BAB II

TINJAUAN PUSTAKA

II.1. Sistem Pakar

Sistem pakar merupakan cabang dari *Artificial Intelegence (AI)* yang cukup tua karena sistem ini mulai dikembangkan pada pertengahan 1960. Sistem pakar yang muncul peretama kali adalah *General Purpose Solver (GPS)* yang dikembangkan oleh *Newel* dan *Simon.* Sampai saat ini sudah banyak sistem pakar yang dibuat seperti MYCIN untuk diagnosis penyakit, DENDRAL untuk mengidentifikasi struktur molekul campuran yang tak dikenal, XCON & XSEL untuk memebantu konfigurasi sistem komputer besar, SOPHIE untuk analisis sirkuit elektronik, *Prospector* digunakan dibidang geologi untuk membantu mencari dan menemukan deposit, FOLIO digunakan untuk membantu memberikan keputusan bagi seorang manager dalam stok dan investasi, DELTA dipakai untuk pemeliharaan lokomotif listrik diesel, dan sebagainya.

Istilah sistem pakar berasal dari istilah *knowledge based expert system*, istilah ini muncul karena untuk memecahkan masalah, sistem pakar menggunakan pengetahuan seorang pakar yang diasukkan ke dalam komputer. Seseorang yang bukan pakar menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar menggunakan sistem pakar untuk *knowledge – assistant*. Berikut adalah beberapa pengertian sistem pakar. (T.Sutojo, dkk ; 2011 : 160).

1. *Turban*

   "Sistem pakar adalah sebuah sistem yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan ke dalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia."

2. *Jackson*

   "Sistem pakar adalah program komputer yang mempresentasikan dan melakukan penalaran dengan pengetahuan beberapa pakar untuk memecahkan masalah atau memberikan saran".

3. *Luger* dan *Stubbleefield*

   "Sistem pakar adalah program yang berbasiskan pengetahuan yang menyediakan solusi 'kualitas pakar' kepada masalah-masalah dalam bidang (domain) yang spesifik".

II.1.1. Ciri-Ciri Sistem Pakar

Ciri-ciri dari Sistem Pakar adalah sebagai berikut :

1. Terbatas pada domain keahlian tertentu
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami
4. Bekerja bedasarkan kaidah/rule tertentu
5. Mudah dimodifikasi
6. Basis pengetahuan dan mekanisme inferensi terpisah

7. Keluarannya bersifat anjuran
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna. (T.Sutojo, dkk ; 2011 : 162).

II.1.2. Manfaat Sistem Pakar

Sistem pakar menjadi sangat populer karena sangan banyak kemampuan dan manfaat yang diberikannya, di anataranya :

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Andal, sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Bisa digunakan sebagai meddia pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
9. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar (T.Sutojo, dkk ; 2011 : 160-161).

II.1.3. Kekurangan Sistem Pakar

Selain manfaat, ada juga beberapa kekurangan yang ada pada sistem pakar, diantaranya :

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100% bernilai benar (T.Sutojo, dkk ; 2011 : 161).

II.1.4. Pemindahan Kepakaran *(Transferring Expertise)*

Tujuan dari sistem pakar adalah memindahkan kepakaran dari seorang pakar ke dalam komputer, kemudian ditransferkan kepada orang lain yang bukan pakar. Proses ini melibatkan empat kegiatan, yaitu :

1. Akuisisi pengetahuan (dari pakar atau sumber lain)
2. Representasi pengetahuan (pada komputer)
3. Inferensi pengetahuan
4. Pemindahan pengetahuan ke pengguna (T.Sutojo, dkk ; 2011 : 163-164).

II.1.5. Inferensi *(Inferencing)*

Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dalam melalukan penalaran. Inferensi ditampilkan pada suatu komponen yang disebut mesin inferensi yang mencakup prosedur-prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki oleh seorang pakar disimpan pada basis pengetahuan oleh sistem pakar. Tugas mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimiliki.

II.1.6. Aturan-Aturan *(Rule)*

Kebanyakan software sistem pakar komersial adalah sistem yang berbasis *rule (rule-based systems)*, yaitu pengetahuan disimpan terutama dalam bentuk *rule*, sebagai prosedure-prosedure pemecahan masalah.

II.1.7. Kemampuan Menjelaskan *(Explanation Capability)*

Fasilitas lain dari sistem pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang berikutnya. Penjelasan dilakukan dalam subsistem yang disebut subsistem penjelasan *(Explanation).* Bagian dari sistem ini memungkinkan sistem untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operasi-operasinya (T.Sutojo, dkk ; 2011 : 164-165).

Tabel II.1 Perbandingan Antara Sistem Konvensional Dengan Sistem Pakar

Sistem Konvensional	Sistem Pakar
Informasi dan pemprosesannya biasanya digabungkan dalam satu program.	Basis pengetahuan dipisahkan secara jelas dengan mekanisme infensi.
Program tidak membuat kesalahan (yang membuat kesalahan : pemprograman atau pengguna).	Program dapat berbuat kesalahan.
Biasanya tidak mejelaskan mengapa data masukan diperlukan atau bagaimana output dihasilkan.	Pejelasan merupakan bagian terpenting dari semua sistem pakar.
Perubahan program sangat menyulitkan.	Perubahan dalam aturan-aturan mudha untuk dilakukan.
Sistem hanya bisa beroperasi setelah lengkap atau selesai.	Sistem dapat beroperasi hanya dengan aturan-aturan yang sedikit (sebagai prototipe awal).

Eksekusi dilakukan langkah demi langkah (algoritmik).	Eksekusi dilakukan dengan menggunakan heuristik dan logika pada seluruh basis pengetahuan.
Perlu informasi lengkap agar bisa beroperasi.	Dapat beroperasi dengan informasi yang tidak lengkap atau mengandung ketidakpastian.
Menipulasi efektif dari basis data yang besar.	Manipulasi efektif dari basis pengetahuan yang besar.
Menggunakan data.	Menggunakan pengetahuan.
Tujuan utama : efisiensi.	Tujuan utama : efektivitas.
Mudah berurusan dengan data kuantitatif.	Mudah berurusan dengan data kualitatif.
Menangkap, menambah, dan mendistribusikan akses ke data numerik atau informasi.	Menangkap, menambah, dan mendistribusikan akses ke pertimbangan dan pengetahuan.

Sumber : T.Sutojo, dkk (2011 : 165)

II.1.8. Kepakaran

Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca, dan pengalaman. Kepakaran inilah yang memungkinkan para ahli dapat mengambil keputusan lebih cepat dan lebih baik dari pada seseorang yang bukan pakar. Kepakaran itu sendiri meliputi pengetahuan tentang :

1. Fakta-fakta tentang bidang permasalahan tertentu
2. Teori-teori tentang bidang permasalahan tertentu
3. Aturan-aturan dan prosedur-prosedur menurut bidang permasalahn umumnya
4. Aturan *heuristic* yang harus dikerjakan dalam situasi tertentu
5. Strategi *global* untuk memecahkan permasalahan

6. Pengetahuan tentang pengetahuan (*meta knowledge*) (T.Sutojo, dkk ; 2011 : 163).

II.1.9. Pakar *(Expert)*

Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasehat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal baru yang berkaitan dengan topik permasalahan, jika perlu harus mampu menyusun kembali pengetahuan-pengetahuan yang didapatkan, dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakarannya. Jadi seseorang pakar harus mampu melakukan kegiatan-kegiatan berikut :

1. Mengenali dan memformulasikan permasalahan
2. Memecahkan permasalahan secara cepat dan tepat
3. Menerangkan pemecahannnya
4. Belajar dari pengalaman
5. Merestrukturisasi pengetahuan
6. Memecahkan aturan-aturan
7. Menentukan relevansi (T.Sutojo, dkk ; 2011 : 163).

II.2. *Metode Certainty Factor*

Teori *Certainty Factor (CF)* diusulkan oleh *Shortliffe* dan *Buchannan* pada 1975 untuk mengakomodasi ketidakpastian pemikiran seorang pakar. Seorang pakar, (misalnya dokter) sering kali menganalisis informasi yang ada

dengan ungkapan seperti “mungkin”, “kemungkinan besar”,’hampir pasti”. Untuk mengakomodasi hal ini kita menggunakan *certainty factor (CF)* guna menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi. (T.Sutojo, dkk ; 2011 : 194).

Ada dua cara dalam mendapatkan tingkat keyakinan (CF) dari sebuah rule, yaitu :

1. Metode ‘Net Belief’ yang diusulkan oleh E.H Shortlife dan B.G. Buchanan

   CF(Rule) = MB(H,E) – MD(H,E)

   $$MB(H,E) = \begin{cases} 1 & P(H) = 1 \\ \frac{\max[P(H|E),P(H)] - P(H)}{\max[1,0] - P(H)} & \end{cases}$$

   $$MD(H,E) = \begin{cases} 1 & P(H) = 0 \\ \frac{\min[P(H|E),P(H)] - P(H)}{\max[1,0] - P(H)} & \end{cases}$$

   Di mana :

   CF (Rule) : faktor kepastian

   MB(H,E) : *measure of belief* (ukuran kepercayaan) terhadap hipotesis H, jika diberikan *evidence* E (antara 0 dan 1)

   MD(H,E) : measure of *disbelief* (ukuran ketidakpercayaan) terhadap evidence H, jika diberikan *evidence* E (antara 0 dan 1)

   P(H) : probabilitas kebenaran hipotesis H

   P(H|E) : probabilitas bahwa H benar karena fakta E

Contoh 1 :

Seandainya seorang pakar penyakit kelamin menyatakan bahwa probabilitas seseorang berpenyakit phimosis adalah 0,02. Dari data lapangan menunjukkan bahwa dari 100 orang penderita penyakit phimosis, 40 orang memiliki gejala kulup berminyak. Dengan menganggap H=Phimosis dan E = Kulup Berminyak, hitung faktor kepastian bahwa phimosis disebabkan oleh adanya kulup berminyak.

Jawab :

Dik : P(*phimosis*) = 0.02

P(*phimosis*|kulup berminyak) = 40/100 = 0.4

$$MB(H,E) = \begin{cases} 1 \\ \dfrac{\max[P(H|E),P(H)] - P(H)}{\max[1,0] - P(H)} \end{cases} \qquad P(H) = 1$$

Jawab: $\frac{\max[0,4|0.02]-P(0.2)}{1-0.02} = \frac{0.4-0.02}{1-0.02} = 0.39$

: $\frac{\min[0,4|0.02]-P(0.2)}{0-0.02} = \frac{0.4-0.02}{-0.02} = 0$

CF = 0.39 – 0 =0.39

Rule : **If** (Gejala= kulup berminyak) **then** Penyakit = *phimosis* (CF=0.39)

2. Dengan cara mewawancarai seorang pakar

Nilai CF(Rule) didapat dari interpretasi "*term*" dari pakar, yang diubah menjadi nilai CF tertentu sesuai tabel berikut.

Uncertainty Term	**CF**
Definitely (pasti tidak)	- 1,0
Almost certainly not (hampir pasti tidak)	- 0,8
Probably not (kemungkinan besar tidak)	- 0,6
May be not (mungkin tidak)	- 0,4
Unknown (tidak tahu)	- 0,2 to 0,2
May be (mungkin)	0,4
Probably (kemungkinan besar)	0,6
Almost certainty (hampir pasti)	0,8
Definately (pasti)	1,0

Contoh 2 :

PAKAR :

Jika batuk dan panas, maka ‘hampir dipastikan’ (*Almost certainty*) penyakitnya adalah influenza.

RULE : IF (batuk *AND* panas) *THEN* penyakit = influenza (CF=0,8)

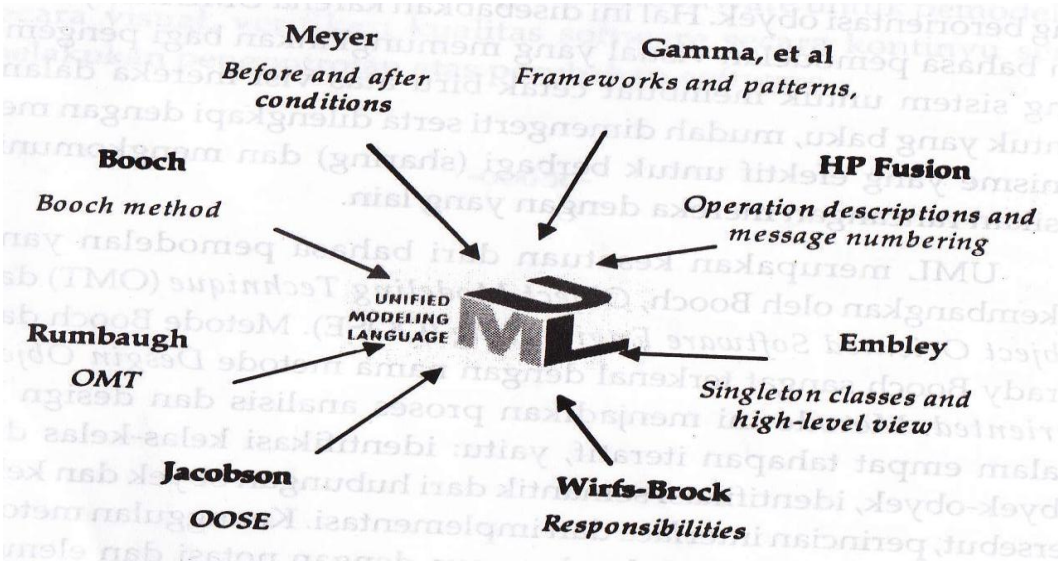
II.3. UML *(Unified Modeling Language)*

UML adalah sebuah bahasa yang telah menjadi standar dalam sebuah industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam

bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti *C++*, *Java, C#* atau *VB.NET*. walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. (Yuni Sugiarti ; 2013 : 34).

Seperti bahasa-bahasa lainnya, UML mendefenisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefenisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya *Grady Booch OOD* (*Object-Oriented Software Engineering).*



Gambar II.1 : Metodologi Pemodelan Berorientasi Objek

Sumber : Yuni Sugiarti (2013 : 35)

II.3.1. Deskripsi UML

Konsep dasar UML terdiri dari *structural ckassification, dynamic behavior*, dan model *management*. (Yuni Sugiarti ; 2013 : 36).

UML biasa digunakan untuk :

1. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagram*.
3. Menggambarkan representasi struktur statik sebuah sistem dalam bentuk *class diagrams*.
4. Membuat model *behavior* yang menggambarkan kebiasaan atau sifat sebuah sistem dengan *state transition diagrams*.
5. Menyatakan arsitektur implementasi fisik menggunakan *component and develepment diagrams*.
6. Menyampaikan atau memperluas *fungsionality* dengan *stereotypes*.

UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunisasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. UML mempunyai banyak diagram

yang dapat mengakomodasi berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. (Yuni Sugiarti ; 2013: 37).

Diagram-diagram tersebut digunakan untuk :

1. Mengakomodasikan ide
2. Melahirkan ide-ide baru dan peluang-peluang baru.
3. Menguji ide dan membuat prediksi memahami struktur dan relasi-relasinya.

II.3.2. Diagram Dasar Dalam UML

Terdapat sembilan jenis diagram UML, namun Penulis akan menjabarkan empat jenis diantaranya :

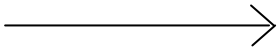
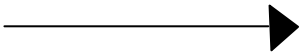
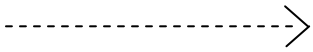
1. *Class Diagram*

   Bersifat statis. Menggambarkan struktur object sistem. Diagram ini menunjukkan *class object* yang menyusun sistem dan juga hubungan antara *class object* tersebut. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

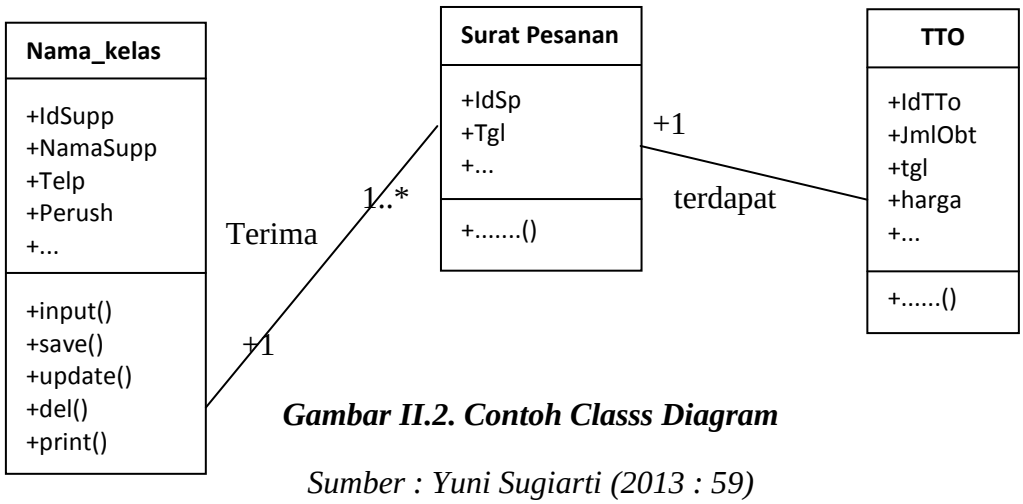
   a. Atribut merupakan *varabel-variabel* yang dimiliki oleh suatu kelas.
   b. Operasi atau *metode* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

   Berikut adalah simbol-simbol yang ada pada diagram kelas :

Tabel II.2. Simbol-simbol Class Diagram

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / Interface  Nama_interface	Sama dengan konsep interface dalam pemrograman berorientasi objek
asosiasi / association	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*.
Asosiasi berarah / directed association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity.*
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
Kebergantungan / *dependency* 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi / *aggregation*	Relasi antar kelas dengan makna

Sumber : Yuni Sugiarti (2013 : 59)



Gambar II.2. Contoh Classs Diagram

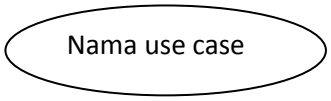
Sumber : Yuni Sugiarti (2013 : 59)

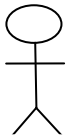
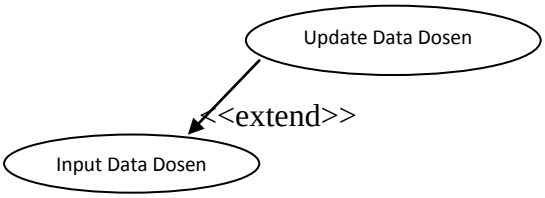
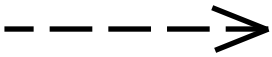
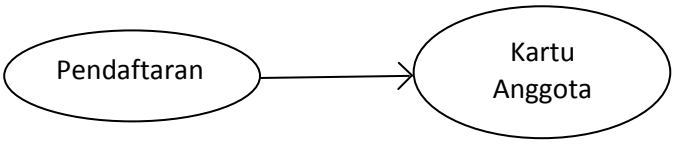
2. Diagram *Use-Case*

Use case diagram secara grafis menggambarkan interaksi antara sistem, sistem *eksternal*, dan pengguna. Dengan kata lain *Use Case* diagram secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna *(user)* mengharapkan interaksi dengan sistem itu. *Use Case* secara naratif digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari setiap interaksi. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas).

Berikut adalah simbol-simbol yang ada pada *diagram* Use Case :

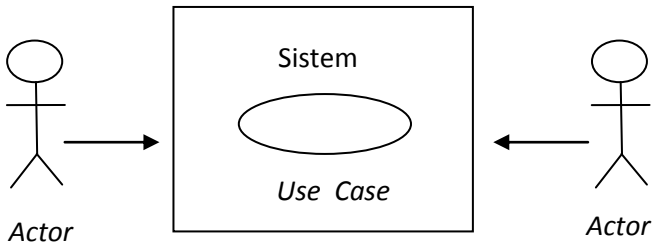
Tabel II.3. Simbol-simbol Use Case

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 diawal *frase* nama *use case.*

Actor  *Nama Actor*	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 actor adalah gambar orang , tapi aktor belum tentu merupakan orang, biasanya dinyatakan meggunakan kata benda di awal frase nama aktor.
asosiasi / association _____	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
Extend  <<extend>>	Relasi *use case* tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan, arah panah menunjuk pada *use case* yang dituju. 
Include <<include>> 	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 dua sudut pandang yang cukup besar mengenai include di use case, include berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* tambahan dijalankan, contoh : 

Sumber : Yuni Sugiarti (2013 : 42)

Berikut contoh gambar use diagram :

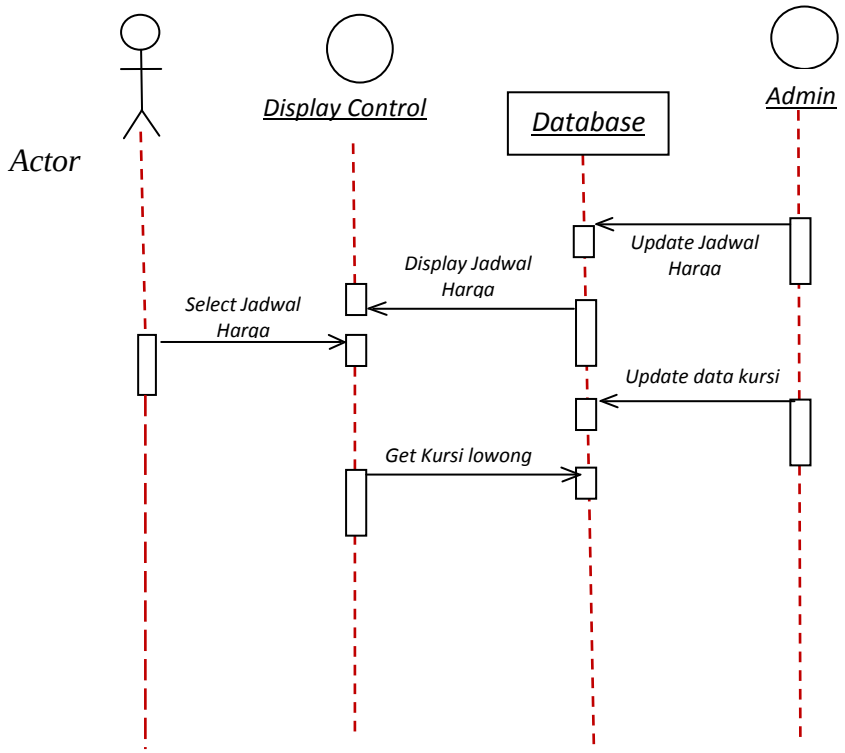


Gambar II.3. Use Case Model

Sumber : Yuni Sugiarti (2013 : 56)

3. Diagram interaksi dan *sequence* (urutan)

Diagram *sequence* meggambarkan kelakuan/prilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram *sequences* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansi menjadi objek itu.



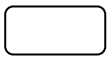
Gambar II.4. Display Sequence Diagram

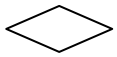
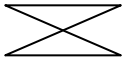
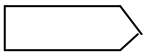
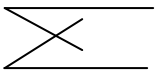
Sumber : Yuni Sugiarti (2013 : 59)

4. Diagram Aktifitas (*Activity Diagram*)

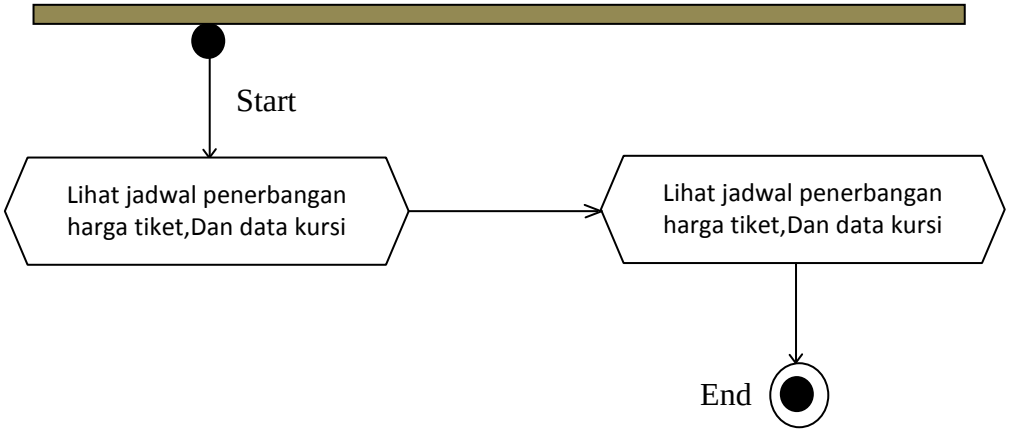
Bersifat dinamis. Diagra aktifitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Berikut adalah contoh *Activity diagram.*

Tabel II.4. Simbol-simbol Activity Diagram

Simbol	Keterangan
	Titik awal
	Titik Akhir
	Activity

	Pilihan untuk pengambilan keputusan
	Fork ; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	Rake; menunjukkan adanya dekomposisi
	Tanda Waktu
	Tanda Pengiriman
	Tanda Penerimaan
	Aliran akhir (Flow Final)

Berikut contoh dari Activity Diagram :



Gambar II.5. Contoh Activity Diagram

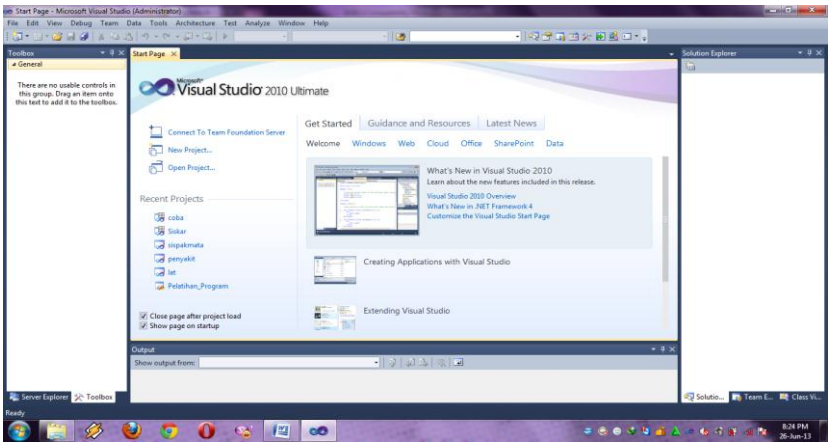
Sumber : Yuni Sugiarti (2013 : 78)

II.4. VB.NET 2010

Visual Basic 2010 merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh *Microsoft*, yaitu *Microsoft Visual Studio 2010*. *Visual studio* merupakan produk pemrograman andalan dari *microsoft corporation*, dimana di dalamnya beberapa jenis IDE pemrograman seperti *Visual Basic*, *Visual C++*, *Visual Web Developer*, *Visual C#*, dan *Visual F#*. (Muhammad Sadeli ; 2010 : 2).

Semua IDE pemrograman tersebut sudah mendukung penuh implementasi *.Net Framework 4.0* yang merupakan pengembangan dari *.Net Framework 3.5*. adapun *database* standar yang disertakan adalah *Microsoft SQL Server 2008 express*. *Visual Basic* 2010 merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu *Visual Basic* 2008. Beberapa pengembangan yang terdapat didalamnya antara lain dukungan terhadap *library* terbaru dari *Microsoft*, yaitu *.Net Framework 4.0*, dukungan terhadap pengembangan aplikasi berbasis *SilverLight*, dukungan terhadap aplikasi berbasis *Cloud Computing*, serta perluasan dukungan terhadap *database-database*, baik *standalone* maupun *database server*. Bahasa *Visual Basic 2010* sendiri awalnya berasal dari bahasa pemrograman yang sangat populer dikalangan programmer komputer, yaitu bahasa *BASIC*, yang oleh *Microsoft* di adaptasi dalam program *Microsoft Quick BASIC*. Seiring berkembangnya teknologi komputasi dan desain, *Microsoft* mengeluarkan produk yang dinamakan *Microsoft Visual Studio* dengan *Visual Basic* di dalamnya. Saat ini versi *Microsoft Visual studio* yang beredar adalah

versi 10 yang populer dengan nama *Microsoft Visual Studio 2010*, yang di dalamnya termasuk *Microsoft Visual Basic 2010*. (Muhammad Sadeli ; 2010).



Gambar II.6 Tampilan Microsoft Visual Basic 2010

II.5. *SQL Server* 2008

SQL Server 2008 adalah sebuah terobosan baru dari *Microsoft* dalam bidang database. *SQL Server* adalah sebuah DBMS (*Database Management System*) yang dibuat oleh *microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan *Oracle*. *SQL Server 2008* dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat (Wahana Komputer ; 2010 : 2).

II.5.1. *Database*

Database atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan di atur dengan susunan tertentu serta disimpan dalam

media penyimpanan komputer. Data itu sendiri adalah representasi dari semua fakta yang ada pada dunia nyata. (Wahana Komputer ; 2010 : 24).

Dalam *database* ada sebutan-sebutan untuk satuan data yaitu :

- Karakter, ini adalah satuan data terkecil. Data terdiri atas susunan karakter yang pada akhirnya mewakili data yang memiliki arti dari sebuah fakta.
- *Field*, adalah sekumpulan dari karakter yang mewakili fakta tertentu misalnya seperti nama siswa, tanggal lahir, dan lain-lain.
- *Record*, adalah sekumpulan dari field.
- Tabel, adalah sekumpulan dari *record-record* yang memiliki kesamaan *entity* dalam dunia nyata. Kumpulan dari tabel adalah *database*, wujud fisik sebuah *database* dalam komputer adalah sebuah *file.*
- *File*, adalah bentuk fisik dari penyimpanan data.

II.5.2. Sistem Database Relasional

SQL Sever 2008 termasuk salah satu mesin *database relasional.* Ide tentang sistem *database relasional* pertama kali dicetuskan oleh seorang yang bernama E.F.Codd lewat sebuah artikel yang berjudul "*A Relational Model of Data for Large Shared Data Banks"*. Artikel tersebut ditulis tahun 1970. Sistem *database relational* berdasarkan pada *metode* matematika. Sistem ini menggantikan *metode* lama yang berdasarkan *network* dan *hirarki*. *Metode relasional* ini bekerja berdasarkan keterkaitan antartabel. (Wahana Komputer ; 2010 : 25).

II.5.3. *Entity Relationship Diagram*

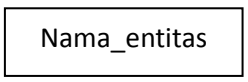
Pada dasarnya ERD adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan. ERD ini digunakan untuk melakukan pemodelan terhadap struktur data dan hubungannya. Penggunaan ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah database yang baik.

Entity dapat berarti sebuah *obyek* yang dapat dibedakan dengan *obyek* lainnya. *Obyek* tersebut harus memiliki komponen-komponen data (*atribut* atau *field*) yang membuatnya dapat dibedakan dari obyek yang lain. Dalam dunia database *entity* memiliki *atribut* yang menjelaskan karakteristik dari *entity* tersebut. Ada dua macam *atribut* yang dikenal dalam *entity* yaitu *atribut* yang berperan sebagai *kunci primer* dan atribut *deskriptif*. Hal ini bearti setiap *entity* memiliki himpunan yang diperlukan sebuah *primary key* untuk membedakan anggota-anggota dalam himpunan tersebut. (Wahana Komputer ; 2010 : 30).

Secara umum ERD terdiri dari 3 komponen, yaitu :

1. Entitas *(Entity)*

   Merupakan suatu “objek nyata” yang mampu dibedakan dengan objek yang lain. Objek tersebut dapat berupa orang benda ataupun hal yang lainnya. Penggambaran entitas dalam ERD seperti pada gambar.

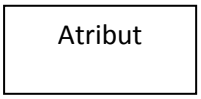


Gambar II.7. Entitas

Sumber : Ema Utami dan Anggit Dwi Hartanto (2012 : 19)

2. Atribut *(Attribute)*

Merupakan semua informasi yang berkaitan dengan entitas. Di dalam dunia pemograman, *atribut* adalah properti dari suatu objek. Penggambaran atribut dalam ERD seperti pada gambar.

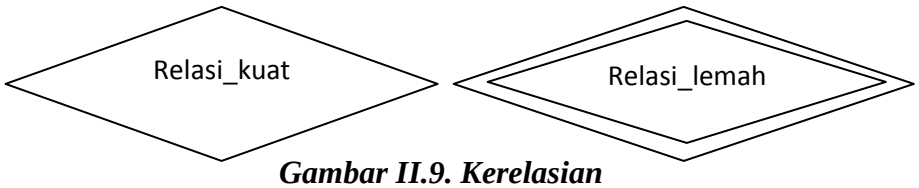


Gambar II.8. Atribut

Sumber : Ema Utami dan Anggit Dwi Hartanto (2012 : 20)

3. Relasi *(Relationship)*

Belah ketupat merupakan penggambaran hubungan (relasi) antar entitas atau sering disebut kerelasian. Ada dua macam penggambaran relasi, yakni relasi kuat dan relasi lemah. Relasi kuat biasanya untuk menghubungkan antar entitas kuat, sedangkan realsi lemaha untuk menghubungkan antar entitas kuat dengan entitas lemah. Penggambarn kerelasiaan seperti gambar.



Gambar II.9. Kerelasian

Sumber : Ema Utami dan Anggit Dwi Hartanto (2012 : 24)

II.5.4. Normalisasi Data

Normalisasi data adalah proses dimana tabel-tabel pada *database* dites dalam hal saling ketergantungan di antara *field-field* pada sebuah tabel. Misalnya

jika pada sebuah tabel terdapat ketergantungan terhadap lebih dari satu *field* dalam tabel tersebut, maka tabel tersebut harus dipecah menjadi banyak tabel. Banyaknya tabel pecahannya bergantung pada seberapa banyak ketergantungannya. Tiap tabel hanya boleh memiliki sebuah *field* kunci yang menjadi ketergantungan dari *field* lainnya dalam tabel tersebut.

Pada proses normalisasi data, aturan yang dijadikan acuan adalah *metode* ketergantungan fungsional. Teorinya adalah bahwa tiap kolom dalam sebuah tabel selalu memiliki hubungan yang unik dengan sebuah kolom kunci. Misalnya pada sebuah tabel data_siswa ada *field* nomor induk dan *field* nama siswa serta *field* tanggal lahir. Maka ketergantungan fungsionalnya dapat dinyatakan sebagai berikut : nmr_induk $\longrightarrow$ nm_siswa dan nmr_induk $\longrightarrow$ Tgl_lahir. Artinya nm_siswa memiliki ketergantungan fungsional terhadap nmr_induk. *Field* nm_siswa isinya ditentukan oleh *field* nmr_induk. Maksud dari semua itu adalah nmr_induk adalah *field* kunci yang menentukan karena tidak ada nomor induk yang sama pada satu sekolah, jadi field nmr_induk dapat dijadikan patokan untuk mengisi nm_siswa dan *field* lainnya. (Wahana Komputer ; 2010 : 32).

Ada beberapa langkah dalam normalisasi tabel, yaitu:

- *Decomposition,* dekomposisi adalah proses mengubah bentuk tabel supaya memenuhi syarat tertentu sebagai tabel yang baik. Dekomposisi dapat dikatakan berhasil jika tabel yang dikenai dekomposisi bila digabungkan kembali dapat menjadi tabel awal sebelum di-dekomposisi.

- Bentuk tidak normal, pada bentuk ini semua data yang ada pada tiap *entity* (di ambil atributnya) masih ditampung dalam satu tabel besar.
- *Normal form* pertama (*1st Normal Form),* pada tahapan ini tabel di-dekomposisi dari tabel bentuk tidak *normal* yang kemudian dipisahkan menjadi tabel-tabel kecil yang memiliki kriteria tidak memiliki atribut yang bernilai ganda dan komposit. Semua atribut harus bersifat atomik.
- *Normal Form* kedua (*2nd Normal Form),* pada tahapan ini tabel di anggap memenuhi normal kedua jika pada tabel tersebut semua atribut yang bukan primer bergantung penuh terhadap kunci primer tabel tersebut.
- *Normal Form* ketiga (*3rd Normal Form),* setiap atribut pada tabel selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama.

II.6. Penyakit Malaria

Penyakit malaria masih merupakan masalah kesehatan masyarakat di Indonesia sampai saat ini. Angka kesakitan penyakit ini masih cukup tinggi, terutama di daerah luar Jawa dan Bali. Di daerah transmigrasi dimana terdapat campuran penduduk yang berasal dari daerah yang endemis dan yang tidak endemis malaria, masih seringnya terjadi letusan atau kejadian luar biasa (KLB) malaria, kadang-kadang disertai adanya kematian. (Mardiana, Dwi Fibrianto ; 2009 ; 12)

Penyakit ini ditularkan oleh vektor nyamuk *(anopheles* betina*)* malaria yang semula banyak ditemukan di daerah rawa-rawa. Di Indonesia penyakit

tersebut merupakan penyakit rakyat yang endemis, oleh karena penyakit tersebut sudah lama diderita oleh banyak penduduk di daerah pantai, daerah persawahan, perkebunan, dan daerah hutan. (Mardiana, Dwi Fibrianto ; 2009 ; 12)

Penyakit Malaria merupakan penyakit infeksi yang tidak hanya menyerang manusia melainkan mahluk hidup lainnya seperti unggas, primata, hewan melata bahkan hewan pengerat. Secara *epidemiologi,* infeksi malaria terhadap manusia dapat menyerang tanpa memandang usia dan jenis kelamin karena penularan penyakit malaria merupakan penyakit infeksi yang ditularkan melalui gigitan nyamuk *Anopheles* yang membawa *mikroorganisme uniselular* berupa *protozoa* parasit yang tergolong dalam golongan *Plasmodium*. Penyebab Malaria adalah infeksi oleh parasit *Plasmodium* yang ditularkan dari satu manusia yang lain dengan gigitan nyamuk malaria yang dikenal dengan nyamuk *Anopheles*. Pada manusia, parasit tersebut bermigrasi ke hati di mana mereka melepaskan bentuk lain. Jika ini terjadi, mereka dapat memasuki aliran darah dan menginfeksi sel-sel darah merah.

Ada empat jenis *plasmodium* yaitu *plasmodium vivax, plasmadium ovale, malariae plasmodium dan plasmodium falciparum* yang menyebabkan penyakit malaria. Khusus untuk *plasmodium falciparum* sering menjurus kepada sakit malaria berat yang sangat sering menyebabkan kematian (pada tahun 2010 diperkirakan 90% angka kematian akibat malaria terjadi di Sub-Sahara Afrika dimana *plasmodium falciparum* bertanggung jawab atas sebagian besar kasus malaria yang terjadi), sedangkan tiga jenis *plasmodium* lainnya adalah penyakit ringan yang sangat jarang menjurus pada Penyakit Malaria akut. Selain itu

adapula *plasmodium knowlesi* yang umumnya menyebabkan malaria pada *spesies* hewan kera tetapi dapat juga menginfeksi manusia walaupun sangat kecil kemungkinannya. (Kemenkes RI ; 2011 : 3).

Plasmodium falciparum, salah satu organisme penyebab malaria, merupakan jenis yang paling berbahaya dibandingkan dengan jenis *Plasmodium* lain yang menginfeksi manusia, yaitu *P.vivax, P.malariae,* dan *P.ovale*. saat ini, *P.falciparum* merupakan salah satu *species* penyebab malaria yang paling banyak diteliti. Hal tersebut karena spesies ini banyak menyebabkan angka kesakitan dan kematian pada manusia, selain juga karena dapat ditumbuhkan dalam jangka waktu yang lama secara *in vitro*.(Harijanto,P.N ; 2008 : 17).

II.6.1 Jenis-Jenis Penyakit Malaria

Ada 4 jenis penyakit malaria pada manusia :

a. Malaria Tertiana

Malaria Tertiana disebabkan oleh parasit ***Plasmodium vivax*** yang ditularkan oleh penyakit *Anopheles*. Pada hari-hari pertama panas irregular, kadang-kadang remiten atau intermiten. Pada saat itu perasaan dingin atau menggigil jarang terjadi. Pada akhir minggu tipe panas menjadi intermiten dan periodik setiap 48 jam dengan gejala klasik *trias malaria.* Serangan paroksismal biasanya terjadi waktu sore hari. Kepadatan parasit mencapai maksimal dalam waktu 7-14 hari. Pada minggu kedua limpa mulai teraba. *Parasitemia* mulai menurun setelah 14 hari, limpa masih membesar dan panas masih berlangsung. Pada akhir minggu kelima, panas mulai turun secara krisis. *Manivestasi* klinis *malaria vivaks* dapat berat, tetapi tidak

terlalu berbahaya, limpa dapat membesar sampai derajat 4 atau 5 (ukuran *Hackett*). *Malaria selebral* dapat terjadi walaupun jarang (Pada *P.vivax multinucleatum).*(Harijanto,P.N ; 2008 : 92).

Terdapat 3 tipe *relaps* pada *malaria vivaks* yang bergantung pada sub *spesies Plasmodium* :

Tipe I : Inkubasi pendek (12-20 hari), *relaps* sering terjadi dan *periode laten* tidak memanjang

Tipe II : Inkubasi pendek (2-20 hari), *periode laten* panjang (7-13 bulan), diikuti satu atau lebih *relaps* selama *periode laten*.

Tipe III : Inkubasi panjang (6-9 bulan), *periode laten* panjang (7-13 bulan), *relaps* terjadi sesudah serangan primer yang terlambat atau selama *periode laten.*

b. *Malaria Quartana*

Malaria quartana disebeabkan oleh infeksi parasit ***Plasmodium malariae***. Banyak dijumpai di Afrika, Amerika Latin, dan sebagian Asia. Penyebaran tidak seluas *P. Vivax* dan *P.falciparum.* masa inkubasi 18 hari atau lebih panjang (30-40 hari). Manifestasi klinis pada *malaria quartana* hanya berlangsung lebih ringan, parasit dapat dijumpai di darah sebelum gejala timbul. Gejala mulai sering *insidous*, sering terjadi mual dan muntah, *herpes labialis* sering ditemukan, *anemia* jarang, splenomegali sering dijumapi walaupun pembesaran ringan. Serangan *paroksismal* terjadi tiap 3-4 hari, biasanya pada waktu sore dan *parasitemia* sangat rendah <1%. (Harijanto,P.N; 2008 : 93).

c. Malaria Tropica

Malaria Tropika – Plasmodium falciparum. Malaria tropica adalah jenis penyakit malaria yang disebabkan oleh parasit *Plasmodium falcifarum. Malaria tropica* merupakan bentuk yang paling berat, ditandai dengan panas *iregular, anemia, splenomegali, parasitemia* yang banyak, dan sering terjadi komplikasi. Masa inkubasi 9-14 hari. *Malaria tropika* mempunyai perlangsungan yang cepat, dan *parasitemia* tinggi dan menyerang semua bentuk *eritrosit*.

Gejala *prodromal* yang sering dijumpai, yaitu sakit kepala, nyeri belakang/tungkai, lesu, perasaan dingin, mual, muntah, dan diare. Parasit sulit ditemukan pada penderita dengan pengobatan *supresif*. Panas biasanya iregular dan tidak periodik, sering terjadi *hiperpireksia* dengan temperatur di atas 40°C. Gejala lain berupa *konvulsi, pneumonia aspirasi* dan banyak keringat walaupun temperatur normal. Jika infeksi memberat nadi cepat, mual, muntah, diare menjadi berat diikuti kalainan paru (batuk). *Splenomegali* dijumpai lebih sering dan nyeri pada perabaan, hati membesar dan timbul *ikterus*. Kelainan *urine* dapat berupa *albuminuria, hialin*, dan *kristal* yang *granular*. *Anemia* lebih menonjol dengan *leukopenia* dan *monositosis*. (Harijanto,P.N ; 2008 : 94).

d. Malaria Pernisiosa

Malaria Pernisiosa - *Plasmodium Ovale*. *Malaria Pernisiosa* di sebabkan oleh parasit *Plasmodium ovale*. Merupaka bentuk yang paling ringan diantara

semua jenis malaria. Masa inkubasi 11-15 hari, walaupun periode laten dapat sampai 4 tahun, serangan *paroksismal* 3-4 hari dan jarang lebih dari 10 kali walaupun tanpa terapi dan serangan *paroksismal* terjadi malam hari. Jika terjadi infeksi campuran dengan *Plasmodium* lain, *maka P.ovale* tidak akan tampak di daerah tepi, tetapi *Plasmodium* lain yang akan ditemukan. Gejala klinis hampir sama dengan *malaria vivaks*, lebih ringan, puncak panas lebih rendah dan perlangsungan lebih pendek, dan dapat sembuh spontan tanpa pengobatan. Serangan menggigil jarang terjadi dan *splenomegali* jarang sampai dapat diraba. *Parasitemia* seperti pada *malaria vivaks*, dan *gametosit* terlihat pada minggu pertama. (Harijanto,P.N ; 2008 : 94).