

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

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

II.1.1. Data

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 (Kusrini ; 2009 : 3).

II.1.2. Informasi

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 (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 (Kusrini ; 2009 : 4).

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 (Kusrini ; 2009 : 5).

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 tersisadi gudang. Kenyataanya, 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 (Kusrini ; 2009 : 6).

II.1.4. Sistem Informasi

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 Kelompok (*Group Support System*) atau GSS
7. 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 (Kusrini ; 2009 : 11).

II.2. Pengertian Sistem Pakar

Sistem Pakar (*Expert System*) adalah sistem komputer yang ditujukan untuk meniru semua aspek (*emulates*) kemampuan pengambilan keputusan (*decision making*) seorang pakar. Sistem pakar memanfaatkan secara maksimal pengetahuan khusus selayaknya seorang pakar untuk memecahkan masalah (Rosnelly Rika ; 2012 : 2)

Sistem Pakar dibuat hanya pada domain pengetahuan tertentu untuk suatu kepakaran tertentu yang mendekati kemampuan manusia di salah satu bidang saja. Sistem pakar mencoba mencari penyelesaian yang memuaskan, yaitu sebuah penyelesaian yang cukup bagus agar pekerjaan dapat berjalan walaupun itu bukan pekerjaan yang optimal

Pengetahuan sistem pakar dibentuk dari kaidah atau pengalaman tentang perilaku elemen dari domain bidang pengetahuan tertentu. Pengetahuan pada sistem pakar diperoleh dari orang yang mempunyai pengetahuan pada suatu bidang (pakar bidang tertentu), buku-buku, jurnal ilmiah, majalah, maupun dokumentasi yang tercetak lainnya. Sumber pengetahuan tersebut direpresentasikan dalam format tertentu, dan dihimpun dalam suatu basis pengetahuan. Basis pengetahuan ini selanjutnya dipakai sistem pakar untuk menentukan penalaran atas problem yang dihadapinya (Sri Hartati ; 2008 : 3)

II.2.1. Komponen Sistem Pakar

Sistem Pakar sebagai sebuah program yang difungsikan untuk menirukan pakar manusia harus bisa melakukan hal-hal yang dapat dikerjakan oleh seorang

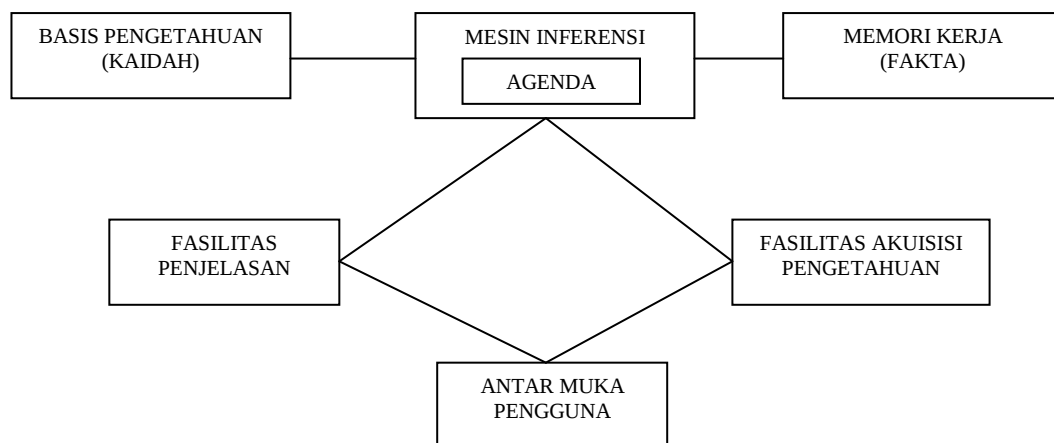
pakar. Untuk membangun sistem yang seperti itu maka komponen-komponen yang harus dimiliki adalah sebagai berikut : (Sri Hartati ; 2008 :3)

- a. Antar Muka pengguna (*User Interface*)
- b. Basis Pengetahuan (*Knowledge Base*)
- c. Mekanisme Inferensi (*Inference Machine*)
- d. Memori Kerja (*Working Memory*)

Sedangkan untuk menjadikan sistem pakar menjadi lebih menyerupai seorang pakar yang berinteraksi dengan pemakai, maka dilengkapi dengan fasilitas berikut

- a. Fasilitas Penjelasan (*Explanatoin Facility*)
- b. Fasilitas Akuisisi Pengetahuan (*Knowledge Acquisition Facility*)

Hal ini terlihat dalam struktur sistem pakar pada gambar II.1. Berikut ini



Gambar II.1 : Struktur Sistem Pakar

(Sumber : Sri Hartati ; 2008 :3)

II.3. Runut Maju (*Forward Chaining*)

Runut maju merupakan proses perunutan yang di mulai dengan menampilkan kumpulan data atau fakta yang meyakinkan menuju konklusi akhir. Runut maju bias juga disebut sebagai penalaran forward (*forward reasoning*) atau pencarian yang dimotori data (*data driven search*). Jadi dimulai dari premis-premis atau informasi masukan (*if*) dahulu kemudian menuju konklusi atau *derived information* (*then*) atau dapat dimodelkan sebagai berikut.

IF (informasi masukan)

THEN (konklusi)

Informasi masukan dapat berupa data, bukti, temuan, atau pengamatan. Sedangkan konklusi dapat berupa tujuan, hipotesa, penjelasan, atau diagnosa. Sehingga jalannya penalaran runut maju dapat dimulai dari data menuju tujuan, dari bukti menuju hipotesa, dari temuan menuju penjelasan, atau dari pengamatan menuju diagnosa (Sri Hartati, 2008 : 45)

Dalam penalaran maju, aturan-aturan diuji satu demi satu dalam urutan tertentu. Urutan itu mungkin berupa urutan pemasukan aturan kedalam basis aturan atau juga urutan lain yang ditentukan oleh pamakai. Saat tiap aturan diuji, sistem pakar akan mengevaluasi apakah kondisinya benar atau salah. Jika kondisi benar, maka aturan itu disimpan kemudian aturan berikutnya diuji. Sebaliknya kondisinya salah, aturan itu tidak disimpan dan aturan berikutnya diuji. Proses ini akan berulang (*iterative*) sampai seluruh basis aturan teruji dengan berbagai kondisi (Tim Penerbit Andi, 2009 : 15)

II.4. Visual Basic 2010

Visual Basic diturunkan dari bahasa *BASIC*. *Visual Basic* terkenal sebagai bahasa pemrograman yang mudah untuk digunakan terutama untuk membuat aplikasi yang berjalan diatas *platform windows*.

Pada tahun 90an, *Visual Basic* menjadi bahasa pemrograman yang paling populer dan menjadi pilihan utama untuk mengembangkan program berbasis *windows*. Versi *Visual Basic* terakhir sebelum berjalan diatas *.NET Framework* adalah VB6 (*Visual Studio* 1998).

Visual Basic .NET dirilis pada februari 2002 bersamaan dengan platform *.NET Framework* 1.0. Kini sudah ada beberapa versi dari *Visual Basic* yang berjalan pada platform *.NET*, yaitu VB 2002 (VB7), VB 2005 (VB8), VB 2008 (VB9), dan terakhir VB 2010 (VB10) yang dirilis bersamaan dengan *Visual Studio* 2010.

Selain *Visual Basic* 2010, *Visual Studio* 2010 juga mendukung beberapa bahasa lain, yaitu C#, C++, F# (bahasa baru untuk *functional programming*). IronPhyton dan IronRuby (bahasa baru untuk *dynamic programming*) (Eric Kurniawan, 2011 : 1).

II.5. Database

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.

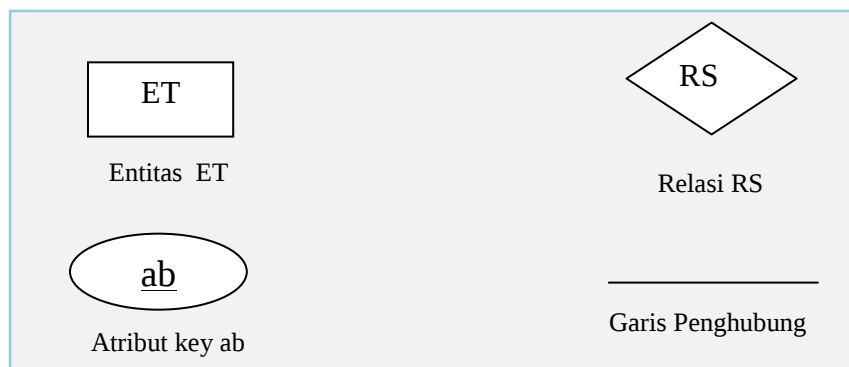
Selain berisi data, *database* juga berisi *metadata*. Metadata adalah data yang menjelaskan tentang struktur dari data itu sendiri. Sebagai contoh, Anda dapat memperoleh informasi tentang nama-nama kolom dan tipe yang ditampilkan tersebut disebut *metadata* (Budi Raharjo ; 2011 : 3)

II.5.1. Pemodelan Data

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 (Yudi Priyadi ; 2014 : 10)

II.5.2. Notasi Diagram E-R

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.2, terdapat suatu simbol/notasi dasar yang digunakan pada Diagram E-R, yaitu entitas, relasi, atribut, dan garis penghubung (Yudi Priyadi ; 2014 : 20)



Gambar II.2 : Notasi Dasar Diagram E-R

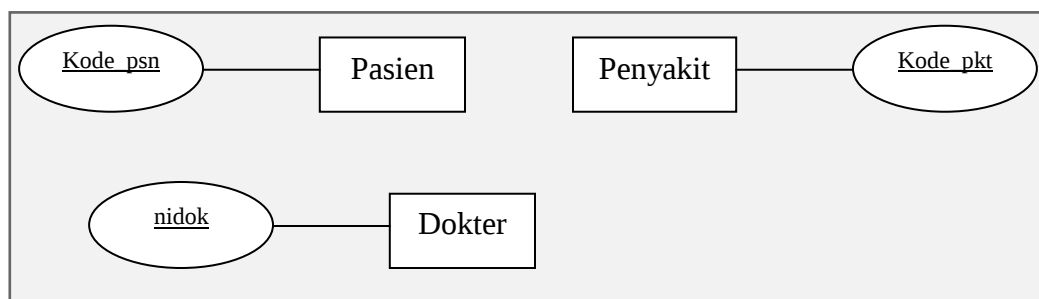
(Sumber : Yudi Priyadi ; 2014 : 20)

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.

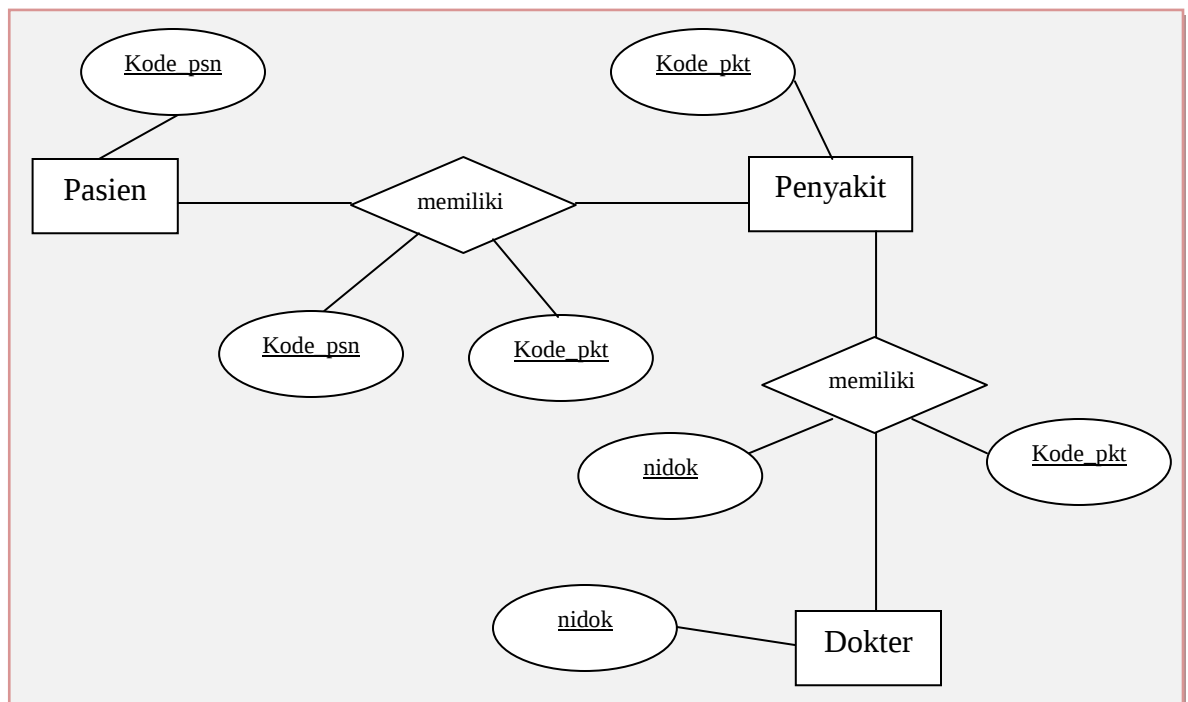


Gambar II.3 : Atribut Key pada Entitas

(Sumber : Yudi Priyadi ; 2014 : 23)

3. Relasi

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



Gambar II.4 : Pemilihan Relasi untuk Entitas

(Sumber : Yudi Priyadi ; 2014 : 25)

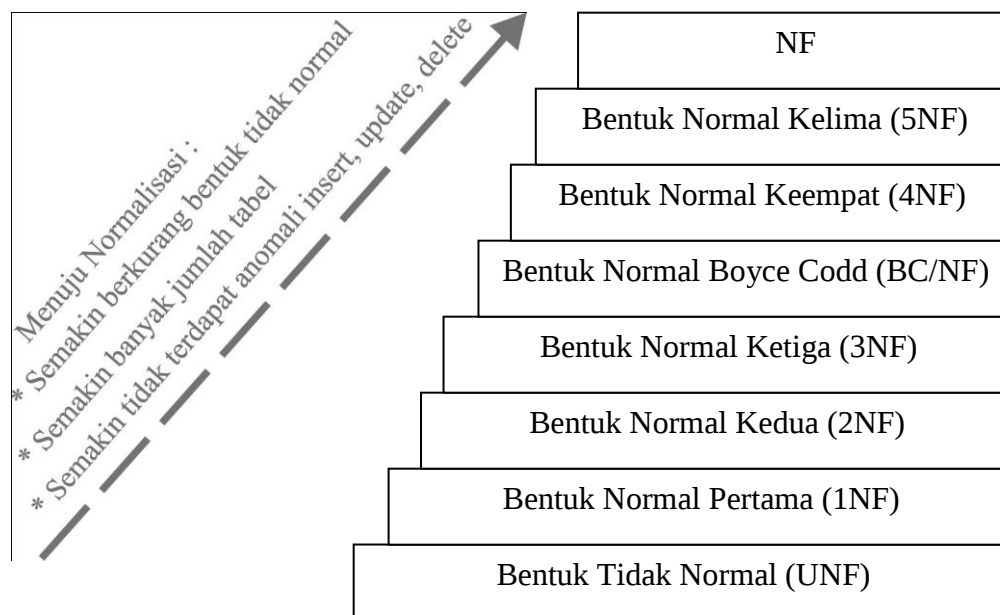
4. Garis penghubung

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

II.5.3. Normalisasi

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*) (Yudi Priyadi ; 2014 : 67)



Gambar II.5 : 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 berubahan 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. (Yudi Priyadi ; 2014 : 68)

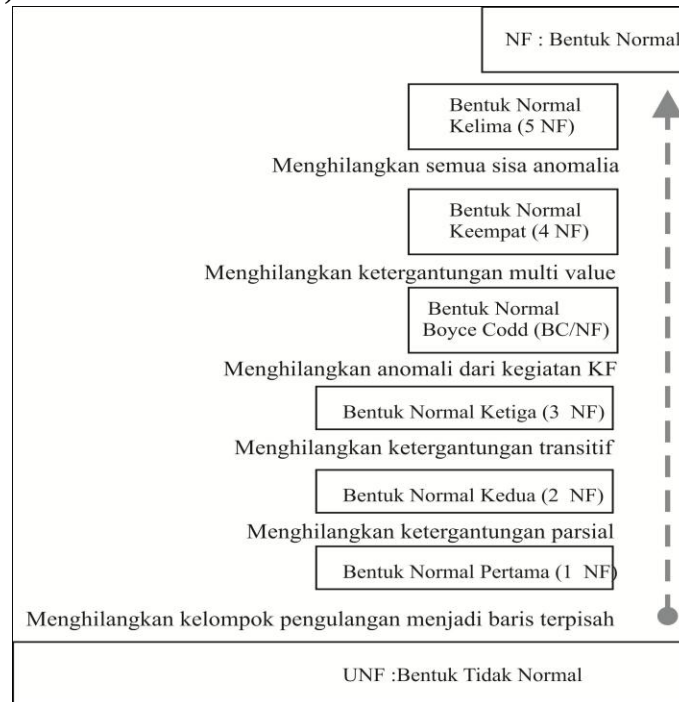
II.5.4. Aturan Proses Normalisasi

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* (Yudi Priyadi ; 2014 : 68)

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* (Yudi Priyadi ; 2014 : 69)



Gambar II.6 : Tahapan Aturan Proses Normalisasi

(Sumber : Yudi Priyadi ; 2014 : 69)

II.6. Unified Modeling Language (UML)

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakanteknik 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 (Rosa A.S & M. Shalahuddin ; 2011 : 118)

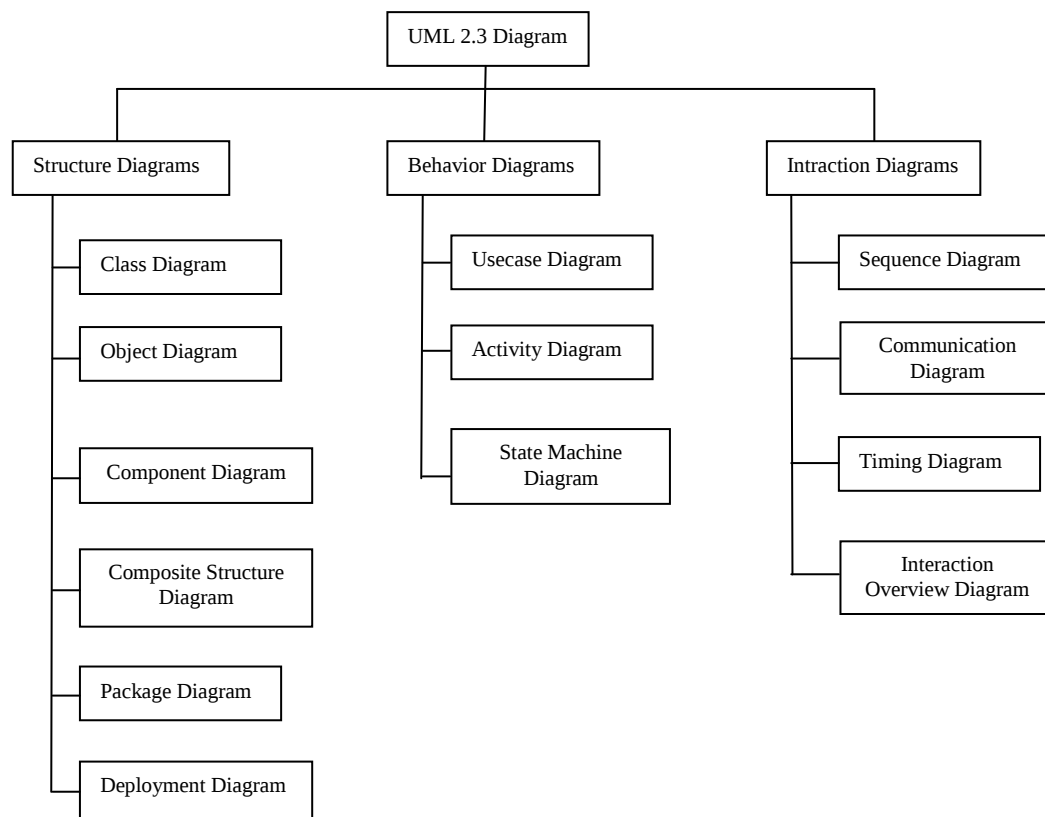
UML diaplikasikan untuk maksud tertentu, biasanya antara lain :

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 (Prabowo Pudjo Widodo & Herlawati ; 2011 : 6)

II.6.1 Diagram-Diagram UML

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.7 di bawah ini (Rosa A.S & M. Shalahuddin ; 2011 : 120)



Gambar II.7 : 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.

A. *Class Diagram*

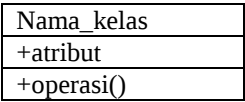
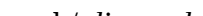
Diagram kelas atau *Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

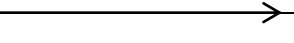
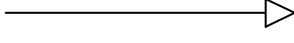
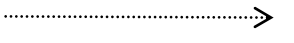
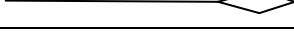
Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- 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>  Nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</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
Agregasi / <i>aggregation</i> 	Semua bagian (<i>whole part</i>)

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

B. 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 didefenisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefenisian kelas itu tidak dapat dipertanggungjawabkan. Untuk apa mendefenisikan 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)

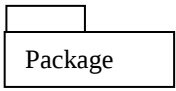
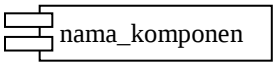
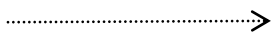
C. *Component Diagram*

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 procesiing* 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
 nama_komponen	Komponen Sistem
 Kebergantungan / <i>dependency</i>	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
 nama_interface	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)

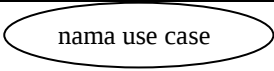
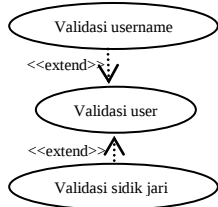
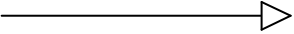
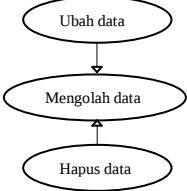
D. *Use Case Diagram*

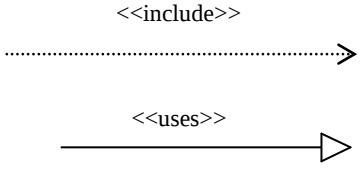
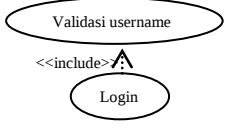
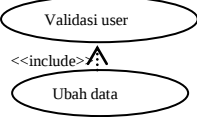
Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang 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 / <i>actor</i>  nama aktor	Orang, proses, atau sistem yang lain berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan di buat itu sendiri
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 subuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanppa <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 use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya misalnya : 

	Arah panah mengarah pada use case yang menjadi generalisasinya (umum)
Menggunakan / <i>include</i> / <i>uses</i> 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini Ada 2 sudut pandang yang cukup besar mengenai include di usecase 1. include berarti use case yang ditambahkan akan selalu dipanggil saat use case dijalankan misal pada kasus berikut :  2. include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang ditambahkan telah di jalankan sebelum use case tambahan di jalankan, misal pada kasus berikut :  Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada yang dibutuhkan.

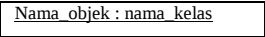
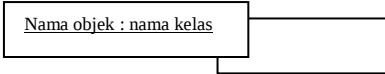
(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 131)

E. 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 	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, jika pada suatu link ada dua arah pesan yang berbeda, maka arah juga digambarkan dua arah pada dua sisi link 

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

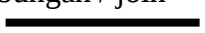
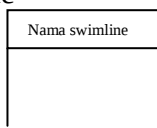
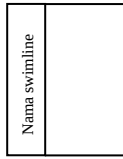
F. Activity Diagram

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

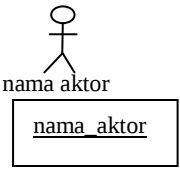
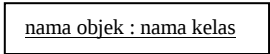
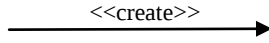
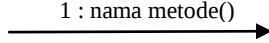
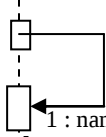
(Sumber : Rosa A.S & M. Shalahuddin ; 2011 : 134)

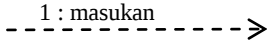
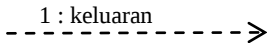
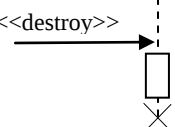
G. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message 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 didefinisikan interaksi jalanya pesan sudah dicakup dapa diagram sekuen sehingga semakin banyak use case

yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Berikut adalah tabel II.7 yang menerangkan simbol-simbol yang ada pada diagram sekuen :

Tabel II.7 : Diagram Sequence

Simbol	Deskripsi
Aktor  atau tampa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol 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 	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
Pesan tipe create 	Objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe call 	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 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainya, arah panah mengarah pada objek yang dikirim
Pesan tipe return 	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)

II.7. Penyakit Akibat Gigitan Nyamuk

Nyamuk adalah serangga yang memiliki dua sayap bersisik yang mampu mengepak 1000 kali per menit, tubuh yang langsing dan enam kaki panjang. Nyamuk memiliki ukuran yang berbeda-beda tetapi jarang sekali ukurannya melebihi 15 mm. Dalam bahasa Inggris, nyamuk dinamakan *Mosquito*, yang berasal dari bahasa yol atau portugis yang berarti *lalat kecil*. Penggunaan kata *Mosquito* berawal sejak tahun 1583. Di negri inggris nyamuk dikenal sebagai *gnats*. Tercatat lebih dari tiga ribu species nyamuk yang beterbangan di muka bumi ini, baik di tempat yang beriklim panas maupun dingin. Meskipun mampu hidup di kutub, sebagian besar nyamuk lebih suka hidup di daerah yang beriklim tropis dengan kelembaban tinggi seperti di Indonesia.

Dalam hidupnya nyamuk mengalami metamorfosa mulai dari telur, larva, pupa, hingga nyamuk dewasa. Nyamuk betina memerlukan protein untuk meningkatkan produksi telurnya karena itu nyamuk mengisap darah manusia yang mengandung protein tersebut. Tempat yang disukainya untuk bertelur adalah rawa-rawa dan genangan air. Selain itu, nyamuk juga menyukai kamar yang lembab dan pakaian kotor.

Penyakit akibat gigitan nyamuk cukup beragam. Nyamuk pembawa virus (*vektor*) juga beragam. Cara hidup dan cara menggigit nyamuk berbeda-beda pula. Ada nyamuk betina yang suka menggigit dalam posisi menungging alias posisi badan, mulut dan jarum yang dibenamkan ke kulit manusia dalam keadaan segaris. Nyamuk ini berasal dari golongan *Anopheles* yang menyebabkan penyakit malaria. Adapula yang suka menggigit dalam posisi mendatar sesuai dengan posisi pendaratan di permukaan kulit korbannya. Nyamuk ini adalah nyamuk *Aedes aegypti* yang menjadi biang penyakit demam berdarah dan *Chikungunya*. Nyamuk lain adalah nyamuk *culex* yang menyebabkan penyakit radang otak atau biasa disebut *west nile virus* alias virus Nil Barat. Penamaan tersebut disesuaikan dengan asal penyakit tersebut dari belahan benua Afrika (<http://artlink17.wordpress.com/2014/02/13/nyamuk-si-pembawa-penyakit/>)

Berikut ini sejumlah jenis penyakit yang diakibatkan gigitan nyamuk pembawa virus.

1. Demam berdarah

Penyakit demam berdarah mengakibatkan gangguan pada pembuluh darah kapiler dan sistem pembekuan darah yang dapat menimbulkan risiko kematian.

- Cara Penularan

Demam berdarah ditularkan oleh nyamuk *Aedes aegypti* pembawa virus Dengue yang menggigit di pagi dan sore hari. Ciri-cirinya berwarna hitam dengan belang- belang putih pada seluruh tubuh.

- Gejala

Penderita mengalami demam tinggi selama 2-7 hari dengan suhu 38-40° C. Muncul bintik-bintik merah pada kulit. Terjadi pula mimisan, muntah darah, nyeri ulu hati dan sebagainya.

- Pengobatan

Penderita mesti ditangani tim medis. Upaya pengobatan terutama ditujukan untuk mengatasi perdarahan dan keadaan syok.

2. Malaria

Malaria merupakan salah satu penyakit yang tersebar luas di dunia. Penyebabnya adalah empat jenis virus, yaitu vivax, falciparum, malariae dan ovale.

- Cara Penularan

Melalui nyamuk *Anopheles* betina. Kurang lebih 10 hari setelah gigitan, virus penyebab malaria masuk ke dalam aliran darah, lalu berkembang biak dan mengganti semua hemoglobin di dalam sel darah merah.

- Gejala

Demam dengan suhu mencapai 40° C, lemas, sakit kepala hebat, mengigau dan menggigil yang menyerupai serangan influenza. Akibatnya selain gangguan metabolisme, penderita akan mengeluarkan banyak keringat.

- Pengobatan

Pengobatan intensif dilakukan di rumah sakit. Namun penderita disarankan untuk mengonsumsi banyak cairan dan memperoleh asupan makanan bergizi.

3. Chikungunya

Virus chikungunya termasuk arbovirus (*arthropod borne virus*) dari genus *Alphavirus*. Nyamuk pembawa penyakit ini dari jenis *Aedes aegypti*, *Aedes africanus*, dan nyamuk *Mansonia*

- Cara Penularan

Penyebaran virus terjadi melalui gigitan nyamuk yang terinfeksi dengan masa inkubasi 1 – 12 hari.

- Gejala

Rasa linu di persendian tangan dan kaki serta pergelangan lutut. Demam tinggi dan muntah-muntah, menggigil, sakit kepala, bintik merah pada kulit seperti penderita demam berdarah. Mimisan bisa terjadi pada pasien anak-anak.

- Pengobatan

Tak ada vaksin maupun obat khusus untuk chikungunya. Umumnya pengobatan bersifat simptomatis. Penderita akan diberi obat penurun panas

dan penghilang rasa sakit. Dianjurkan banyak beristirahat dan banyak minum serta mengonsumsi menu makanan bergizi. Vitamin bisa digunakan untuk menambah daya tahan tubuh.

4. Filariasis

Filariasis atau penyakit kaki gajah adalah penyakit menular yang disebabkan oleh cacing *Filaria* yang ditularkan oleh berbagai jenis nyamuk seperti *Anopheles*, *Culex*, *Mansonia*, *Aedes* dan *Armigeres*. Penyakit ini bersifat menahun (kronis) dan bila tak mendapatkan pengobatan dapat menimbulkan cacat menetap berupa pembesaran kaki, lengan, dan alat kelamin baik perempuan maupun laki-laki.

- Cara penularan

Penderita awalnya digigit nyamuk yang sudah “terkontaminasi” larva stadium III. Siklus penularan penyakit kaki gajah melalui dua tahap. Pertama, tahap perkembangan dalam tubuh nyamuk. Kedua, tahap perkembangan dalam tubuh manusia.

- Gejala

Demam selama 3–5 hari, pembengkakan kelenjar getah bening, panas dan sakit terasa menjalar dari pangkal kaki atau pangkal lengan.

- Pengobatan

Dititikberatkan untuk mencegah kecacatan dan mengurangi sakit.

5. West Nile Virus

Virus Nil Barat (*west nile virus*/WNV) termasuk arbovirus yang ditularkan nyamuk. Tiga faktor utama kemunculan WNV adalah perubahan iklim, populasi burung yang terinfeksi dan penyebaran populasi nyamuk, khususnya nyamuk *culex*.

- Cara penularan

Pada awalnya virus tersebut hidup pada tubuh burung. Dari burung lalu “disebarluaskan” oleh nyamuk. Virus ini dapat menimbulkan penyakit bagi manusia dan sejumlah mamalia.

- Gejala

Saat terinfeksi WNV tidak terlihat gejala-gejala tertentu kecuali seperti orang terkena flu. Namun akibatnya bisa sangat serius termasuk *encephalitis* (radang otak).

- Pengobatan

Sejauh ini belum ada pengobatan khusus atau vaksin untuk mengatasi infeksi WNV.

6. Japanese Encephalitis

Japanese Encephalitis adalah penyakit yang disebabkan oleh virus. Virus ini disebarkan melalui gigitan nyamuk *Culex*.

- Cara penularan

Awalnya virus *Japanese Encephalitis* berkembang biak dalam tubuh babi. Lalu, nyamuk betina *Culex* mengisap darah babi dan menularkan virus ini saat menggigit manusia.

- Gejala

Demam, sakit kepala, lemah, mengingau, mengantuk, lumpuh, bahkan pingsan

- Pengobatan

Tidak ada pengobatan khusus untuk penyakit ini. Pengobatan yang dilakukan sekadar simptomatis untuk mengurangi gejala sakit