BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Tempat Penelitian

2.1.1 Sejarah Singkat dan Perkembangan Umum Perusahaan [8]

Perusahaan Daerah Bank Perkreditan Rakyat Kabupaten Bandung Cabang Batujajar (PD BPR Batujajar) didirikan berdasarkan Surat Keputusan Gubernur Jawa Barat No.25/Reg/9/A/A/1970, yaitu tentang pendirian Bank Karya Produksi Desa (BKPD) pada tanggal 1 Juli 1979 dengan modal dasar pertama sebesar Rp.5.000.000.- (lima juta rupiah) hasil dari penyisihan pemasukan Ipeda sebesar 9% dari jumlah Ipeda yang diterima di desa-desa. Sejak tahun 1979 status kepemilikannya telah berubah menjadi milik Pemerintah Daerah Kabupaten Bandung berdasarkan peraturan pemerintah No.VI 1976.

Sejalan dengan perkembangan lembaga keuangan yang terus mengalami perubahan yang mengarah kepada peningkatan kinerja lembaga maka status BKPD berubah menjadi Perusahaan Daerah Bank Perkreditan Rakyat (PD BPR) sesuai dengan undang-undang nomor 7 tahun 1992 yang pada saat ini telah dirubah ke dalam undang-undang nomor 10 tahun 1998 tanggal 10 Nopember 1998 tentang perbankan. Perubahan ini pun diresmikan dengan dikeluarkannya Peraturan Daerah (PERDA) Kabupaten Bandung Nomor 23 tahun 1996 tanggal 7 Nopember 1996 tentang Perusahaan Daerah Bank Perkreditan Rakyat (PD BPR) yang kemudian disahkan dengan Surat Keputusan Gubernur Jawa Barat nomor 188.342/SK.231-Huk/1997 tanggal 4 maret 1997.

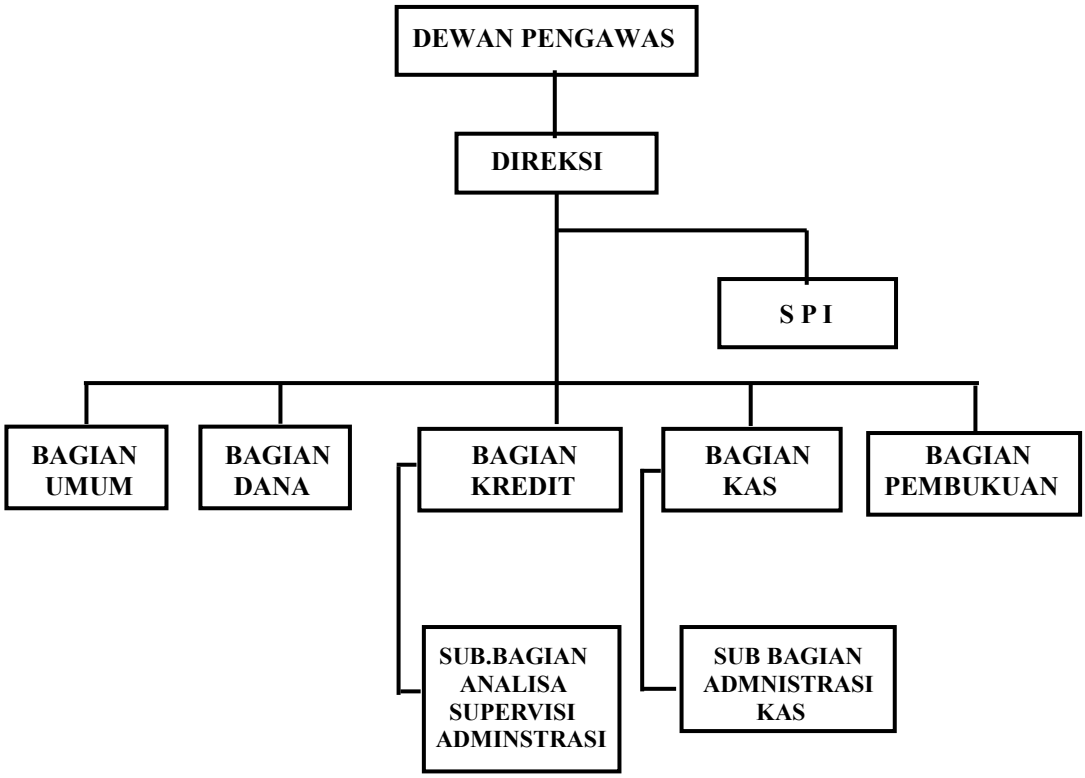
2.1.2 Visi dan Misi [8]

Visi PD BPR Batujajar adalah membantu dan mendorong pertumbuhan perekonomian dan pembangunan daerah di segala bidang serta berperan sebagai salah satu sumber pendapatan asli daerah dalam rangka meningkatkan taraf hidup rakyat. Adapaun misi PD BPR Batujajar adalah :

1 Menghimpun dana dari masyarakat dalam bentuk tabungan dan deposito berjangka maupun bentuk simpanan yang lain yang bisa dijadikan sebagai salah satu sumber dana bagi kalangan bank.
2 Memberikan kredit kepada masyarakat dan melakukan pembinaan terhadap pengusaha golongan ekonomi lemah.
3 Melakukan kerjasama antara PD BPR dengan lembaga perbankan atau lembaga keuangan lainnya.

2.1.3 Struktur Organisasi [8]

Secara umum struktur organisasi PD BPR Batujajar terdiri dari Dewan Pengawas, Direksi, Satuan Pengawas Intern, Bagian Umum, Bagian Dana, Bagian Kredit, Bagian Kas, dan Bagian Pembukuan. Seiring dengan perkembangan perusahaan termasuk produk pelayanan yang diberikan yang terus meningkat perlu diimbangi dengan organisasi perusahaan yang terstruktur sehingga mudah mengatasi berbagai hal. Struktur organisasi memerlukan adanya pengelompokan dari berbagai kegiatan yang perlu dilakukan secara terarah dan terpimpin untuk mencapai tujuan perusahaan. Oleh karena itu struktur organisasi di PD BPR Batujajar berorientasikan kepada tujuan dan fungsi organisasi.



Gambar 2.1 Struktur Organisasi PD BPR Batujajar (berdasarkan Surat Keputusan Gubernur Jawa Barat nomor 188.342/SK.231-Huk/1997 tanggal 4 maret 1997)

2.1.4 Uraian Tugas

2.1.4.1 Direksi

Tugas dan tanggung jawab direksi adalah :

1. Membantu kepala daerah dalam pengolahan keuangan bank di bidang permodalan berdasarkan asas-asas usaha yang layak sehat dan menguntungkan.
2. Melakukan pengolahan kekayaan yang bertanggung jawab atas kelangsungan perusahaan.

3. Menghimpun, mengkoordinasi dan mengendalikan kegiatan perusahaan.
4. Melakukan kebijakan perusahaan sesuai dengan prinsip perbankan dan peraturan-peraturan lainnya.
5. Menjaga stabilitas kemampuan perusahaan.

2.1.4.2 Satuan Pengawasan Intern (SPI) [8]

Tugas dan tanggung jawab satuan pengawas intern adalah :

1. Melaksanakan pengawasan intern atas kegiatan-kegiatan di PD BPR.
2. Pengawasan atas pelaksanaan anggaran pendapatan dan belanja PD BPR, menyelenggarakan tata kerja dan prosedur kerja unit organisasi menurut ketentuan yang berlaku serta pengawasan keamanan dan ketertiban PD BPR.
3. Pengawasan kegiatan operasional PD BPR dan pelaksanaan penelitian, pembahasan secara periodik atau berkala
4. Pelaksanaan audit atas administrasi keuangan dan pengelolaan dana seluruh kekayaan milik PD BPR.
5. Pelaksanaan supervisi atas bagian secara keseluruhan.
6. Pelaksanaan supervisi atas anggunan-anggunan dan jaminan-jaminan lain yang diterima oleh PD BPR dan melikuidir serta merealisir anggunan atau jaminan.
7. Pemberian saran atau pertimbangan kepada direksi secara hirarki tentang langkah dan tindakan yang perlu diambil dibidang tugasnya.

2.1.4.3 **Bagian Umum** [8]

Tugas dan tanggung jawab bagian umum adalah :

1. Merencanakan, mengkoordinir dan mengevaluasi serta melaporkan kebijaksanaan penyelenggaraan kegiatan PD.BPR
2. Membantu direksi sesuai dengan hilarki dibidang tugasnya.
3. Pemberian saran atau pertimbangan kepada direksi secara hirarki tentang langkah dan tindakan yang perlu diambil di bidang tugasnya.
4. Melakukan urusan surat menyurat, administrasi perjalanan dinas dan urusan kepegawaian.
5. Pelaksanan administrasi surat menyurat baik surat masuk maupun surat keluar dan kearsipan.
6. Pelaksanaan administrasi perjalanan dinas termasuk melayani tamu untuk kepentingan dinas.
7. Pelaksanaan administrasi kepegawaian termasuk memegang buku absensi.
8. Pelaksanaan penyusunan rencana kebutuhan perlengkapan PD BPR.
9. Pelaksanaan pengadaan, penyimpanan, dan pendistribusian perlengkapan.
10. Pelaksanaan penyusunan bahan pembinaan dan pengendalian perlengkapan PD BPR.
11. Pengadministrasian pembayaran gaji pegawai setiap bulannya.
12. Membantu kepala bagian kas dalam pengadministrasian kas yaitu :

a. Pengelolaan perlengkapan serta melaksanakan administrasi pembukuan keluar dan masuk keuangan.
b. Pelaksanaan administrasi pembukuan keluar dan masuknya uang.
c. Pemberian saran dan pertimbangan kepada kepala bagian secara hirarki tentang langkah dan tindakan yang perlu diambil di bidang tugasnya.

2.1.4.4 **Bagian Dana** [8]

Tugas dan tanggung jawab bagian dana adalah :

1. Mengusahakan dan mengkoordinasikan pengembangan dana PD.BPR.
2. Pengkoordinasian dan pengawasan serta pengarahan kegiatan dan pelaksanaan tugas dibawahnya.
3. Penyelenggaraan pengembangan dana dan pengevaluasian keluar masuknya dana.
4. Penelitian dan penandatanganan *voucher* laporan mutasi.
5. Mencari nasabah dan atau menghimpun dana masyarakat berupa dana tabungan dan deposito serta simpanan atau pinjaman lainnya.
6. Pelaksanaan administrasi lengkap yang berkenaan dengan penerimaan dan pengeluaran transaksi dana tabungan dan deposito serta simpanan atau pinjaman lainnya
7. Pelaksanaan perhitungan bunga tabungan dan deposito.
8. Menyusun rekapitulasi (nominatif) tabungan dan deposito serta simpanan atau pinjaman lainnya

9. Pemberian saran atau pertimbangan kepada kepala bagian secara hirarki tentang langkah dan tindakan yang perlu diambil diambil dibidang tugasnya.
10. Menyimpan dan memelihara dokumen transaksi pinjaman yang diterima atau kredit kerjasama.

2.1.4.5 Bagian Kredit [8]

Tugas dan tanggun jawab bagian kredit adalah :

1. Bagian kredit mempunyai tugas pokok yakni menyalurkan dana dan menyelenggarakan penyaluran kredit kepada nasabah.
2. Pengkoordinasian dan pengawasan serta pengarahan kegiatan dan pelaksanaan tugas dibawahnya.
3. Penyelenggaraan penjelasan dan penelitian syarat dan prosedur kredit serta pemberian putusan kredit sebagaimana yang ditentukan oleh direksi kepada para pemohon.
4. Melaksanakan administrasi lengkap untuk persiapan realisasi kredit yang diberikan sesuai dengan kebijakan atau putusan dari direktur.
5. Menyelenggarakan pembinaan dan pengawasan terhadap nasabah peminjam (debitur).
6. Melaksanakan penagihan pengembalian kredit.
7. Usaha-usaha penyelamatan kredit bagi nasabah peminjam yang dalam keadaan posisi debius.
8. Pemberian saran atau pertimbangan kepada direksi secara hirarki tentang

langkah dan tindakan yang perlu diambil dibidang tugasnya melakukan pencatatan registrasi penyaluran kredit.

9. Pelaksanaan pencatatan registrassi permohonan pinjaman.
10. Pelayanan administrasi pemberian kredit dengan mengadakan proses perjanjian kredit, baik pengikatan yang dilakukan secara notaril maupun dibawah tangan.
11. Pelayanan penerimaan pembayaran angsuran atau pengembalian kredit dan perhitungan kewajiban membayar bunga dan lainnya.
12. Menyusun rekapitulasi nominatif kredit.
13. Meneliti syarat-syarat yang diajukan oleh calon nasabah.
14. Pemeriksaan dan penilaian ke tempat usaha peminjam proyek atau tempat lain yang ada kaitannya dengan usaha peminjam, untuk memperoleh keyakinan bahwa usaha kegiatan peminjam sesuai dengan laporannya.
15. Melaksanakan administrasi buku catatan atas usaha peminjam sehingga teruji kebenarannya, mengumpulkan data baik data intern maupun data eksternal lainnya secara sistematis dan aman agar jangan sampai bocor kepada pihak lain.
16. Penyampaian dokumen hasil analisa kepada kepala bagian kredit.untuk mendapat pertimbangan dan putusan dari direktur.
17. Pemberian saran atau pertimbangan kepada kepala bagian secara hirarki tentang langkah dan tindakan yang perlu diambil dibidang tugasnya.
18. Penyusunan evaluasi tingkat kolektibilitas kredit dengan kriteria lancar,

kurang lancar, diragukan dan macet masing-masing kriteria ini didasarkan pada peraturan yang dikeluarkan oleh Bank Indonesia

19. Pengawasan dan pembinaan terhadap kredit-kredit, baik kredit baru maupun kredit yang sedang berjalan, agar kredit tersebut tetap berjalan lancar.
20. Mengadakan penagihan yang efektif bagi kredit yang tergolong kriteria kurang lancar, diragukan dan macet.
21. Usaha-usaha penyelamatan kredit bagi nasabah yang sudah dalam posisi debius.

2.1.4.6 **Bagian Kas** [8]

Tugas dan tanggung jawab bagian kas adalah :

1. Mengkoordinasikan kegiatan pemasukan dan pengeluaran uang.
2. Pelaksanaan dan pengkoordinasian kegiatan-kegiatan pengeluaran uang.
3. Pemberian saran atau pertimbangan kepada direksi secara hirarki tentang langkah dan tindakan yang perlu diambil dibidang tugasnya.
4. Melaksanakan administrasi penerimaan dan pengeluaran uang.
5. Pelaksanaan perhitungan penerimaan dan pengeluaran uang berdasarkan berdasarkan bukti-bukti yang sah.
6. Membuat atau membukukan penerimaan dan pembayaran
7. menyimpan dan memelihara uang kas secara teliti pada tempat penyimpanan uang (Brankas)
8. Membuat berita acara penyimpanan uang.

2.1.4.7 **Bagian Pembukuan** [8]

Tugas dan tanggung jawab bagian pembukuan :

1. Melaksanakan pembukuan dan menerima laporan dari bagian-bagian lain untuk dilaporkan kepada direksi.
2. Pelaksanaan pembukuan dan evaluasi laporan dari setiap bagian.
3. Pembuatan rencana kerja dan anggaran perusahaan setiap tahun.
4. Menyusun dan mengkoordinasikan kegiatan penyusunan anggaran serta mengevaluasi hasil kegiatan dari masing-masing bagian dan hasil dari pelaksanaan anggaran.
5. Pelaksanaan penyusunan anggaran PD.BPR.
6. Pengkoordinasian kegiatan penyusunan anggaran PD.BPR.
7. Pelaksanaan evaluasi anggaran PD.BPR yang telah dilaksanakan sesuai dengan ketentuan Bank Indonesia.
8. Pemberian saran atau pertimbangan kepada direksi secara hirarki tentang langkah dan tindakan yang perlu diambil dibidang tugasnya.
9. Menghimpun data pelaporan dari setiap bagian.
10. Pelaksanaan penyusunan laporan PD.BPR dari setiap bagian.
11. Membuat dan menyampaikan laporan pertanggungjawaban keuangan PD.BPR sesuai dengan laporan yang harus dilaporkan.
12. Membantu bagian kredit dalam hal analisa atau supervisi yang mempunyai tugas pokok meneliti data calon nasabah yang akan diberi

kredit serta melakukan pengawasan terhadap debitur atau nasabah yang diberi pinjaman.

2.2 Landasan Teori

2.2.1 Data mining

2.2.1.1 Pengertian Data mining

Ada beberapa definisi yang berkaitan dengan *data mining*, diantaranya adalah [5]:

1. *Data mining* adalah proses mengekstraksi pola yang tersembunyi dari dalam kumpulan data. Karena semakin banyak data yang diperoleh, dan kira-kira jumlahnya akan menjadi 2x lipat tiap 3 tahun, *data mining* berubah menjadi sarana penting untuk mengolah data mentah menjadi suatu informasi.
2. *Data mining* merupakan disiplin ilmu yang menggabungkan statistika, *machine learning*, *database* dan visualisasi. Kini sangat diperlukan baik dalam industri perbankan sampai mikrobiologi.
3. *Data mining* didefinisikan sebagai satu set teknik yang digunakan secara otomatis untuk mengeksplorasi secara menyeluruh dan membawa ke permukaan relasi-relasi yang kompleks pada set data yang sangat besar. Set data yang dimaksud di sini adalah set data yang berbentuk tabulasi, seperti yang banyak diimplementasikan dalam teknologi manajemen basis data relasional. Akan tetapi, teknik-teknik *data mining* dapat juga diaplikasikan pada representasi data yang lain, seperti *domain data spatial*, berbasis text, dan *multimedia* (citra). *Data mining* dapat juga didefinisikan

sebagai “pemodelan dan penemuan pola-pola yang tersembunyi dengan memanfaatkan data dalam volume yang besar”.

4. *Data mining* (kadang disebut juga *data or knowledge discovery*) adalah proses menganalisa data dari perspektif yang berbeda dan menyimpulkannya menjadi informasi-informasi penting yang dapat dipakai untuk meningkatkan keuntungan, memperkecil biaya pengeluaran, atau bahkan keduanya. Dengan *data mining* kita dapat menganalisis data dari berbagai dimensi dan sudut, mengelompokkannya, dan menyimpulkan relasi yang terbentuk. Secara teknis, *data mining* dapat disebut sebagai proses untuk menemukan korelasi atau pola dari ratusan atau ribuan *field* dari sebuah relasional *database* yang besar.

2.2.1.2 Tujuan *Data mining*

Dengan kemampuan *Data mining* (penambangan data) untuk mencari informasi bisnis yang berharga dari basis data yang sangat besar, yang dilakukan dapat dianalogikan dengan penambangan logam mulia dari lahan sumbernya, teknologi ini dipakai untuk [5] :

1. Prediksi *trend* dan sifat-sifat bisnis.

   Data mining mengotomatisasi proses pencarian informasi pemprediksi di dalam basis data yang besar. Pertanyaan-pertanyaan yang berkaitan dengan prediksi ini dapat cepat dijawab langsung dari data yang tersedia. Contoh dari masalah prediksi ini misalnya target pemasaran, peramalan kebangkrutan, dan bentuk-bentuk kerugian lainnya.

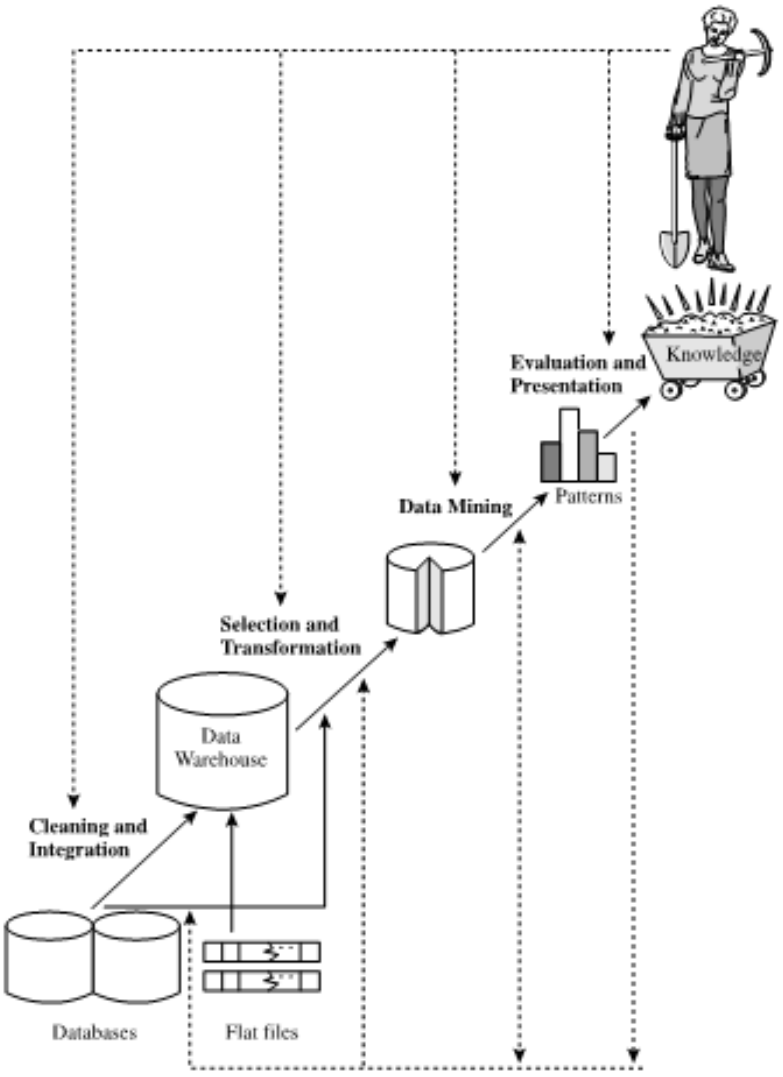
2. Penemuan pola-pola yang tidak diketahui sebelumnya.

   Kakas *data mining* "menyapu" basis data, kemudian mengidentifikasi pola-pola yang sebelumnya tersembunyi dalam satu sapuan. Contoh dari penemuan pola ini adalah analisis pada data penjualan ritel untuk mengidentifikasi produk-produk, yang kelihatannya tidak berkaitan, yang seringkali dibeli secara bersamaan oleh customer.

2.2.1.3 Tahapan *Data mining*

Dalam aplikasinya, *data mining* sebenarnya merupakan salah satu bagian proses *Knowledge Discovery in Database* (KDD) bukan sebagai teknologi yang utuh dan berdiri sendiri. Definisi KDD adalah proses *nontrivial* untuk mengidentifikasi pola dari data, yang valid, baru, berpotensi menjadi pengetahuan yang bermanfaat, dan dapat dimengerti [4], Sedangkan *data mining* sendiri merupakan salah satu bagian dari proses KDD yang bertugas untuk mengekstrak pola atau model dari data dengan menggunakan suatu algoritma yang spesifik.

Data mining merupakan suatu bagian yang penting dalam proses KDD terutama berkaitan dengan ekstrasi dan perhitungan pola-pola dari data yang dianalisis. Berikut tahapan-tahapan dalam proses KDD seperti yang ditunjukan gambar 2.2



Gambar 2.2 Tahapan proses *Knowledge Discovery in Database*

(Sumber [6] : *Data Mining Concepts and Techniques, 2nd edition,* Han, Kamber)

Adapun proses KDD diuraikan sebagai berikut : [1]

1. *Data Selection*

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

2. *Preprocessing*

   Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses mencakup antara lain cleaning membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (tipografi). Juga dilakukan proses *enrichment*, yaitu proses "memperkaya" data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal.

3. *Transformation*

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

4. *Data mining*

   Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan

5. *Interpretation / Evaluation*

   Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut dengan *interpretation*.

Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya atau tidak.

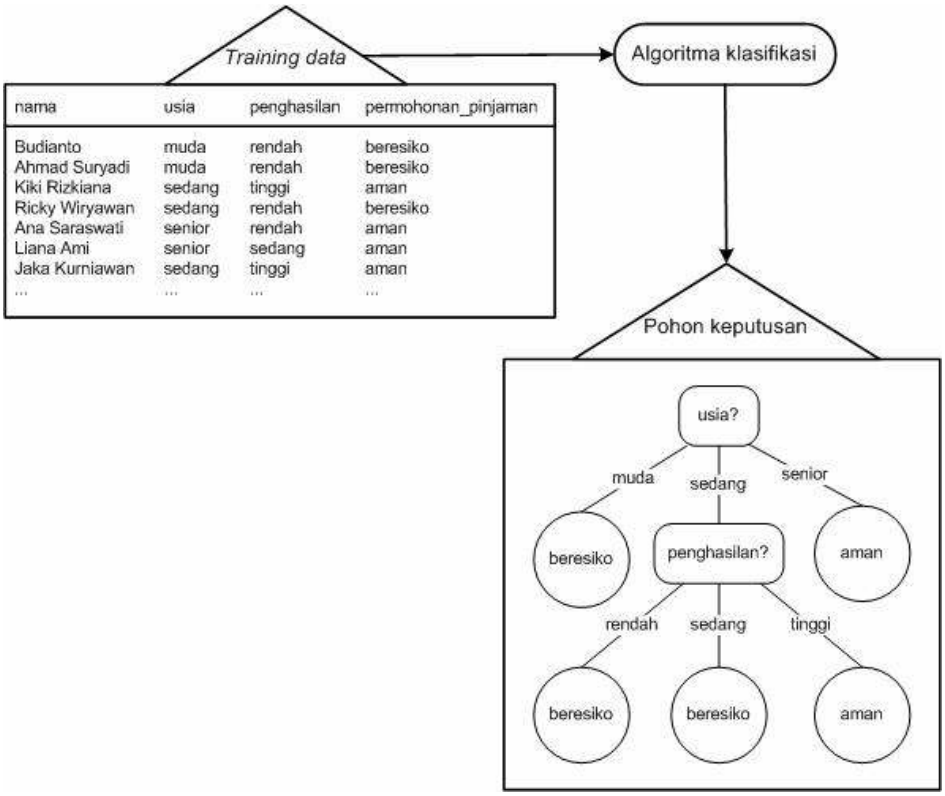
2.2.1.4 Klasifikasi

Klasifikasi adalah suatu fungsionalitas *data mining* yang akan menghasilkan model untuk memprediksi kelas atau kategori dari objek-objek di dalam basis data. Klasifikasi merupakan proses yang terdiri dari dua tahap, yaitu tahap pembelajaran dan tahap pengklasifikasian.

Pada tahap pembelajaran, sebuah algoritma klasifikasi akan membangun sebuah model klasifikasi dengan cara menganalisis training data. Tahap pembelajaran dapat juga dipandang sebagai tahap pembentukan fungsi atau pemetaan Y=F(X) di mana Y adalah kelas hasil prediksi dan X adalah tuple yang ingin diprediksi kelasnya. Selanjutnya, pada tahap pengklasifikasian, model yang telah dihasilkan akan digunakan untuk melakukan klasifikasi. [9]

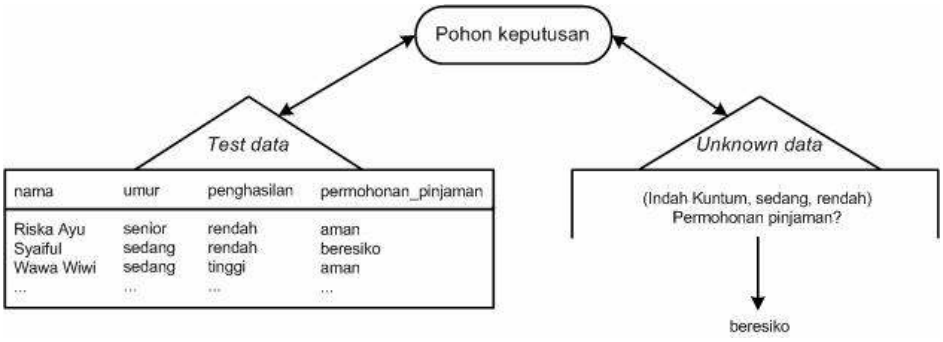
2.2.1.5 Pohon Keputusan (*decision tree*)

Pohon keputusan merupakan salah satu bentuk penggambaran model klasifikasi. Pada pohon keputusan, simpul dalam menyatakan pengujian terhadap suatu atribut, cabang menyatakan hasil dari suatu pengujian, sementara daun menyatakan kelas yang diprediksi [9]. Contoh pohon keputusan untuk kasus permohonan pinjaman pada tahap pengklasifikasian dapat dilihat pada gambar 2.3 dan pada tahap pembelajaran dapat dilihat pada gambar 2.4.



Gambar 2.3 Tahap Pengklasifikasian

(Sumber [9] : Penggunaan Pohon Keputusan untuk Data Mining, Khairina, I.K.)



Gambar 2.4 Tahap Pembelajaran

(Sumber [9] : Penggunaan Pohon Keputusan untuk Data Mining, Khairina, I.K.)

2.2.1.6 Algoritma C4.5

Salah satu algoritma induksi pohon keputusan yaitu ID3 (*Iterative Dichotomiser 3*). ID3 dikembangkan oleh J. Ross Quinlan. Dalam prosedur algoritma ID3, input berupa sampel training, label training dan atribut. Algoritma C4.5 merupakan pengembangan dari ID3 [10]. Secara umum algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut [7]:

1. Pilih atribut sebagai *root*
2. Buat cabang untuk masing-masing nilai
3. Bagi kasus dalam cabang
4. Ulangi proses untuk masing-masing cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Untuk memilih atribut sebagai akar, didasarkan pada nilai gain tertinggi dari atribut-atribut yang ada. Untuk menghitung gain digunakan rumus seperti tertera dalam rumus 1

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i) \quad \text{.....(1)}$$

Dengan,

S : Himpunan kasus

A : Atribut

n : Jumlah partisi A

$|S_i|$: Jumlah kasus pada partisi ke I

| S | : Jumlah kasus dalam S

Sedangkan untuk perhitungan nilai entropy dapat dilihat pada rumus 2

$$Entropy(S) = \sum_{i-1}^{n} - pi * \log_2 pi \quad \text{.....(2)}$$

dengan,

S : Himpunan kasus

A : Fitur

n : Jumlah partisi S

pi : Proporsi dari Si terhadap S

2.2.1.7 Contoh Kasus

Untuk memudahkan penjelasan mengenai algoritma C4.5 berikut ini disertakan contoh kasus yang dituangkan dalam Tabel 2.1 [7].

Tabel 2.1 Keputusan Bermain Tenis

NO	OUTLOOK	TEMPERATURE	HUMIDIY	WINDY	PLAY
1	Sunny	Hot	High	False	No
2	Sunny	Hot	High	True	No
3	Cloudy	Hot	High	False	Yes
4	Rainy	Mild	High	False	Yes
5	Rainy	Cool	Normal	False	Yes
6	Rainy	Cool	Normal	True	Yes
7	Cloudy	Cool	Normal	True	Yes
8	Sunny	Mild	High	False	No
9	Sunny	Cool	Normal	False	Yes
10	Rainy	Mild	Normal	False	Yes
11	Sunny	Mild	Normal	True	Yes

Bersambung,

Tabel 2.1 Keputusan Bermain Tenis (Lanjutan)

12	Cloudy	Mild	High	True	Yes
13	Cloudy	Hot	Normal	False	Yes
14	Rainy	Mild	High	True	No

Dalam kasus yang tertera pada Tabel 2.1, akan dibuat pohon keputusan untuk menentukan main tenis atau tidak dengan melihat keadaan cuaca, temperatur, kelembaban dan keadaan angin. Adapun langkah-langkahnya sebagai berikut :

1. Menghitung jumlah kasus, jumlah kasus untuk keputusan *Yes*, jumlah kasus untuk keputusan *No*, dan *Entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut *OUTLOOK*, *TEMPERATURE*, *HUMIDITY* dan *WINDY*. Setelah itu lakukan penghitungan *Gain* untuk masing-masing atribut. Hasil perhitungan ditunjukkan oleh Tabel 2.2

Tabel 2.2 Perhitungan Node 1

Node			Jumlah Kasus (S)	Tidak (S_1)	Ya (S_2)	Entropy	Gain
1	TOTAL		14	4	10	0.863120569	
	OUTLOOK						0.258521037
		CLOUDY	4	0	4		
		RAINY	5	1	4	0.721928095	
		SUNNY	5	3	2	0.970950594	
	TEMPERATURE						0.183850925
		COOL	4	0	4	0	
		HOT	4	2	2	1	
		MILD	6	2	4	0.918295834	
	HUMIDITY						0.370506501
		HIGH	7	4	3	0.985228136	
		NORMAL	7	0	7	0	
	WINDY						0.005977711
		FALSE	8	2	6	0.811278124	
		TRUE	6	4	2	0.918295834	

Baris TOTAL kolom *Entropy* pada Tabel 2.2 dihitung dengan rumus 2, sebagai berikut:

$$Entropy(Total) = (-\frac{14}{14} * \log_2(\frac{4}{14})) + (-\frac{10}{14} * \log_2(\frac{10}{14}))$$
$$Entropy(Total) = 0.863120569$$

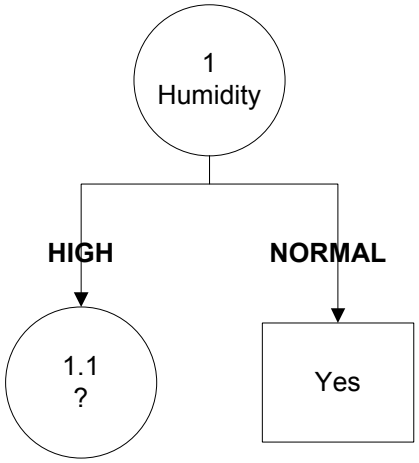
Sementara itu nilai Gain pada bari OUTLOOK dihitung dengan menggunakan rumus 1 sebagai berikut :

$$Gain(Total, Outlook) = Entropy(Total) - \sum_{i-1}^{n} \frac{|Outlook_i|}{|Total|} * Entropy(Outlook_i)$$
$$Gain(Total, Outlook) = 0.863120569 - ((\frac{4}{14} * 0) + (\frac{5}{14} * 0.723) + (\frac{5}{14} * 0.97))$$
$$Gain(Total, Outlook) = 0.23$$

Dari hasil pada Tabel 2.2 dapat diketahui bahwa atribut dengan *Gain* tertinggi adalah *HUMIDITY* yaitu sebesar 0.37. Dengan demikian *HUMIDITY* dapat menjadi node akar. Ada 2 nilai atribut dari *HUMIDITY* yaitu *HIGH* dan NORMAL. Dari kedua nilai atribut tersebut, nilai atribut NORMAL sudah mengklasifikasikan kasus menjadi satu yaitu keputusannya *Yes*, sehingga tidak perlu dilakukan perhitungan lebih lanjut, tetapi untuk nilai atribut *HIGH* masih perlu dilakukan perhitungan lagi. Dari hasil tersebut dapat digambarkan pohon keputusan sementaranya seperti tampak pada Gambar 2.5 berikut,

2. Menghitung jumlah kasus, jumlah kasus untuk keputusan *Yes*, jumlah kasus untuk keputusan *No*, dan *Entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut *OUTLOOK*, *TEMPERATURE* dan *WINDY* yang dapat menjadi node akar dari nilai atribut *HIGH*. Setelah itu lakukan penghitungan *Gain* untuk masing-masing atribut. Hasil perhitungannya

ditunjukkan oleh Tabel 2.3.

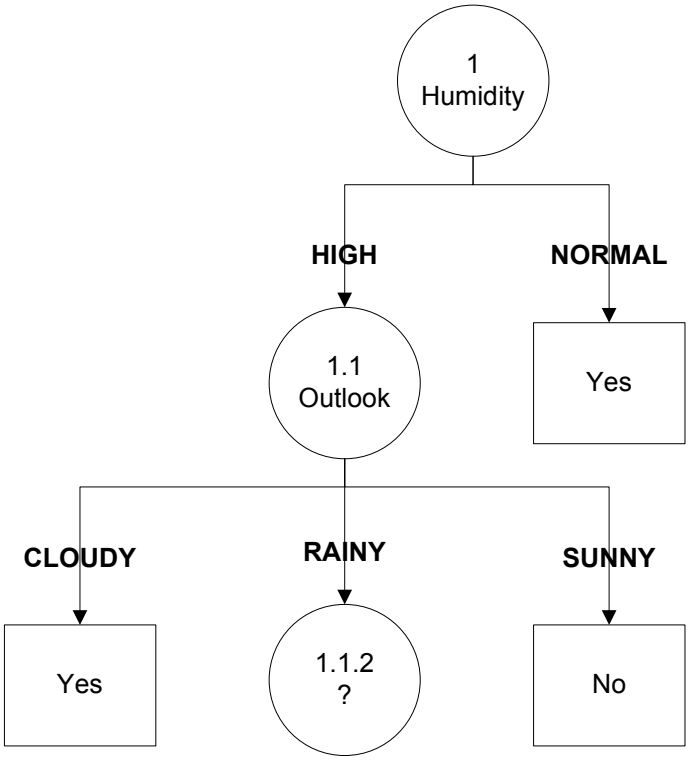


Gambar 2.5 Pohon Keputusan Hasil Perhitungan Node 1

Tabel 2.3 Perhitungan Node 1.1

NODE			Jumlah Kasus (S)	Tidak (S1)	Ya (S2)	Entropy	Gain
	HUMIDITY HIGH		7	4	3	0.985228136	
	OUTLOOK						0.69951385
		CLOUDY	2	0	2	0	
		RAINY	2	1	1	1	
		SUNNY	3	3	0	0	
	TEMPERATURE						0.020244207
		COOL	0	0	0	0	
		HOT	3	2	1	0.918295834	
		MILD	4	2	2	1	
	WINDY						0.020244207
		FALSE	4	2	2	1	
		TRUE	3	2	1	0.918295834	

Dari hasil pada Tabel 2.3 dapat diketahui bahwa atribut dengan *Gain* tertinggi adalah *OUTLOOK* yaitu sebesar 0.67. Dengan demikian *OUTLOOK* dapat menjadi node cabang dari nilai atribut *HIGH*. Ada 3 nilai atribut dari *OUTLOOK* yaitu *CLOUDY*, *RAINY* dan *SUNNY*. Dari ketiga nilai atribut tersebut, nilai atribut *CLOUDY* sudah mengklasifikasikan kasus menjadi 1 yaitu keputusan-nya *Yes* dan nilai atribut *SUNNY* sudah mengklasifikasikan kasus menjadi satu dengan keputusan *No*, sehingga tidak perlu dilakukan perhitungan lebih lanjut, tetapi untuk nilai atribut *RAINY* masih perlu dilakukan perhitungan lagi. Pohon keputusan yang terbentuk sampai tahap ini dapat dilihat pada Gambar 2.6



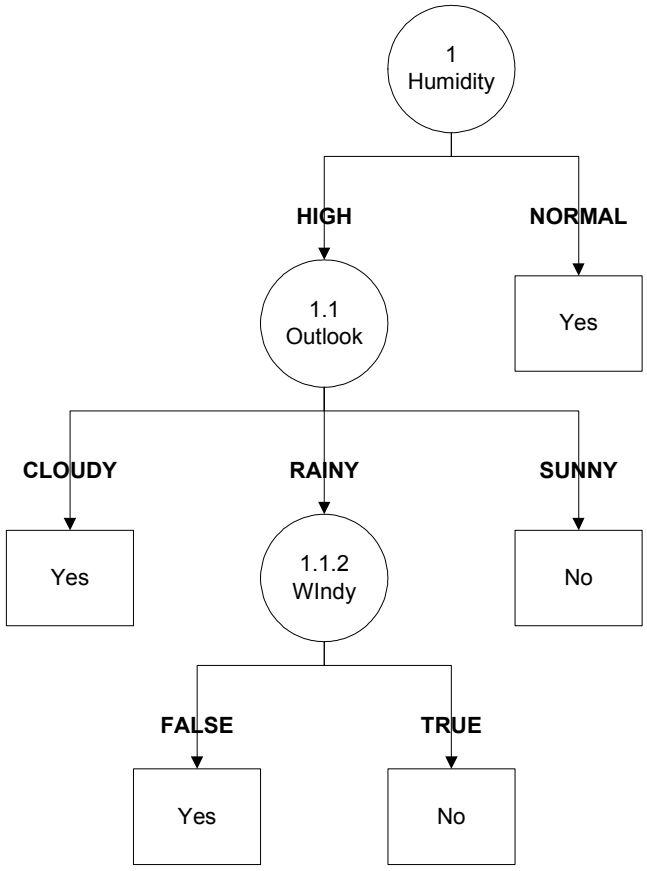
Gambar 2.6 Pohon Keputusan Hasil Perhitungan Node 1.1

3. Menghitung jumlah kasus, jumlah kasus untuk keputusan *Yes*, jumlah kasus untuk keputusan *No*, dan *Entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut *TEMPERATURE* dan *WINDY* yang dapat menjadi node cabang dari nilai atribut *RAINY*. Setelah itu melakukan penghitungan *Gain* untuk masing-masing atribut. Hasil perhitungan ditunjukkan oleh Tabel 2.4.

   Dari hasil pada tabel 2.4 dapat diketahui bahwa atribut dengan *Gain* tertinggi adalah *WINDY* yaitu sebesar 1. Dengan demikian *WINDY* dapat menjadi node cabang dari nilai atribut *RAINY*. Ada 2 nilai atribut dari *WINDY* yaitu *FALSE* dan *TRUE*. Dari kedua nilai atribut tersebut, nilai atribut *FALSE* sudah mengklasifikasikan kasus menjadi 1 yaitu keputusan-nya *Yes* dan nilai atribut *TRUE* sudah mengklasifikasikan kasus menjadi satu dengan keputusan *No*, sehingga tidak perlu dilakukan perhitungan lebih lanjut untuk nilai atribut ini. Pohon keputusan yang terbentuk sampai tahap ini dapat dilihat pada gambar 2.7. Dengan memperhatikan pohon keputusan pada Gambar 2.7, diketahui bahwa semua kasus sudah masuk dalam kelas. Dengan demikian, pohon keputusan pada Gambar 2.7 merupakan pohon keputusan terakhir yang terbentuk.

Tabel 2.4 Perhitungan Node 1.1.2

NODE			Jml Kasus (S)	Tidak (S1)	Ya (S2)	Entropy	Gain
1.1.2	HUMIDITY-HIGH dan OUTLOOK-RAINY		2	1	1	1	
	TEMPERATURE						0
		COOL	0	0	0	0	
		HOT	0	0	0	0	
		MILD	2	1	1	1	
	WINDY						1
		FALSE	1	0	1	0	
		TRUE	1	1	0	0	



Gambar 2.7 Pohon Keputusan Hasil Perhitungan Node 1.1.2

2.3 Metodologi Yang Digunakan

2.3.1 Pengertian UML

Unified Modeling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem [3].

2.3.2 Jenis diagram UML

UML memperkenalkan beberapa jenis diagram diantaranya, [3]

1. *Use Case Diagram*

   Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang ataupun sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

2. *Class Diagram*

   Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan atribut atau properti suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan

tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.. Class memiliki tiga area pokok :

a. Nama (dan *stereotype*)
b. Atribut
c. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

a. *Private*, tidak dapat dipanggil dari luar class yang bersangkutan
b. *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
c. *Public*, dapat dipanggil oleh siapa saja

3. Statechart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram).* Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk

lingkaran berwarna penuh dan berwarna setengah.

4. *Activity Diagram*

   Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

5. *Sequence Diagram*

   Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri dari dimensi *vertikal* (waktu) dan dimensi *horizontal* (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *respons*

dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

6. *Collaboration Diagram*

   Collaboration diagram menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. *Messages* dari level yang sama memiliki *prefiks* yang sama.

7. *Component Diagram*

   Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

8. *Deployment Diagram*

   Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras), bagaimana kemampuan

jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal.