BAB II

LANDASAN TEORI

II.1. Kecerdasan Buatan

Kecerdasan buatan adalah mesin yang mampu berpikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan manusia. Menurut John McCarthy dari Stanfort mendefinisikan kecerdasan adalah kemampuan untuk mencapai sukses dalam menyelesaikan suatu permasalahan. Kecerdasan buatan juga merupakan aplikasi, dan intruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia. Yang bertujuan membuat mesin menjadi lebih pintar, memahami apa itu kecerdasan, membuat mesin lebih bermanfaat. (T.Sutojo, Edi Mulyanto;2011:1).

II.2. Tipe Pengetahuan dalam Penjelasan Sistem Pakar

Secara khusus ada tiga tipe pengetahuan yang akan dibahas yaitu :

1. ***Resoning Domain Knowledge*** (***RDK***), merupakan doamain pengetahuan yang dikodekan oleh domain pakar dalam sistem pakar yang sesuai.
2. ***Communication Doamain Knowledge (CDK),*** merupakan pengetahuan tentang doamain pakar yang diperlukan untuk komunikasi tentang domain itu.

3. ***Domain Communication Knowledge (DCK),*** merupakan pengetahuan tentang bagaimana cara mengkomunikasikan doamain itu. (Rika Rosnelly; 2012:108).

II.3. Sistem Pakar

Sistem pakar adalah merupakan cabang dari *Artificial Intelligence* yang sampai saat ini sudah banyak sistem pakar yang dibuat untuk mendiagnosis penyakit, dan untuk mengidentifikasi struktur molekul campuran yang tidak dikenal, dan untuk membantu konfigurasi sistem komputer besar, dan untuk membantu memberikan keputusan bagi seorang pakar.

Sistem pakar merupakan sebuah sistem yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan kedalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia. (T.Sutojo;2011:160).

II.3.1. Kelebihan Sistem Pakar

Sistem pakar memiliki beberapa fitur menarik yang merupakan kelebihannya adalah :

1. Meningkatkan produktifitas.
2. Membuat orang awam seperti orang pakar.
3. Meningkatkan kualitas.
4. Dapat beroperasi dilingkungan berbahaya.

5. Memudahkan akses pengetahuan seorang pakar.
6. Meningkatkan kapabilitas sistem komputer.
7. Mampu bekerja dengan informasi yang tidak lengkap.
8. Bisa digunakan sebagai media pelengkap. (T.Sutojo;2011:161).

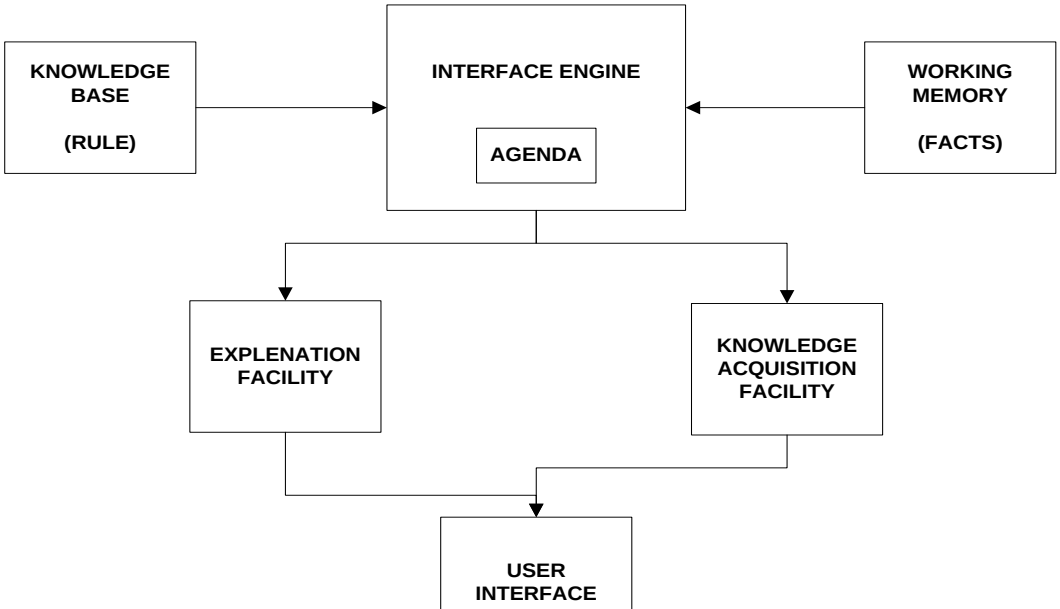
II.3.2. Karakteristik Sistem Pakar

Sistem pakar umumnya dirancang untuk memenuhi beberapa karakteristik umum berikut ini :

1. Terbatas.
2. Dapat memberi penawaran.
3. Mudah dimodifikasi.
4. Keluarannya bersifat anjuran. (T.Sutojo;2011:162).

II.3.3. Struktur Sistem Pakar

Adapun struktur sistem pakar dapat dilihat pada Gambar II.1.



Gambar II.1 : Struktur Sistem Pakar
(Sumber : Rika Rosnelly; 2012:13)

Komponen yang terdapat pada dalam struktur sistem pakar ini adalah *knowledge base (rules), inference engine, working memory, explanation facility, knowledge acquisition facility, dan user interface.* (Rika Rosnelly; 2012: 13).

II.3.4. Pakar

Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus serta mampu menerapkannya untuk memecahkan masalah atau memberi nasihat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal baru yang berkaitan dengan topik permasalahan, dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakarannya. (T.Sutojo;2011:163).

Seorang pakar memiliki kemampuan kepakaran, yaitu :

1. Mengenali dan memformulasikan permasalahan.
2. Menyelesaikan masalah dengan cepat dan tepat.
3. Menerangkan pemecahannya.
4. Restrukturisasi pengetahuan.
5. Belajar dari pengalaman.
6. Memecahkan aturan-aturan. (T.Sutojo; 2011: 164).

II.3.5. Perceptron

Secara umum teori perceptron termasuk salahsatu bentuk Jaringan Syaraf Tiruan *(Neural Network)* yang sederhana. Perceptron biasanya digunakan untuk mengklasifikasi suatu tipe pola tertentu yang sering dikenal dengan istilah pemisahan secara linear. Pada dasarnya perceptron pada Jaringan Syaraf Tiruan *(Neural Network)* dengan satu lapisan memiliki bobot yang bisa diatur dan suatu nilai ambang. Algoritma yang digunakan oleh aturan perceptron ini akan mengatur parameter-parameter bebasnya melalui proses pembelajaran. Fungsi aktivasi dibuat sedemikian rupa sehingga terjadi pembatasan antara daerah positif dan daerah negatif. Untuk mengontrol perubahan bobot pada setiap iterasi, besarnya nilai lebih besar dari 0 (nol) dan maksimal 1. Besarnya perubahan bobot yang terjadi pada setiap iterasi adalah :

$\Delta w_i = \alpha . x_i . y_i$
$\Delta b = \alpha . y_i$
dengan, w_i = sinyal input
α = laju pembelajaran
y_i = sinyal output

Dengan demikian bobot yang baru adalah :

$$w_i(baru) = w_i(lama) + \alpha . x_i . y_i$$
$$b(baru) = b(lama) + \alpha . y_i$$

(Sumber : Yani Immarul Za'iim;2013:4)

Inisialisasi semua bobot dan bias (dimanimya w_1 = 0) Tentukan laju pembelajaran l a elemen vektor input yang respon unit outputnya tidak sama dengan target 09k, lakukan Set aktivasi input xi = si dengan s adalah vector input, Hitung respon unit output, Perbaiki bobot pola yang mengandung kesalahan (y ≠ t) menurut persamaan :

$$w_i(baru) = w_i(lama) + \Delta w$$
dengan $\Delta w = \alpha . t . x_i$
$$b(baru) = b(lama) + \Delta b$$

(Sumber : Yani Immarul Za'iim;2013:4)

Iterasi dilakukan terus hingga semua pola memiliki output jaringan yang sama dengan targetnya (jaringan sudah memahami pola), Perubahan bobot hanya dilakukan pada pola yang mengandung kesalahan (output jaringan target). Perubahan tersebut merupakan hasil kali unit input dengan target dan laju pembelajaran. Perubahan bobot hanya akan terjadi kalau unit input 0.

II.4.1. Jaringan Saraf Tiruan

II.4.1. Jaringan Saraf Tiruan Perceptron

Perceptron pada Jaringan Syaraf Tiruan *(Neural Network)* termasuk kedalam salah satu bentuk Jaringan Syaraf *(Neural Network)* yang sederhana. Perceptron biasanya digunakan untuk mengklasifikasikan suatu tipe pola tertentu yang sering dikenal dengan istilah pemisahan secara linear. Pada dasarnya perceptron pada Jaringan Syaraf Tiruan *(Neural Network)* denga lapisan memiliki bobot yang bisa diatur dan suatu nilai ambang. Algoritma yang digunakan oleh aturan perceptron ini akan mengatur parameter-parameter bebasnya melalui proses pembelajaran. Fungsi aktivasi dibuat sedemikian rupa sehingga terjadi pembatasan antara daerah positif dan daerah negatif. (Yani Immarul Za'iim;2013:3).

II.4.2. Jaringan Saraf Tiruan Manusia

Jaringan saraf manusia adalah organ yang berfungsi untuk mengatur, mengkoodinir ,dan menggabungkan segala fungsi dan aktifitas tubuh, serta adaptasi terhadap lingkungan di sekitar tubuh. Terhadap lingkungan di sekitar tubuh. Jaringan saraf terdiri dari gabungan dan susunan dari sel-sel saraf yang

kemudian membentuk jaringan dan mengolah sumber informasi. Neuron dalam jaringan saraf jumlahnya mencapai 10 sampai 12 buah dengan jumlah yang begitu banyak otak dapat mengenali objek, melakukan perhitungan, dan mengontrol organ-organ tubuh dengan sangat cepat. (Yani Immarul Za'iim;2013:2).

II.6. Kusta

Kusta merupakan suatu penyakit yang mendatangkan stigma sehingga menimbulkan masalah psikososial bagi penderita kusta, keluarga maupun masyarakat. Keluarga sebagai *support system* diharapkan mampu memberikan dukungan psikososial bagi penderita kusta sehingga masalah psikososial yang muncul dapat teratasi. Psikoedukasi keluarga merupakan suatu strategi yang dapat diterapkan untuk membantu keluarga dalam memberikan dukungan psikososial keluarga kepada penderita kusta. Penelitian ini bertujuan untuk mengidentifikasi pengaruh psikoedukasi keluarga terhadap dukungan psikososial keluarga kepada anggota keluarga yang menderita kusta. (Desi Ariyana Rahayu ; 2011).

Penyakit kusta adalah penyakit kronik yang disebabkan oleh kuman *Mycobacterium leprae* yang pertama kali menyerang susunan saraf tepi, selanjutnya dapat menyerang kulit, mukosa (kulit), saluran pernafasan bagian atas, mata, otot, tulang dan testis. Kusta atau *lepra* atau disebut juga *Morbus Hansen* merupakan penyakit menular kronis yang disebabkan oleh bakteri *Mycobacterium leprae* , melalui mulut dan mukosa hidung. Penyakit kusta pertama menyerang saraf tepi, kulit serta organ tubuh yang lain kecuali susunan saraf pusat yang apabila tidak didiagnosi dan diobati secara dini dapat menimbulkan kecacatan.

Penyakit kusta adalah penyakit infeksi yang kronik, penyebabnya ialah *Mycobacterium leprae,* lalu mulut dan mukosa traktus respiratorius bagian atas, kemudian dapat ke organ-organ lain kecuali susunan saraf pusat. (Desi Ariyana Rahayu ; 2011).

II.7. Pemrograman Visual Basic 2010

Visual basic 2010 merupakn salah satu bagian dari produk pemograman terbaru yang dikeluarkan oleh Microsoft, sebagai produk lingkungan pengembangan terintegrasi atau IDE andalan yang dikeluarkan Microsoft. Visual Studio merupakan produk pemograman andalan dari Microsoft Corporation, yang didalamnya berisi beberap IDE pemograman seperti Visual Basic, Visual C++, Visual Web Developer, Visual C# dan Visual F#. Semua IDE tersebut sudah mendukung penuh implementasi. (Wahana Komputer; 2010: 2).

II.8. *Unified Modelling Language* (UML)

UML sesuai dengan kata terakhir dari kepanjangannya, UML itu adalah salah satu bentuk *language* atau bahasa. Menurut pencetusnya, UML didefenisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, membuat model, dan mendokumentasikan adpek-aspek dari sebuah sistem. Karena tergolong bahasa visual, UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari sistem yang sedang dimodelkan (Yuni Sugiarti; 2013: 34).

Unified Modelling Language (UML) biasanya digunakan untuk :

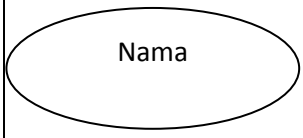
1. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksakan secara umum.
3. Menggambarkan representasi stuktur statik sebuah sistemdalam bentuk *class diagram*.
4. Membuat model *behavior* "yang menggambarkan sifat atau sebuah sistem.
5. Menyatakan arsitektur implementasi fisik menggunakan *component* dan *development diagram.*
6. Menyampaikan atau memperluas *fungsionalty* dengan *stereotypes.* (Yuni Sugiarti; 2013: 34).

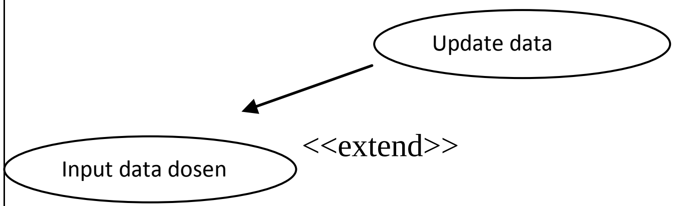
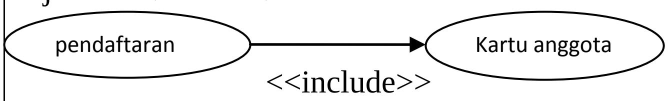
II.8.1. *Use Case Diagram* (Diagram *Use Case*)

Use case diagram secara garis besar menggambarkan interaksi antara sistem, sistem ekstrenal, dan pengguna. Dengan kata lain secara garis besar *use case diagram* secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna (*user*) mengharapkan interaksi dengan sistem itu. Terdapat beberapa simbol dalam menggambarkan diagram use case, yaitu use case, actor, relasi.

Berikut ini adalah simbol-simbol yang ada pada diagram *use case* :

Tabel II.1 Simbol-Simbol *Use case Diagram*

Simbol	Deskripsi
Use case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dengan menggunakan kata kerja di awal frase nama *use case*.
Aktor Nama actor	Orang, proses, atau lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu adalah orang, biaanya dinyatakan menggunakan kata benda di awal fase nama aktor.
Asosiasi / association _____	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case miliki interaksi dengan actor

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 cccase yang di bambahkan, arah panah menunjukkan pada use case yang dituju.contoh: 
Include <<include>>	Relasi use case tambahan sebuah use case di mana 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 use case, include berarti use case yang ditambahkan akan selalu di panggil saat use case tambahan dijalankan. Contoh: 

II.8.2.

(Sumber: Yuni Sugiarti; 2013: 42)

Diagram kelas atau c*lass diagram* menggambarkan struktur sistem dari pendefenisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang dimaksud dengan atribut dan metode atau operasi.

Diagram kelas menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewaris, asosiasi, dan lain-lain.

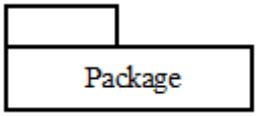
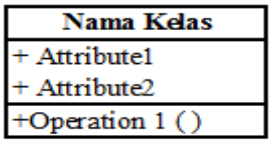
Kelas memiliki tiga area pokok :

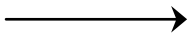
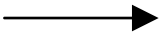
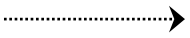
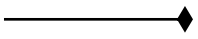
1. Nama
2. Atribut
3. Operasi

Contoh kelas “Manusia”, Atribut: nama, manusia, tanggal lahir, Method/operasi: berjalan, makan, minum

Berikut ini adalah simbol-simbol yang ada pada *class* diagram :

Tabel II.2 Simbol-Simbol *Class Diagram*

Simbol	Deskripsi
Package 	Package merupakan bungkusan dari satu atau lebih kelas
Operasi 	Kelas pada struktur sistem
Antarmuka/interface	Sama dengan konsep *interface* dalam pemograman berorientasi objek

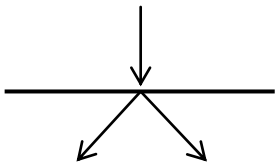
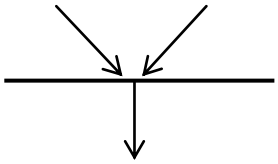
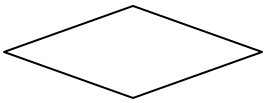
interface	
Asosiasi 1 1..*	Relasi antar kelas dengan maknaumum, asosiasi biasanya juga disertai dengan *multiplicty*
Asosiasi berarah / directed asosiasi 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicty*
Generalisasi 	Relasi antar kelas dengan makna *generalisasi-spesialisasi* (umum-khusus)
Kebergantungan (*defedency)* 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi 	Relasi antar kelas dengan makna semua-bagian (*whole-part*)

(Sumber: Yuni Sugiarti; 2013: 59)

II.8.3. *Actifity Diagram* (Diagram Aktivitas)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. Simbol *Activity Diagram*

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

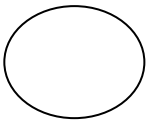
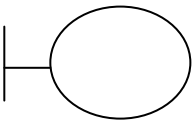
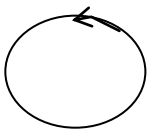
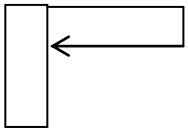
(*Sumber : Windu Gata ; 2013 : 6*)

II.8.4. *Sequence Diagram* (Diagram Rangkaian)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan

dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.4 dibawah ini :

Tabel II.4. Simbol *Sequence Diagram*

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

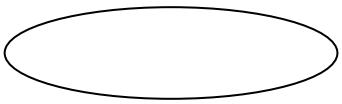
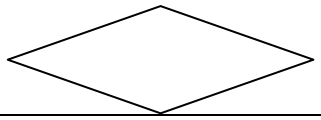
(*Sumber : Windu Gata ; 2013 : 7*)

II.9. *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) Al Fatta (2007:121) mengemukakan *Entity Relationship Diagram* adalah gambar atau diagram yang menunjukkan

informasi dibuat, disimpan, dan digunakan dalam sistem bisnis.Adapun simbolnya sebagai berikut :

Tabel II.5 Simbol-Simbol *Entity Relationship Diagram* (*ERD)*

No	Simbol	Keterangan Fungsi
1	Entitas	Persegi panjang menyatakan himpunan entitas adalah orang, kajadian, atau berada dimana data akan dikumpulkan.
2	Atribut 	Atribut merupakan informasi yang diambil tentang sebuah entitas.
3	Relasi 	Belah ketupat menyatakan himpunan relasi merupakan hubungan antar entitas.
4	Link	Garis sebagai penghubung antara himpunan, relasi, dan himpunan entitas dengan atributnya.

II.10. Teknik Normalisasi

(Sumber: Ibnu Aqil; 2010: 6)

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi

adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional, bentuk-bentuk normalisasi yaitu :

1. **Bentuk normal tahap pertama (1" normal form)**

   Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

2. **Bentuk normal tahap kedua (2nd normal form)**

   Bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. bentuk normal kedua jika dia berada pada bentuk normal kedua berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

3. **Bentuk normal tahap ketiga (3rd normal form)**

   Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

4. **Boyce Code Normal Form (BCNF)**

   Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup

karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

5. Bentuk Normal Tahap Keempat dan Kelima

Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalue merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalue (MVD). Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Janner Simarmata ; 2010 : 76).