

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Menurut (Kusrini; 2009:11) Sistem merupakan kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan (*input*) sehingga menghasilkan keluaran (*output*).

II.1.1 Data

Menurut (Kusrini; 2009:3) Data merupakan representasi dari fakta atau gambaran mengenai suatu objek atau kejadian, ambil contoh fakta mengenai biodata mahasiswa yang meliputi nama, alamat, jenis kelamin, agama yang dianut, dan lain-lain. Contoh lain dari fakta mengenai kejadian / transaksi dalam sebuah perusahaan dagang adalah seperti transaksi penjualan yang meliputi waktu transaksi, pelaku transaksinya (pelanggan, kasir), barang yang ditransaksikan, serta jumlah dan harganya. Data dinyatakan dengan nilai yang berbentuk angka, deretan karakter, atau simbol.

II.1.2. Informasi

Menurut (Kusrini ; 2009:4) informasi merupakan hasil olahan data, di mana data tersebut sudah diproses dan diinterpretasikan menjadi sesuatu yang bermakna untuk pengambilan keputusan. Informasi juga diartikan sebagai himpunan dari data yang relevan dengan satu atau beberapa orang dalam suatu waktu.

Menurut (Kusrini; 2009:4) Suatu informasi berguna bagi Pembuat keputusan karena informasi bisa menurunkan ketidakpastian (meningkatkan pengetahuan) tentang hal yang sedang dipikirkan. Makna dari sebuah informasi tentu berbeda-beda antara seorang dengan lainnya, tergantung pada tingkat kepentingannya, misalnya informasi daftar pelanggan yang potensial akan sangat dibutuhkan oleh bagian marketing di suatu perusahaan guna meningkatkan penjualan produk, tetapi barangkali tidak akan menjadi perhatian dibagian personalia

Kegunaan informasi bagi seseorang juga sangat tergantung pada waktu. Pada suatu waktu tertentu informasi tersebut mungkin sangat diperlukan di lain hari, mungkin saja hal tersebut sudah tidak berguna sama sekali. Contohnya, informasi perbandingan harga barang akan sangat dibutuhkan oleh seseorang yang akan membeli barang tersebut. Namun saat ini dia sedang tidak mempertimbangkan untuk membeli barang tersebut, informasi tersebut menjadi kurang bermakna

II.1.3. Kualitas Informasi

Agar bisa menyediakan keluaran yang berguna untuk membantu manager atau para pengambil keputusan, sebuah sistem informasi harus mampu mengumpulkan data dan mentransformasikan data tersebut kedalam informasi yang memiliki kualitas-kualitas tersebut

Berikut karakteristik informasi yang berkualitas :

- 1. Relevan.** Informasi yang disajikan sebaiknya terkait dengan keputusan yang akan diambil oleh pengguna informasi tersebut. Misalnya, seorang manager

yang akan memberikan kredit kepada pelanggan bisa melihat laporan keuangan pelanggan tersebut karena laporan tersebut terkait dengan keputusan yang akan dibuat, yaitu memberikan atau tidak memberikan kredit kepada pelanggan tersebut.

2. **Akurat.** Kecocok antara informasi dengan kejadian-kejadian atau objek-objek yang diwakilinya. Misalnya, laporan inventaris yang tidak akurat menyebutkan bahwa terdapat 15 unit barang yang tersisa di gudang. Kenyataannya, masih ada 51 unit barang di dalam gudang.
3. **Lengkap.** Merupakan derajat sampai seberapa jauh informasi menyertakan kejadian-kejadian atau objek-objek yang berhubungan. Misalnya, penjualan selama satu hari yang seharusnya ada 150 transaksi di laporan hanya tercatat sebanyak 145 transaksi.
4. **Tepat waktu.** Informasi yang tidak tepat waktu akan menjadi informasi yang tidak berguna atau tidak dapat di gunakan untuk membantu pengambilan keputusan. Misalnya, informasi jadwal ujian seorang mahasiswa disampaikan setelah kegiatan ujian diselenggarakan. Informasi ini menjadi tidak berguna lagi.
5. **Dapat dipahami.** Hal tersebut terkait dengan bahasa dan cara penyajian informasi agar pengguna lebih mudah mengambil keputusan.
6. **Dapat dibandingkan.** Sebuah informasi yang memungkinkan seorang pemakai untuk mengidentifikasi persamaan dan perbedaan antara dua objek

atau kejadian yang mirip. Misalnya, membandingkan laporan pendapatan antara tahun 2006 dan 2007,

II.1.4. Sistem Informasi

Menurut (Kusrini; 2009:11) suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan merupakan kegiatan strategi dari suatu organisasi, serta menyediakan laporan-laporan yang diperlukan untuk pihak luar.

Berdasarkan dukungan kepada pemakainya, sistem informasi dibagi menjadi

1. Sistem Pemrosesan Transaksi (*Transaction Processing System*) atau TPS
2. Sistem Informasi Manajemen (*Management Information System*) atau MIS
3. Sistem Otomasi Perkantoran (*Office Automation System / OAS*)
4. Sistem Pendukung Keputusan (*Decision Support System*) atau DSS
5. Sistem Informasi Eksekutif (*Executive Information System*) atau EIS
6. Sistem Pendukung Cerdas (*Intelligent Support System*) atau ISS

Mengingat bahwa EIS, DSS, dan MIS digunakan untuk pendukung manajemen, maka ketiga sistem tersebut sering disebut Sistem Pendukung Manajemen (*management support system*) atau MSS

II.2 Manajemen Pemasaran

Menurut (Nembah F H Ginting, 2011 : 8) pada hakikatnya manajemen pemasaran memiliki 3 kelompok subjek, yakni : komponen utama, bauran

pemasaran, dan strategi pemasaran. Agar dapat memasarkan produknya dengan memperoleh yang dikehendaki, seyogianya seorang pemasar mengenal betul komponen utama pemasaran dan unsur-unsur bauran pemasaran sehingga dapat menerapkan strategi tepat.

Untuk memberi gambaran menyeluruh, berikut ini akan diperkenalkan secara sepintas tentang tiga kelompok subjek itu dan kaitanya satu sama lain

1. Komponen Utama Studi Pemasaran

Studi manajemen pemasaran memiliki tiga komponen utama, yaitu : industri/khususnya bagian pemasar, konsumen/pasar, dan lingkungan pemasaran.

a. Industri/Pemasar

Kita mengenal adanya produsen-produsen individu, baik yang merupakan usaha perorangan ataupun organisasi usaha, yang bertujuan mendapatkan keuntungan maupun yang tidak mencari keuntungan. Kumpulan produsen individual inilah yang secara umum disebut industri. Dalam ilmu manajemen pemasaran, yang dimaksud dengan industri adalah kumpulan produsen individual yang menghasilkan produk yang sama atau serupa, sehingga produk satu produsen dapat mengganti produk produsen yang lain.

b. Konsumen Pasar

Kita mengenal adanya pembeli-pembeli produk individual, baik yang berupa perorangan maupun organisasi, yang merupakan pengguna akhir tanpa memperoleh laba maupun pembeli yang menggunakan untuk proses

produksi lebih lanjut dengan berharap memperoleh laba. Secara umum kedua kelompok pembeli ini, yang tidak mencari dan mencari laba, disebut Konsumen. Adapun kumpulan konsumen ini disebut Pasar

c. Lingkungan Pemasaran

Kegiatan pemasaran yang dilakukan oleh industri atau pemasar itu dikelilingi sesuatu yang berlaku individual, khususnya bagi setiap unit usaha dan berlaku umum dalam suatu wilayah tertentu. Sesuatu yang mengelilingi ikhtiar pemasaran itu disebut lingkungan pemasaran. Lingkungan ini bukan lingkungan hampa udara, dalam pengertian bukan lingkungan tidak terpengaruh, bukan lingkungan pasif, tetapi lingkungan aktif, lingkungan yang mempengaruhi hasil ikhtiar pemasaran.

Komponen ini sama pentingnya dengan dua komponen terdahulu. Karena itu perhatian kita curahkan kepada tiga komponen ini sama besarnya.

2. Unsur Bauran Pemasaran

Dengan pengetahuan tentang komponen utama, pemasaran akan dapat menentukan bauran pemasaran (*marketing mix*), yakni kombinasi penggunaan peralatan (*tools*) dan taktik pemasaran yang terkendali guna keberhasilan pemasaran. Terdapat empat kelompok peralatan atau 4 unsur atau variabel bauran pemasaran itu yaitu :

- a. *Product* (produk) adalah kombinasi "barang dan jasa" yang ditawarkan kepada pasar. Misalnya kita membeli komputer, maka apa yang kita peroleh bukan sekedar komputer sebagai *hardware* termasuk didalamnya juga

software-nya, dan bahkan komputer diantar kerumah, dipasangkan sampai dapat kita pakai. Jadi konsumen memperoleh produk berupa barang fisik komputer dan jasa hantaran serta pemasangan.

- b. *Price* (harga) adalah sejumlah uang yang harus dibayar oleh konsumen untuk mendapatkan produk. Perusahaan biasanya menyarankan harga penjualan kepada *dealer*, namun *dealer* itu sering menentukan harga sendiri.
- c. *Place* (tempat) mencakup semua kegiatan perusahaan yang mengusahakan agar produknya tersedia bagi konsumen yang dituju. Untuk keperluan sehari-hari, misalnya, produk tersebut harus dengan mudah dapat dibeli di warung sebelah.
- d. *Promotion* (promosi) berarti semua kegiatan perusahaan produsen untuk meningkatkan mutu produknya dan membujuk/merayu konsumen agar membeli produknya. Pabrik mobil Ford misalnya, mengeluarkan lebih dari \$600 juta per tahun untuk kegiatan promosinya.

3. Strategi Pemasaran

Strategi pemasaran merangkum cara-cara dimana bauran pemasaran disyukuri untuk menarik dan memuaskan sasaran pasar sekaligus mewujudkan tujuan perusahaan. Ada tiga strategi pemasaran yang akan di ketengahkan yaitu :

a. Strategi Inti

Yaitu strategi paling penting yang harus diterapkan oleh semua bisnis dengan memanfaatkan bauran pemasaran melalui penerapat STP (*segmentation, tergeting and positioning*).

b. Strategi Dasar

Setelah menerapkan strasategi inti, yaitu setelah memposisikan produknya, perusahaan masih harus menerapkan strategi dasar, yaitu : strategi kepemimpinan harga, strategi diferensiasi serta strategi fokus dan satu strategi untuk kalah, yaitu strategi di tengah-tengah jalan (*on the middle of the roader*)

c. Posisi Strategis

Akibat penerapan strategi inti dan strategi dasar, perusahaan akan sampai kepada posisi persaingan strategi tertentu, ada pengelompokan posisi strategi perusahaan atas dasar pangsa pasar menjadi tiga kelompok yaitu : pemimpin pasar, penantang pasar, perelung pasar.

II.3 PHP

Menurut (Rosa & M. Shalahuddin; 2011 : 85) PHP dibuat pertama kali oleh seorang perekrayasa perangkat lunak (*software engineering*) yang bernama Rasmus Lerdoff. Rasmus Lerdoff membuat halaman web PHP pertamanya pada tahun 1994. PHP4 dengan versi-versi akhir menuju PHP5 sudah mendukung pemrograman berorientasi objek. PHP merupakan bahasa pemrograman yang digunakan untuk pemrograman web. PHP singkatan dari *hypertext preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan *web* yang disisipkan pada dokumen HTML. Penggunaan PHP memungkinkan *web* dapat dibuat dinamis sehingga *maintanance* situs *web* tersebut lebih mudah dan efisien. PHP merupakan

software *open-source* yang disebar dan dilisensikan secara gratis dan dapat didownload dari situs resminya yaitu : <http://www.php.net>.

II.4 Database

Menurut (Budi Raharjo, 2011: 3) istilah database banyak memiliki definisi. Untuk sebagian kalangan sederhana *database* diartikan sebagai kumpulan data (buku, nomor telepon, daftar pegawai, dan lain sebagainya). Ada juga yang menyebut *database* dengan definisi lain yang lebih formal dan tegas. *Database* didefinisikan sebagai kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil dan dicari secara cepat.

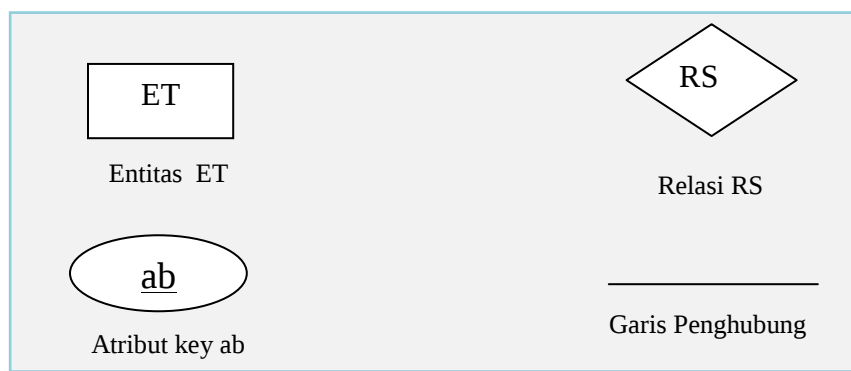
Selain berisi data, *database* juga berisi *metadata*. Metadata adalah data yang menjelaskan tentang struktur dari data itu sendiri. Sebagai contoh, Anda dapat memperoleh informasi tentang nama-nama kolom dan tipe yang ditampilkan tersebut disebut *metadata*.

II.4.1 Pemodelan Data

Menurut (Yudi Priyadi ; 2014 : 10) Terdapat beberapa penjelasan mengenai pemodelan basis data. Suatu basis data dapat digunakan secara bebas untuk menggambarkan dan memberikan deskripsi mengenai kumpulan informasi yang tersimpan dalam *data storage* komputer. Secara sederhana, definisi untuk model basis data adalah sekumpulan notasi atau simbol untuk menggambarkan data dan relasinya, berdasarkan suatu konsep dan aturan tertentu suatu pemodelan.

II.4.2 Model E-R

Menurut (Yudi Priyadi ; 2014 : 20) Pemodelan basis data dengan menggunakan diagram relasi antar entitas, dapat dilakukan dengan menggunakan suatu pemodelan basis data yang bernama Diagram *Entity-Relational* (selanjutnya disingkat Diagram E-R). Pada Gambar II.1, terdapat suatu simbol/notasi dasar yang digunakan pada Diagram E-R, yaitu entitas, relasi, atribut, dan garis penghubung.



Gambar II.1 : Notasi Dasar Diagram E-R

(Sumber : Yudi Priyadi ; 2014 : 20)

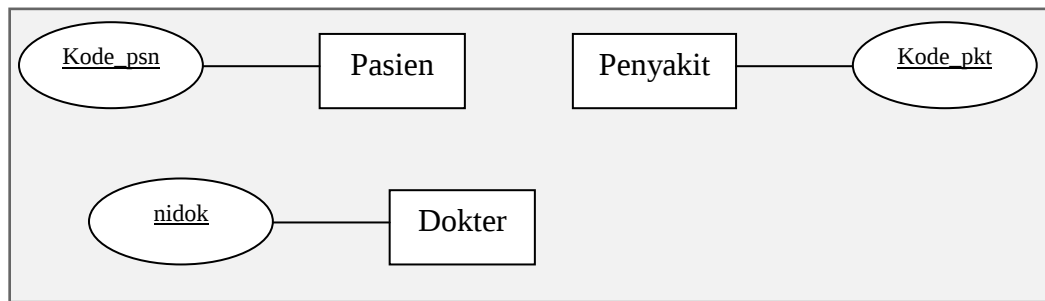
1. Entitas

Merupakan notasi untuk mewakili suatu objek dengan karakteristik sama, yang dilengkapi oleh atribut, sehingga pada suatu lingkungan nyata setiap objek akan berbeda dengan objek lainnya. Pada umumnya, objek dapat berupa benda, pekerjaan, tempat dan orang.

2. Atribut

Merupakan notasi yang menjelaskan karakteristik suatu entitas dan juga relasinya. Atribut dapat sebagai key yang bersifat unik, yaitu *Primary Key* atau

Foreign Key. Selain itu, atribut juga dapat sebagai atributdeskriptif saja, yaitu sebagai pelengkap deskripsi suatu entitas dan relasi.

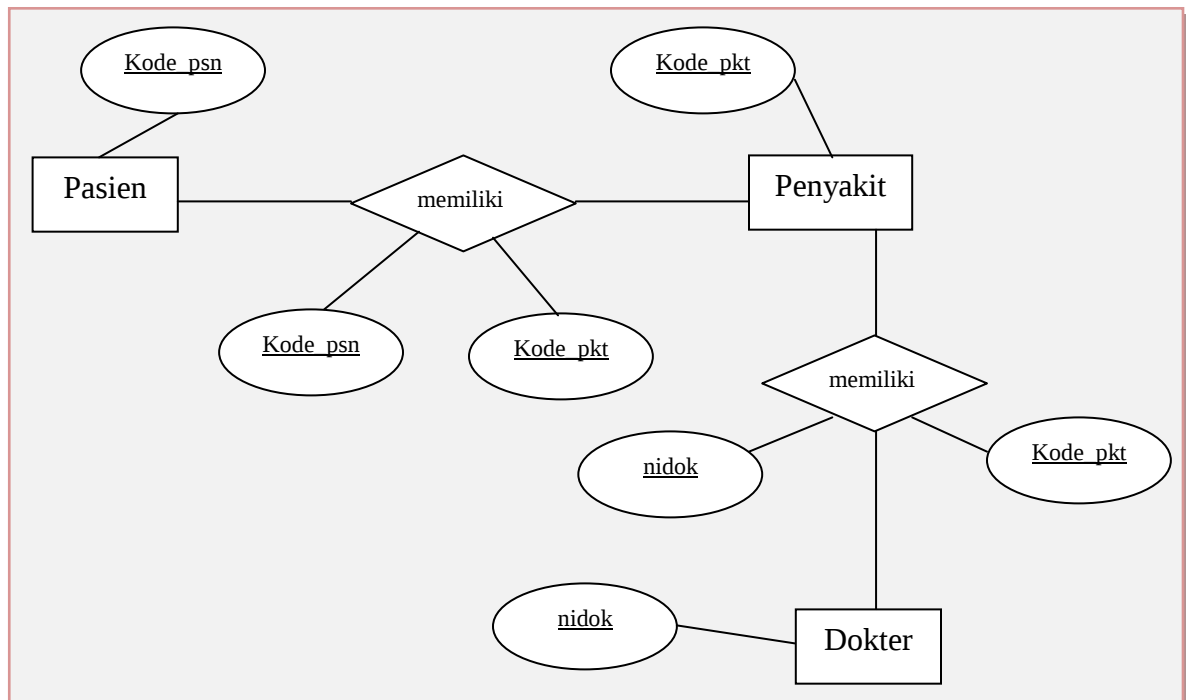


Gambar II.2 : Atribut Key pada Entitas

(Sumber : Yudi Priyadi ; 2014 : 23)

3. Relasi

Merupakan notasi yang digunakan untuk menghubungkan beberapa entitas berdasarkan fakta pada suatu lingkungan.



Gambar II.3 : Pemilihan Relasi untuk Entitas

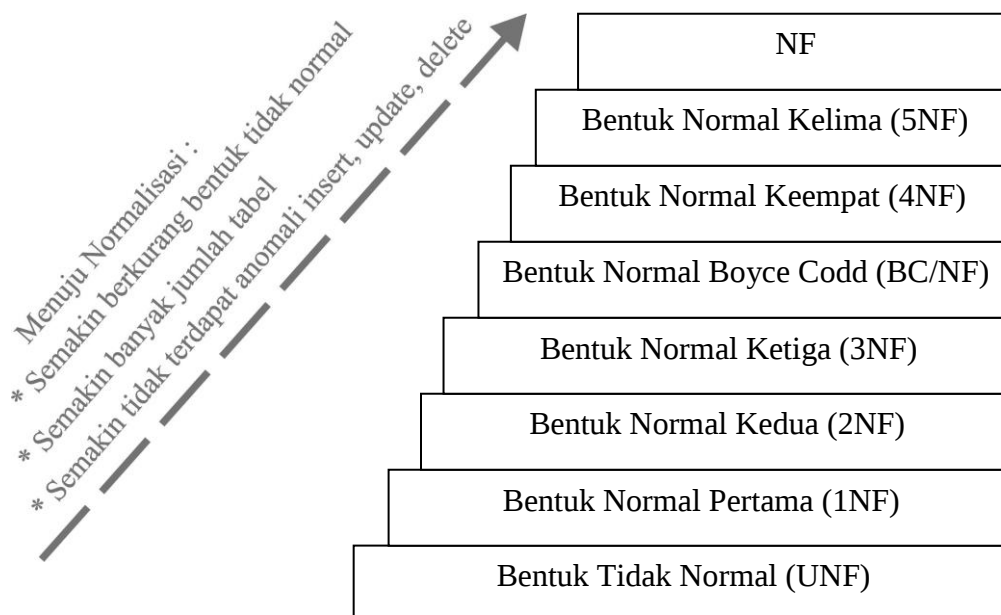
(Sumber : Yudi Priyadi ; 2014 : 25)

4. Garis penghubung

Merupakan notasi untuk merangkaikan keterkaitan antar notasi yang digunakan dalam Diagram E-R, yaitu entitas, relasi dan atribut.

II.4.3 Normalisasi

Menurut (Yudi Priyadi ; 2014 : 67) Normalisasi merupakan proses sistematis yang dilakukan pada struktur tabel basis data menjadi struktur tabel yang memiliki integritas data, sehingga tidak memiliki data anomali pada saat melakukan *insert*, *delete*, dan *update*. Pada Gambar II.4, tahapan proses sistematis yang dilakukan mulai dari bentuk tidak normal menjadi bentuk normal memiliki suatu syarat yang harus dipenuhi pada saat menuju suatu bentuk yang lebih baik (*well structured relation*).



Gambar II.4 : Tahapan Proses Bentuk Normalisasi

(Sumber : Yudi Priyadi ; 2014 : 67)

Setiap syarat dalam tahapan suatu bentuk normal memiliki keterkaitan, hal ini disebabkan karena pada setiap bentuk normal mengalami penyempurnaan untuk bentuk normal selanjutnya. Bentuk tidak normal akan semakin berkurang, setelah melalui tahapan perubahan bentuk normalisasi, sehingga berdampak pada jumlah tabel yang semakin banyak, tetapi menuju perbaikan ke dalam bentuk *well structured relation*. Hal ini terjadi akibat dari pengelompokan data suatu tabel agar memiliki ketergantungan secara fungsional.

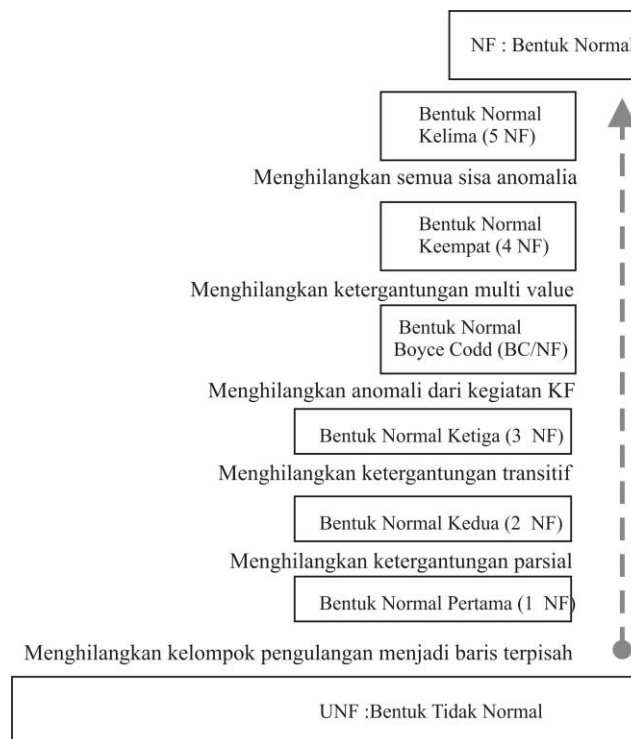
II.4.4 Aturan Proses Normalisasi

Menurut (Yudi Priyadi ; 2014 : 68) Secara sederhana, kegiatan normalisasi adalah melakukan dekomposisi atau penguraian tabel beserta datanya, menjadi tabel yang normal menurut konsep RDBMS. Merujuk pada gambar II.6, dekomposisi diawali dengan melakukan analisis pada suatu tabel atau beberapa contoh formulir yang sudah memiliki data lengkap dalam basis data, tetapi masih dalam bentuk yang tidak normal (UNF). Oleh karena itu agar dapat memenuhi syarat bentuk normal pertama (1NF), pada setiap barisnya diisikan suatu *value* dengan kelompok data yang sama, berdasarkan suatu atribut key. Dengan demikian, kelompok pengulangan dalam suatu baris dapat dihilangkan, karena sudah tidak terdapat *value* yang kosong untuk setiap *field* dan *recordnya*

Setelah memenuhi syarat bentuk normal pertama (1NF), proses berikutnya adalah menghilangkan ketergantungan secara parsial, yaitu dengan cara melakukan dekomposisi tabel menjadi beberapa kelompok tabel berdasarkan field yang memiliki status sebagai key. Hal ini dapat dilakukan oleh salah satu field saja,

dengan tetap tidak mengubah arti relasi dan ketergantungannya. Oleh sebab itu, disebut ketergantungan fungsional sebagian (*partiallly functional*), sehingga syarat bentuk normal kedua (2NF) sudah tercapai.

Bentuk normal kedua (2NF) merupakan syarat yang harus dimiliki untuk menuju bentuk normal ketiga (3NF). Pada proses ini, dilakukan dengan menghilangkan ketergantungan secara transitif, yaitu suatu konsep untuk tabel dari hasil relasi yang didalamnya terdapat ketergantungan secara tidak langsung pada beberapa atributnya. Pada umumnya proses normalisasi sudah dapat tercapai pada bentuk normal ketiga (3NF), yaitu dengan menghasilkan tabel yang tidak mengalami anomali basis data pada saat proses *insert*, *delete*, dan *update*.



Gambar II.5 : Tahapan Aturan Proses Normalisasi

(Sumber : Yudi Priyadi ; 2014 : 69)

II.5 MySQL

Menurut (Budi Raharjo, 2011:21) *MySQL* merupakan *software* RDBMS (atau *server database*) yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user* (*multi user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*). Saat ini *MySQL* banyak digunakan di berbagai kalangan untuk melakukan penyimpanan dan pengolahan data, mulai dari kalangan akademis sampai ke industri, baik industri kecil, menengah, maupun besar.

Lisensi *MySQL* terbagi menjadi dua. Anda dapat menggunakan *MySQL* sebagai produk *open source* dibawah *GNU general Public License* (gratis) atau dapat membeli lisensi dari versi komersialnya. *MySQL* versi komersial tentu memiliki nilai lebih atau kemampuan-kemampuan yang tidak disertakan pada versi gratis. Pada kenyataannya, untuk keperluan industri menengah kebawah, versi gratis masih dapat digunakan dengan baik.

II.6 Unified Modeling Language (UML)

Menurut (Rosa A.S & M. Shalahuddin 2011 : 118) Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem

perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML hanya berfungsi untuk melakukan pemodelan, jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metode berorientasi objek.

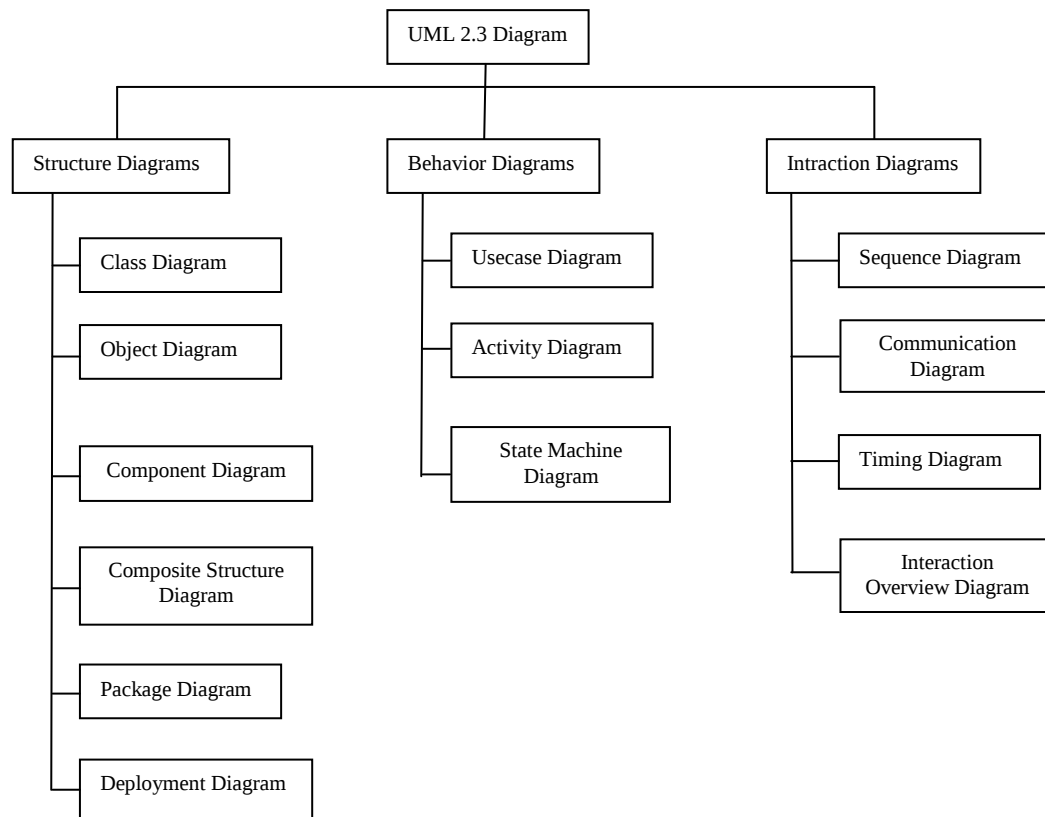
Menurut (Prabowo Pudjo Widodo & Herlawati, 2011 : 6) UML diaplikasikan untuk maksud tertentu, biasanya antara lain :

1. Merancang perangkat Lunak.
2. Sarana Komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

Blok pembangunan utama UML adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. UML memungkinkan para anggota team untuk bekerja sama dengan bahasa model yang sama dengan mengaplikasikan beragam sistem. Intinya UML merupakan alat komunikasi yang konsisten dalam mendukung para pengembang sistem saat ini.

II.6.1 Diagram-Diagram UML

Menurut (Rosa A.S & M. Shalahuddin, 2011 : 120) Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar II.6 di bawah ini



Gambar II.6 : Diagram UML

Sumber : Rosa A.S & M. Shalahuddin (2011 : 121)

Berikut ini penjelasan singkat dari pembagian kategori tersebut

1. *StructureDiagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

2. *Behavior Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

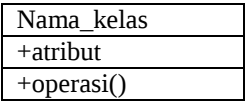
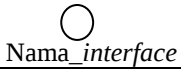
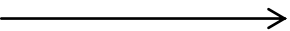
II.6.2 Class Diagram

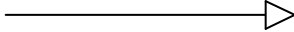
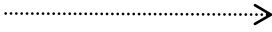
Diagram kelas atau *Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Berikut tabel II.1 menerangkan simbol-simbol pada diagram kelas :

Tabel II.1 : Diagram Kelas

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i>  Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas

Sumber : Rosa A.S & M. Shalahuddin (2011 : 124)

II.6.3 Object Diagram

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggungjawabkan. Untuk apa mendefinisikan sebuah kelas sedangkan pada jalannya sistem, objeknya tidak pernah dipakai.

Berikut adalah tabel II.2 menerangkan simbol-simbol diagram objek

Tabel II.2 : Diagram Paket

Simbol	Deskripsi
Objek Nama_objek : nama_kelas Atribut = nilai	Objek dari kelas yang berjalansaat sistem dijalankan
Link 	Relasi antar objek

Sumber : Rosa A.S & M. Shalahuddin (2011 : 124)

II.6.4 Component Diagram

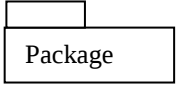
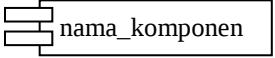
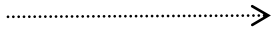
Diagram komponen atau component diagram dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem.

Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem. Komponen dasar yang biasanya ada dalam suatu sistem adalah sebagai berikut :

1. Komponen *user interface* yang menangani tampilan
2. Komponen *bussiness procesing* yang menangani fungsi-fungsi proses bisnis
3. Komponen data yang menangani manipulasi data
4. Komponen *security* yang menangani keamanan sistem

Komponen lebih terfokus pada penggolongan secara umum fungsi-fungsi yang diperlukan, berikut tabel II.3 yang menerangkan simbol-simbol yang ada pada diagram komponen

Tabel II.3 : Diagram Komponen

Simbol	Deskripsi
Package 	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen
Komponen 	Komponen Sistem
Kebergantungan / <i>dependency</i> 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
Antar muka / <i>interface</i> 	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen
Link 	Relasi antar komponen

Sumber : Rosa A.S & M. Shalahuddin (2011 : 126)

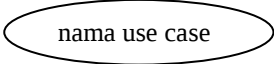
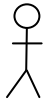
II.6.5 Use Case Diagram

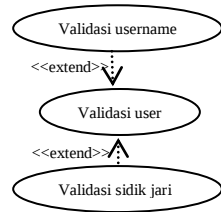
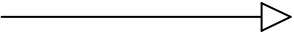
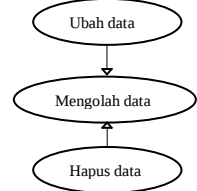
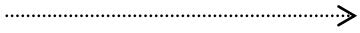
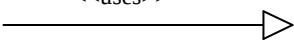
Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

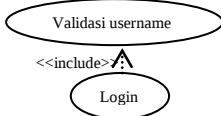
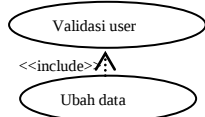
1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut tabel II.4 menerangkan simbol-simbol pada diagram *use case*

Tabel II.4 : Diagram Use case

Simbol	Deskripsi
Use case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
Aktor / actor 	Orang, proses, atau sistem yang lain berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan di buat itu sendiri

nama aktor	
Asosiasi / <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> , atau <i>usecase</i> memiliki interaksi dengan aktor
Ekstensi / <i>extend</i> 	Relasi usecase tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip inheritance pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan misal  arah panah mengarah pada <i>use case</i> yang ditambahkan
Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya misalnya :  Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)
Menggunakan / <i>include / uses</i>  	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini Ada 2 sudut pandang yang cukup besar mengenai include di usecase 1. include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat

	use case dijalankan misal pada kasus berikut :  <pre> graph TD A([Validasi username]) -- "<<include>>" --> B([Login]) </pre> 2. include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang di tambahkan telah di jalankan sebelum use case tambahan di jalankan, misal pada kasus berikut :  <pre> graph TD A([Validasi user]) -- "<<include>>" --> B([Ubah data]) </pre> Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.
--	--

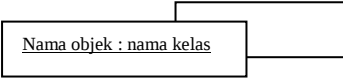
Sumber : Rosa A.S & M. Shalahuddin (2011 : 131)

II.6.6 Communication Diagram

Diagram komunikasi mengelompokkan message pada kumpulan diagram sekuen menjadi sebuah diagram. Dalam diagram komunikasi yang dituliskan adalah operasi / metode yang di jalankan antara objek yang satu dengan objek lainnya secara keseluruhan, oleh karna itu dapat di ambil dari jalanya interaksi pada semua diagram sekuen. Berikut adalah tabel II.5 yang menerangkan simbol-simbol yang ada pada diagram komunikasi :

Tabel II.5 : Diagram Komunikasi

Simbol	Deskripsi
Objek <pre> classDiagram class NamaObjek { nama_kelas } </pre>	Objek yang melakukan interaksi pesan

Link 	Relasi antar objek yang menghubungkan objek satu dengan lainnya atau dengan dirinya sendiri 
Arah pesan / stimulus 	Arah pesan yang terjadi, pada suatu link ada dua arah pesan yang berbeda 

Sumber : Rosa A.S & M. Shalahuddin (2011 : 140)

II.6.7 Activity Diagram

Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem/user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Berikut adalah tabel II.6 yang menggambarkan simbol-simbol yang ada pada diagram aktivitas :

Tabel II.6 : Diagram Aktivitas

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja

Percabangan / decesion 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber : Rosa A.S & M. Shalahuddin (2011 : 134)

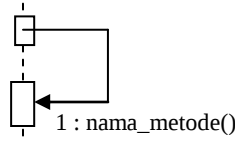
II.6.8 Sequence Diagram

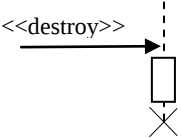
Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan messasge yang dikirimkan dan diterima antar objek. Banyaknya diagram objek yang digambarkan adalah sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefenisikan interaksi jalanya pesan sudah dicakup dapa diagram sekuen sehingga semakin banyak use case yang didefenisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah tabel II.7 yang menerangkan simbol-sombol yang ada pada diagram sekuen :

Tabel II.7 : Diagram Squence

Simbol	Deskripsi
Aktor  atau nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol

nama aktor tampa waktu aktif	dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya di nyatakan menggunakan kata benda di awali frase nama aktor
Garis hidup / lifeline ⋮	Menyatakan kehidupan suatu objek
Objek nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan
Waktu aktif ▮	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
Pesan tipe create → <<create>> →	Objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe call 1 : nama metode()	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri  Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode maka operasi / metode yang di panggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
Pesan tipe send 1 : masukan ----->	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan tipe return 1 : keluaran ----->	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian

Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy
---	--

Sumber : Rosa A.S & M. Shalahuddin (2011 : 138)