BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Tempat Penelitian

Roti Riki Bandung adalah perusahaan yang bergerak dibidang industri *bakery* yang berada di kota Bandung, Jawa Barat. Roti Riki Bakery memproduksi produk olahan tepung yaitu roti manis (meses, *mocca*, coklat, kopyor dan *blueberry*) dan kue pia (pia kacang ijo dan pia coklat).

2.1.1 Profil Perusahaan

Roti Riki Bandung berdiri sejak tahun 1990 oleh Bapak H. Sudiarto yang berlokasi di Antapani Bandung. Pada saat itu, proses produksinya masih sedikit karana proses pengolahannya masih dilakukan secara manual. Produk yang pertama kali dihasilkan oleh Roti Riki Bandung yaitu roti manis. Roti manis yang diproduksi pada saat itu tidak banyak dan baru memiliki 3 varian rasa yaitu coklat, keju dan Blueberry.

Pada tahun 1992 Roti Riki Bandung berpindah lokasi ke daerah Cicaheum Bandung, Roti Riki Bandung mengontrak tempat selama 6 tahun. Pada saat itu Roti Riki Bandung mengalami perkembangan yang cukup baik dimana roti-roti yang diproduksi dapat diterima dan disukai oleh masyarakat. Seiring perkembangan perusahan membuat Roti Riki Bandung memerlukan tenaga kerja yang cukup banyak. Hingga akhirnya pada tahun 1998, Roti Riki Bandung dipindahkan ke tempat yang lebih luas yaitu di Jalan Raya Radio 2 Kp.Sukamanah, Kec. Rancaekek Wetan 40394 Bandung, Jawa Barat Indonesia. Proses produksi pun mulai menggunakan mesin.

Pada tahun 1999, Roti Riki Bandung Mulai mencoba menambah varian rasa roti yaitu kopyor dan durian. Selain itu, Roti Riki Bandung Mengembangkan Produksinya dengan menambahkan mesin-mesin baru untuk mempermudah produksi roti. Penambahan mesin baru membuat produksi di Roti Riki Bandung meningkat dan Roti Riki Bandung pada tahun 2002 menambahkan varian rasa baru yaitu *strawberry*, *mocca*, dan cream meses. Roti Riki Bandung mencoba menambah produk hasil olahan tepung terigu lainnya, dengan memproduksi kue pia pada tahun 2014. Kue pia yang diproduk si mempunyai 2 varian rasa yaitu

kacang hijau dan coklat. Produk roti yang dihasilkan oleh Roti Riki Bandung yaitu roti manis dan kue pia mulai menigkat produksinya dan sudah banyak diminati oleh konsumen hingga sekarang.

2.1.2 Logo Perusahaan

Roti Riki Bandung memiliki sebuah logo. Logo merupakan simbol, tanda gambar, merek dagang (*trandmark*) yang berfungsi sebagai lambang identitas dari suatu badan usaha dan tanda pengenal yang merupakan ciri khas perushaan. Gambar 2.1 merupakan logo dari Roti Riki Bandung.



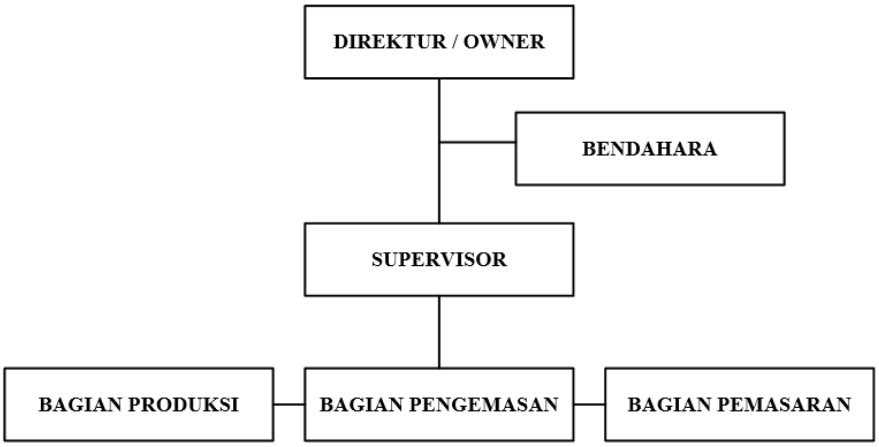
Gambar 2.1 Logo perusahaan roti riki

Penjelasan Logo :

1. Seorang *chef* dengan pakaian lengkap : Menggambarkan kesungguhan Roti Riki dalam memproduksi roti layaknya seorang chef yang membuat makanan.
2. Tangan kiri yang menggenggam roti : Menjelaskan bahwa perusahaan Roti Riki memproduksi hanya Roti.
3. Kepalan tangan dan jempol yang diangkat ke atas : Menjelaskan bahwa roti dan olahan tepung lainnya diproduksi dengan baik dan memiliki citarasa yang baik pula.
4. Tulisan Riki Bakery : adalah *trademark* nama perusahaan Roti Riki.

2.1.2 Struktur Organisasi Perusahaan

Untuk menjelaskan struktur organisasi yang menjadi sasaran kegiatan penelitian, maka struktur organisasi dari Roti Riki Bandung dapat dilihat pada gambar 2.2



Gambar 2.2 Struktur Organisasi Roti Riki

Berikut adalah *job description* dari masing-masing jabatan yang tertera di struktur organisasi.

1. Direktur

   Seorang direktur bertugas dan bertanggung jawab memimpin jalannya kelangsungan perusahaan, mengelola dan mengoperasikan aset perusahaan, membuat kebijakan-kebijakan dengan memperhatikan saran-saran dari masing-masing bagian yang ada serta melaksanakan koordinasi yang baik dengan instansi yang berkepentingan.

2. Bendahara

   Bendahara atau bagian keuangan bertugas dan bertanggung jawab atas kegiatan-kegiatan di lingkungan administrasi dang keuangan seperti proses penggajian, mengusahakan agar manajemen keuagan dapat dilaksanakan sesuai yang diharapkan, merencanakan jumlah dana yang akan disediakan dan mengamankan agar administrasi kelayakan terjamin.

3. Supervisor

   Supervisor atau pengawas perusahaan bertugas dan bertanggung jawab atas kegiatan-kegiatan di lingkungan produksi, pengemasan, dan

pemasaran seperti mengoordinasikan kegiatan dan tugas agar berjalan lancar, menempatkan setiap karyawan yang sesuai pada tugas dan pekerjaannya masing-masing, mengarahkan karyawan agar tugas dan pekerjaannya dapat berjalan lancar serta melakukan kontrol terhadap kegiatan atau pekerjaan yang dilakukan oleh karyawan.

4. Bagian Produksi

   Bagian produksi bertanggung jawab dalam hal produksi beserta penyediaan bahan baku untuk kegiatan produksi

5. Bagian Pengemasan

   Bagian pengemasang mempunyai tugas pokok untuk mengemas roti-roti yang telah diproduksi. Sebelum dikemas, bagian pengemasan wajib memeriksa keadaan roti (*quality control*) agar roti yang dikemas memenuhi standar yang telah ditetapkan.

6. Bagian Pemasaran

   Bagian pemasaran bertugas untuk melakukan kegiatan pemasaran, di mana bagian ini lah yang berinteraksi dengan distributor-distributor yang akan mengambil roti.

2.1.3 Visi dan Misi

Berikut adalah visi dan misi dari Roti Riki :

Visi : Menjadi produsen roti kemasan terkemuka di Indonesia

Misi : Sebagai perusahaan yang memproduksi roti kemasan murah dan citarasa yang tidak murahan.

2.2 Landasan Teori

2.2.1 Data

Data adalah suatu nilai yang merepresentasikan fakta di dunia nyata dan mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dsb. Nilai tersebut lalu direkam dalam bentuk angka, huruf, simbol, teks, gambar, atau kombinasinya [3].

2.2.2 Basis Data

Basis data adalah sekumpulan data yang saling terhubung dan mempunyai skema tertentu.

Menurut Fathansyah [6], basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file* atau tabel atau arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

2.2.3 *Database Management System*

Database management system adalah sistem yang digunakan untuk membuat dan mengelola basis data. Tujuan dari *DBMS* adalah menyediakan lingkungan yang mudah dan aman untuk penggunaan dan perawatan *database*. Dalam *DBMS* pengguna difasilitasi untuk mendefinisikan skema basis data yang disebut sebagai *data definition language (DDL).* Selain itu pengguna juga dapat membuat, mengubah, dan membuat *query* data yang biasanya disebut dengan *data manipulation language* (DML) [4].

2.2.4 *Data Mining*

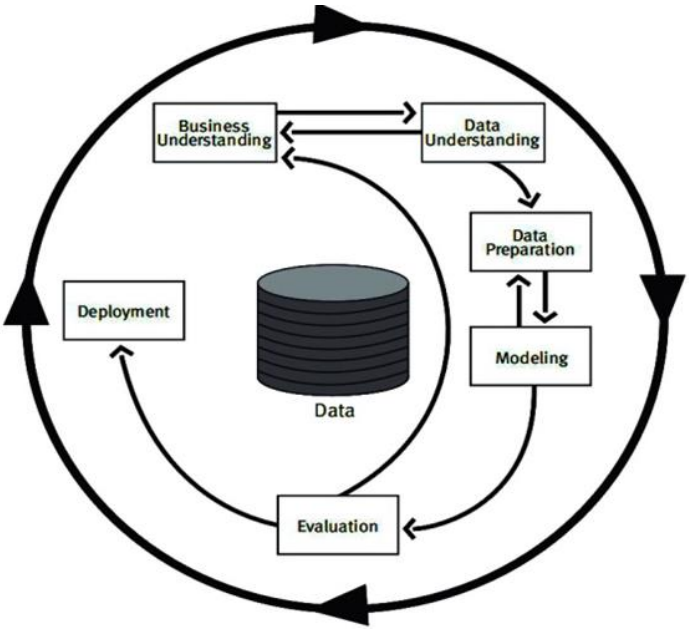
Data mining adalah kegiatan mengekstraksi atau menambang pengetahuan dari data yang berukuran atau berjumlah besar [7]. Nama lain yang sering digunakan dan memiliki makna yang sama dengan data *mining* adalah *knowledge mining from data, knowledge extraction, pattern analysis, data archaeology, data dredging*.

Data mining berusaha memecahkan masalah dengan menganalisis sekumpulan data yang sudah ada dalam basis data. Sedangkan hasil dari proses dari data mining adalah penemuan pengetahuan, informasi, pola, atau model.

2.2.5 Metode CRISP-DM

CRISP – DM (*Cross – Industry Standard Process of Data Mining*). CRISP – DM merupakan metodologi data *mining* yang digambarkan dengan model proses dan di dalamnya terdapat tugas-tugas yang harus dilakukan untuk

melakukan proyek data *mining* [2]. Tahapan dari CRISP – DM dapat dilihat pada gambar 2.3 :



Gambar 2.3 Model CRISP-DM [5]

Berikut penjelasan tahapan-tahapan yang terdapat dalam CRISP-DM :

1. *Business Understanding*

Tahap awal ini berfokus pada pemahaman tujuan dan persyaratan proyek dari perspektif bisnis, kemudian mengubah pengetahuan tersebut ke dalam definisi masalah data mining dan rencana awal dirancang untuk mencapai tujuan tersebut. Tahap pemahaman bisnis dibagi menjadi beberapa langkah, antara lain :

a. Penentuan Tujuan Bisnis

Pada tahap ini akan menghasilkan keluaran berupa latar belakang dan tujuan bisnis.

b. Penilaian Situasi

Tahapan ini menjelaskan situasi tentang semua sumber daya, kendala, asumsi, dan faktor lain yang harus dipertimbangkan dalam menentukan tujuan analisis data dan rencana proyek.

c. Penentuan Tujuan Data Mining

Pada tahap ini akan menghasilkan output berupa sasaran data mining.

d. Membuat Perencanaan Proyek

Pada tahap ini dijelaskan perencanaan proyek yang akan dilakukan.

2. Data Understanding

Pada fase pemahaman data dimulai dengan pengumpulan data awal dan memahami data yang akan digunakan dalam penelitian. Tahap pemahaman data dibagi menjadi beberapa langkah, antara lain :

a. Pengumpulan Data Awal

Pada langkah ini akan dikumpulkan data-data yang akan digunakan dalam data mining.

b. Penjelasan Data

Pada langkah ini akan dijelaskan data yang telah dikumpulkan di tahap awal. Dalam penjelasan data akan dijelaskan format dari data, kuantitas data, jumlah *record* dan *field* dalam setiap tabel, dsb.

3. *Data Preparation*

Tahap persiapan data mencakup semua kegiatan yang diperlukan untuk membangun data yang akan digunakan ke dalam pemodelan. Tahap persiapan data dibagi menjadi beberapa langkah, antara lain :

a. Pemilihan Data

Tahap pemilihan data bertujuan menentukan data yang akan digunakan. Pemilihan data ini meliputi pemilihan atribut ataupun pemilihan *record*.

b. Pembersihan Data

Tahap pembersihan data bertujuan untuk menghilangkan data berdasarkan data yang telah diversifikasi pada tahap verifikasi kualitas data

c. Penyusunan Data

Pada tahap ini akan disiapkan data yang telah dipilih akan diturunkan atribut atribut yang akan dipakai untuk proses selanjutnya.

d. Integrasi Data

Pada tahap ini atribut-atribut yang telah dipilih akan digabungkan dengan atribut-atribut dari data lainnya jika analisis yang dilakukan terhadap lebih dari satu data.

4. Modelling

Pada tahap ini akan dilakukan pemilihan dan penerapan model yang sesuai berdasarkan tujuan yang akan dicapai. Tahap pemodelan dibagi menjadi beberapa langkah, antara lain :

a. Pemilihan Teknik Pemodelan

Tahapan ini digunakan untuk memilih teknik pemodelan yang sesuai dengan permasalahan dan tujuan yang ingin dicapai.

b. Pembuatan Model

Pada tahap ini akan dijelaskan teknik pemodelan yang telah dipilih pada tahap pemilihan teknik pemodelan.

c. Pengujian Model

Pada tahap ini model yang telah dipilih akan diterapkan terhadap kasus uji.

5. Evaluation

Pada tahap ini akan dilakukan evaluasi terhadap model yang telah dibuat dalam tahap pemodelan, untuk memastikan model benar-benar telah mencapai tujuan bisnis. Pada tahap evaluasi terdapat beberapa langkah, antara lain :

a. Evaluasi Hasil

Pada tahap ini akan dievaluasi hasil dari pemodelan apakah sudah sesuai dengan tujuan bisnis atau tidak.

b. Peninjauan Proses

Pada tahap ini akan dilakukan peninjauan menyeluruh proses *data mining* yang telah dilakukan untuk memastikan adanya faktor penting atau tugas yang terabaikan.

c. Menentukan Langkah Selanjutnya

Pada tahap ini akan diputuskan langkah selanjutnya, apakah hasil pemodelan akan dilanjutkan ke tahap *deployment* atau tidak.

6. Deployment

Pembuatan model umumnya bukan akhir dari sebuah proyek, bahkan jika tujuan dari model ini adalah untuk meningkatkan pengetahuan dari data, maka harus disajikan dalam bentuk yang dapat digunakan oleh pengguna.

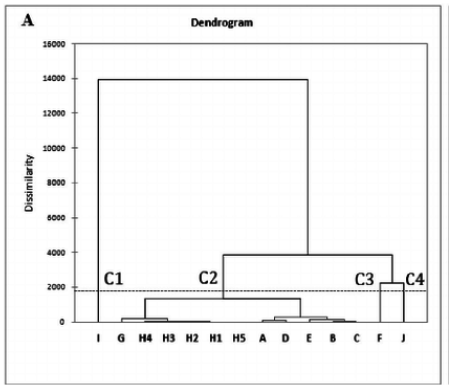
2.2.6 Teknik *Clustering*

Clustering adalah proses mengelompokkan objek berdasarkan informasi yang diperoleh dari data yang menjelaskan hubungan antar objek dengan prinsip untuk memaksimalkan kesamaan antar anggota satu kelas dan meminimumkan kesamaan antar kelas atau *cluster* [4]. Tujuannya menemukan *cluster* yang berkualitas dalam waktu yang layak. *Clustering* dalam data *mining* berguna untuk menemukan pola distribusi di dalam sebuah data set yang berguna untuk proses analisa data.

Kesamaan objek biasanya diperoleh dari kedekatan nilai-nilai atribut yang menjelaskan objek-objek data, sedangkan objek-objek data biasanya direpresentasikan sebagai sebuah titik dalam ruang multidimensi. Dengan menggunakan *clustering*, dapat diidentifikasi daerah yang padat, pola-pola distribusi secara keseluruhan dan keterkaitan yang menarik antara atribut-atribut data. Dalam data *mining* usaha difokuskan pada metode-metode penemuan untuk *cluster* pada basis data berukuran besar secara efektif dan efisien.

2.2.7 Analisis *Cluster* Berbasis Hierarki (*Hierarchical Clustering*)

Membuat hierarki berupa *dendrogram* di mana data yang mirip akan disimpan di hierarki yang berdekatan. Metode ini terbagi menjadi dua yaitu *buttom-up* yang menggabungkan *cluster* kecil menjadi *cluster* lebih besar dan *topdown* yang memecah *cluster* besar menjadi *cluster* yang lebih kecil.



Gambar 2.4 Dendrogram Hasil Pengelompokan Metode AHC [8]

Dalam metode *clustering* hal yang penting adalah mengukur kemiripan atau ketidakmiripan data yang akan dikelompokkan. *Similaritas* (kemiripan) antara dua objek merupakan ukuran numerik dari seberapa mirip dua buah objek. *Similaritas* mempunyai nilai yang lebih tinggi jika objek-objek tersebut sama. *Range* nilai *similaritas* adalah [0,1]. *Dissimilaritas* (ketidakmiripan) antara dua objek merupakan ukuran numerik dari seberapa berbeda dua buah objek. *Dissimilaritas* bernilai lebih rendah jika objek-objek tersebut mirip. Minimum *dissimilaritas* adalah 0 dan batas atasnya bervariasi. *Dissimiliratas* sering disebut juga dengan *distance* (jarak) [10]. Salah satu cara pengukuran jarak adalah menggunakan jarak *Euclidian* (*Euclidian Distance*), yang dinyatakan dengan persamaan (2.1) berikut :

$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2} \qquad \textbf{..... persamaan (2. 1)}$$

Di mana :

n adalah jumlah dimensi (*attribute*)

pk dan qk adalah atribut (komponen) ke-k atau objek data p dan q

Jika skala nilai dari objek-objek yang diukur berbeda maka diperlukan standarisasi.

2.2.8 *Agglomerative Hierarchical Clustering* (AHC)

Pengelompokan data menggunakan *agglomerative hierarchical clustering* (AHC) diawali dengan merepresentasikan setiap objek pada data sebagai satu kelompok, kemudian dilakukan perhitungan jarak (*distance measure*) antar kelompok tersebut. Setelah itu dua kelompok yang memiliki jarak terdekat digabungkan menjadi sebuah kelompok baru. Kemudian jarak antara kelompok yang baru dengan kelompok lain dihitung menggunakan metode perhitungan jarak antar kelompok *complete linkage*. Proses perhitungan jarak dan penggabungan dua kelompok dilakukan secara iteratif. *Complete linkage* disebut juga maksimum *link*, di mana *similaritas* dari dua *cluster* didasarkan pada dua titik paling dekat dari dua *cluster* yang berbeda. Untuk mencari *complete linkage* dapat menggunakan persamaan (2.2) berikut ini :

$$d_{uv} = \max\{d_{uv}\}, d_{uv} \in D \qquad \textbf{..... persamaan (2.2)}$$

Berikut ini merupakan langkah-langkah dalam algoritma *Agglomerative Hierarchical Clustering* (AHC) [4]:

1. Setiap objek data dianggap sebagai *cluster*. Jika n = jumlah data dan k = jumlah *cluster*, maka k = n.
2. Menghitung matriks jarak antar *cluster* (menggunakan *euclidean distance*)
3. Cari dua *cluster* yang mempunyai jarak paling dekat (*complete linkage*) lalu gabungkan dua *cluster* tersebut, sehingga k = k-1.
4. Jika k > 1, ulangi langkah 2 dan 3
5. Jika k = 1, perulangan selesai.
6. *Cluster* dihasilkan dari pemotongan dendrogram pada level yang tepat. Pemotongan dendrogram dapat dilakukan dengan berdasarkan pada *agglomeration schedule*. Jumlah *cluster* bisa didapatkan dengan mencari perbedaan koofisien jarak dari setiap iterasinya.

2.2.9 *Agglomeration Schedule*

Agglomeration schedule adalah ringkasan numerik dari solusi *cluster* yang dihasilkan dari proses *clustering* menggunakan metode hierarki [4]. Berikut ini merupakan contoh dari *agglomeration schedule* :

Tabel 2.1 Contoh Tabel *Agglomeration Schedule*

Stage	***Cluster Combined***		***Coefficients***	***Stage Cluster First Appears***		***Next Stage***
	Cluster 1	***Cluster 2***		***Cluster 1***	***Cluster 2***	
1	8	11	1.260	0	0	7
2	6	7	1.579	0	0	4
3	2	9	1.625	0	0	7
4	1	6	2.318	0	2	6
5	3	5	2.619	0	0	8
6	1	10	3.670	4	0	10
7	2	8	4.420	3	1	8
8	2	3	4.505	7	5	9
9	2	4	4.774	8	0	10
10	1	2	5.718	6	9	0

Penjelasan tabel *agglomeration schedule* :

1. *Stage* merupakan tahapan atau iterasi yang telah dilakukan pada proses *clustering* menggunakan metode hierarki.
2. *Cluster combined* merupakan *cluster* yang digabungkan pada setiap *stage*nya.
3. Cooficients merupakan jarak dari dua *cluster* yang digabungkan pada setiap *stage*nya yang dicari dengan menggunakan rumus jarak.
4. *Stage cluster first appear* merupakan *stage* apabila *cluster* yang digabungkan telah digabungkan dengan *cluster* lain pada *stage* sebelumnya. Jika *cluster* yang digabungkan belum pernah digabungkan pada *stage* sebelumnya, maka akan diisi dengan nilai 0.
5. *Next stage* merupakan *stage* selanjutnya saat *cluster* akan digabungkan dengan *cluster* lain.

Berikut ini contoh penjelasan tahapan pada *agglomeration schedule* : Pada tahap pertama, *cluster* yang digabungkan adalah *cluster* 8 dan *cluster* 11, karena memiliki jarak yang paling kecil.

Tabel 2.2 ***Agglomeration Schedule*** **Tahap Pertama**

Stage	*Cluster Combined*		*Coefficients*	Stage Cluster First Appears		Next Stage
	Cluster 1	*Cluster 2*		*Cluster 1*	*Cluster 2*	
1	8	11	1.260	0	0	7
2	6	7	1.579	0	0	4
3	2	9	1.625	0	0	7
4	1	6	2.318	0	2	6
5	3	5	2.619	0	0	8
6	1	10	3.670	4	0	10
7	2	8	4.420	3	1	8
8	2	3	4.505	7	5	9
9	2	4	4.774	8	0	10
10	1	2	5.718	6	9	0

Cluster yang telah digabungkan pada *stage* pertama akan digabungkan kembali pada *stage* ke-7.

Tabel 2.3 ***Agglomeration Schedule Next Stage***

Stage	*Cluster Combined*		*Coefficients*	*Stage Cluster First Appears*		*Next Stage*
	Cluster 1	*Cluster 2*		*Cluster 1*	*Cluster 2*	
1	8	11	1.260	0	0	7
2	6	7	1.579	0	0	4
3	2	9	1.625	0	0	7
4	1	6	2.318	0	2	6
5	3	5	2.619	0	0	8
6	1	10	3.670	4	0	10
7	2	8	4.420	3	1	8
8	2	3	4.505	7	5	9
9	2	4	4.774	8	0	10
10	1	2	5.718	6	9	0

Pada *stage* 7, *cluster* yang telah digabungkan pada *stage* 1 dan *stage* 3 akan digabungkan kembali. *Cluster* yang akan dihasilkan berikutnya akan muncul pada *stage* 8.

Tabel 2.4 ***Agglomeration Schedule Stage 7***

Stage	*Cluster Combined*		*Coefficients*	*Stage Cluster First Appears*		*Next Stage*
	Cluster 1	*Cluster 2*		*Cluster 1*	*Cluster 2*	
1	*8*	*11*	*1.260*	*0*	*0*	*7*
2	*6*	*7*	*1.579*	*0*	*0*	*4*
3	*2*	*9*	*1.625*	*0*	*0*	*7*
4	*1*	*6*	*2.318*	*0*	*2*	*6*
5	*3*	*5*	*2.619*	*0*	*0*	*8*
6	*1*	*10*	*3.670*	*4*	*0*	*10*
7	*2*	*8*	*4.420*	*3*	*1*	*8*
8	*2*	*3*	*4.505*	*7*	*5*	*9*
9	*2*	*4*	*4.774*	*8*	*0*	*10*
10	*1*	*2*	*5.718*	*6*	*9*	*0*

Ketika *cluster* yang akan digabungkan berjumlah sangat banyak, maka tabel *agglomeration schedule* yang dihasilkan akan bertambah panjang, tetapi akan mempermudah untuk melihat jarak terbesar dari seluruh *cluster* dibandingkan dengan melihat dari dendrogram. Jumlah *cluster* terbaik adalah selisih jarak terbesar.

Tabel 2.5 Selisih Jarak Pada ***Agglomeration Schedule***

Stage	*Cluster Combined*		*Coefficients*	*Stage Cluster First Appears*		*Next Stage*
	Cluster 1	*Cluster 2*		*Cluster 1*	*Cluster 2*	
1	8	11	1.260	0	0	7
2	6	7	1.579	0	0	4
3	2	9	1.625	0	0	7
4	1	6	2.318	0	2	6
5	3	5	2.619	0	0	8
6	1	10	3.670	4	0	10
7	2	8	4.420	3	1	8
8	2	3	4.505	7	5	9
9	2	4	4.774	8	0	10
10	1	2	5.718	6	9	0

Selisih jarak terbesar terdapat pada antara *stage* 5 dan *stage* 6, yang menunjukkan solusi jumlah *cluster* terbaik adalah sebanyak 6 *cluster*. Hal yang sama dapat dilihat juga dari dendrogram yang dihasilkan dari proses *clustering* menggunakan metode hierarki.

2.2.10 *Object Oriented Programing*

Pemograman berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan

objek yang berisi data dan operasi yang diberlakukan terhadapnya. Menggunakan pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek dunia nyata.

Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancangan, pemograman, dan pengujian perangkat lunak. Berikut adaah beberapa konsep dasar yang harus dipahami pemograman berorientasi objek :

1. Kelas (*Class*)

   Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dari kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan dan kelas semula dapat diwariskan ke kelas yang baru.

2. Objek (*Object)*

   Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, strutur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

3. Metode (*Method*)

   Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada terstruktur. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri.

4. Atribut (*Attribute*)

   Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki

oleh objek dalam kelas objek. Atribut secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.

5. Abstraksi (*Abstraction*)

   Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapulasi (*Encapsulation*)

   Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerja.

7. Pewarisan (*Inheritance*)

   Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya.

8. Antarmuka (*Interface*)

   Antarmuka atau *interface* sangat mirip dengan kelas, tetapi tanpa atribut kelas dan tanpa memiliki metode yang dideklarasikan. Antarmuka biasanya digunakan agar kelas lain tidak langsung mengakses ke suatu kelas.

9. *Reusability*

   Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

    Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor, dan kereta.

11. Komunikasi antar objek

    Komunikasi antar-objek dilakukan lewat pesan (*message*) yang dikirim dan satu objek ke objek lainnya.

12. Polimorfisme (*Polymorphism*)

Kemampuan suatu objek untuk digunakan dibanyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

13. *Package*

Package adalah sebuah kontainer atau kemasan yang daoat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda.

2.2.11 *Unified Modeling Language* (UML)

Unified modeling language adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berorientasi objek [9]. Beberapa model yang akan digunakan untuk menggambarkan sistem yang akan dibangun dalam penelitian ini adalah :

1. *Use Case Diagram*

Use case diagram digunakan untuk memodelkan fungsionalitas-fungsionalitas sistem atau perangkat lunak dilihat dari pengguna yang ada di luar sistem (yang sering dinamakan sebagai *actor*). Kegunaan dari *use case diagram* adalah untuk mendaftarkan aktor-aktor dan *use case* serta memperlihatkan aktor mana yang berpartisipasi dalam masing-masing *use case*. Aktor merupakan representasi dari orang yang berada di luar sistem dan akan berinteraksi dengan sistem. *Use case* adalah fungsionalitas sistem atau perangkat lunak.

2. *Class Diagram*

Class diagram merupakan diagram yang menggambarkan hubungan antara *class* yang satu dengan *class* yang lain. *Class* didefinisikan dengan tiga area pokok, yaitu nama *class*, atribut, dan operasi atau *method*. Pada *class*, atribut dan *method* akan dilakukan encapsulation, jenis-jenis *encapsulation* adalah *public*, *private*, dan *protected*.

3. *Activity Diagram*

Activity diagram bertujuan memodelkan komputasi-komputasi dan aliran-aliran kerja yang terjadi dalam sistem atau perangkat lunak. Suatu diagram

aktivitas memuat didalamnya *activity state* di mana suatu *activity state* merepresentasikan eksekusi pernyataan dalam suatu prosedur atau kinerja suatu aktivitas dalam satu aliran kerja.

4. *Sequence Diagram*

Sequence diagram digambarkan secara *vertical* dan *horizontal*. Pada bagian vertikal digambarkan sumbu waktu (*lifeline*), waktu bertambah dari atas ke bawah. Pada bagian horizontal memperlihatkan objek-objek yang terlibat. Selama objek ada, peran digambarkan menggunakan garis tegas. Selama aktivitas prosedur pada objek aktif, garis waktu digambarkan dengan garis ganda. Pesan-pesan digambarkan sebagai suatu tanda panah dari garis waktu suatu objek ke garis waktu objek lainnya.

2.2.12 MySQL

Database MySQL merupakan sistem manajemen basis data SQL yang sangat terkenal dan bersifat *open source*. MySQL dibangun, didistribusikan, dan didukung oleh MySQL AB. MySQL AB merupakan perusahaan komersial yang dibiayai oleh pengembang MySQL. MySQL mempunyai dua macam lisensi yaitu lisensi yang bersifat *open source* dengan menggunakan GNU (*General Public License*) dan lisensi kedua berupa *Standard Commercial License* yang dapat dibeli dari MySQL AB.

Dalam konteks bahasa SQL, informasi disimpan dalam tabel-tabel yang secara logis merupakan struktur dua dimensi yang tersimpan atas baris-baris data (*row* atau *record*) yang berada dalam satu atau lebih kolom (*column*). Baris pada tabel disebut *insance* dari data sedangkan kolom sering disebut sebagai *atributes* atau *field*. MySQL yang berjalan di sistem operasi *Windows* terbukti sangat stabil. MySQL versi *Windows* mempunyai fitur yang sama dengan MySQL versi *Unix* [10].