisis data, yang bertujuan untuk mempartisi *dataset* yang diberikan ke dalam beberapa *cluster* yang mana *dataset* yang berada dalam suatu *cluster* memiliki kemiripan dan berbeda dari *dataset* yang berada di *cluster* lain (Raval dan Jani, 2016; Y. Rani dan Rohil, 2013). *Clustering* berakar dari berbagai bidang seperti *data mining*, statistik, biologi dan *machine learning* (Patel, Modi, dan Sarvakar, 2014).

Metode ini adalah metode yang tidak diawasi dari aplikasi *machine learning*. Dalam *clustering*, pengelompokan dibagi menurut variabel kelas. Dua hal yang harus diperhatikan, yaitu, perbedaan cara dalam mengelompokkan satu *set* objek ke dalam *set cluster*. Kemudian, jenis *cluster*. Analisis *cluster* menghasilkan sejumlah kelompok heterogen dengan konten yang homogen (Thakare, Mohod, dan Thakare, 2016).

Berdasarkan strategi bagaimana data dibedakan, teknik *clustering* dapat dibagi menjadi 5 teknik, seperti yang telah dijelaskan oleh Madhulatha (2012) yaitu: *hierarchical*, *partitional*, *density-based*, *grid-based* dan *model-based*.

2.6.1 Hierarchical Clustering

Hierarchical clustering adalah metode analisis klaster yang digunakan untuk membangun hierarki klaster. Dalam *data mining*, cara kerja *hierarchical clustering* ialah dengan mengelompokkan objek-objek data ke dalam pohon *cluster* (Y. Rani dan Rohil, 2013), yang mana struktur pohon *cluster* ini biasa disebut

dendogram (Rafsanjani, Varzaneh, dan Chukanlo, 2012) yang biasa disajikan sebagai hasil dari *Hierarchical clustering* (Sasirekha dan Baby, 2013). Pohon ini menampilkan proses penggabungan dan gugus menengah secara grafis. Struktur grafis ini menunjukkan bagaimana titik dapat digabung menjadi satu kelompok (Y. Rani dan Rohil, 2013).

Dendogram mewakili urutan dari sarang *cluster* yang dibangun dari atas ke bawah atau dari bawah ke atas. Akar pohon melambangkan satu *cluster*, berisi semua titik data, sementara pada daun pohon, ada n *cluster*, masing-masing berisi satu titik data. Dengan memotong pohon pada tingkat yang diinginkan, *clustering* dari titik data diuraikan ke dalam grup yang diperoleh (Pham dan Afify, 2007).

Metode hierarkis dapat diklasifikasikan berdasarkan bagaimana penguraian hierarkis terbentuk. Ada dua pendekatan yaitu pendekatan *Agglomerative* dan pendekatan *Divisive* (Reddy, Vivekananda, dan Satish, 2017):

1. *Agglomerative hierarchical clustering*.

Pendekatan aglomeratif secara populer dikenal sebagai pendekatan *bottom-up* karena dalam hal ini, satu dimulai dengan setiap objek membentuk kelompok terpisah. Itu terus menggabungkan benda-benda atau kelompok yang dekat satu sama lain. Itu terus melakukannya sampai semua kelompok digabung menjadi satu atau sampai kondisi penghentian berlaku. Contoh klasik dari ini adalah taksonomi spesies. Data ekspresi gen mungkin juga menunjukkan kualitas hierarkis yang sama. Aglomeratif pengelompokan hierarkis dimulai dengan setiap objek tunggal atau sampel dalam satu kelompok, kemudian di setiap iterasi yang berurutan, menggumpalkan pasangan terdekat gugus dengan memenuhi beberapa kriteria kemiripan, kecuali semua data dalam satu kelompok

2. *Divisive hierarchical clustering*.

Divisive hierarchical clustering adalah pendekatan *top-down* yang dimulai dengan satu *cluster* dan membagi *cluster* menjadi dua kelompok berbeda secara rekursif sampai kondisi tertentu terpenuhi (Yildirim dan Birant, 2017). Metode pengelompokan *top-down* bekerja dengan cara yang mirip dengan pengelompokan *agglomerative* tetapi dalam arah yang berlawanan (Sasirekha dan Baby, 2013).

2.6.2 *Partitional Clustering*

Di antara banyaknya jenis *clustering* yang ada, *clustering* partisi adalah yang paling banyak digunakan karena perhitungan yang digunakan tidak terlalu rumit (Aparna dan Nair, 2014).

Algoritma *Partitional Clustering* membagi titik data ke dalam partisi k , di mana setiap partisi mewakili sebuah *cluster*. Partisi dilakukan berdasarkan fungsi obyektif tertentu (Elavarasi, Akilandeswari, dan Sathiyabhama, 2011) dan $k \leq n$, di mana n adalah jumlah titik data. Metode partisi didasarkan pada gagasan bahwa klaster dapat diwakili oleh titik pusat (Chitra dan Maheswari, 2017).

Cluster harus menunjukkan dua properti sebagai syarat dalam mengklasifikasikan data ke dalam kelompok k , yaitu:

1. Setiap kelompok harus mengandung setidaknya satu objek.
2. Setiap objek harus milik tepat satu kelompok (Selvakumar, 2013).

Kelemahan dari algoritma ini adalah setiap jarak antara dua titik dari pusat berdekatan dengan kelompok lain, hasilnya akan menjadi buruk atau tidak akurat karena saling tumpang tindih dari titik data (Saket dan Pandya, 2016).

2.6.3 *Density-Based Clustering*

Algoritma *Density-Based Clustering* adalah algoritma *clustering* utama lainnya yang telah lama diusulkan (Ester, Kriegel, dan Xu, 1996). Algoritma Ini dapat menemukan *cluster* yang berbentuk acak dan dapat menangani *noise*, dan juga merupakan algoritma *satu-scan* yang mana hanya melakukan satu kali *scanning* ketika memeriksa data mentah. Dalam algoritma *Density-Based Clustering*, area padat objek dalam ruang data dianggap sebagai kelompok, yang dipisahkan oleh daerah dengan kepadatan rendah (*noise*). Oleh karena itu, metode *Density-Based Clustering* merupakan algoritma pengelompokan dasar yang menarik untuk aliran data (Amini, Wah, Saybani, dan Yazdi, 2011).

Dalam Metode *Density-Based Clustering*, dalam mempartisi objek *cluster* didasarkan pada jarak antar objek. Metode ini dapat menemukan gugus yang berbentuk acak. Metode ini dapat digunakan untuk menyaring *noise* atau *outlier* (S. Shah dan Singh, 2012).

Algoritma *Density-Based Clustering* adalah algoritma yang terkenal dikarenakan penjelasannya yang mudah dan oleh karena itu pengimplementasiannya pun relatif mudah. Dua kelebihan lainnya dari algoritma ini adalah kemampuannya untuk menemukan *cluster* dalam berbagai bentuk dan ukuran yang berbeda bahkan dalam kumpulan data *outlier* dan pengguna tidak perlu untuk menentukan jumlah *cluster* (Nidhi dan Patel, 2016). Contoh algoritma *Density-Based Clustering* ialah DBSCAN, OPTICS, LDBSCAN, dll (Kisilevich, Mansmann, dan Keim, 2010).

Algoritma ini mengaitkan setiap objek dengan nilai kepadatan yang ditentukan oleh jumlah objek tetangganya dalam radius tertentu. Suatu objek yang kepadatannya lebih besar dari ambang batas yang telah ditentukan didefinisikan sebagai

objek padat dan membuat gugusnya sendiri. Kelompok-kelompok digabungkan jika saling berbagi terhadap tetangga yang sama yang juga padat (Liao, Liu, dan Choudhary, 2004).

2.6.4 *Grid-Based Clustering*

Algoritma *Grid-Based Clustering* adalah tipe yang paling penting dalam algoritma *clustering* hirarkis. Pendekatan *Grid-Based Clustering* lebih mengangap sel dibandingkan titik data. Karena sifat algoritma *Grid-Based Clustering* ini umumnya lebih efisien secara komputasi apabila dibandingkan dengan semua jenis algoritma *clustering*. Bahkan, sebagian besar algoritma *Grid-Based Clustering* mencapai kompleksitas waktu di mana n adalah jumlah objek data. Sehingga memungkinkan semua operasi *clustering* untuk tampil di ruang data *grid* (N. A. Shah dan Paul, 2017).

Algoritma berbasis *grid* menggunakan struktur data *grid* multiresolusi. Kerumitan yang muncul tergantung pada jumlah sel *grid* terpopulasi bukan berdasarkan pada jumlah objek dalam *dataset* (Suman dan Rani, 2017).

Beberapa metode *Grid-Based Clustering* yang populer adalah:

1. *A Statistical Information Grid Approach to Spatial Data Mining* (STING).
2. *Clustering In Quest* (CLIQUE).
3. *WaveCluster*.

Semua metode ini menggunakan jaring-jaring yang seragam untuk menutupi seluruh masalah. Untuk masalah yang berhubungan dengan distribusi data yang sangat tidak teratur, resolusi dari *grid mesh* harus cukup baik untuk mendapatkan kualitas pengelompokan yang baik. Jaring yang lebih halus dapat menghasilkan ukuran *mesh* mendekati atau bahkan melebihi ukuran objek data, yang dapat signifikan meningkatkan beban komputasi dalam proses *clustering*. Karakteristik umum dari semua metode berbasis *grid*, seperti waktu pemrosesan cepat, kepekaan terhadap urutan *input* data, dan kemampuan untuk memisahkan dari *noise* (Liao dkk., 2004).

Langkah-langkah dasar Algoritma *Grid-Based Clustering* adalah sebagai berikut:

1. Definisikan satu set sel-sel *grid*.
2. Tetapkan objek ke sel *grid* yang sesuai dan hitung kepadatan setiap sel.
3. Hilangkan sel, yang kerapatannya di bawah ambang batas tertentu.
4. Membentuk klaster dari kelompok sel padat yang berdekatan (Parikh dan Varma, 2014).

2.6.5 *Model-Based Clustering*

Algoritma *Model-Based Clustering* adalah algoritma yang didasarkan pada konsep bahwa data yang diamati berasal dari populasi dengan beberapa sub-populasi. Konsep umumnya adalah masing-masing sub-populasi dimodelkan secara terpisah dan keseluruhan populasi sebagai campuran dari sub-populasi ini, dan menggunakan model campuran terbatas (Raftery dan Dean, 2006).

2.7 *Algoritma K-Medoids*

K-Medoids atau *Partitioning Around Medoids* (PAM) adalah algoritma *clustering* yang mirip dengan *K-Means*. Perbedaan dari kedua algoritma ini yaitu pada algoritma K-Medoids atau PAM objek digunakan sebagai perwakilan (*medoid*) sebagai pusat dari *cluster* untuk setiap *cluster*, sedangkan *K-Means* yang menjadi pusat *cluster* ialah nilai rata-rata (*mean*) (Kaur, Kaur, dan Singh, 2014) atau lebih mudahnya, pada algoritma K-Medoids objek yang menjadi perwakilan yang disebut *medoid* dianggap sebagai pengganti *centroid* (pusat dari *cluster*). Karena menggunakan objek yang paling terpusat dalam sebuah *cluster*, K-Medoids dianggap kurang sensitif terhadap *outlier* jika dibandingkan dengan algoritma *K-Means* (Park, Lee, dan Jun, 2014).

Perbedaan metode *K-Means* dengan metode K-Medoids yaitu metode K-Medoids tidak menentukan nilai rata-rata dari objek yang ada dalam *cluster* sebagai titik acuan, tetapi menggunakan *medoid* (*median*), yang merupakan objek yang paling terpusat suatu *cluster*. Dengan demikian, metode partisi masih tetap dapat dilakukan berdasarkan prinsip jumlah dari ketidaksamaan antara setiap objek dan titik acuan yang sesuai (*medoid*) di minimalisir yang mana hal tersebut merupakan dasar dari metode K-Medoids (Defiyanti, Jajuli, dan Rohmawati, 2017).

Menemukan *cluster k* pada objek *n* dengan terlebih dahulu menemukan objek awal (*medoid*) secara acak sebagai perwakilan untuk setiap *cluster* adalah strategi dasar dari algoritma K-Medoids. Kemudian, setiap objek yang tersisa dikelompokkan berdasarkan *medoid* yang memiliki tingkat kemiripan yang tinggi. Pada metode ini, objek representatif digunakan sebagai titik yang menjadi referensi dan kunci dari metode ini adalah tidak rata-ratanya objek dari setiap *cluster*. K-Medoids akan mengambil parameter masukan *k* jumlah *cluster* yang akan di partisi antara himpunan jumlah *n* objek (Defiyanti dkk., 2017).

Langkah-langkah dari algoritma K-Medoids yaitu (Pramesti, Furqon, dan Dewi, 2017):

1. Inisialisasi pusat *cluster* sebanyak *k* (jumlah *cluster*).
2. Alokasikan setiap data (objek) ke *cluster* terdekat menggunakan persamaan

ukuran jarak *Euclidian Distance* dengan Persamaan 2.1.

$$d(x,y) = \|x - y\| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad ; 1, 2, 3, \dots, n \quad (2.1)$$

3. Pilih secara acak objek pada masing-masing *cluster* sebagai kandidat *medoid* baru.
4. Hitung jarak setiap objek yang berada pada masing-masing *cluster* dengan kandidat *medoid* baru.
5. Hitung total simpangan (S) dengan menghitung nilai total *distance* baru – total *distance* lama. Jika $S < 0$, maka tukar objek dengan data *cluster* untuk membentuk sekumpulan k objek baru sebagai *medoid*.
6. Ulangi langkah 3 sampai 5 hingga tidak terjadi perubahan *medoid*, sehingga didapatkan *cluster* beserta anggota *cluster* masing-masing.

K-Medoids juga merupakan teknik *clustering* partisi yang mengelompokkan kumpulan data dari n objek menjadi k *cluster* dengan k diketahui sebagai prioritas. K-Medoids lebih kuat dibandingkan dengan K-Means karena K-Medoids meminimalkan jumlah ketidaksamaan objek data bukan jumlah jarak *Euclidean* kuadrat untuk objek data. Dan metrik jarak ini mengurangi *noise* dan *outliers* (S. Shah dan Singh, 2012). Adapun kelemahan dari algoritma K-Means (Arora, Deepali, dan Varshney, 2016):

1. Untuk menemukan *K-Value* adalah tugas yang sulit.
2. Tidak efektif bila digunakan dengan *cluster* global.
3. Jika partisi awal yang berbeda telah dipilih dapat memvariasikan hasil dari *cluster*.
4. Perbedaan ukuran dan kerapatan *cluster* tidak ditangani oleh algoritma.

2.8 Asosiasi (*Association Rules*)

Aturan asosiasi (*Association Rules*) adalah salah satu teknik *data mining* yang sering digunakan. Proses *data mining* dengan menggunakan aturan asosiasi yaitu menemukan hubungan asosiatif antara data dan menyimpulkan *rules* yang terbentuk dari data-data yang ada. Contohnya, seorang investor bertanya “apakah nilai *return* perusahaan berhubungan dengan volume permintaan ataupun volume penawaran?”. Menemukan pola-pola, asosiasi, hubungan antara data dan menemukan fakta yang sebelumnya tidak diketahui atau bahkan tidak diperhatikan oleh sebuah perusahaan adalah fungsi aturan asosiasi (Azhari dan Anshori, 2009).

Aturan asosiasi adalah teknik *data mining* untuk menemukan aturan asosiatif antara kombinasi *item*. Untuk mengetahui kepentingan suatu aturan asosiasi

dapat diketahui dengan dua parameter, yaitu *support* dan *confidence*. *Support* yaitu persentase kombinasi *item* tersebut dalam *database*. Sedangkan *confidence* yaitu kuatnya hubungan antara *item* dalam *association rule* (Azhari dan Anshori, 2009).

Algoritma apriori adalah algoritma yang paling populer dalam *Association Rules* yang memiliki paradigma *generate and test*, yaitu kandidat kombinasi *item* yang mungkin akan dibentuk berdasarkan aturan tertentu, kemudian kombinasi *item* tersebut diuji apakah sudah memenuhi syarat *minimum support* yang telah ditentukan. *Frequent item set* atau *large item set* adalah sebutan kombinasi *item* yang telah memenuhi syarat tersebut, kemudian digunakan untuk membuat aturan-aturan (*rules*) yang memenuhi syarat *minimum confidence*. Contoh algoritma yang lain dari aturan asosiasi adalah algoritma *predictive apriori* dan tertius (Azhari dan Anshori, 2009).

Aturan asosiasi dapat dihasilkan dalam dua langkah proses. Pertama, *minimum support* digunakan untuk menghasilkan himpunan semua *itemset* yang sering digunakan dalam kumpulan data dalam suatu *database*. *Itemset* yang sering digunakan dapat diketahui dengan melihat apakah *itemset-itemset* tersebut telah memenuhi batasan *minimum support*. Kemudian, langkah kedua yaitu setiap *itemset* yang sering digunakan atau *frequent itemset* tersebut digunakan untuk menghasilkan semua kemungkinan aturan (*rules*) yang terbentuk dari kandidat *itemset* tersebut berdasarkan *minimum confidence* yang telah ditentukan, dan semua aturan (*rules*) yang tidak memenuhi batasan *minimum confidence* dieliminasi atau dihilangkan. Peningkatan nilai *minimum support* biasanya digunakan untuk menjaga agar jumlah aturan asosiasi (*Association rules*) yang ditemukan pada taraf ukuran yang dapat dikelola. Namun, hal tersebut juga dapat berpotensi menghilangkan aturan (*rules*) yang menarik dengan nilai *support* yang lebih kecil (Hahsler, 2017).

Dalam aturan asosiasi digunakan istilah *antedecent* untuk mewakili bagian “jika” dan *consequent* untuk mewakili bagian “maka” (Listriani, Setyaningrum, dan Eka, 2016).

Untuk memulai pembentukan aturan asosiasi yaitu dimulai dengan mencari *frequent itemset* yang merupakan kombinasi paling umum dalam *itemset* dan harus memenuhi syarat *minimal support* (minsup). Untuk mendapatkan nilai dukungan *item A* dapat menggunakan Persamaan 2.2 (Wijaya dan Fitriana, 2018).

$$Support(A) = \frac{JumlahTransaksiyangmengandungItemA}{TotalTransaksi} \times 100\% \quad (2.2)$$

Kemudian, untuk mendapatkan nilai *support* dua *item* dapat digunakan Persamaan 2.3.

$$Support(A, B) = P(A \cap B) = \frac{JumlahTransaksiyangmengandungItemAdanB}{TotalTransaksi} \times 100\% \quad (2.3)$$

Setelah semua *item* yang sering muncul dan *itemset* besar diperoleh, langkah selanjutnya yaitu menemukan kondisi *minimum confidence* (minconf) dengan Persamaan 2.4.

$$Confidence(A \rightarrow B) = P(A|B) = \frac{JumlahTransaksiyangmengandungItemAdanB}{JumlahTransaksiyangmengandungItemA} \times 100\% \quad (2.4)$$

Adapun untuk mencari nilai *lift ratio* dapat digunakan Persamaan 2.5.

$$Lift\ Ratio = \frac{Confidence(A, B)}{NilaiPatokanConfidence(A, B)} \quad (2.5)$$

Untuk mendapatkan patokan nilai *confidence* itu sendiri dapat dihitung menggunakan Persamaan 2.6.

$$Benchmark\ Confidence = \frac{Confidence(A, B)}{Benchmark\ Confidence(A, B)} \quad (2.6)$$

Menurut Anggraeni, Saputra, dan Noranita (2013) langkah-langkah dalam pembentukan *rules* asosiasi yaitu:

1. Analisis pola frekuensi tinggi.

Tahap ini mencari kombinasi *item* yang memenuhi dari syarat nilai *minimum support* dalam *database*. *Support* untuk aturan “X → Y” adalah probabilitas atribut atau kumpulan atribut X dan Y yang terjadi bersamaan dalam suatu transaksi, dapat dilihat pada Persamaan 2.7.

$$Support(X \rightarrow Y) = P(X \cap Y) \quad (2.7)$$

Keterangan:

X → Y adalah item yang muncul bersamaan

P (X ∩ Y) adalah probabilitas transaksi yang mengandung X dan Y dibagi dengan jumlah transaksi seluruhnya.

2. Pembentukan aturan asosiasi.

Setelah semua pola frekuensi tinggi ditemukan, kemudian dicari aturan asosiasi yang memenuhi syarat *minimum confidence* dengan menghitung *confidence* dari aturan *if X then Y*. Bentuk rumus matematika dari *confi-*

dence adalah pada Persamaan 2.8.

$$\text{Confidence}(X \rightarrow Y) = P(X|Y) \quad (2.8)$$

Keterangan:

$X \rightarrow Y$ adalah item yang muncul bersamaan

$P(X | Y)$ adalah probabilitas jumlah transaksi yang mengandung X dan Y dibagi dengan jumlah transaksi yang mengandung X.

2.9 Algoritma FP-Growth

FP-Growth adalah sebuah metode pada teknik asosiasi dalam *data mining* untuk mencari *frequent itemset* tanpa menggunakan *candidate generation*. FP-Growth merupakan pengembangan dari algoritma Apriori. Pada algoritma FP-Growth, kekurangan dari algoritma Apriori diperbaiki dengan menghilangkan *candidate generation*, karena dalam FP-Growth digunakan konsep pembangunan pohon, yang biasa disebut FP-Tree dalam pencarian *frequent itemset*. Selain itu, *scanning database* hanya dilakukan sebanyak dua kali. Sehingga membuat algoritma FP-Growth memiliki performa yang lebih cepat jika dibandingkan dengan algoritma Apriori (Fitria dkk., 2017; Sinthuja, Puviarasan, dan Aruna, 2018; Vijayarani dan Prasannalakshmi, 2015).

Metode FP-Growth dibagi menjadi tiga tahap utama (Wijaya dan Fitriyah, 2018):

1. Tahap Pembangkitan *Conditional Pattern Base*.
2. Tahap Pembangkitan *Conditional FP-Tree*.
3. Tahap pencarian *frequent itemset*.

Pendekatan FP-Growth dirancang untuk melakukan penambangan guna menemukan *itemset* yang sering muncul dalam analisis keranjang pasar (*Market Basket Analysis*). Dalam membangun FP-Tree tidaklah mahal secara komputasi, namun jika basis data melibatkan ribuan *item* maka pohon yang terbentuk menjadi besar (Soni, Gupta, dan Sinhal, 2013).

FP-Growth pertama kali dikemukakan oleh Han. Dalam studinya, Han membuktikan bahwa karena metode *Divide and Conquer* dan metode lainnya, algoritma ini lebih efisien dibandingkan dengan metode populer lainnya dalam mencari pola *frequent itemset*, misalnya. Algoritma Apriori (Sidhu, Meena, Nawani, Gupta, dan Thakur, 2014).

Alasan dari algoritma FP-Growth menjadi lebih efisien daripada algoritma lainnya adalah:

1. *Divide and Conquer*: *Data mining* didekomposisi menjadi *sub-dataset* sesuai dengan *frequent pattern* yang telah diidentifikasi. Sehingga pencarian lebih terfokus karena basis data lebih kecil.
2. Tidak ada generasi kandidat. Akibatnya, tidak diperlukan tes kandidat.
3. Tidak ada *scanning* ulang seluruh basis data (Sidhu dkk., 2014).

Dalam *FP-Growth*, pertama-tama mengkompres *database* yang merepresentasikan *item-item* yang sering muncul menjadi *Frequent Pattern Tree* atau *FP-Tree*. Hal ini dapat mempertahankan informasi mengenai asosiasi *itemset* dan *database* dikompres dan dibagi menjadi satu *set database* kondisional, yang mana masing-masing terkait dengan *item* yang sering muncul. Setiap *node* pada *FP-Tree* mewakili satu *item* dan setiap *path* mewakili rangkaian transaksi yang melibatkan *item* tertentu. Semua *node* yang mengacu pada *item* yang sama dihubungkan bersama dalam daftar, sehingga semua transaksi yang berisi *item* yang sama dapat dengan mudah ditemukan dan dihitung (Farooq dan Sharma, 2016).

FP-Tree dibangun dengan menggunakan 2 tahapan adalah sebagai berikut:

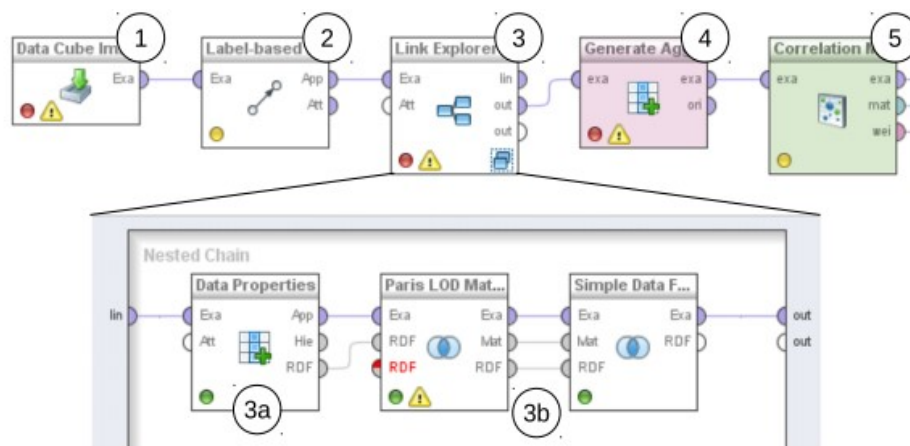
1. Tahap 1:
 - (a) *Scan* data dan temukan nilai *support* untuk setiap *item*.
 - (b) Buang *item* yang tidak sering muncul.
 - (c) Urutkan *item* yang sering dalam urutan menurun berdasarkan pada nilai *support* mereka. Dengan menggunakan urutan ini kita dapat membangun *FP-Tree*, sehingga awalan umum dapat dibagi.
2. Tahap 2:
 - (a) Di sini *node* atau simpul berhubungan dengan *item* dan memiliki hitungan.
 - (b) *FP-Growth* membaca satu transaksi dalam satu waktu dan kemudian memetakannya ke dalam *path* atau jalur.
 - (c) Urutan tadi tetap digunakan, sehingga *path* dapat saling tumpang tindih ketika memiliki transaksi *item* yang sama. Dalam hal ini, hitungan menjadi bertambah. Beberapa pointer dipertahankan antara *node* yang berisi *item* yang sama, dengan membuat daftar yang terhubung satu sama lain. Semakin banyak *path* yang tumpang tindih, semakin tinggi kompresi. *FP-Tree* dapat masuk dalam memori. Terakhir, *itemset* yang sering muncul diekstrak dari *FP-Tree* (Farooq dan Sharma, 2016).

2.10 RapidMiner

RapidMiner adalah *platform* perangkat lunak ilmu data yang dikembangkan oleh perusahaan RapidMiner yang menyediakan lingkungan terpadu untuk *machine*

learning, deep learning, text mining, dan predictive analytics. Yang digunakan untuk aplikasi bisnis dan komersial, penelitian, pendidikan, pelatihan, *rapid prototyping*, dan pengembangan aplikasi. Selain itu, juga mendukung semua langkah dari proses *machine learning* termasuk persiapan data, visualisasi hasil, validasi dan optimalisasi (Kori, 2017).

RapidMiner adalah *platform data mining* tanpa pemrograman, di mana proses *mining* dan analisis data dirancang dari blok bangunan dasar, yang disebut operator. Setiap operator mengeksekusi perintah tertentu pada data, seperti, *load* dan *save* data, mengubah data, atau menyimpulkan model pada data. Pengguna dapat mendesain atau merancang suatu proses analisis data dari operator-operator yang telah disediakan dengan menempatkannya pada kanvas dan kemudian memasang *port input* dan *output*, seperti ditunjukkan pada Gambar 2.2 (Ristoski dkk., 2015).



Gambar 2.2. Contoh penggunaan RapidMiner (Ristoski dkk., 2015)

Selain itu, pada RapidMiner dapat menambahkan fungsionalitas dengan mengembangkan ekstensi, yang tersedia di RapidMiner *Marketplace*. Ekstensi RapidMiner *Linked Open Data* menambahkan operator-operator yang belum tersedia secara *default* pada RapidMiner untuk memuat data dari *dataset*. Selain itu, ekstensi ini mendukung skema yang cocok untuk data yang dikumpulkan dari *dataset* yang berbeda (Ristoski dkk., 2015).

2.11 Penelitian Terdahulu

Penerapan algoritma asosiasi dalam data yang telah di klusterisasi terlebih dahulu telah dilakukan oleh Fauzanu dkk pada tahun 2017 yaitu tentang menganalisa *Web Usage Mining*. Fauzanu dkk menggunakan algoritma *K-Means* untuk klusterisasi dan algoritma *Apriori* untuk asosisasi pada penelitiannya terhadap data *web log file*. Tujuan dari penelitian ini adalah untuk memperoleh pola perilaku

pengunjung. Hasil dari penelitian ini yaitu analisis kecenderungan perilaku pengunjung, yang mana pengunjung *website* cenderung membutuhkan informasi mengenai Owlexa beserta layanannya dengan akurasi pada *cluster* 1 dengan *confidence* 60%, *cluster* 4 dengan *confidence* 82%, dan *cluster* 7 dengan *confidence* 60% (Fauzanu, Darwiyanto, dan Wisudiawan, 2017).

Penelitian lain yaitu yang telah dilakukan oleh Hermanto pada tahun 2015, dalam mengimplementasikan algoritma asosiasi yaitu Apriori dan algoritma *clustering* yaitu *K-Means* dengan mengolah data transaksi pembelian dan data pelanggan. Yang mana pengujian pada data transaksi penjualan barang menggunakan lima kali perbandingan nilai *support* dan nilai *confidence* sebesar 5%, 6%, 7%, 8% dan 9%. Adapun hasil pengolahan data transaksi melalui beberapa tahap diatas yakni pembentukan kandidat *itemset* dan pengukuran *frequent itemset*. Penerapan metode *K-Means clustering* dalam mengolah data pelanggan berdasarkan *recency*, *frequency*, dan *monetary* dapat membentuk sebuah *cluster* atau pengelompokan pelanggan menjadi empat kelompok yaitu *migrator*, *most growable*, *below zeros* dan *most valuable*. Hasil penelitian yaitu rekomendasi produk pada *website* penjualan online (Hermanto, 2015).

Dan penelitian oleh Santosa dan Jadi tahun 2016 yang menganalisis data penjualan menggunakan strategi klaster CLARANS dan asosiasi FP-Growth. Penelitian ini menyajikan analisis keranjang pasar yang bertujuan untuk rekomendasi produk. Penggunaan algoritma CLARANS dapat mengklaster transaksi penjualan berdasarkan jumlah barang yang dibeli dan barang yang tidak dibeli menjadi tiga *cluster*. Penggunaan algoritma FP-Growth pada *Minimum Support* 70%-100% dan berdasarkan *Minimum Confidence* 70%-100% tidak dapat menghasilkan *Lift Ratio* sehingga tidak ditemukan *association rules* (Santosa dan Jadi, 2016).