

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Berdasarkan Data Mining

Berdasarkan penelitian yang dilakukan oleh Anindya Khrisna Wardhani (2016), dengan judul “Implementasi algoritma K-Means untuk pengelompokan penyakit pasien pada puskesmas kaje pekalongan (Jurnal Transformatika, Volume 14, Nomor 1, Juli 2016)”. Penelitian ini menghasilkan suatu sistem data mining yang mampu menampilkan hasil pengelompokan penyakit pasien berdasarkan data penyakit yang ada di puskesmas kaje pekalongan.

Berdasarkan penelitian yang dilakukan oleh Fakihatin Wafiyah, ddk (2017), dengan judul “Implementasi Algoritma Modified K-Nearest Neighbor (MKNN) untuk Klasifikasi Penyakit Demam”. Penelitian ini menghasilkan aplikasi data mining untuk mengklasifikasikan pasien yang mengalami demam menggunakan metode *K-Nearest Neighbour*.

Berdasarkan penelitian yang dilakukan oleh Aline Embun Pramadhani, dkk (2014), dengan judul “Penerapan Data Mining Untuk Klasifikasi Prediksi Penyakit Ispa (Infeksi Saluran Pernapasan Akut) Dengan Algoritma Decision Tree (ID3)”. Penelitian ini menghasilkan sistem data mining yang dapat mengklasifikasi penderita penyakit ispa atau infeksi saluran pernapasan akut di klinik Dharma Husada menggunakan metode *K-Nearest Neighbour*.

Berdasarkan penelitian yang dilakukan oleh Olha Musa, dkk (2017) dengan judul “Analisis Penyakit Paru-Paru Menggunakan Algoritma K-Nearest Neighbors Pada Rumah Sakit Aloe Saboe Kota Gorontalo”. Penelitian ini menghasilkan aplikasi data mining yang mampu mengklasifikasikan penderita penyakit paru-paru di Rumah Sakit Aloe Saboe Kota Gorontalo menggunakan metode *K-Nearest Neighbour*.

Berdasarkan penelitian yang dilakukan oleh Mei Lestari (2014), dengan judul “Penerapan Algoritma Klasifikasi Nearest Neighbor (K-NN) Untuk Mendeteksi Penyakit Jantung”. Penelitian ini menghasilkan sistem data mining yang digunakan untuk membantu mengklasifikasi penderita penyakit jantung menggunakan metode *K-Nearest Neighbour*.

II.2. Landasan Teori

II.2.1. Data

Data merupakan komponen paling dasar atau masih mentah dari suatu informasi yang akan diproses lebih lanjut agar dapat berarti dan dapat menghasilkan informasi (Nurul Alifah Rahmawati, dkk, 2018).

Data adalah sesuatu yang belum mempunyai arti bagi penerimanya dan masih memerlukan adanya suatu pengolahan. Data bisa berwujud suatu keadaan, gambar, suara, huruf, angka, matematika, bahasa ataupun simbol-simbol lainnya yang bisa kita gunakan sebagai bahan untuk melihat lingkungan, obyek, kejadian ataupun suatu konsep (Eka Iswandy, 2015).

Data merupakan bentuk jamak dari bentuk tunggal datum atau data-item. Data adalah kenyataan yang menggambarkan suatu kejadian dan kesatuan nyata. Kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu. Di dalam dunia bisnis, kejadian- kejadian nyata adalah perubahan dari suatu nilai yang disebut dengan transaksi. Indikator yang dapat digunakan untuk mengukur kualitas informasi adalah *relevance*, *accurate*, *completeness*, *timeliness*, dan *understandability* dari informasi yang dihasilkan (Fanny Andalia, dkk, 2015).

II.2.2. Variabel

Variabel adalah suatu sifat yang dapat memiliki bermacam nilai. Variabel adalah sesuatu yang bervariasi. Variabel adalah simbol yang ada padanya kita lekatkan nilai. Ada tiga jenis variabel yang penting yaitu variabel bebas dan terikat, variabel aktif dan atribut, serta variabel kontinu dan variabel kategori (Freddy Rangkuti, 2015 ; 64).

Variabel adalah gejala yang bervariasi, yang menjadi objek penelitian. Adapun variabel dalam penelitian ini ada dua, yaitu:

1. Variabel *Independent* (Variabel Bebas)

Variabel bebas adalah variabel yang akan diselidiki pengaruhnya. Dalam penelitian ini yang menjadi variabel bebas adalah pembelajaran dengan menggunakan strategi *Student Team Heroic Leadership*.

2. Variabel *Dependent* (Variabel Terikat)

Variabel terikat adalah variabel yang dipengaruhi oleh variabel bebas. Disini yang menjadi variabel terikat adalah kreativitas belajar matematika siswa (Samidi, 2015).

II.2.3. Tipe Data

Tipe data merupakan sebuah kalsifikasi untuk menentukan suatu dari beberapa tipe data yang tersedia seperti *integer*, *string* atau *boolean* yang menjelaskan *value* yang memungkinkan untuk *variable* tersebut. Hal ini berkaitan dengan *value* yang dapat disisipkan pada *variable* dan operasi yang memungkinkan untuk dijalankan terhadap *value* dari sebuah *variable*. Secara umum, tipe data pada *database engine* dapat dibedakan menjadi (*Numerik*, *date time*, *string*, dan *json*). Menentukan sebuah tipe data yang tepat nantinya akan memudahkan dalam pengaturan tabel, seperti penggunaan tipe data *date* akan secara otomatis membuat format standar *date* seperti '2017-09-01'. Tipe data sebaiknya ditentukan pada saat membuat tabel, karena jika diubah saat sudah ada isinya dikhawatirkan akan mengacaukan isi tabel yang berakibat kehilangan data (Alim Imaduddin, 2017).

II.2.4. Sistem

Ada dua kelompok pendekatan dalam mendefinisikan sistem. Ada yang menekankan pada prosedurnya dan ada yang menekankan pada komponen atau elemennya, diantaranya yaitu Pendapat pertama menekankan sistem pada

komponennya, Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Pendapat kedua menekankan sistem pada prosedurnya, Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu (Fanny Andalia, dkk, 2015).

Pengertian Sistem menurut Mulyadi, Sistem adalah sekelompok dua atau lebih komponen-komponen yang saling berkaitan (sub sistem yang bersatu untuk mencapai tujuan yang sama). Pengertian Sistem menurut Winarno, Sistem adalah Sekumpulan komponen yang saling bekerja sama untuk mencapai tujuan tertentu. Sistem adalah sekelompok elemen yang terintegritasi dengan maksud yang sama untuk mencapai suatu tujuan (Rini Asmara, dkk, 2016).

Sebuah sistem terdiri dari berbagai unsur yang saling melengkapi dalam mencapai tujuan dan sasaran. Unsur-unsur yang terdapat dalam sistem itulah yang disebut dengan subsistem. Subsistem-subsistem tersebut harus saling berhubungan dan berinteraksi melalui komunikasi yang relevan sehingga sistem dapat bekerja secara efektif dan efisien (Eka Iswandy, 2015).

II.2.5. Informasi

Informasi menurut Turban et al merupakan data yang telah diorganisir sehingga memberikan arti dan nilai kepada penerimanya. Sedangkan menurut Jogiyanto, Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Dapat dikatakan bahwa data merupakan

bahan mentah, sedangkan informasi adalah bahan jadi atau bahan yang telah siap digunakan, Jadi, sumber dari informasi adalah data (Fanny Andalia, dkk, 2015).

Informasi adalah data yang diproses kedalam bentuk yang lebih berarti bagi penerima dan berguna dalam pengambilan keputusan, sekarang atau untuk masa yang akan datang (Rini Asmara, dkk, 2016).

Menurut Davis dalam Abdul Kadir, Informasi merupakan data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat bagi pengambilan keputusan saat ini atau saat mendatang. Sumber dari informasi adalah data (Eka Iswandy, 2015).

II.2.6. Sistem Informasi

Definisi Sistem Informasi menurut Azhar Susanto, Sistem Informasi adalah kumpulan dari subsistem apapun baik fisik ataupun *non fisik* yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi yang berarti dan berguna (Rini Asmara, dkk, 2016).

Sistem informasi adalah kegiatan yang mencakup semua aspek dalam perolehan (*gathering*), pengkombinasian (*combining*), penyimpanan (*storing*) dan penggunaan (*using*) informasi untuk mencapai tujuan tertentu dari dibuatnya suatu sistem (Ganda Yoga Swara, dkk, 2016).

Sistem informasi merupakan suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu

organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan.

Tujuan Perancangan Sistem Informasi adalah sebagai berikut :

1. Untuk memenuhi kebutuhan pemakai sistem informasi.
2. Untuk memberikan gambaran yang jelas dan rancang bangun lengkap kepada pemrogram komputer dan ahli teknik lain yang terlibat.
3. Untuk mendukung pengolahan pelaporan manajemen dan mendukung perusahaan (Eka Iswandy, 2015).

II.2.7. Aplikasi

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju.. Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia, Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu (Andi Juansyah, 2015).

Menurut Jogiyanto, Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi *output*. Menurut Kamus Kamus Besar Bahasa Indonesia, Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat

untuk mengerjakan dan melaksanakan tugas khusus dari pengguna (Bella Chintya Neyfa, dkk, 2016).

II.2.8. Algoritma

Algoritma adalah sistem kerja komputer memiliki *brainware*, *hardware*, dan *software*. Tanpa salah satu dari ketiga sistem tersebut, komputer tidak akan berguna. Kita akan lebih fokus pada *software computer*. *Software* terbangun atas susunan program) dan *syntax* (cara penulisan atau pembuatan program). Untuk menyusun program atau *syntax*, diperlukannya langkah-langkah yang sistematis dan logis untuk dapat menyelesaikan masalah atau tujuan dalam proses pembuatan suatu *software* (Gun Gun Maulana, 2017).

Menurut Rinaldi Munir, algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis. Berdasarkan dua pengertian algoritma di atas, dapat disimpulkan bahwa algoritma merupakan suatu istilah yang luas, dan masih banyak lagi pengertian-pengertian tentang algoritma. Algoritma tidak hanya berkaitan dengan dunia komputer akan tetapi algoritma juga berlaku dalam kehidupan sehari-hari (Ritayani, 2017).

II.2.9. Normalisasi

Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar *ambiguity* bisa dihilangkan. Normalisasi merupakan sebuah teknik dalam *logical* desain sebuah basis data relasional yang mengelompokkan atribut dari suatu tabel sehingga membentuk struktur tabel yang normal. Adapun kriteria

tabel dikatakan normal adalah ketika tidak ada kerangkapan data (redudansi data).

Tujuan dari normalisasi adalah untuk :

1. Untuk menghilangkan kerangkapan data sehingga meminimumkan pemakaian *storage* yang dipakai oleh *base relations (file)*.
2. Untuk mengurangi kompleksitas.
3. Untuk mempermudah pemodifikasian data (Dwi Puspitasari, dkk, 2016).

Normalisasi adalah suatu teknik untuk mengorganisasikan data kedalam tabel-tabel untuk memenuhi kebutuhan pemakai didalam suatu organisasi.

Proses normalisasi antara lain :

1. Data diuraikan dalam bentuk tabel, selajutnya dianalisis berdasarkan persyaratan tertentu kebeberapa tingkat.
2. Apabila tabel yang diuji belum memenuhi persyaratan tertentu maka tabel tersebut perlu dipecah menjdi beberapa tabel yang lebih sederhana sampai memenuhi data yang optimal.

Bentuk-bentuk dari normalisasi adalah :

1. Bentuk tidak normal (*unformalized form*) Bentuk ini merupakan bentuk data yang direkam, tidak ada keharusan untuk mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi.
2. Bentuk normal pertama (1NF atau *first normal form*) Bentuk normal pertama mempunyai ciri-ciri yaitu setiap data dibentuk dalam *flat file (file dasar)* dan data dibentuk dalam satu record demi satu record. Tidak ada set atribut yang berulang-ulang atau atribut yang bernilai ganda.

3. Bentuk normal kedua (2NF atau *second normal form*) Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal pertama, atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama, atau *primary key*, sehingga untuk bentuk normal (Ganda Yoga Swara, dkk, 2016).

II.2.10. Data Mining

Data mining merupakan proses iteratif dan interaktif untuk menemukan pola-pola atau model baru yang shahih (sempurna), bermanfaat dan dapat di mengerti dalam suatu *database* yang sangat besar (*massive databases*). Data mining berisi pencarian *trend* atau pola yang diinginkan dalam *database* besar untuk membantu pengambilan keputusan di waktu yang akan datang (Sheih Al Syahdan, dkk, 2018).

Data Mining yang juga dikenal dengan istilah *Pattern Recognition* merupakan suatu metode yang digunakan untuk pengolahan data guna menemukan pola yang tersembunyi dari data yang diolah. Data yang diolah dengan teknik data mining ini kemudian menghasilkan suatu pengetahuan baru yang bersumber dari data lama, hasil dari pengolahan data tersebut dapat digunakan dalam menentukan keputusan di masa depan (Anindya Khrisna Wardhani, 2016).

Data mining adalah penambangan atau penemuan informasi baru dengan mencari pola atau aturan tertentu dari sejumlah data yang sangat besar. Data mining, sering juga disebut sebagai *Knowledge Discovery In Database* (KDD)

adalah kegiatan yang meliputi pengumpulan, pemakaian data, historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar. Data mining didefinisikan sebagai proses menemukan pola-pola dalam data. Proses ini otomatis atau seringnya semiotomatis. Pola yang ditemukan harus penuh arti dan pola tersebut memberikan keuntungan, biasanya keuntungan secara ekonomi (Heroe Santoso, dkk, 2016).

II.2.11. Metode K-Nearest Neighbor (KNN)

Algoritma *k-Nearest Neighbor* (kNN) merupakan sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Kelas yang paling banyak muncul itu yang akan menjadi kelas hasil klasifikasi. Tujuan dari algoritma ini adalah mengklasifikasikan objek baru berdasarkan atribut dan *training sample*. Algoritma *k-Nearest Neighbor* menggunakan klasifikasi ketetanggaan (*neighbor*) sebagai nilai prediksi dari *query instance* yang baru. Algoritma ini sederhana, bekerja berdasarkan jarak terpendek dari *query instance* ke *training sample* untuk menentukan ketetanggaannya (Mutiar Ayu Banjarsari, dkk, 2015).

Algoritma *K-Nearest Neighbor* (KNN) adalah suatu metode yang menggunakan algoritma *supervised*. KNN termasuk kelompok *instancebased learning*. Algoritma ini juga merupakan salah satu teknik *lazy learning*. KNN dilakukan dengan mencari kelompok k objek dalam data training yang paling dekat (mirip) dengan objek pada data baru atau data testing. Secara umum untuk mendefinisikan jarak antara dua objek x dan y, digunakan rumus jarak *Euclidean*

pada persamaan berikut ini (Mustakim, dkk, 2016) :

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2} \dots\dots\dots(1)$$

Confusion matrix adalah *tool* yang digunakan untuk evaluasi model klasifikasi untuk memperkirakan objek yang benar atau salah. Sebuah matriks dari prediksi yang akan dibandingkan dengan kelas yang asli dari inputan atau dengan kata lain berisi informasi nilai aktual dan prediksi pada klasifikasi. Adapun perhitungan tingkat akurasi pada *confusion* matriks (Mustakim, dkk, 2016) :

$$Accuracy = \frac{\text{Jumlah data benar}}{\text{Jumlah data keseluruhan (n)}} 00\% \dots\dots\dots(2)$$

Langkah-langkah untuk menghitung metode *k-Nearest Neighbor* antara lain :

1. Menentukan parameter k.
2. Menghitung jarak antara data yang akan dievaluasi dengan semua pelatihan.
3. Mengurutkan jarak yang terbentuk.
4. Menentukan jarak terdekat sampai urutan k.
5. Memasangkan kelas yang bersesuaian.
6. Mencari jumlah kelas dari tetangga yang terdekat dan tetapkan kelas tersebut sebagai kelas data yang akan dievaluasi (Mutiara Ayu Banjarsari, dkk, 2015).

II.2.12. Demam Berdarah

Demam Berdarah *Dengue* (DBD) merupakan penyakit yang disebabkan oleh virus *Dengue* yang ditularkan dari orang ke orang melalui gigitan nyamuk *Aedes* (Ae). Sampai saat ini penyakit demam berdarah masih menjadi masalah kesehatan masyarakat dan endemis di Indonesia dan dapat mengakibatkan Kejadian Luar Biasa (KLB) di beberapa daerah endemis yang terjadi hampir setiap tahunnya pada musim penghujan. Diperlukan deteksi dini apakah seseorang terinfeksi atau tidaknya penyakit DBD (Chainur Arrasyid Hasibuan, dkk, 2017).

II.2.13. Microsoft Visual Basic

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *Console*, aplikasi *Windows*, ataupun aplikasi Web. Visual Studio mencakup kompiler, *Software Development Kit* (SDK), *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual *Basic*, Visual *Basic .NET*, Visual *InterDev*, Visual J++, Visual J#, Visual *FoxPro*, dan Visual *SourceSafe* (Herpendi, 2016).

Microsoft Visual Basic adalah program yang digunakan untuk membuat aplikasi berbasis *Microsoft Windows* secara cepat dan mudah. *Visual basic* menyediakan *tool* untuk membuat aplikasi yang sederhana sampai aplikasi yang kompleks atau rumit baik untuk keperluan pribadi atau untuk keperluan

perusahaan atau instansi dengan sistem yang lebih besar. *Microsoft Visual Basic* merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment* (IDE) visual untuk membuat program perangkat lunak berbasis sistem operasi *Microsoft Windows* dengan menggunakan model pemrograman (Rita Irviani, dkk, 2017).

II.2.14. Basis Data

Basis data adalah kumpulan data yang saling berhubungan (punya relasi). Relasi biasanya ditunjukkan dengan kunci (*key*) dari tiap *file* yang ada. Basis data diartikan sebagai markas atau gedung, sedangkan data merupakan representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, konsep, keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya (Eko Putra Membara, dkk, 2014).

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Sebuah basis data adalah tempat penyimpanan file data. Sebagai file data, suatu basis data tidak dapat menyajikan informasi secara langsung kepada pengguna. Pengguna harus menjalankan aplikasi untuk mengakses data dari basis data dan menyajikannya dalam bentuk yang bisa dimengerti. Basis data (*database system*) adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang

lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi (Muhammad Agus Suropto, dkk, 2017).

II.2.15. Database

Database merupakan kumpulan *file-file* yang saling berkaitan dan berinteraksi, relasi tersebut bila ditunjukkan dengan kunci dari tiap-tiap *file* yang ada. Satu database menunjukkan suatu kumpulan data yang dipakai dalam suatu lingkup perusahaan, instansi. Pengolahan database merupakan suatu cara yang dilakukan terhadap file-file yang berada di suatu instansi yang mana file tersebut dapat disusun, diurut, diambil sewaktu-waktu serta dapat ditampilkan dalam bentuk suatu laporan sehingga dapat mengolah *file-file* yang berisikan informasi tersebut secara rapi (Rini Asmara, 2016).

Database secara umum merupakan susunan atau kumpulan dari *record* data yang disimpan dalam komputer yang saling berhubungan dan dapat dijadikan sebagai salah satu sumber dari sistem informasi yang sedang berjalan sehingga mampu memenuhi informasi yang optimal yang dibutuhkan oleh pengguna. *Record database* masih sering ditampilkan dalam bentuk teks sebagai informasi bagi pengguna, sehingga dapat mempermudah kriptanalisis untuk mengakses serta memberi peluang untuk melakukan pembocoran, mendistribusikan maupun memodifikasi *record database* tersebut (Neti Rusri Yant, dkk, 2018).

II.2.15.1. Tabel

Tabel merupakan penyimpanan data. Data dalam tabel juga bisa dimanipulasi, dilakukan operasi-operasi dasar database, diurutkan, diperbaiki dan lain sebagainya. Jadi fungsi dari tabel adalah tempat menampung dan menyimpan data yang kemudian dapat di proses lebih lanjut. Dalam tabel terdapat field, record dan data. Tabel dalam database fungsinya cukup vital. Tabel adalah objek utama yang mesti ada, karena tabel adalah tempat menyimpan data yang sebenarnya (Beranda Agency, 2015 ; 5).

II.2.15.2. Record

Record adalah kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu *entity* secara lengkap satu *record* mewakili satu data atau informasi (Mhd Bustanur Rahmad, skk, 2014).

II.2.15.3. Field

Field merupakan karakter-karakter yang membentuk suatu arti tertentu, misalnya *field* untuk nomor mahasiswa, *field* untuk nama mahasiswa, *field* untuk mata pelajaran, dan lainnya (Muhammad Sobri, dkk, 2017 ; 29).

II.2.16. SQL Server

Bahasa *SQL* adalah bahasa yang bersifat *requestoriented* dan bersifat non procedural sehingga lebih mudah dipelajari, karena sintak yang digunakan

dalam bahasanya menyerupai bahasa yang digunakan dalam komunikasi setiap hari. *SQL* terdiri dari dua bagian yaitu :

1. DDL (*Data Definition Language*), adalah bahasa yang memiliki kemampuan untuk mendefinisikan data yang berhubungan dengan pembuatan dan penghapusan obyek seperti tabel, indeks dan lainnya seperti *CREATE*, *DROP*, dan *ALTER*.
2. DML (*Data Manipulation Language*), adalah bahasa yang berhubungan dengan proses manipulasi data pada *table* dan *record*, seperti : *INSERT*, *UPDATE*, *SELECT*, dan *DELETE* (Sri Handayani, dkk, 2014).

SQL (*Structured Query Language*) adalah bahasa *non procedural* untuk mengakses data pada database relasional. *SQL* adalah bahasa *database* yang dipergunakan dalam menyelesaikan permasalahan dalam *database* serta mempunyai kelebihan dalam mengolah data. Standar *SQL* mula-mula didefinisikan oleh ISO (*International Standards Organization*) dan ANSI (*the American National Standards Institute*) yang dikenal dengan sebutan *SQL86*. Dengan menggunakan *SQL*, kita dapat melakukan hal-hal berikut:

1. Memodifikasi struktur *database*.
2. Mengubah, mengisi, menghapus isi *database*.
3. Mentransfer data antara *database* yang berbeda.

SQL ada yang dikembangkan untuk PC dan ada juga yang dikembangkan untuk dapat mengakomodasi database yang sangat besar (Eka Iswandy, 2015).

II.2.17. UML (*Unified Modeling Language*)

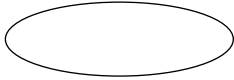
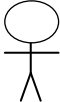




Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Ade Hendeni, 2016).

Unified Modelling Language merupakan salah satu alat bantu yang dapat digunakan dalam bahasa pemrograman yang berorientasi objek. Menurut Rosa, Salahuddin (2016:137), UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak (Humuntal Rumapea, dkk, 2017). Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram yaitu :

Tabel II.3. Simbol *Use Case*




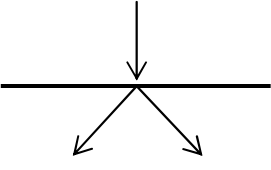
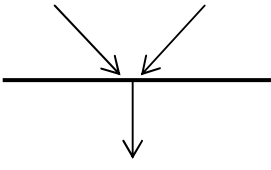
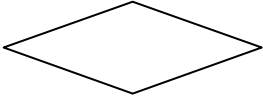

Gambar	Keterangan
	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Ade Hendini; 2016)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* yaitu :

Tabel II.4. Simbol *Activity Diagram*

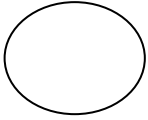
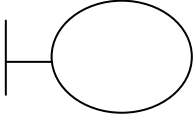
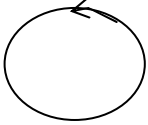

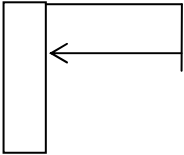

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
 New Swimline	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.


(Sumber : Ade Hendini; 2016)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* yaitu :

Tabel II.5. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.

	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .
---	---

(Sumber : Ade Hendini; 2016)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel dibawah ini :

Tabel II.6. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n.n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Ade Hendini; 2016)