BAB II

KAJIAN LITERATUR

2.1 Reduksi Data

Reduksi data merupakan langkah penting pada preprocessing data mining dengan tujuan memperoleh keakuratan hasil, kecepatan, dan kemudahan dalam beradaptasi pada kompleksitas data yang ditandai dengan kecepatan objek dalam merespon perubahan data (Ramírez-Gallego, *et al*., 2017). Reduksi data merupakan proses analisis untuk memilih, memusatkan perhatian, penyederhanaan abstraksi serta mentransformasi data yang diperoleh dari catatan – catatan lapangan dengan tujuan untuk lebih mudah dipahami dan tetap mempertahankan informasi yang terkandung dengan berfokus pada hal-hal yang dianggap penting, penentuan pola dan mencari tema serta membuang data yang dianggap tidak penting (Shiyue, L., *et al* 2018). Pada perusahaan penerapan reduksi data dilakukan berbasis data mining dan machine learning untuk mencapai berbagai tujuan dalam menciptakan value yang berbeda (Rágyanszki A, *et al*., 2015). Beberapa pendekatan untuk melakukan reduksi data dapat menggunakan feature reduksi seperti: *T-Distrib Stochastic Neib Embedding, Principle Component Analysis, Canonical Correlation Analysis, Linier Discriminant* Analysis (Ramírez-Gallego, *et al*., 2017). Tahap terpenting sebelum melakukan reduksi adalah tahap persiapan data, dimana operasi *preprocessing* serta data yang saling terintegrasi dapat meningkatkan kualitas data tersebut. Berikut tahapan untuk melakukan operasi *preprocessing* yang mencakup dalam beberapa metode yaitu (ur Rehman, *et al*., 2016):

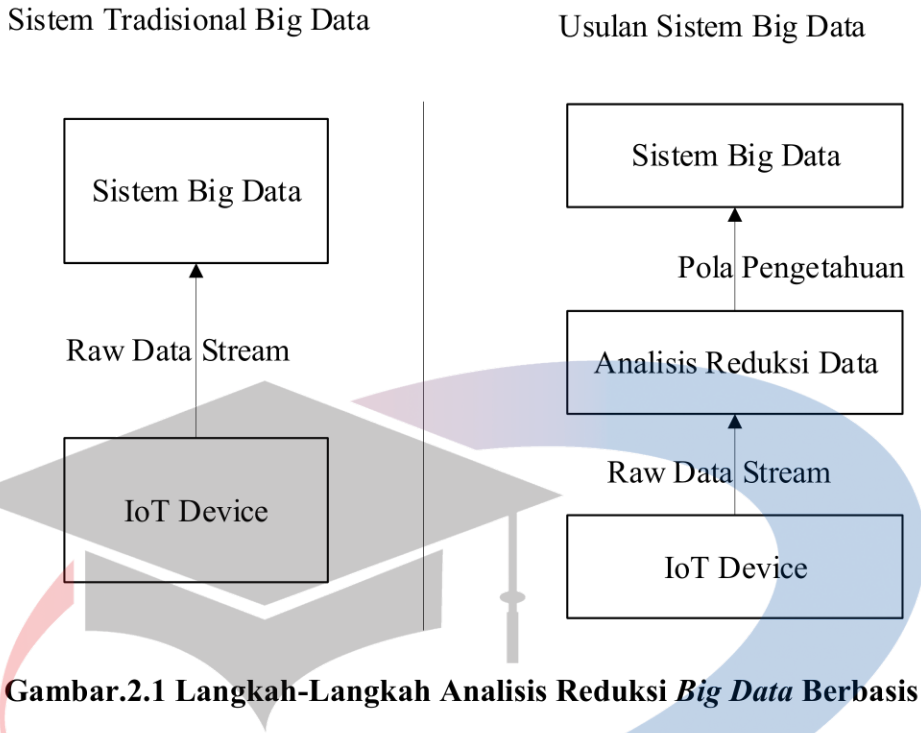
1. *Noise Reduction*, merupakan pengumpulan data mining dengan sumber data berupa sensorik berbasis IoT dan internet melalui aliran data media social untuk memperkenalkan sejumlah informasi yang tidak terstruktur dan berisik. Penerapan metode ini untuk menghilangkan *noise* dan data yang tidak relevan.

2. *Detecting Outliers*, merupakan keberadaan *outliers* (atribut/ nilai yang tidak diinginkan) dalam jumlah data yang besar sehingga menurunkan kualitas pola pengetahuan, metode ini umumnya digunakan pada untuk mendeteksi dan penghapusan data outlier untuk menghasilkan dataset yang berkualitas.
3. *Removing Anomalies*, terdapat ketidakteraturan nilai data, data yang tidak diinginkan pada dataset sehingga mempengaruhi nilai kualitas pada dataset tersebut.
4. *Extracting Features*, merupakan aliran data yang tidak terstruktur dan berkelanjutan. *Extracting Fiture* digunakan untuk memisahkan data terstruktur dari data mentah tergantung pada sifat dan jenis data, metode ini digunakan untuk mengidentifikasi fitur domain waktu dan frekuensi dari data mining tersebut.
5. *Fusing Data Streams from Multiple Data Source*, merupakan variasi data mining berdasarkan kecepatan dan tipe data sehingga intelligent operasi fusi data untuk mengintegrasi dan meningkatkan kualitas data.
6. *Creating Uniform Dataset*, merupakan pengumpulan aliran data dari berbagai sumber dan format, *preprocessing* dilakukan untuk mengubah aliran data mentah menjadi format terstruktur.
7. *Reducing Dimensions*, merupakan kumpulan data mining (missal atribut / kolom dalam table data) digunakan untuk mengurangi data sehingga menghasilkan dataset yang relevan.
8. *Handling Missing Value*, sejumlah data yang hilang dapat menurunkan kualitas pola pengetahuan terhadap data mining. *Preprocessing* digunakan untuk menangani nilai yang hilang dalam data mining dengan teknik eliminasi, sketsa dan imputasi.

2.1.1 Pendekatan Reduksi Atribut

Permasalahan yang sering terjadi ketika lamanya proses pengambilan keputusan karena data yang digunakan sangat besar serta adanya redudansi data

sehingga terjadinya peningkatan beban kerja yang mengakibatkan waktu tunggu layanan membutuhkan waktu yang lebih lama (Shah N B, *et al*., 2015). Pada proses mengurangi beban kerja memori Xu C, *et al*, 2018 melakukan penelitian dengan melakukan pengurangan dimensi video namun tetap mempertahankan kualitas video melalui pendekatan metode *Convolutional Neural Network* mendapatkan hasil pengoptimalan algoritma pengambilan keputusan lebih cepat dan mengurangi jumlah redudansi dengan mempertahankan kualitas video tersebut (Xu C, *et al*, 2018). Dengan perkembangan data yang semakin besar proses *preprocessing* dan reduksi data menjadi menjadi hal penting dalam menemukan model-model yang dapat di dilakukan penyederhaan dengan tujuan untuk mengurangi komopleksitas yang terdapat pada dataset tersebut (Ramírez-Gallego, *et al*., 2017). Pada penelitian sebelumnya dilakukan efisiensi transmisi data masal *multi-source* dengan teknik kombinasi yaitu reduksi atribut dan *gene expression programming algorithm* (GEP) untuk melindungi data dalam jaringan distribusi aktif dengan hasil rentang waktu lebih cepat dan diperoleh nilai akurasi lebih baik (Shiyue, *et.al*, 2018). Penelitian sejenis juga dilakukan oleh Tao *et.al* 2016, dengan melakukan penyaringan pada data *cloud computing* sebagai strategi dan perawatan dalam peningkatan efisiensi dan keamanan dengan hasil klasifikasi pada penerapan *dynamic blocking algorithm* secara efektif sehingga dapat meningkatkan kesulitan dalam mengembalikan data asli (Tao *et.al* 2016). Penerapan konsep analisis reduksi data juga digunakan pada perusahaan seperti yang dilakukan oleh ur Rehman, *et al*., 2016 tentang menentukan pola pengetahuan tersembunyi yang terdapat pada data operasional pelanggan dalam mengurangi biaya pemanfaatan data mining pada layanan *cloud* dengan menggunakan pemodelan *business canvas* sehingga memperoleh peningkatan pengatahuan, pengawasan privasi dalam membangun kepercayaan antara pelanggan dengan perusahaan (ur Rehman, *et al*., 2016). Berikut langkah-langkah analisis reduksi *big data* berbasis IoT (Rehman & Batool, 2015) yang dikutip kembali oleh (ur Rehman*, et al*., 2016).



Gambar.2.1 Langkah-Langkah Analisis Reduksi *Big Data* Berbasis Iot

Sumber: (ur Rehman, *et al*., 2016), Hal: 12

Kemajuan teknologi *cloud computing* merupakan peluang bagi perusahaan dalam mengurangi *flow* data sebelum adanya penyimpanan yang berpusat pada *cloud computing* dengan mengadopsi mobile edge cloud computing terdapat tiga lapisan yaitu: pemprosesan *mobile device*, penyediaan fasilitas *computing* dan melakukan reduksi secara *local* (Akhbar, Chang, Yao, & Munoz., 2016 yang dikutip kembali oleh ur Rehman *et al*., 2016).

2.2. Principal Component Analysis

Principal component analysis (PCA) telah banyak digunakan dalam berbagai bidang seperti pengolahan citra, pengenalan pola, kompresi data, *data mining*, *machine learning* dan *computer vision* untuk mengurangi sebuah dimensi (Larue, *et al*., 2016). *Principal Component Analysis* (PCA) adalah sebuah teknik menganalisa pada sebuah table data observasi ke dalam sebuah data baru yang memiliki kemiripan korelasi dengan tujuan untuk menyederhanakan data observasi

yang sebelumnya kompleks agar menjadi lebih sederhana sehingga mudah untuk di proses atau dianalisis (Zhao, Z., *et a*l., 2016). Menurut Jolliffe, I. T., & Cadima, J., *principal component analysis* adalah teknik statistic yang secara linier mengubah sekumpulan bentuk variable asli menjadi variable yang lebih kecil atau sederhana yang tidak berkorelasi yang dapat mewakili informasi dari sekumpulan variable asli (Jolliffe, I. T., & Cadima, J., 2016).

2.2.1 Prosedur Pengerjaan Reduksi Menggunakan PCA

Kompleksitas data semakin umum ditemukan dan sulit dalam pengolahannya, dalam meningkatkan interpretabilitas tanpa menghilangkan informasi yang terkandung didalam data membutuhkan reduksi dimensi dalam menentukan variabel-variabel utama. Berikut langkah – langkah pengerjaan dalam mereduksi atribut sebagai berikut (Metsalu, T., & Vilo, J.,2015).

a. Pembentukan Matriks

Tahapan awal dari ekstraksi PCA adalah pembentukan matriks. Data dipresentasikan dalam ukuran matrik mxn, dimana m adalah jumlah citra yang dilatih dan n merupakan dimensi dari citra tersebut, kemudian dilakukan proses ekstraksi menjadi citra dengan dimensi yang lebih kecil yang hasilnya diproyeksikan menjadi sebuah matriks seperti pada persamaan berikut (Yi, S *et al.,* 2017):

$$X = \begin{vmatrix} X_{11} & X_{12} & X_{1j} & X_{1N} \\ X_{21} & \dots & \dots & X_{2N} \\ X_{31} & \dots & \dots & X_{3N} \\ X_{i1} & \dots & \dots & X_{iN} \\ X_{M1} & X_{M2} & X_{Mj} & X_{MN} \end{vmatrix}$$

…….…… (2.1)

Dimana x = matrik citra. Setelah matrik data citra terbentuk, maka proses berikutnya adalah proses perhitungan untuk mencari rata-rata hasil seluruh citra. Pencarian nilai rata-rata ini bertujuan untuk mengetahui noise atau persamaan tiap

vektor yang dapat mengganggu keakuratan perhitungan pada PCA, yang dapat dihitung dengan menggunakan rumus (Yi, S *et al.,* 2017):

$$[A - \lambda I]\,[X] = [0] \quad \text{.....(2.2)}$$

b. Menghitung Matrik Kovarian

Input vektor x_t (t=1, ⋯,1 dan $x_t = 0$) dengan dimensi m $x_t = [x_t(1), x_t(2), \cdots, x_t(m)]^T$ biasanya m < 1, setiap vektor x_t ditransformasikan secara linier kedalam satu vektor baru *s* yang dinyatakan sebagai berikut :

$$S_t = U^T . x_i \quad \text{..... (2.3)}$$

Dimana *U* adalah matrik orthogonal m x m dengan kolom ke i, u_i adalah nilai eigenvector dari sampel matrik kovarian

$$C = \frac{1}{l} \sum_{t=1} l \; Xt. x_t^T \quad \text{.....(2.4)}$$

c. Menghitung Eigenvalue dan Eigenvector Dari Matrik Kovarian

$$\lambda_t U_t = C.u_i, \; i=1, \cdots m \quad \text{..... (2.5)}$$

Dimana *i* adalah salah satu eigenvalue dari C, u_i adalah nilai eigenvector.

d. Menghitung nilai transformasi orthogonal

Berdasarkan nilai estimasi u_i dan komponen s_t, yang kemudian dihitung sebagai transformasi orthogonal dari x_t

$$S_t(i) = u_i^T x_t \text{ , i = 1, ..., m} \quad \text{.....(2.6)}$$

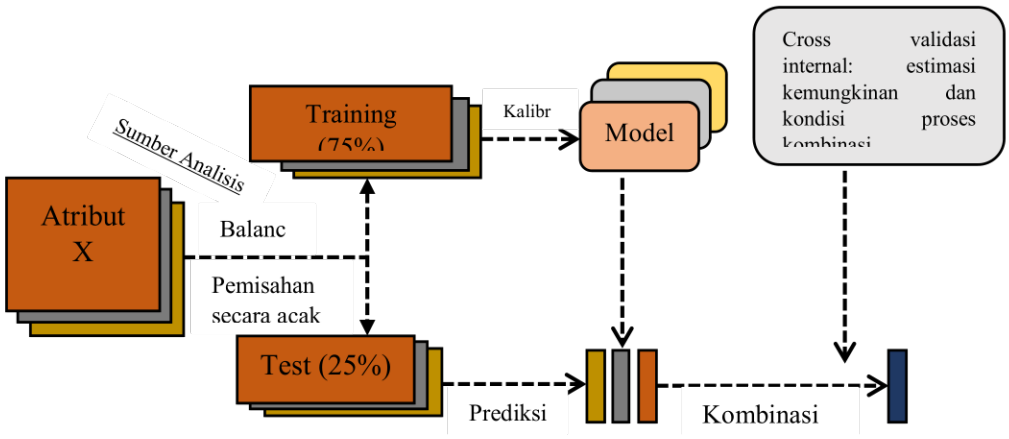
Komponen yang baru tersebut disebut dengan principal component. Dengan menggunakan hanya beberapa nilai pertama eigenvector yang telah diurutkan berdasarkan nilai eigennya, jumlah principal component dari s_t (*i*) yang tidak saling terkorelasi, mempunyai varian maksimum yang berurutan dan estimasi error rata-rata dari representasi data input asli adalah minimal.

1.2.2 Pendekatan Principal Component Analysis

Pada penelitian sebelumnya yang telah dilakukan oleh Ng, S. C. 2017 tentang penurunan dimensi file gambar dengan membandingkan kualitas gambar sebelum dan sesudah dilakukan reduksi menggunakan PCA dengan hasil mencapai 35,3% pada pengurangan ukuran file gambar dengan tetap mempertahankan property utama pada gambar asli (Ng, S. C. 2017). Pada penelitian sebelumnya yang telah dilakukan oleh Yi S, *et al*., 2017 tentang menguji sensitivitas outlier data terhadap dimensi rendah menghasilkan pendekatan lebih efektif (Yi S, *et al*., 2017). Pada penelitian sebelumnya yang telah dilakukan oleh Zhao Z, *et al*., 2016 tentang keakuratan dan efektitfitas PCA terhdap gambar 2D dalam mengurangi *noisy* data (Zhao Z, et al., 2016).

2.3. Klasifikasi

Klasifikasi merupakan proses pencarian sekumpulan model atau fungsi yang menggambarkan dan membedakan kelas data dengan adalah untuk memprediksi kelas dari suatu objek yang belum diketahui kelasnya (Constantinou AC, et al., 2016). Klasifikasi sendiri memiliki dua proses yaitu membangun model klasifikasi dari sekumpulan kelas data yang sudah didefinisikan sebelumnya (training data set) dan menggunakan model tersebut untuk klasifikasi data test serta mengukur akurasi dari model (Martinez, *et al*., 2016). Proses pengklasifikasian dapat dilakukan dengan berbagai metode seperti *Bayesian Network*, Support Vector Machine, K-Nearest Neighbour Classifier, Neural Network (Luo Y, et al., 2017). Berikut tahapan dalam melakukan klasifikasi (Ballabio et al., 2019).



Gambar.2.2 Skema Klasifikasi Validasi Prosedur

Sumber: (Ballabio et al., 2019), Hal: 18

Metode pelatihan data mining memiliki 3 kelompok, seperti: *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning*. 3 kelompok tersebut memiliki definisi, sebagai berikut (Hui, S. K. A., & Grandner, M. A. 2015):

1. *Supervised Learning*

Kumpulan record dari inputan yang digunakan dan telah diketahui output, dengan kata lain variable yang menjadi target telah ditentukan dalam *dataset* yang sedang dianalisis. Sebagian besar algoritma dalam kelompok tersebut terdiri dari: klasifikasi, estimasi, dan prediksi. Algoritma yang digunakan akan melakukan *process* pembelajaran yang berdasarkan *value* dari *variable* sasaran yang telah terasosiasi dengan *value* pada varibel *predictor*. Beberapa contoh algoritma yang menerapkan metode supervised learning adalah *Bayesian Network*, *Nearest – Neighbor Classifier, Artificial Neural Network, Support Vector Machine, Fuzzy K-Nearest Neighbor.*

2. *Unsupervised Learning*

Pada klasifikasi unsupervised learning data tersebut dapat dianalisa diterapkan tanpa adanya guru serta pelatihan pada data lampau, dengan kata lain diartikan sebagai pencarian pola pada setiap atribut yang digunakan. Tidak termasuk penetapan atribut atau kelas pada sasaran. Contoh algoritma yang menerapkan metode *unsupervised learning* adalah *K-means, hierarchical clustering, DBSCAN, Fuzzy C-Means, Self-Organizing Map.*

3. *Reinforcement Learning*

Berbeda dengan dua kelompok yang terdapat di atas, pada model ini mempunyai tujuan untuk mencari atribut yang muncul pada transaksi yang sama. *Reinforcement Learning* biasanya berfungsi untuk mencari dan menganalisa transaksi belanja dengan konsep mencari produk yang dibeli secara bersamaan dalam satu transaksi yang sama algoritma yang digunakan dalam kelompok *Reinforcement Learning* adalah Apriori.

2.3.1 Bayesian Network

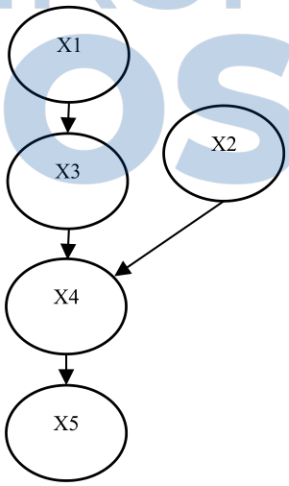
Bayesian Network merupakan salah satu *Probabilistic Graphical Model* (PGM) sederhana yang dibangun dari teori *probabilistic* berhubungan langsung dengan data, sedangkan teori *graf* berhubungan langsung dengan representasi yang ingin didapatkan (You, Y., Li, J., & Xu, N., 2017). Terdapat empat hal yang dapat ditawarkan oleh *Bayesian Network* yaitu; pertama, *Bayesian Network* dapat dengan mudah menangani ketidaklengkapan maupun masalah pada data. Kedua, *Bayesian Network* memungkinkan seseorang untuk belajar tentang hubungan kausal. Proses pembelajaran menjadi penting ketika kita mencoba untuk memahami domain dari permasalahan. Ketiga, *Bayesian Network* dapat memfasilitasi kombinasi dari pengetahuan domain dan data. Terakhir, *Bayesian Network* menawarkan pendekatan yang efisien dan berprinsip menghindari *over fitting* pada data. Pembuatan model dalam *Bayesian Network* melibatkan dua langkah yaitu: membuat struktur jaringan dan mengestimasi nilai probabilitas setiap *node* (Prima Sari D, *et al.,* 2016). Sebagai contoh, sebuah *Bayesian Network* dapat mewakili hubungan antara penyakit dan gejala. *Bayesian Network* dapat digunakan untuk menghitung probabilitas dari kehadiran berbagai gejala penyakit.

Bayesian Network memiliki dua tugas pembelajaran melalui DAG dan struktur dari *Bayesian Network* berupa jaringan (Izadi, M., Charland, K., & Buckeridge, D. L. 2017). *Bayesian Network / Belief Netwok / Probalilistik Network* merupakan sebuah model grafik untuk merepresentasikan sebuah interaksi antar variable. Hubungan sebab akibat antara variable dapat dinyatakan dalam

probabilitas kondisional. *Bayesian Network* mampu mengkodekan keduanya secara kualitatif yaitu: (rendah / sedang / tinggi), ataupun secara *Boolean* seperti: (ya / tidak, benar / salah) atau variable kontinyu. Variable – variable untuk menggambarkan data dapat berasal dari data historis, seorang pakar, atau kombinasi dari keduanya (Hosseini, S., & Barker, K. 2016).

Adapun *Bayesian Network* merupakan *acyclic graphs* dengan seperangkat variable (node) yang diwakili oleh $V = \{X_1, X_2, \cdots, X_n\}$, dan struktur set tepi untuk menentukan ketergantungan antar variable. Tepi luar dari X_i ke X_j menampilkan hubungan nilai variable X_j bergantung dari nilai X_i. Berikutnya, jika terdapat tepi luar dari X_i ke X_j, maka X_i merupakan simpul induk dari X_j, dan X_j adalah simpul anak X_i pada *Bayesian Network* terdapat tiga kelas simpul yaitu Hosseini, S., & Barker, K. (2016):

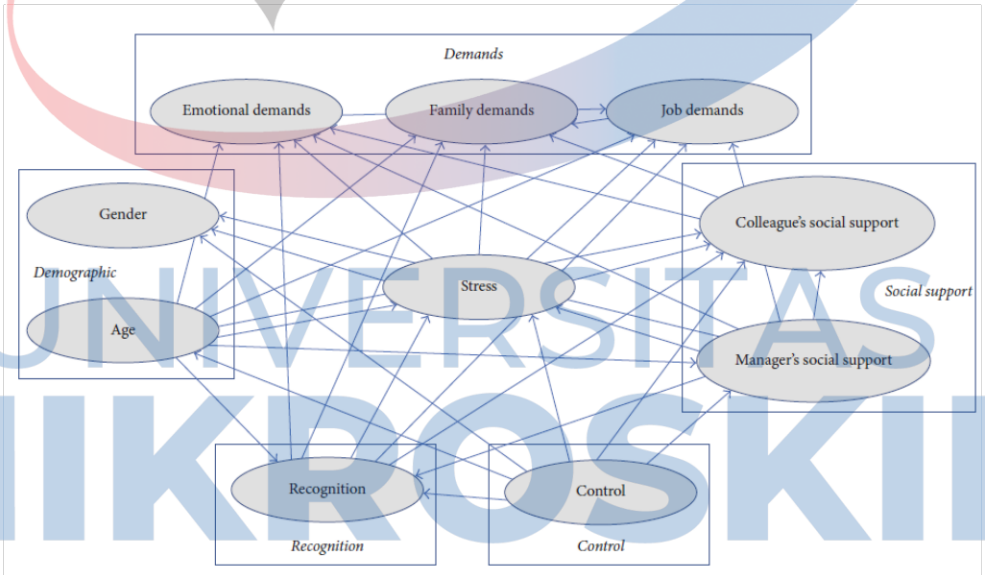
a. Simpul tanpa memiliki simpul anak disebut dengan simpul *leaf*.
b. Simpul tanpa simpul orang tua disebut dengan simpul *root*.
c. Simpul dengan simpul anak dan orang tua disebut simpul *intermediate.* Berikut contoh gambar.2.3 simpul X1 dan X2 merupakan simpul *root*, simpul X3 dan X4 merupakan simpul *intermediate*, dan X5 merupakan simpul leaf (Hosseini, S., & Barker, K. 2016).



Gambar. 2.3 Contoh *Bayesian Network* dengan Lima Variable (Node)

Sumber: (Hosseini, S., & Barker, K. 2016).

Adapun Bayesian Netwok itu sendiri digambarkan seperti graf yang terdiri dari simpul (*node*) dan busur (*arc)*. Simpul akan menunjukan variable, misalnya X beserta nilai probalilitasnya p(x) dan busur akan menunjukan hubungan antar simpul. Jika ada hubungan dari simpul X ke simpul Y, ini akan mengindikasikan bahwa variable X ada pengaruhnya terhadap variable Y, dan pengaruh itu dinyatakan dengan peluang bersyarat P(Y|X). Perbedaan dari *Naïve Bayes* dengan *Bayesian Network* adalah pada Naïve Bayes mengabaikan korelasi antar variable, sedangkan pada *Bayesian Network* merupakan variable input yang bisa saling dependen (berhubungan) (Martinez AM, *et al*., 2016). Seperti contoh kasus pada gambar 2.4, terdapat 9 variabel yang menggambarkan saling ketergantungan hubungan antar variable yang ada dengan menggunakan faktorisasi probabilitas gabungan untuk menyerderhanakan beberapa pengetahuan variable dalam memprediksi keadaan hasilnya (García-Herrero, *et al*., 2017):



Gambar: 2.4 Faktor-Faktor Yang Mempengaruhi Penurunan Kinerja Karyawan.

Sumber: (García-Herrero, *et al*., 2017)

Bayesian Network didasarkan pada Teorema Bayes yaitu conditional probability (peluang bersyarat) yang dinotasikan dengan P(A|B) artinya peluang

keadaan A jika keadaan B telah terjadi (García-Herrero, *et al*., 2017). Berbeda dari naive bayes yang mengabaikan hubungan antar atribut atau variabel, pada *Bayesian Network* antar variabel atau atribut bisa saling dependent atau berhubungan, berikut Rumus Teorema Bayes yaitu (Ballabio, D., Todeschini, R., & Consonni, V. 2019):

$$P\left(A|\mathrm{B}\right) = \frac{P(B|A)P(A)}{P(B)} \quad \text{.....(2.7)}$$

Atau

$$P\left(A|\mathrm{B}\right) = \frac{P(A)P(B\,|\,A)}{P(B|A)P(A)+P(B\,|\,\hat{\mathrm{A}})P(\hat{\mathrm{A}})} \quad \text{.....(2.8)}$$

Dimana:

$P\left(A|\mathrm{B}\right)$ = disebut posterior probability, yaitu peluangA terjadi setelah B terjadi

$P\left(A \cap \mathrm{B}\right)$ = peluang B dan A terjadi bersamaan

$P\left(B|\mathrm{A}\right)$ = disebut juga likehood, yaitu peluang B terjadi setelah A terjadi

$P\left(\mathrm{A}\right)$ = disebut juga prior, yaitu peluang kejadian A

$P\left(B\right)$ = peluang kejadian B

Adapun langkah-langkah untuk menerapkan *Bayesian Network* yaitu (Ballabio, D., Todeschini, R., & Consonni, V. 2019):

1. Membangun struktur *Bayesian Network*
2. Menentukan parameter
3. Membuat *Conditional Probability Table* (CPT)
4. Membuat *Joint Probability Distribution* (JPD), untuk menghitung Joint Probability Distribution adalah mengalikan nilai *Conditional Probability* dengan *Prior Probability*.
5. Menghitung *Posterior Probabilistik*, didapatkan dari hasil JPD yang telah diperoleh.
6. Inferensi Probabilistik yaitu penelusuran yang dilakukan berdasarkan variabel input yang diberikan pengguna sehingga menghasilkan suatu nilai probabilitas.

Berikut contoh kasus dengan *Bayesian Network* (Murphy K, 2006), missal: diberikan lima variable, yaitu: Pencuri (*Burglary*), Gempa Bumi (*Earthquake)*,

Alarm Mati, John menelpon (John *Calls*), dan Mary menelpon (Mary *Calls)*. Selanjutnya definisikan lima variable diatas sebagai berikut:

B = Seorang pencuri masuk ke dalam rumah

E = Gempa bumi yang terjadi di rumah

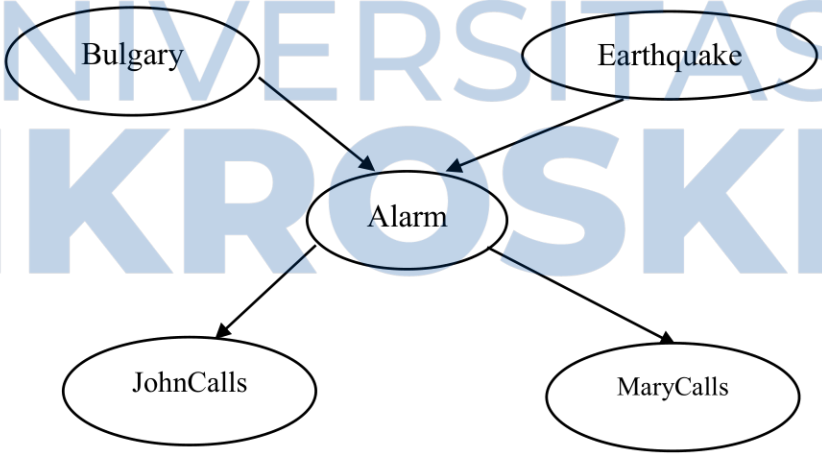
A = Alarm mati

J = John menelpon untuk melaporkan jika mendengar suara alarm

M = Mary menelpon untuk melaporkan jika mendengar suara alarm

Misal kita akan menghitung P(B | M, J) dengan menggunakan joint probability, maka akan dibangun *Bayesian Network* dengan langkah-langkah sebagai berikut:

1. Terdapat lima kejadian. Setiap kejadian mempunyai dua nilai kebenaran, yaitu benar dan salah, sehingga kemungkinannya adalah $2^5 = 32$.
2. Selanjutnya menggunakan prior untuk setiap variable (gambar)



Gambar.2.5 Contoh *Bayesian Network*

Sumber: (Lee, J., Henning, R., & Cherniack, M. 2019)

3. Membangun *Bayesian Network*

   a. Pada gambar: dapat kita tuliskan partisi ordernya sebagai berikut:

      Contoh:

      {E,B} →{A} →{J,M}

   b. P(J,M,A,E,B) = P(J,M | A,E,B) P(A | E,B) P(E,B)

      ≈ P(J,M | A).P(A|E,B).P(E).P(B)

      ≈ P(J|A).P(M|A).P(A|E,B).P(E) P(B)

Ini adalah asumsi untuk *conditional independence* yang direpresentasikan dalam bentuk struktur *graf* dari *Bayesian Network*

   c. P(J,M,A,E,B) = P(J|A) P(M|A) P(A|E, B) P(E) P(B)

   d. Missal terdapat tiga *conditional probability table* yang akan ditentukan, yaitu: P(J|A), P(M|A), P(A|E, B). maka memiliki kemungkinan 2+2+4 = 8

   e. Selanjutnya dua *marginal probabilities* P(E), P(B) memiliki dua lagi kemungkinan

   f. Untuk menghitung P(B|M, J), kita asumsikan P(b|m, ¬J) dimana P(B=True|M=True ∧ J = Falses). Dari definisi, kita dapatkan

$$P(b|m,\neg j = \frac{P(b,\ m,\neg J)}{P(m,\neg J)}$$

   g. Kemudian kita dapatkan *marginal probability* sebagai berikut:

      P(b|m, ¬j) =

$$\sum A\ \epsilon\{a,\neg a\} \sum E\ \epsilon\ \{e,\neg e\}\ P(\neg j,m,A,E,B)$$

   h. Dari *conditional independence* tersebut, kita dapat tuliskan sebagai berikut:

      P(J,M,A,E,B) ≈P(J|A) P(M|A) P(A|E,B) P€ P(B)

      P(¬j, M,A,E,B) ≈P(¬j|A) P(m|a) P(A|E, b) P(E) P(b)

         a. Untuk kasus A = a ∧ E = ¬e

P(¬$j, m, a, \neg e, b$) ≈P(¬j|A) P(m|a) P(A|E, b) P(E) P(b)

$$\approx 0.1 \times 0.70 \times 0.94 \times 0.998 \times 0.001$$

i. Begitu juga untuk kasus a ∧ e, $\neg a \wedge e, \neg a \wedge \neg e$

j. Sama dengan P(m, $\neg j$), dapat kita partisi untuk mendapatkan P(b|m, $\neg j$)

2.4. RMSE *(Root Mean Square Error)*

Kriteria yang digunakan untuk mengukur kebaikan model setelah diperoleh suatu model adalah *root mean square* error (RMSE). RMSE merupakan alat seleksi model berdasarkan pada *error* hasil estimasi. *Error* yang ada menunjukkan seberapa besar perbedaan hasil estimasi dengan nilai yang akan diestimasi (Yi, S *et al.,* 2017). Nilai error tersebut akan digunakan untuk menentukan model mana yang terbaik. Definisi RMSE dapat ditulis sebagai berikut (Martinez, *et al*., 2016).

$$RMSE = \sqrt{\frac{\sum_{i}^{n}=1\ (yi\ Yi)^2}{n}} \quad \text{.....(2.9)}$$

Dimana:

RMSE : *Root Mean Square Error*

n : Jumlah Sampel

yi : Nilai Aktual

Yi : Nilai Prediksi

2.5. Kerangka Pikir / Pola Pemecahan Masalah

Beberapa faktor ketidakhadiran karyawan beragam seperti; sakit, masalah keluarga, faktor usia, lingkungan kerja, lokasi kerja, transportasi kerja dan lainnya (Kocakulah MC, *et al*., 2016). Hal tersebut mempengaruhi kinerja perusahaan baik secara positif maupun *negative* sehingga terganggunya aktivitas seperti; penugasan, pengambilan keputusan, yang mengakibatkan peningkatan biaya, peningkatan beban kerja karyawan yang berdampak pada penurunan motivasi, kualitas kinerja,

ketidakhadiran karyawan dan produktivitas ditempat kerja (Desmeles, F., *et al* 2016).

Namun masih terdapat kelemahan dari penelitian sebelumnya seperti, keterbatasan data ketidakhadiran karyawan, hasil akurasi ketidakhadiran karyawan hal ini disebabkan oleh berbagai faktor seperti kurangnya data pengetahuan yang dimiliki sistem, dan metode klasifikasi yang kurang tepat. Pada penelitian ini dilakukan reduksi data dengan menggunakan metode Principal Component Analysis dalam menentukan variable yang paling berpengaruh pada data ketidakhadiran karyawaan. Selanjutnya dilakukan klasifikasi kembali terhadap atribut menggunakan *Bayesian Network* untuk melihat keterhubungan faktor sebab akibat ketidakhadiran karyawan selanjutnya dilakukan penilaian tingkat akurasi sebelum dan setelah dilakukannya reduksi atribut yang diharapakan mampu mengoptimalisasi sebelum dan sesudah dilakukannya reduksi atribut.