BAB 2
TINJAUAN PUSTAKA

2.1 Ruang Lingkup Objek Penelitian

Pada tahap ini merupakan tahap peninjauan terhadap tempat penelitian yaitu Max Print yang berlokasi di jalan PHH. Mustofa No. 20 Kota Bandung.

2.1.1 Sejarah Perusahaan

Max Print yang memiliki nama perusahaan CV. Maxprint Ganesa Optindo adalah sebuah perusahaan yang bergerak di bidang *digital printing*. Max Print melayani print photo, kartu nama, *x-banner*, baligho, spanduk, sticker dan lain-lain. Saat ini Max Print telah memiliki kurang lebih 5 cabang yang terletak di kota Bandung, Garut dan Tasikmalaya.

Dalam pelayanan pelanggannya, Max Print mengelompokkan konsumen ke dalam 2 jenis, yaitu konsumen mitra dan konsumen individu. Konsumen mitra adalah konsumen yang telah menjalin kerja sama dengan max print. Konsumen mitra terdiri dari perusahaan konveksi, kantor-kantor dan lain-lain. Sedangkan konsumen individu adalah konsumen biasa yang tidak terikat kerja sama dengan Max Print.

Max Print menjadikan pelanggan sebagai prioritas utama dalam proses bisnis yang dilakukan. Hal ini sesuai dengan visi dan misi yang dijunjung tinggi oleh Max Print. Adapun visi Max Print yaitu menjadi perusahaan *printing* yang eksis, terdepan dalam layanan dan kinerja guna mencapai keberkahan. Sementara itu terdapat 2 misi Max Print yaitu yang pertama adalah memberikan layanan prima dan solusi yang bernilai tambah. Misi yang kedua yaitu berorientasi pada kepentingan pelanggan.

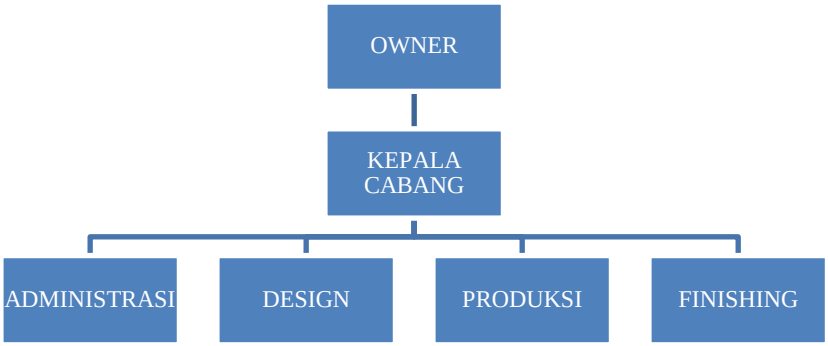
Max Print memiliki sebuah logo perusahaan yang digunakan sebagai identitas perusahaan yang akan membedakan Max Print dengan perusahaan lain. Adapun logo yang dimiliki oleh Max Print dapat dilihat pada gambar 2.1.



Gambar 2.1 Logo Max Print

2.1.2 Struktur Organisasi Perusahaan

Struktur organisasi Max Print dapat dilihat pada gambar 2.2.



Gambar 2.2 Struktur Organisasi Max Print

2.2 Landasan Teori

Landasan teori merupakan kumpulan dari beberapa definisi dan konsep yang disusun secara sistematis mengenai variabel-variabel dalam penelitian. Landasan teori ini akan menjadi acuan dasar penulis dalam melakukan penelitian melalui uraian-uraian secara teoritis.

2.2.1 Konsep Dasar Sistem

Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sebuah komponen fungsional (dengan satu fungsi/tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses/pekerjaan tertentu. Informasi adalah data yang telah diolah menjadi sebuah bentuk yang

berarti bagi penerimanya dan bermanfaat dalam pengambian keputusan saat ini atau saat mendatang [6].

Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada para pemakai [6].

Sistem informasi dibentuk dari dua kata yaitu sistem dan informasi. Sistem adalah hubungan satu unit dengan unit-unit lainnya yang saling berhubungan satu sama lain dan tidak dapat dipisahkan serta menuju suatu kesatuan untuk mencapai tujuan tertentu, apabila terdapat gangguan atau kendala maka tujuan yang tercapai pun akan terganggu [3].

Kualitas informasi dapat dipengaruhi oleh enam hal, yaitu :

a. Relevan
b. Akurat
c. Tepat waktu
d. Ekonomis
e. Efisien
f. Dapat dipercaya

2.2.2 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen, batas sistem, lingkungan luar sistem, penghubung, masukan, keluaran, pengolah, dan sasaran atau tujuan [6].

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai karakteristik dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini dapat dipandang sebagai satu kesatuan dan menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga merugikan. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, jika tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem (*Interface*)

Merupakan suatu media yang menghubungkan antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya. Keluaran dari satu subsistem akan menjadi masukan untuk subsistem lainnya melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Masukan (*input*) adalah energi yang dimasukkan ke dalam sistem yang berupa masukan perawatan (*maintenance input*) agar sistem tersebut dapat beroperasi dan masukan sinyal yang diproses untuk menghasilkan keluaran. Sebagai contoh di dalan sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputer dan data adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang telah diproses dan dapat diklasifikasikan antara lain menjadi *output* yang langsung digunakan, *output* yang bisa dijadikan *input* untuk subsistem yang lain atau kepada supra sistem serta *output* yang bagi

sistem tidak digunakan lagi (sisa pembuangan) tapi bagian sistem lain sangat berguna.

7. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merupakan masukan menjadi keluaran. Sistem akuntansi akan megolah data-data transaksi menjadi laporan-laporan keuangan dan laporan-laporan yang dibutuhkan oleh manajemen.

8. Sasaran dan Tujuan

Suatu sistem dikatakan berhasil jika mengenai sasaran atau tujuannya, kalau sistem tidak mempunyai sasaran maka sistem tidak akan ada. Sasaran sangat berpengaruh pada masukan dan keluaran yang dihasilkan.

2.2.3 Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lainnya. Karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi yang ada didalam sistem tersebut. Oleh karena itu sistem dapat diklasifikasikan kedalam beberapa sudut pandang [6].

1. Sistem diklasifikasikan sebagai sistem alamiah dan sistem buatan manusia.

Sistem alamiah adalah sistem yang terjadi karena proses alam tidak dibuat oleh manusia (ditentukan dan tunduk kepada kehendak sang pencipta alam). Sedangkan sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut dengan *human-machine system* atau ada yang menyebut dengan *machine system*.

2. Sistem diklasifkasikan sebagai sistem tertutup dan sistem terbuka.

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak berpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan pihak luarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyatannya tidak ada sistem yang benar-benar tertutup, yang ada hanya *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan tepengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk

lingkungan luar atau subsistem yang lain. Karena sistem sifatnya terbuka dan terpengaruh lingkungan luarnya, maka suatu sistem harus mempunyai suatu pengendalian yang baik. Sistem yang baik harus dirancang sedemikian rupa, sehingga secara relatif tertutup karena sistem tertutup akan bekerja secara otomatis dan terbuka hanya untuk pengaruh yang baik saja.

2.2.4 *Customer Relationship Management* (CRM)

CRM adalah singkatan dari *customer relationship management*. CRM memiliki arti sebagai suatu konsep manajemen secara menyeluruh yang berpusat pada pelanggan sebagai faktor utama untuk meningkatkan keunggulan bersaing dan bertujuan memperoleh tingkat kepuasan pelanggan secara maksimum yang akan membawa keuntungan bagi perusahaan dengan biaya yang relatif lebih rendah. CRM adalah strategi bisnis dengan menerapkan teknologi informasi pada perusahaan secara komprehensif, handal, dan terintegrasi berdasarkan sudut pandang pelanggan, dengan begitu semua proses dan interaksi saling membantu mempertahankan dan mengembangkan hubungan yang bermanfaat [9].

Customer Relationship Management (CRM) merupakan strategi yang diimplementasikan secara luas untuk mengatur hubungan antara perusahaan dengan pelanggan, *client*, dan calon pelanggan. CRM menggunakan suatu teknologi untuk mengorganisasi, otomasi, dan sinkronisasi semua proses bisnis, terutama kegiatan–kegiatan penjualan; tetapi juga kegiatan–kegiatan pemasaran, pelayanan pelanggan dan dukungan teknis. Prinsip dalam CRM ialah memperoleh pelanggan, mempertahankan pelanggan, dan memberikan pelayanan yang berkualitas sehingga pelanggan merasa puas dan menjadi loyal terhadap perusahaan. Biasanya CRM mengacu pada tiga hal, pertama, CRM merupakan bidang yang berfokus pada pelanggan seperti pemasaran penjualan, dan dukungan pelanggan. Kedua, CRM menggunakan alat yang memiliki fungsi sebagai *sales force automation* (SFA). Ketiga, CRM merupakan proses yang mengelola hubungan dengan pelanggan.

2.2.5 Jenis-Jenis *Customer Relationship Management* (CRM)

Ada beberapa jenis dari *Customer Relationship Management* (CRM) yaitu[1]:

1. Strategis (*Strategic*)

   Perspektif strategis berfokus pada pengembangan budaya bisnis berorientasi pelanggan. Budaya ini diterapkan untuk dapat memenangkan dan menjaga pelanggan, dengan memberikan pelayanan yang lebih baik dibandingkan pesaing. Budaya ini juga mengharuskan perusahaan untuk mengetahui lebih banyak mengenai pelanggan supaya dapat melayani pelanggan secara lebih intim untuk menghasilkan keuntungan lebih bagi perusahaan.

2. Operasional (*Operational*)

   Perspektif operasional berfokus pada otomatisasi dan meningkatkan tatap muka dengan pelanggan dan proses bisnis penunjang bagi pelanggan. Fokus otomatisasi ini memungkinkan adanya integrasi antara fungsional perusahaan seperti fungsi pemasaran, penjualan, dan pelayanan.

3. Analitis (*Analytical*)

   Pada perspektif ini berfokus pada proses menangkap, menyimpan, mengekstrak, mengintegrasi, memproses, menerjemahkan, mendistribusikan, menggunakan dan melaporkan data yang terkait dengan pelanggan untuk meningkatkan nilai pelanggan dan perusahaan. Data yang digunakan untuk menganalisis informasi mengenai pelanggan seperti data pemasaran, data pembayaran, data pengiriman, data pribadi pelanggan, data interaksi pelanggan dalam sistem, dan data transaksi pelanggan.

4. Kolaborasi (*Collaborative*)

   Fokus kolaborasi pada CRM memungkinkan perusahaan yang bekerja pada sebuah *cluster* bisnis atau dalam proses kerja sama dalam rantai pasok untuk dapat berbagi data lintas fungsional dan lintas perusahaan. Persebaran informasi yang terjadi memungkinkan perusahaan untuk dapat mengidentifikasi permasalahan yang terjadi dan mengembangkan pelanggan dengan melayani mereka lebih baik.

2.2.6 *Zachman Framework*

Zachman Framework adalah skema - persimpangan antara dua klasifikasi historis yang telah digunakan untuk ribuan tahun. Klasifikasi pertama adalah dasar-dasar komunikasi yang ditemukan di interogatif primitif: Apa, Bagaimana, Kapan, Siapa, Dimana, dan Mengapa. Ini adalah integrasi jawaban atas pertanyaan-pertanyaan yang memungkinkan deskripsi yang komprehensif, komposit ide yang kompleks. Klasifikas kedua berasal dari reifikasi, transformasi ide abstrak menjadi instansiasi yang awalnya didalilkan oleh filsuf Yunani kuno dan diberi label dalam *Zachman Framework*: Identifikasi, Definisi, Representasi, Spesifikasi, Konfigurasi dan Instansiasi[4].

Zachman Framework biasanya digambarkan sebagai matriks 6 x 6 dengan kata tanya komunikasi sebagai kolom dan transformasi reifikasi sebagai baris. Klasifikasi kerangka diwakili oleh sel, yaitu, persimpangan antara kata tanya dan transformasi. Matriks ini tentu merupakan total set representasi deskriptif yang relevan untuk menggambarkan sesuatu hal khususnya dalam suatu perusahaan. Matriks *Zachman Framework* dapat dilihat pada gambar2.3[4]

	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*
Objective/Scope (contextual) *Role: Planner*	List of things important in the business	List of Business Processes	List of Business Locations	List of important Organizations	List of Events	List of Business Goal & Strategies
Enterprise Model (conceptual) *Role: Owner*	Conceptual Data/ Object Model	Business Process Model	Business Logistics System	Work Flow Model	Master Schedule	Business Plan
System Model (logical) *Role:Designer*	Logical Data Model	System Architecture Model	Distributed Systems Architecture	Human Interface Architecture	Processing Structure	Business Rule Model
Technology Model (physical) *Role:Builder*	Physical Data/Class Model	Technology Design Model	Technology Architecture	Presentation Architecture	Control Structure	Rule Design
Detailed Reprentation (out of context) *Role: Programmer*	Data Definition	Program	Network Architecture	Security Architecture	Timing Definition	Rule Speculation
Functioning Enterprise *Role: User*	Usable Data	Working Function	Usable Network	Functioning Organization	Implemented Schedule	Working Strategy

Gambar 2.3 Matriks *Zachman Framework*

Pada dimensi pertama, Zachman menggambarkannya sebagai baris yang terdiri dari 6 perspektif yaitu[4]:

1. *The Planner Perspective (Scope Context)* : Daftar lingkup penjelasan unsur bisnis yang dikenali oleh para ahli strategi sebagai ahli teori.
2. *The Owner Perspective (Enterprise Concept)* : Model semantik keterhubungan bisnis antara komponen-komponen bisnis yang didefenisikan oleh pimpinan eksekutif sebagai pemilik.
3. *The Designer Perspective (System Model)* : Model logika yang lebih rinci yang berisi kebutuhan dan desain batasan sistem yang direpresentasikan oleh para arsitek sebagai desainer.
4. *The Builder Perspective (Technology Model)* : Model fisik yang mengoptimalkan desain untuk kebutuhan spesifik dalam batasan teknologi spesifik, orang, biaya dan lingkup waktu yang dispesifikasikan oleh engineer sebagai builder.
5. *The Implementer Perspective (Detailed Reprentation)* : Teknologi khusus, tentang bagaimana komponen dirakit dan dioperasikan, dikonfigurasikan oleh teknisi sebagai implementator.
6. *The Participant Perspective (Functioning Enterprise)* : Kejadian-kejadian sistem berfungsi nyata yang digunakan oleh para teknisi sebagai participant.

Untuk dimensi kedua, setiap isu perspektif membutuhkan cara yang berbeda untuk menjawab pertanyaan fundamental : who, what, why, when, where and how. Setiap pertanyaan membutuhkan jawaban dalam format yang berbeda. Zachman menggambarkan setiap pertanyaan fundamental dalam bentuk kolom/ fokus[4].

1. *What* (kolom data) : material yang digunakan untuk membangun sistem (inventory set).
2. *How* (kolom fungsi) : melaksanakan aktivitas (process transformations).
3. *Where* (kolom jaringan) : lokasi, tofografi dan teknologi (network nodes).
4. *Who* (kolom orang) : aturan dan organisasi (organization group).
5. *When* (kolom waktu) : kejadian, siklus, jadwal (time periods).
6. *Why* (kolom tujuan) : tujuan, motivasi dan inisiatif (motivation reason).

2.2.7 Pemanfaatan *Zachman Framework* dalam Perancangan CRM

Berdasarkan hasil penelitian Jaja yang ditulis dalam jurnal penelitian yang berjudul "Perancangan Customer Relationship Management Dengan Menggunakan Zachman Framework", dijelaskan bahwa dalam proses implementasi CRM, ada 3 aspek yang perlu diperhatikan yaitu [5]:

1. Orang

   Aspek orang meliputi internalisasi cara berpikir orang tentang bagaimana memberikan pelayanan kepada konsumen.

2. Proses

   Prose yang memiliki interaksi langsung dnegan pelanggan harus ditangani sebagai prioritas ketika mengintegrasikan dan mengotomatisasi proses bisnis.

3. Sistem dan Teknologi

   Teknologi mengacu pada kemampuan komputasi yang memungkinkan perusahaan untuk mengumpulkan, mengatur, menyimpan dan menggunakan data tentang pelanggan.

Pada jurnal tersebut diatas juga dipetakan *Customer Relationship Management* (CRM) kedalam kerangka *Zachman Framework* berukuran 3 x 4 yang meliputi baris pertama pendefinisian enterprise, baris kedua pemodelan enterprise dan baris ketiga arsitektur enterprise. Dari semua baris dapat menggambarkan tentang data, proses, sistem/teknologi dan organisasi. Hasil dari pemetaan tersebut dapat digunakan sebagai dasar investasi teknologi yang dibutuhkan untuk mendukung pengelolaan hubungan pelanggan.

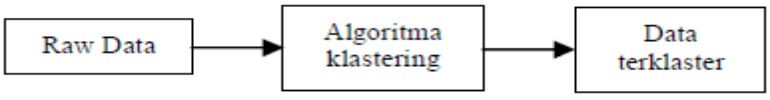
Berdasarkan hasil penelitian Sri Karnila yang ditulis dalam jurnal penelitian yang berjudul "Pemodelan Customer Relationship Management (CRM) Berbasis Web dan SMS dalam Pengelolaan Lowongan Pekerjaan Bagi Alumni Perguruan Tinggi Dengan Framework Zachman", dijelaskan bahwa dengan *framework Zachman* maka CRM memiliki keterkaitan yang erat secara langsung dengan metode dalam membangun sebuah integrasi data dan informasi[7]. Melalui pemetaan *framework Zachman* yang didefinisikan perbaris, kemudian tiap baris menggambarkan komponen dan perspektif yang berbeda serta gambaran aspek-aspek development proses. Dari setiap perspektif memberikan deskripsi

secara lengkap bagaimana model CRM dibangun. *Framework Zachman* menjadi kunci tahapan pembangunan dengan urutan kronologis dari perencana sampai ke pengguna.

Berdasarkan hasil penelitian Vidila Rosalina yang ditulis dalam jurnal penelitian yang berjudul "Pemodelan Customer Relationship Management Pada Perusahaan Petrokimia Menggunakan Zachman Framework", dijelaskan bahwa *Zachman Framework* tidak memberikan model dan arsitektur khusus yang dapat digunakan untuk memberikan penjelasan lengkap. Pemakai *Zachman Framework* bebas memilih alat yang akan digunakan untuk menerapkan model yang akan dibuat. Salah satu alat yang dapat digunakan adalah *Unified Modelling Language* (UML). UML dapat memberikan penjelasan yang baik terhadap permintaan dari *Zachman Framework*[8].

2.2.8 Algoritma *K-Means*

Berdasarkan hasil penelitian Feri Sulianta yang ditulis dalam jurnal penelitian yang berjudul "Customer Profiling Pada Supermarket Menggunakan Algoritma K-Means Dalam Memilih Produk Berdasarkan Selera Konsumen Dengan Daya Beli Maksimum", dijelaskan bahwa klasterisasi merupakan *Knowledge Discovery Database* (KDD) pada data mining yang akan membagi data kedalam kelompok-kelompok dimana pengelompokan yang terbentuk didasari pada obyek yang memiliki kesamaan karakteristik, sedemikian sehingga masing-masing elemen grup memiliki pembeda dengan grup lain. Klaster akan mengungkapkan hubungan dan struktur di dalam data yang sebelumnya bias dan tersembunyi. Tetapi sewaktu ditemukan korelasi item-item dalam klaster, hasilnya sangat relevan dengan mempertimbangkan karakteristik data-data yang terklaster[11].



Gambar 2.4 Proses pengelompokan data secara umum kedalam klaster

Algoritma k-means mendasari metode operasinya berdasarkan namanya yaitu ‘mean’ atau rerata. Dimana akan dilakukan pengamatan ke dalam kelompok k, dimana k diberikan sebagai parameter masukan. Kemudian dilakukan pengamatan pola untuk setiap kelompok berdasarkan kedekatan pada rerata jarak klaster yang adalah titik tengah, selanjutnya titik pusat atau *centroid* akan dikalkulasi ulang dan proses dimulai kembali[11].

K-Means adalah teknik yang dikategorikan *greedy*, komputasi dilakukan dengan teknik yang efisien, menjadi salah satu algoritma klastering yang paling banyak digunakan untuk klasterisasi[11].



Gambar 2.5 Flowchart Algoritma K-Means

Algoritma *K-Means* dapat dijabarkan sebagai berikut[11]:

1. Algoritma secara acak akan memilih beberapa k titik sebagai pusat klaster.
2. Setiap titik dalam dataset didedikasikan untuk klaster tertutup yang didasarkan pada jarak *Euclidean* antara setiap titik dan masing-masing pusat klaster.
3. Setiap pusat klaster dikalkulasi ulang sebagai rerata dari titik-titik pada klaster.
4. Langkah 2 dan 3 diulangi hingga klaster berkumpul dan tidak diapati perubahan ketika langkah 2 dan 3 diulang atau bahwa perubahan tidak

membuat perbedaan karakteristik nilai data pada member klaster. Pada kondisi ini klaster mencapai taraf stabil, dalam pengertian tidak ada lagi obyek yang dapat dipindahkan.

Jarak *Euclidean* antara dua titik atau obyek atau item x dan y, dimana jika x = (x1, x2,..., xn) dan y = (y1, y2,..., yn) adalah dua titik pada ruang lingkup *Euclidean* n-area, maka jarak dari x ke y, atau dari y ke x ditulis dalam formula sebagai berikut[8]:

$$d_{xy} = \sqrt{(|Y_1 - X_1|^2 + \cdots + |Y_n - X_n|^2)} \qquad \text{(2.1 Rumus Jarak Euclidean)}$$

Keterangan:

a. Y: Centroid yang menjadi pusat cluster
b. n: Jumlah dimensi dari pola masukan

2.2.9 *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *Class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET [13].

Pada pemodelan UML terdapat beberapa diagram yaitu *Use Case Diagram*, *Sequence Diagram*, *Class Diagram*, *Activity Diagram*[13].

1. *Use Case Diagram*

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah *Use Cas*e merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use Case* merupakan sebuah pekerjaan tertentu, misalnya *Login* ke sistem, mengbuat sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor

adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu[10].

Use Case Diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *Use Case* dapat meng-*include* fungsionalitas *Use Case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *Use Case* yang di-*include* akan dipanggil setiap kali Use Case yang meng-*include* dieksekusi secara normal. Sebuah *Use Case* dapat di-*include* oleh lebih dari satu *Use Case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang sama. Sebuah *Use Case* juga dapat meng-*extend Use Case* lain dengan kebutuhannya sendiri. Sementara hubungan generalisasi antar *Use Case* menunjukkan bahwa *Use Case* yang satu merupakan spesialisasi dari yang lain[13].

2. *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 atar dimensi vertikal (waktu) dan dimensi *horizonta*l (objek-objek yang terkait)[13].

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. Masing-masing objek, termasuk aktor, memiliki *lifeline vertikal*. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi / metode dari *Class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk objek *boundary*, *controller* dan *persistent entity*[13].

3. *Class Diagram*

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/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[13].

Hubungan antar *Class diagram* dapat dilihat dibawah ini[13]:

a. Asosiasi, yaitu hubungan statis antar *Class*. Umumnya menggambarkan Class yang memiliki atribut berupa *Class* lain, atau *Class* yang harus mengetahui eksistensi *Class* lain. Panah *navigability* menunjukkan arah *query* antar *Class*.
b. Agregasi, yaitu hubungan yang menyatakan bagian ("terdiri atas..").
c. Pewarisan, yaitu hubungan hirarkis antar *Class*. *Class* dapat diturunkan dari *Class* lain dan mewarisi semua atribut dan metode *Class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *Class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
d. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *Class* kepada *Class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *Sequence diagram.*

4. *Activity Diagram*

Activity diagram 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[10].

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[13].

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. Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu[13].

2.2.10 Basis Data

Basis data dapat dipahami sebagai suatu kumpulan data yang terhubung (interrelated data) yang disimpan secara bersama–sama pada suatu media, tidak perlu suatu kerangkapan data (walaupun ada harus seminimal mungkin dan terkontrol (controlled redundancy)), data disimpan dengan cara–cara tertentu sehingga mudah digunakan atau ditampilkan kembali. Data dapat digunakan oleh satu atau lebih program–program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan dengan program yang akan menggunakannya, data disimpan sedemikianrupa sehingga proses penambahan, pengambilan dan modifikasi data dapat dilakukan dengan mudah dan terkontrol[2].

Basis data (database) sendiri dapat didefiniskan sebagai berikut :“Basis data (database) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan pada perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya”[2].

2.2.11 MySQL

MySQL adalah multi user database yang menggunakan bahasa *Strucktured Query Language* (SQL). MySQL dalam operasi *client-server* melibatkan *server daemon* MySQL disisi *server* dan berbagai macam program serta *library* yang berjalan disisi *client*. MySQL mampu menangani data yang cukup besar.

Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku menyimpan data lebih dari 40 database, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 Gigabyte data[10].

SQL adalah bahasa standar yang digunakan untuk mengakses database server. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai standar industri. Dengan menggunakan SQL, proses akses database menjadi lebih user-friendly dibandingkan dengan menggunakan dBASE atau Clipper yang masih menggunakan perintah-perintah pemrograman[10].

MySQL merupakan *software database* yang paling populer di lingkungan Linux, kepopuleran ini karena ditunjang performa *query* dari databasenya yang saat ini bisa dikatakan paling cepat dan jarang bermasalah. MySQL ini juga sudah dapat berjalan pada lingkungan Windows[10].

2.2.12 PHP

PHP adalah bahasa pemrograman *server side* yang sudah banyak digunakan saat ini, terutama untuk pembuatan *website* dinamis. Untuk hal-hal tertentu dalam pembuatan *web*, bahasa pemrograman PHP memang diperlukan, misalnya saja pemroses data yang dikirimkan oleh pengunjung *web*. Contoh nyata adalah penggunaanya buku untuk tamu, halaman *login*, dan masih banyak lagi. Agar bisa berjalan dengan baik maka PHP harus dikolaborasikan dengan dengan *database engine*, misalnya saja MySQL, PostgresSQL, dan lain-lain[12].

PHP (*Hypertext Preprocessor*) adalah sebuah bahasa scripting yang dibundel dengan HTML, yang berjalan disisi *server*. Sebagian besar perintahnya berasal dari bahasa C, Java dan Perl dengan beberapa tambahan fungsi khusus PHP. Bahasa ini memungkinkan para pembuat aplikasi *web* yang menyajikan HTML yang dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan server. PHP biasa berinteraksi dengan hampir semua teknologi web yang sudah ada. *Developer* biasa menulis sebuah program PHP yang mengeksekusi suatu program CGI di *server* lain. Fleksibilitas ini amat bermanfaat bagi pemilik situs-situs *web* yang besar dan sibuk, karena pemilik masih menggunakan aplikasi-aplikasi yang sudah terlanjur dibuat dimasa lalu dengan menggunakan CGI, ISAP

atau dengan *script* seperti Perl, awk atau Phyton selama proses migrasi ke aplikasi baru yang dibuat dengan menggunakan PHP[12].