

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Pengertian Sistem**

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan / berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu. Contoh : Sistem komputer, terdiri dari software, hardware, dan brainware ( Asbon Hendra, S.Kom. ; 2012 ; 157 ).

##### **II.1.1. Karakteristik Sistem**

###### a. Komponen (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem, tidak peduli betapa pun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi fungsi sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut supra sistem. Sebagai contoh, suatu perusahaan dapat disebut suatu sistem dan industri merupakan suatu sistem yang lebih besar dapat disebut supra sistem. Kalau dipandang industri sebagai suatu sistem, maka perusahaan dapat disebut sebagai subsistem. Demikian juga

apabila perusahaan dipandang sebagai sistem, maka sistem akuntansi adalah subsistemnya.

b. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari subsistem yang satu dengan yang lainnya berbeda tapi tetap saling berinteraksi. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan Luas Sistem (*Environment*)

*Environment* merupakan segala sesuatu dari luar batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnahkan atau dikendalikan agar tidak mengganggu operasi sistem.

d. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya untuk membentuk satu kesatuan sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. Dengan kata lain, *output* dari suatu subsistem akan menjadi *input* dari subsistem lainnya.

e. Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenanceinput*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan sinyal (*signalinput*) adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh, di dalam sistem komputer, program adalah *maintenanceinput* yang digunakan untuk mengoperasikan komputernya dan data adalah *signalinput* untuk diolah menjadi informasi.

f. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah sistem, meliputi *output* yang berguna, contohnya informasi yang dikeluarkan oleh komputer. Dan *output* yang tidak berguna dikenal sebagai sisa pembuangan, contohnya panas yang dikeluarkan oleh komputer.

g. Pengolah Sistem (*Process*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan. Contoh CPU pada komputer, bagian produksi yang mengubah bahan baku menjadi barang jadi, serta bagian akuntansi yang mengolah data transaksi menjadi laporan keuangan.

h. Tujuan Sistem (*Goal*)

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi input yang dibutuhkan dan output yang dihasilkan. Dengan kata lain, suatu sistem akan dikatakan berhasil kalau

pengoperasian sistem itu mengenai sasaran atau tujuannya. Jika sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya (Asbon Hendra, S.Kom. ; 2012 ; 158 - 160).

## II.2. Ginjal

Ginjal adalah organ ekskresi dalam vertebrata yang berbentuk mirip kacang. Sebagai bagian dari sistem urin, ginjal berfungsi menyaring kotoran (terutama urea) dari darah dan membuangnya bersama dengan air dalam bentuk urin. Cabang dari kedokteran yang mempelajari ginjal dan penyakitnya disebut nefrologi.

Manusia memiliki sepasang ginjal yang terletak di belakang perut atau abdomen. Ginjal ini terletak di kanan dan kiri tulang belakang, di bawah hati dan limpa. Di bagian atas (*superior*) ginjal terdapat kelenjar adrenal (juga disebut *kelenjar suprarenal*).

Ginjal adalah sepasang organ saluran kemih yang terletak di rongga retroperitoneal bagian atas. Bentuknya menyerupai kacang dengan sisi cekungnya menghadap ke medial. Kedua ginjal terletak di sekitar vertebra T12 hingga L3. Ginjal kanan biasanya terletak sedikit di bawah ginjal kiri untuk memberi tempat untuk hati. Sebagian dari bagian atas ginjal terlindungi oleh iga ke sebelas dan duabelas. Kedua ginjal dibungkus oleh dua lapisan lemak (lemak perirenal dan lemak pararenal) yang membantu meredam goncangan.

Berat dan besar ginjal bervariasi hal ini tergantung jenis kelamin, umur, serta ada tidaknya ginjal pada sisi lain. Pada orang dewasa, rata-rata ginjal

memiliki ukuran panjang sekitar 11,5 cm, lebar sekitar 6 cm dan ketebalan 3,5 cm dengan berat sekitar 120-170 gram atau kurang lebih 0,4% dari berat badan.

Ginjal memiliki bentuk seperti kacang dengan lekukan yang menghadap ke dalam. Di tiap ginjal terdapat bukaan yang disebut hilus yang menghubungkan arteri renal, vena renal, dan ureter. Aliran darah ginjal berasal dari arteri renalis yang merupakan cabang langsung dari aorta abdominalis, sedangkan yang mengalirkan darah balik adalah vena renalis yang merupakan cabang vena kava inferior. Sistem arteri ginjal adalah tidak ada anastomosis ke cabang arteri lain bagian paling luar dari ginjal disebut korteks, bagian lebih dalam lagi disebut medulla. Bagian paling dalam disebut pelvis.

Pada bagian medulla ginjal manusia dapat pula dilihat adanya *piramida* yang merupakan bukaan saluran pengumpul. Ginjal dibungkus oleh jaringan fibros tipis dan mengkilap yang disebut kapsula fibrosa ginjal dan diluar kapsul ini terdapat jaringan lemak perirenal. Di sebelah atas ginjal terdapat kelenjar adrenal. Ginjal dan kelenjar adrenal dibungkus oleh fasia gerota. Unit fungsional dasar dari ginjal adalah nefron yang dapat berjumlah lebih dari satu juta buah dalam satu ginjal normal manusia dewasa. Nefron berfungsi sebagai regulator air dan zat terlarut (terutama elektrolit) dalam tubuh dengan cara menyaring darah, kemudian mereabsorpsi cairan dan molekul yang masih diperlukan tubuh. Molekul dan sisa cairan lainnya akan dibuang. Reabsorpsi dan pembuangan dilakukan menggunakan mekanisme pertukaran lawan arus dan kotranspor. Hasil akhir yang kemudian diekskresikan disebut urine. Sebuah nefron terdiri dari sebuah komponen penyaring yang disebut korpuskula (atau badan Malphigi) yang

dilanjutkan oleh saluran-saluran (*tubulus*). Setiap korpuskula mengandung gulungan kapiler darah yang disebut glomerulus yang berada dalam kapsula Bowman. Setiap glomerulus mendapat aliran darah dari arteriaferen. Dinding kapiler dari glomerulus memiliki pori-pori untuk filtrasi atau penyaringan. Darah dapat disaring melalui dinding epitelium tipis yang berpori dari glomerulus dan kapsula Bowman karena adanya tekanan dari darah yang mendorong plasma darah. Filtrat yang dihasilkan akan masuk ke dalam tubulus ginjal. Darah yang telah tersaring akan meninggalkan ginjal lewat arterieferen. Di antara darah dalam glomerulus dan ruangan berisi cairan dalam kapsula Bowman terdapat tiga lapisan:

- a. kapiler selapis sel endotelium pada glomerulus
- b. lapisan kaya protein sebagai membran dasar
- c. selapis sel epitel melapisi dinding kapsula Bowman (*podosit*)

Dengan bantuan tekanan, cairan dalam darah didorong keluar dari glomerulus, melewati ketiga lapisan tersebut dan masuk ke dalam ruangan dalam kapsula Bowman dalam bentuk filtrat glomerular. Filtrat plasma darah tidak mengandung sel darah ataupun molekul protein yang besar. Protein dalam bentuk molekul kecil dapat ditemukan dalam filtrat ini. Darah manusia melewati ginjal sebanyak 350 kali setiap hari dengan laju 1,2 liter per menit, menghasilkan 125 cc filtrat glomerular per menitnya. Laju penyaringan glomerular ini digunakan untuk tes diagnosa fungsi ginjal.

Tubulus ginjal merupakan lanjutan dari kapsula Bowman. Bagian yang mengalirkan filtrat glomerular dari kapsula Bowman disebut tubulus konvolusi

proksimal. Bagian selanjutnya adalah lengkung Henle yang bermuara pada tubulus konvulasi distal. Lengkung Henle diberi nama berdasar penemunya yaitu Friedrich Gustav Jakob Henle pada awal tahun 1860-an. Lengkung Henle menjaga gradien osmotik dalam pertukaran lawan arus yang digunakan untuk filtrasi. Sel yang melapisi tubulus memiliki banyak mitokondria yang menghasilkan ATP dan memungkinkan terjadinya transpor aktif untuk menyerap kembali glukosa, asam amino, dan berbagai ion mineral. Sebagian besar air (97.7%) dalam filtrat masuk ke dalam tubulus konvulasi dan tubulus kolektivus melalui osmosis. Cairan mengalir dari tubulus konvulasi distal ke dalam sistem pengumpul yang terdiri dari:

- a. tubulus penghubung
- b. tubulus kolektivus kortikal
- c. tubulus kolektivus medularis

Tempat lengkung Henle bersinggungan dengan arteri aferen disebut aparatus juxtaglomerular, mengandung macula densa dan sel juxtaglomerular. Sel juxtaglomerular adalah tempat terjadinya sintesis dan sekresi renin. Cairan menjadi makin kental di sepanjang tubulus dan saluran untuk membentuk urin, yang kemudian dibawa ke kandung kemih melewati ureter. Ginjal mengatur pH, konsentrasi ion mineral, dan komposisi air dalam darah.

Ginjal mempertahankan pH plasma darah pada kisaran 7,4 melalui pertukaran ion hidronium dan hidroksil. Akibatnya, urine yang dihasilkan dapat bersifat asam pada pH 5 atau alkalis pada pH 8. Kadar ion natrium dikendalikan melalui sebuah proses homeostasis yang melibatkan aldosteron untuk meningkatkan penyerapan

ion natrium pada tubulus konvolusi. Kenaikan atau penurunan tekanan osmotik darah karena kelebihan atau kekurangan air akan segera dideteksi oleh hipotalamus yang akan memberi sinyal pada kelenjar pituitari dengan umpan balik negatif. Kelenjar pituitari mensekresi hormon antidiuretik (*vasopresin*, untuk menekan sekresi air) sehingga terjadi perubahan tingkat absorpsi air pada tubulus ginjal. Akibatnya konsentrasi cairan jaringan akan kembali menjadi 98% ([id.wikipedia.org/wiki/Ginjal](http://id.wikipedia.org/wiki/Ginjal)).

## **II.2.1. Penyakit Pada Ginjal**

### **a. Batu Ginjal**

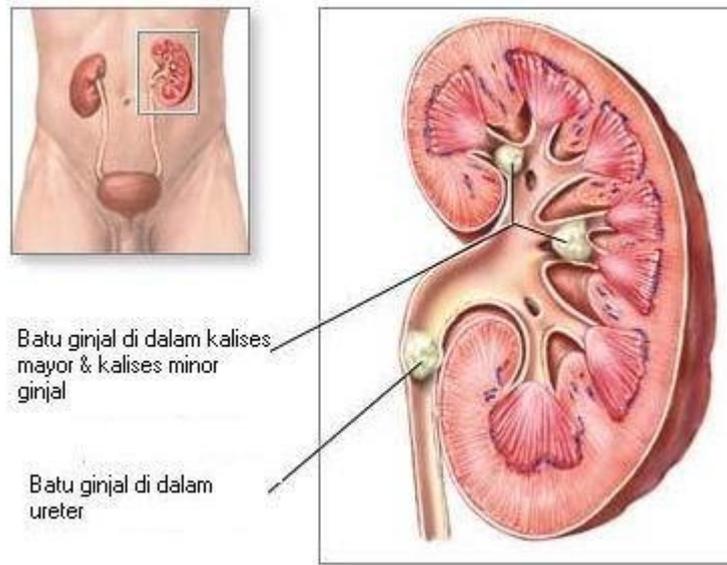
Penyakit batu ginjal merupakan salah satu penyakit paling sering ditemui dan dialami oleh banyak masyarakat Indonesia yang umumnya dialami pria. Pada umumnya penyakit ginjal disebabkan oleh rutinitas pekerjaan yang membuat pola makan menjadi tidak teratur, adanya faktor keturunan yang juga memiliki peranan penting karena jika terdapat keluarga yang memiliki penyakit ginjal, resiko diturunkan penyakit ginjal pada anak 6 kali lebih besar, kurangnya konsumsi air putih, jarang buang air kecil atau sering ditahan, banyak mengonsumsi makanan atau minuman yang mengandung bahan kimia, bahan pengawet dan lingkungan suhu udara disekitar tempat tinggal dan tempat bekerja yang tidak mendukung aktivitas sehari-hari. Penyakit batu ginjal memang banyak melanda orang Asia dan Afrika khususnya Indonesia yang diliputi berbagai macam kultur, suhu udara yang cenderung sering kali berubah tidak menentu, pola

hidup dan gaya hidup yang terkadang salah, dsb. Penyakit ginjal memang lebih dominan menyerang kaum pria dibanding wanita, hal ini terbukti dari survei yang diperkirakan bahwa pria yang berusia 70 tahun keatas memiliki resiko lebih besar terserang penyakit ginjal hingga 80% dibanding wanita.

Batu ginjal terbentuk disebabkan oleh adanya peningkatan pada bakteri dan saluran kandung kemih yang terinfeksi bakteri pemecah urea dan urine yang kemudian membentuk batu pada kandung kemih. Jika tubuh kekurangan cairan atau kurang minum air putih, akan terjadi kepekatan urine yang semakin meningkat yang mempermudah pembentukan batu ginjal.

Batu ginjal memiliki komponen penyusun batu ginjal melalui proses pembentukan batu ginjal yang terdiri dari 80% batu kalsium, kalsium okalat dan kalsium fosfat.

Batu ginjal dapat disebabkan oleh peningkatan pH urine (misalnya batu kalsium bikarbonat) atau penurunan pH urine (misalnya batu asma urat). Konsentrasi bahan-bahan pembentuk batu yang tinggi di dalam darah dan urine serta kebiasaan makan atau obat tertentu, juga dapat merangsang pembentukan batu. Segala sesuatu yang menghambat aliran urine dan menyebabkan stasis (tidak ada pergerakan) urine di bagian mana saja di saluran kemih, meningkatkan kemungkinan pembentukan batu.



**Gambar II.1. Batu Ginjal**

**Sumber : Mary Baradero, SPC, MN, Mary Wilfrid Dayrit, SPC, MAN & Yakobus Siswadi, MSN (2005 : 59-67)**

Batu di ginjal itu sendiri mungkin asimtomatik kecuali apabila batu tersebut menyebabkan obstruksi atau timbul infeksi. Umumnya batu ginjal tidak menimbulkan gejala. Gejala baru nyata ada jika batu tersangkut di saluran kemih ginjal atau kalau turun memasuki ureter atau jika menyumbat muara kandung kemih.

Komplikasi dari batu ginjal itu sendiri dapat disertai oleh batu ginjal yang disertai hipertensi dan batu ginjal disertai diabetes. Jika penderita asam urat memiliki penyakit hipertensi maka tekanan darah haruslah diturunkan hingga kembali ke batas tekanan darah normal dengan tekanan darah yang normal tentunya dapat membantu meringankan batu ginjal yang terjadi di saluran kemih.

**Pencegahannya dapat dilakukan dengan cara :**

- a. Istirahat yang cukup
- b. Kendalikan stress
- c. Minum air putih sekurang-kurangnya 2 liter sehari
- d. Kurangi makanan yang mengandung garam dan banyak minyak
- e. Yang penting adalah mensyukuri yang sudah diterima dan dimiliki
- f. Minum jus mengkudu, mentimun, cincau rambat, labu siam, seledri atau belimbing manis.

Tekanan darah yang tinggi juga memberi pengaruh yang cukup tinggi bagi timbulnya komplikasi pada penyakit lainnya termasuk batu ginjal. Beberapa obat penurun tekanan darah dapat mengakibatkan intensitas berkemih semakin tinggi namun keadaan ginjal yang diliputi oleh batu ginjal dengan gejala yang sama akan semakin memberatkan kerja ginjal untuk mengeluarkan urine dari ginjal ke kandung kemih dan kemudian di buang ( Mary Baradero, SPC, MN, Mary Wilfrid Dayrit, SPC, MAN & Yakobus Siswadi, MSN ; 2005 : 59-67 ).

**b. Infeksi Ginjal**

Infeksi ginjal merupakan infeksi yang terjadi di saluran kencing atau kandung kemih dan bisa merambat hingga ke ginjal. Infeksi ginjal, atau pyelonephritis, adalah salah satu jenis infeksi saluran kencing. Penyakit ini membutuhkan penanganan yang menyeluruh.

Penanganan yang tepat akan mencegah kerusakan ginjal atau menyebar hingga ke aliran darah dan menyebabkan infeksi yang fatal.

Infeksi ginjal ditunjukkan dengan :

- a. Demam.
- b. Nyeri pada punggung, tubuh bagian samping, dan paha.
- c. Nyeri pada perut.
- d. Sering buang air kecil.
- e. Sering mengalami keinginan untuk buang air kecil.
- f. Sensasi terbakar atau nyeri ketika buang air kecil.
- g. Terdapat cairan atau darah di urine (*hematuria*).

Infeksi ginjal biasanya terjadi ketika bakteri masuk ke saluran kencing dan mulai berkembang biak. Bakteri yang berasal dari infeksi di bagian tubuh lain juga bisa menyebar ke aliran darah dan masuk ke ginjal.

Kondisi seperti ini dapat terjadi jika bagian tubuh buatan mengalami infeksi. Bagian tubuh buatan, misalnya katup jantung buatan atau sendi buatan, yang digunakan untuk menggantikan bagian tubuh asli yang rusak.

Infeksi ginjal juga dapat muncul setelah operasi ginjal. Tapi, hal ini jarang sekali terjadi.

Faktor peningkat risiko terkena infeksi ginjal, yaitu :

- a. Perempuan. Saluran kencing perempuan lebih pendek dari laki-laki, sehingga bakteri lebih mudah masuk ke kandung kemih.

- b. Hambatan pada saluran kencing, seperti batu ginjal, kelainan struktur sistem buang air kecil, atau membesarnya kelenjar prostat.
- c. Hambatan ini membuat seseorang tidak bisa mengeluarkan semua urine yang ada di kandung kemih.
- d. Melemahnya sistem kekebalan tubuh.
- e. Kerusakan syaraf di sekitar kandung kemih.
- f. Penggunaan kateter urine yang terlalu lama.
- g. Kondisi yang menyebabkan urine masuk kembali ke kandung kemih (*vesicoureteral reflux*).

Infeksi ginjal yang dibiarkan dapat berujung pada :

- a. Kerusakan ginjal permanen.
- b. Keracunan darah (*septicemia*). Ginjal yang terinfeksi akan menyebarkan infeksinya lewat darah yang masuk ke organ tersebut.
- c. Komplikasi kehamilan.

Dokter akan melihat gejala infeksi ginjal yang terjadi. Setelah itu, dokter akan mengambil sampel urine untuk mengetahui apakah ada bakteri, cairan, atau darah di dalamnya.

Untuk mengatasi infeksi ginjal, biasanya dokter akan memberikan antibiotik. Jenis obat, dosis, dan berapa lama harus diminum bergantung pada kondisi kesehatan dan bakteri yang menginfeksi. Namun untuk infeksi yang cukup parah, penderitanya harus dirawat di rumah sakit. Pengobatannya melibatkan penggunaan infus.

Sedangkan, untuk infeksi berulang atau infeksi yang kondisinya memburuk, penanganannya lebih serius lagi. Infeksi ginjal yang disebabkan kelainan struktur harus diperbaiki dengan operasi.

Penderita infeksi ginjal harus minum air yang banyak. Selain untuk mencegah dehidrasi, sering buang air kecil juga dapat membantu mengeluarkan bakteri dari saluran kencing.

Infeksi ginjal bisa dicegah dengan langkah-langkah berikut :

- a. Minum air yang banyak.
- b. Sering buang air kecil dan jangan menahan keinginan buang air kecil.
- c. Kencinglah setelah berhubungan seksual.
- d. Bagi perempuan, bersihkan dan keringkan vagina setelah buang air kecil atau buang air besar dengan benar. Bersihkan dari depan ke belakang, bukan dari belakang ke depan.
- e. Bersihkan vagina dan anus dengan perlahan-lahan.
- f. Jangan gunakan sabun pembersih khusus yang dijual bebas di pasaran (Prof H. M. Hembing Wijayakusuma ; 2008 : 90-91).

### **c. Ginjal kronis**

Gagal ginjal kronik (GGK) adalah salah satu penyakit tidak menular, merupakan keadaan gangguan fungsi ginjal yang bersifat menahun berlangsung progresif dan irreversible (tidak dapat kembali ke keadaan semula). Dimana kemampuan tubuh gagal untuk mempertahankan metabolisme dan keseimbangan cairan dan elektrolit yang

menyebabkan uremia (retensi urea dan sampah nitrogen lain dalam darah).

Penyakit gagal ginjal kronik memang merupakan masalah kesehatan yang cukup penting di negeri kita. Menurut catatan Sub Bagian ginjal. Penyebab gagal ginjal kronik yang sering dijumpai adalah batu, infeksi yang disebut pirlonefritis, hipertensi, nefropati karena asam urat, nefropati karena lupus dan kencing manis.

Hilangnya fungsi cadangan ginjal seringkali tidak disadari penderita. Pada gagal ginjal kronik gangguan fungsi ginjal acapkali sudah disertai gejala yang nyata dalam aktivitas sehari-hari. Penderita mulai menunjukkan gejala anemia (pada suami ibu sudah terdapat penurunan hemoglobin).

Tes kreatinin klirens dapat membedakan berat ringannya gangguan fungsi ginjal. Pada keadaan normal Tes Kreatinin Klirens (TKK) adalah 100 sampai 125 ml/mm. Pada TKK 75 sampai 100 sudah terjadi hilangnya fungsi cadangan ginjal. Sedangkan TKK 25 sampai 75 disebut keadaan insufisiensi ginjal. Pada TKK 5 sampai 25 digolongkan gagal ginjal kronik. TKK yang di bawah 5 disebut gagal ginjal terminal.

Gagal ginjal kronik dan gagal ginjal terminal memerlukan perhatian khusus karena bila dibiarka dapat menjurus keadaan yang membahayakan jiwa penderita. Pada gagal ginjal kronik dapat dimulai terapi konservatif yang bertujuan menghilangkan gejala yang

mengganggu penderita, sehingga penderita dapat hidup secara normal. Komponen utama terapi konservatif adalah diet yaitu dengan mengatur asupan protein. Di samping itu juga harus diatur air dan garam, vitamin, elektrolit, dan asam amino esensial diberikan jika diperlukan.

Penderita gagal ginjal kronik acapkali mengeluh mual sehingga asupan makannya dapat terbatas. Karena itu evaluasi asupan makanan perlu dilakukan dengan baik. bila terapi konservatif ini dapat dijalankan dengan baik dan fungsi ginjal dapat dipertahankan maka belum diperlukan terapi cuci darah.

Gagal ginjal kronik atau penyakit ginjal tahap akhir adalah penyimpangan progresif, fungsi ginjal yang tidak dapat pulih dimana kemampuan tubuh untuk mempertahankan keseimbangan metabolik dan cairan dan elektrolit mengalami kegagalan yang mengakibatkan uremia. Kondisi ini mungkin disebabkan oleh glomerulonefritis kronis. Preparat lingkungan dan okupasi yang telah menunjukkan mempunyai dampak dalam gagal ginjal kronis termasuk timah, kadmium, merkuri, dan kromium. Pada akhirnya dialisis atau transplantasi ginjal diperlukan untuk menyelamatkan pasien.

Faktor penyebab terjadinya gagal ginjal kronik adalah radang ginjal menahun, batu ginjal dan batub saluran kemih yang kurang mendapat perhatian, obat-obatan modern ataupun tradisional yang digunakan

dalam jangka waktu lama, hipertensi, diabetes, narkoba, serta penyakit ginjal turunan (genetik).

*Faktor penyebab gagal ginjal kronik harus diobati untuk menghambat laju proses gagal ginjal agar tidak menjadi gagal ginjal terminal, atau gagal ginjal tidak dapat berfungsi lagi.* Tekanan darah dan gula darah harus dikendalikan, dan antibiotik secara teratur diberikan bila terjadi infeksi. Jangan sampai terjadi infeksi pada salah satu ginjal yang dapat dengan mudah menular pada ginjal yang lain. Penderita harus menjalaninya dengan kemauan untuk sembuh yang tinggi dan disiplin ketat. Olahraga pun harus dibatasi hanya yang ringan, seperti jalan kaki dan berenang secukupnya.

Di sisi lain, akibat pemecahan protein tubuh yang meningkat penderita gagal ginjal terminal perlu mendapat terapi nutrisi agar kecukupan protein untuk keperluan perbaikan jaringan tubuh. Selain itu, perlu mengatur keseimbangan cairan elektrolit, mencegah penurunan massa tulang dan kelemahan otot, memperbaiki gangguan irama jantung yang tidak seimbang (aritmia), dan menghambat peningkatan lemak tubuh. Penderita juga perlu mempertahankan kekebalan tubuh menghadapi infeksi virus dan bakteri.

Apa penyebab Penyakit Ginjal Kronis ?

- a. Diabetes Melitus (kencing manis)
- b. Hipertensi
- c. Batu ginjal / saluran kemih

- d. Dehidrasi
- e. Makanan olahan yang mengandung zat kimia berbahaya bagi ginjal
- f. Obat-obatan
- g. Radang kronis pada penyaring ginjal
- h. Kelainan (penyakit) ginjal turunan

Bagaimana gejalanya ?

- a. Cepat lelah
- b. Kurang bertenaga
- c. Menurunnya nafsu makan
- d. Mual, muntah
- e. Sulit tidur
- f. Kram otot (terutama pada malam hari)
- g. Pembengkakan pada kaki/pergelangan kaki
- h. Kulit gatal dan kering
- i. Bengkak seputar mata (pada pagi hari)
- j. Seringkali ingin berkemih pada malam hari

Bagaimana diagnosis nya ?

Deteksi dini dilakukan pada mereka yang berisiko tinggi, yaitu penderita diabetes, hipertensi, umur tua (>60 tahun), penderita dengan riwayat sakit ginjal sebelumnya, atau mempunyai keluarga yang menderita sakit ginjal/gangguan fungsi ginjal.

Pemeriksaan yang dilakukan untuk skrining adalah:

Pemeriksaan albumin rutin, Sedimen urin, Kreatinin/Cystatin C

Pengobatan yang dilakukan :

- a. Terapi
- b. Obat
- c. Operasi
- d. Cuci Darah
- e. Cangkok Ginjal

Pencegahan yang dapat dilakukan :

- a. Kontrol diabetes
- b. Kontrol dan jaga tekanan darah
- c. Menurunkan berat badan (bagi yang obesitas), serta kontrol kadar lipid (lemak)
- d. Berhenti merokok

Dampak yang ditimbulkan oleh Penyakit Ginjal Kronis

Komplikasi :

- a. Anemia
- b. Osteodistrofi Ginjal (gangguan fungsi ginjal akibat pengendapan mineral tulang)
- c. Gagal jantung
- d. Disfungsi ereksi (impotensi)

( Diane C. Baugman & JoAnn C. Hackley ; 1996 : 171-174 ).

### **II.3. Pakar**

Menurut Kusrini (2008 : 3), pakar adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Sebagai contoh, dokter adalah seorang pakar yang mampu mendiagnosis penyakit yang diderita pasien serta dapat memberikan penatalaksanaan terhadap penyakit tersebut. Tidak semua orang dapat mengambil keputusan mengenai diagnosis dan memberikan penatalaksanaan suatu penyakit.

### **II.4. Sistem Pakar**

keputusannya Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar, sistem pakar yang mencoba memecahkan masalah yang biasanya hanya bisa dipecahkan oleh seorang pakar, dipandang berhasil ketika mampu mengambil keputusan seperti yang dilakukan oleh pakar aslinya baik dari sisi proses pengambilan maupun hasil keputusan yang diperoleh (Kusrini ; 2008 : 3).

### **II.5. Logika Fuzzy**

Konsep logika Fuzzy dicetuskan oleh Lofty A. Zadeh, seorang profesor *University of California* di Berkeley, dan dipresentasikan bukan sebagai metodologi kontrol, namun sebagai suatu cara pemrosesan data yang memperbolehkan anggota himpunan parsial daripada anggota himpunan kosong atau non-anggota. Pendekatan ini pada teori himpunan tidak diaplikasikan untuk mengontrol sistem sampai tahun 70-an karena kurangnya kemampuan komputer-mini pada saat itu. Profesor Zadeh beralasan bahwa masyarakat tidak butuh ketepatan, *input* informasi numeris, dan mereka belum sanggup dengan kontrol

adaptif yang tinggi. Jika kembalian dari kontroler dapat diprogram untuk menerima *noisy, input* yang tidak teliti, mereka akan lebih efektif dan lebih mudah diimplementasikan (Kusrini ; 2008 : 37 ).

### II.5.1. Metode Fuzzy

Ada beberapa istilah yang perlu diketahui dalam memahami sistem *fuzzy*, yaitu:

a. Variabel *fuzzy*

Variabel *fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*. Contoh: Umur, Temperatur, Permintaan, Persediaan, Produksi, dan sebagainya. (Sri Kusumadewi dan Hari Purnomo, 2004: 6)

b. Himpunan *fuzzy*

Misalkan  $X$  semesta pembicaraan, terdapat  $A$  di dalam  $X$  sedemikian sehingga:  $A = \{ x, \mu_A[x] \mid x \in X, \mu_A : x \rightarrow [0,1] \}$  Suatu himpunan *fuzzy*  $A$  di dalam semesta pembicaraan  $X$  didefinisikan sebagai himpunan yang bercirikan suatu fungsi keanggotaan  $\mu_A$ , yang mengawankan setiap  $x \in X$  dengan bilangan real di dalam interval  $[0,1]$ , dengan nilai  $\mu_A(x)$  menyatakan derajat keanggotaan  $x$  di dalam  $A$  (Athia Saelan, 2009: 2).

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Misalkan  $X = \text{Umur}$  adalah variabel *fuzzy*. Maka dapat didefinisikan himpunan “Muda”, “Parobaya”, dan “Tua” (Jang dkk ,1997:17).

c. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan

himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

Contoh: semesta pembicaraan untuk variabel umur:  $[0, +\infty)$ . Sehingga semesta pembicaraan dari variable umur adalah  $0 \leq \text{umur} < +\infty$ . Dalam hal ini, nilai yang diperbolehkan untuk dioperasikan dalam variable umur adalah lebih besar dari atau sama dengan 0, atau kurang dari positif tak hingga. (Sri Kusumadewi dan Hari Purnomo, 2004:7)

d. Domain

Domain himpunan *fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif. Contoh domain himpunan *fuzzy*: Muda =  $[0,45]$  (Sri Kusumadewi dan Hari Purnomo, 2004: 8).

### II.5.2. Metode Fuzzy Tsukamoto

Inferensi adalah proses penggabungan banyak aturan berdasarkan data yang tersedia. Komponen yang melakukan inferensi dalam sistem pakar disebut mesin inferensi. Dua pendekatan untuk menarik kesimpulan pada *IF-THEN rule* (aturan jika-maka) adalah *forward chaining* dan *backward chaining* (Turban dkk, 2005:726).

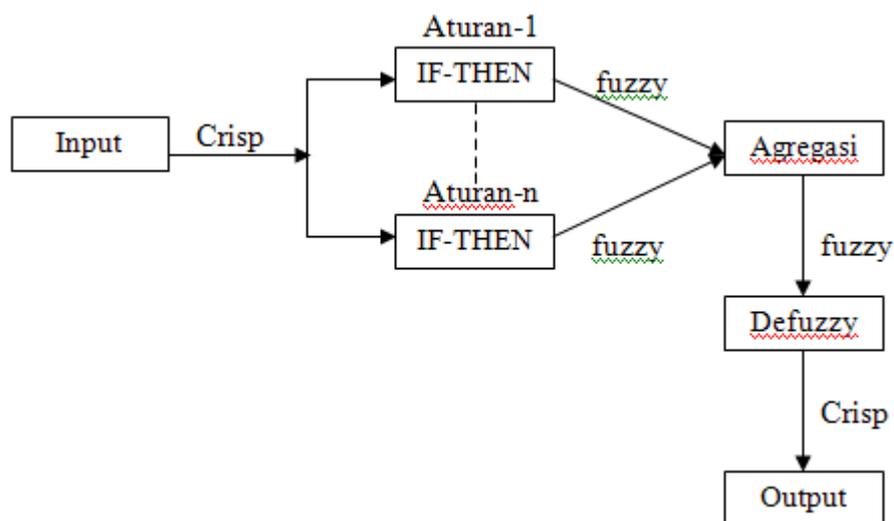
**a. Forward chaining**

*Forward chaining* mencari bagian JIKA terlebih dahulu. Setelah semua kondisi dipenuhi, aturan dipilih untuk mendapatkan kesimpulan. Jika kesimpulan yang diambil dari keadaan pertama, bukan dari keadaan yang terakhir, maka ia akan digunakan sebagai fakta untuk disesuaikan dengan kondisi JIKA aturan yang lain untuk mendapatkan kesimpulan yang lebih baik. Proses ini berlanjut hingga dicapai kesimpulan akhir .

**b. Backward chaining**

*Backward chaining* adalah kebalikan dari *forward chaining*. Pendekatan ini dimulai dari kesimpulan dan hipotesis bahwa kesimpulan adalah benar. Mesin inferensi kemudian mengidentifikasi kondisi JIKA yang diperlukan untuk membuat kesimpulan benar dan mencari fakta untuk menguji apakah kondisi JIKA adalah benar. Jika semua kondisi JIKA adalah benar, maka aturan dipilih dan kesimpulan dicapai. Jika beberapa kondisi salah, maka aturan dibuang dan aturan berikutnya digunakan sebagai hipotesis kedua. Jika tidak ada fakta yang membuktikan bahwa semua kondisi JIKA adalah benar atau salah, maka mesin inferensi terus mencari aturan yang kesimpulannya sesuai dengan kondisi JIKA yang tidak diputuskan untuk bergerak satu langkah ke depan memeriksa kondisi tersebut. Proses ini berlanjut hingga suatu set aturan didapat untuk mencapai kesimpulan atau untuk membuktikan tidak dapat mencapai kesimpulan. Menurut Sri Kusumadewi dan Sri Hartati (2006:34) sistem inferensi *fuzzy* merupakan suatu kerangka komputasi yang didasarkan pada teori himpunan *fuzzy*,

aturan *fuzzy* yang berbentuk IF-THEN, dan penalaran *fuzzy*. Secara garis besar, diagram blok proses inferensi *fuzzy* terlihat pada Gambar 2.3.



**Gambar II.2. Diagram Blok Sistem Inferensi Fuzzy**

**Sumber : (Sri Kusumadewi dan Sri Hartati, 2006: 34)**

Sistem inferensi *fuzzy* menerima input *crisp*. Input ini kemudian dikirim ke basis pengetahuan yang berisi  $n$  aturan *fuzzy* dalam bentuk IF-THEN. *Fire strength* (nilai keanggotaan anteseden atau  $\alpha$ ) akan dicari pada setiap aturan. Apabila aturan lebih dari satu, maka akan dilakukan agregasi semua aturan. Selanjutnya pada hasil agregasi akan dilakukan *defuzzy* untuk mendapatkan nilai *crisp* sebagai *output* sistem. Salah satu metode FIS yang dapat digunakan untuk pengambilan keputusan adalah metode *Tsukamoto*. Berikut ini adalah penjelasan mengenai metode FIS *Tsukamoto*.

Pada metode *Tsukamoto*, implikasi setiap aturan berbentuk implikasi “Sebab-Akibat”/Implikasi “*Input-Output*” dimana antara anteseden dan konsekuen harus ada hubungannya. Setiap aturan direpresentasikan menggunakan himpunan-himpunan *fuzzy*, dengan fungsi keanggotaan yang monoton. Kemudian untuk

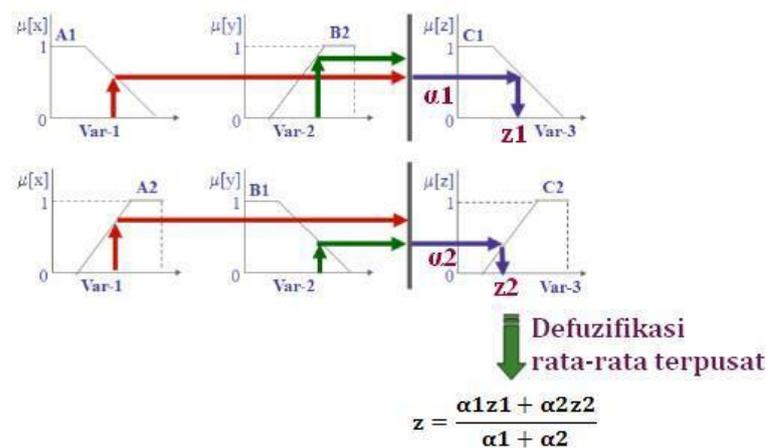
menentukan hasil tegas (*Crisp Solution*) digunakan rumus penegasan (defuzifikasi) yang disebut “Metode rata-rata terpusat” atau “Metode defuzifikasi rata-rata terpusat (*Center Average Defuzzifier*)” (Setiadji, 2009: 200).

Misalkan ada 2 variabel input, Var-1 (x) dan Var-2(x), serta variabel output, Var-3(z), dimana Var-1 terbagi atas 2 himpunan yaitu A1 dan A2. Var-2 terbagi atas 2 himpunan B1 dan B2, Var-3 juga terbagi atas 2 himpunan yaitu C1 dan C2 (C1 dan C2 harus monoton). Ada 2 aturan yang digunakan, yaitu:

[R1] IF (x is A1) and (y is B2) THEN (z is C1)

[R2] IF (x is A2) and (y is B1) THEN (z is C2)

Pertama-tama dicari fungsi keanggotaan dari masing-masing himpunan *fuzzy* dari setiap aturan, yaitu himpunan A1, B2 dan C1 dari aturan *fuzzy* [R1], dan himpunan A2, B1 dan C2 dari aturan *fuzzy* [R2]. Aturan *fuzzy* R1 dan R2 dapat direpresentasikan dalam Gambar 2.4 untuk mendapatkan suatu nilai crisp Z.



**Gambar II.3. Inferensi dengan menggunakan Metode Tsukamoto**

**Sumber : (Sri Kusumadewi dan Hari Purnomo, 2004:34).**

Karena pada metode *Tsukamoto* operasi himpunan yang digunakan adalah konjungsi (*AND*), maka nilai keanggotaan anteseden dari aturan *fuzzy* [R1] adalah

irisan dari nilai keanggotaan A1 dari Var-1 dengan nilai keanggotaan B1 dari Var-2, maka nilai keanggotaan anteseden dari operasi konjungsi (*And*) dari aturan *fuzzy* [R1] adalah nilai minimum antara nilai keanggotaan A1 dari Var-1 dan nilai keanggotaan B2 dari Var-2. Demikian pula nilai keanggotaan anteseden dari aturan *fuzzy* [R2] adalah nilai minimum antara nilai keanggotaan A2 dari Var-1 dengan nilai keanggotaan B1 dari Var-2. Selanjutnya, nilai keanggotaan anteseden dari aturan *fuzzy* [R1] dan [R2] masing-masing disebut dengan  $\alpha_1$  dan  $\alpha_2$ . Nilai  $\alpha_1$  dan  $\alpha_2$  kemudian disubstitusikan pada fungsi keanggotaan himpunan C1 dan C2 sesuai aturan *fuzzy* [R1] dan [R2] untuk memperoleh nilai  $z_1$  dan  $z_2$ , yaitu nilai  $z$  (nilai perkiraan produksi) untuk aturan *fuzzy* [R1] dan [R2]. Untuk memperoleh nilai output *crisp*/nilai tegas  $Z$ , dicari dengan cara mengubah input (berupa himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*) menjadi suatu bilangan pada domain himpunan *fuzzy* tersebut. Cara ini disebut dengan metode defuzifikasi (penegasan). Metode defuzifikasi yang digunakan dalam metode *Tsukamoto* adalah metode defuzifikasi rata-rata terpusat (*Center Average Defuzzyfier*).

## II.6. Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* atau ERD adalah gambar atau diagram yang menunjukkan informasi yang dibuat, disimpan, dan digunakan dalam sistem bisnis. Entitas biasanya menggambarkan jenis informasi yang sama. Dalam entitas digunakan untuk menghubungkan antar entitas yang sekaligus menunjukkan hubungan antar data. Pada akhirnya ERD juga bisa digunakan untuk menunjukkan

aturan-aturan bisnis yang ada pada sistem informasi yang akan dibangun ( Hanif Al Fatta ; 2007 ; 121-122).

## **II.7. Kamus Data**

Menurut Raymond McLeod Jr dan George P Schell dalam buku Sistem Informasi Manajemen, Kamus data (*Data Dictionary*) mencakup definisi-definisi dari data yang disimpan didalam basis data dan dikendalikan oleh sistem manajemen basis data. Struktur basis data yang dimuat dalam basis data adalah kumpulan dari seluruh definisi *field*, definisi tabel, relasi tabel, dan hal-hal lainnya. Nama *field* data, jenis data (seperti teks atau angka atau tanggal), nilai-nilai yang valid untuk data, dan karakteristik-karakteristik lainnya akan disimpan dalam kamus data. Perubahan-perubahan pada struktur data hanya dilakukan satu kali di dalam kamus data, program-program aplikasi yang menggunakan data tidak akan terpengaruh.

## **II.8. Normalisasi**

Normalisasi data adalah proses dimana tabel-tabel pada database dites dalam hal kesalingtergantungan diantara *field-field* pada sebuah tabel. Misalnya jika sebuah tabel terdapat ketergantungan terhadap lebih dari satu *field* dalam tabel tersebut, maka 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 unik dengan sebuah kolom kunci. Misalnya pada tabel *data\_siswa* ada *field* nomor induk dan *field* nama siswa serta *field* tanggal lahir. Maka ketergantungan fungsionalnya dapat dinyatakan sebagai berikut : *nmr\_induk* → *nm\_siswa* dan *nmr\_induk* → *tgl\_lahir*. Artinya *nm\_siswa* memiliki ketergantungan fungsional terhadap *nmr\_induk*. *Field* *nm\_siswa* isinya ditentukan oleh *nmr\_induk*. Demikian juga *field* *tgl\_lahir* isinya juga ditentukan oleh *nmr\_induk*. Maksud dari semua itu, *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 ; 2008 : 32).

### **II.8.1. Tahap - Tahap Normalisasi**

Menurut (Wahana Komputer ; 2008 : 32 - 35) yaitu :

- a. *Decomposition*, dekomposisi adalah proses mengubah bentuk pada 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. Dekomposisi akan sering dilakukan dalam proses normalisasi untuk memenuhi syarat-syaratnya.
- b. Bentuk Tidak Normal, pada bentuk ini semua data yang ada pada tiap *entity* (diambil atributnya) masih ditampung dalam satu tabel besar. data yang ada pada tabel ini masih ada yang redundansi dan ada juga yang kosong. Semuanya masih tidak tertata rapi.

- c. 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.
- d. Normal *Form* Kedua (*2nd Normal Form*), pada tahapan ini tabel dianggap memenuhi normal kedua jika pada tabel tersebut semua atribut yang bukan kunci primer tabel tersebut.
- e. Normal *Form* Ketiga (*3rd Normal Form*), setiap atribut pada tabel selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama. Bentuk normal ketiga biasanya digunakan bila masih ada tabel yang belum efisien. Biasanya penggunaan bentuk normal (normalisasi) hanya sampai pada bentuk ketiga, dan tabel yang dihasilkan telah memiliki kualitas untuk membuat sebuah *database* yang dapat diandalkan.

## II.9. Database

*Database* atau basis *data* adalah sekumpulan *data* yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer. *Data* itu sendiri adalah representasi dari semua fakta yang ada pada dunia nyata. *Database* sering digunakan untuk melakukan proses terhadap data-data tersebut untuk menghasilkan informasi. Dalam *database* ada sebutan-sebutan untuk satuan data yaitu :

1. Karakter, ini adalah satuan data terkecil. *Data* terdiri atas susunan karakter yang pada akhirnya mewakili data yang memiliki arti dari sebuah fakta.

2. *Field*, adalah kumpulan dari karakter yang memiliki fakta tertentu, misalnya seperti nama siswa, tanggal lahir, dan lain-lain.
3. *Record*, adalah kumpulan dari *field*. Pada *record* anda dapat menemukan banyak sekali informasi penting dengan cara mengombinasikan *field-field* yang ada.
4. Tabel, adalah sekumpulan dari *record-record* yang memiliki kesamaan entity dalam dunia nyata. Kumpulan tabel adalah *database* (Wahana Komputer ; 2010 ; 24).

## II.10. Sekilas Tentang Java

Menurut Supriyato (2010 : 2), *java* merupakan sebuah bahasa pemrograman berorientasi objek yang dapat berjalan pada platform yang berbeda, baik di *windows*, *linux*, serta sistem operasi lainnya. Jadi, kita dapat membuat sebuah aplikasi dengan *java* pada sistem operasi *linux* dan selanjutnya dapat menjalankan atau menginstal aplikasi tersebut pada sistem operasi *windows* dan juga sebaliknya tanpa mengalami masalah. Dengan menggunakan *java*, kita dapat mengembangkan banyak aplikasi yang dapat digunakan pada lingkungan yang berbeda, seperti pada : *Desktop*, *Mobile*, *Internet*, dan lain-lain.

Berikut ini uraian singkat mengenai paket aplikasi *java* yang tersedia :

1. J2ME ( *Java 2 Micro Edition* )

Paket instalasi ini dapat digunakan untuk mengembangkan *software* yang berjalan pada perangkat yang memiliki memori dan sumber daya yang kecil, seperti pada *Handphone*, PDA dan *Smartcard*.

2. J2SE ( *Java 2 Standard Edition* )

Paket instalasi ini dapat digunakan untuk mengembangkan aplikasi yang berjalan pada lingkungan workstation, seperti aplikasi *desktop*.

### 3. J2EE (*Java 2 Enterprise Edition*)

Paket instalasi ini dapat digunakan untuk mengembangkan aplikasi pada lingkungan *internet* maupun aplikasi skala *enterprise*.

## II.11. MySQL

Menurut Wahana Komputer (2010 : 5), *MySQL database server* adalah RDBMS (*Relational Database Management System*) yang dapat menangani data yang bervolume besar. Meskipun begitu, tidak menuntut *resource* yang besar. *MySQL* adalah *database* yang paling populer diantara *database-database* yang lain.

*MySQL* adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. *MySQL* memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*.

## II.12. Unified Modeling Language (UML)

UML singkatan dari *Unified Modelling Language* yang berarti bahasa pemodelan standart. (Chonoles; 2003 : 6) mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan *semantic*. Ketika kita membuat model menggunakan konsep *UML* ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat harus berhubungan satu dengan lainnya harus mengikuti standart yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi?

Bagaimana keamanan terhadap sistem yang ada kita buat? Dan sebagainya dapat dijawab dengan *UML*.

*UML* diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

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

*UML* telah diaplikasikan dalam investasi perbankan, lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales, dan supplier.

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 berorientasikan objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para anggota team untuk bekerja sama dalam mengaplikasikan beragam sistem. Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini. Sebagai perancang sistem mau tidak mau pasti menjumpai *UML*, baik kita sendiri yang membuat sekedar membaca diagram *UML* buatan orang lain (Prabowo Pudjo Widodo Herlawati ; 2011 ; 6).

### II.12.1. *Diagram-Diagram UML*

Beberapa literatur menyebutkan bahwa *UML* menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa yang digabung, misalnya diagram komunikasi, diagram urutan, dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

1. Diagram Kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi, serta relasi-relasi diagram. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas.
2. Diagram paket (*PackageDiagram*) bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas merupakan bagian dari diagram komponen.
3. Diagram *Use Case* bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.
4. Diagram interaksi dan *Sequence* (urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam waktu tertentu.
5. Diagram komunikasi (*Communication Diagram*) bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi *UML* yang menekankan

organisasi *structural* dari objek-objek yang menerima serta mengirim pesan.

6. Diagram *Statechart* (*Statechart Diagram*) bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*State*), transisi kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.
7. Diagram aktivitas (*Activity Diagram*) bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali antar objek.
8. Diagram komponen (*Component Diagram*) bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan diagram kelas dimana komponen dipetakan kedalam satu atau lebih kelas-kelas. Antarmuka-antarmuka serta kolaborasi-kolaborasi.
9. Diagram *Deployment* (*Deployment Diagram*) bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run time*). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram *Deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen.

Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan.

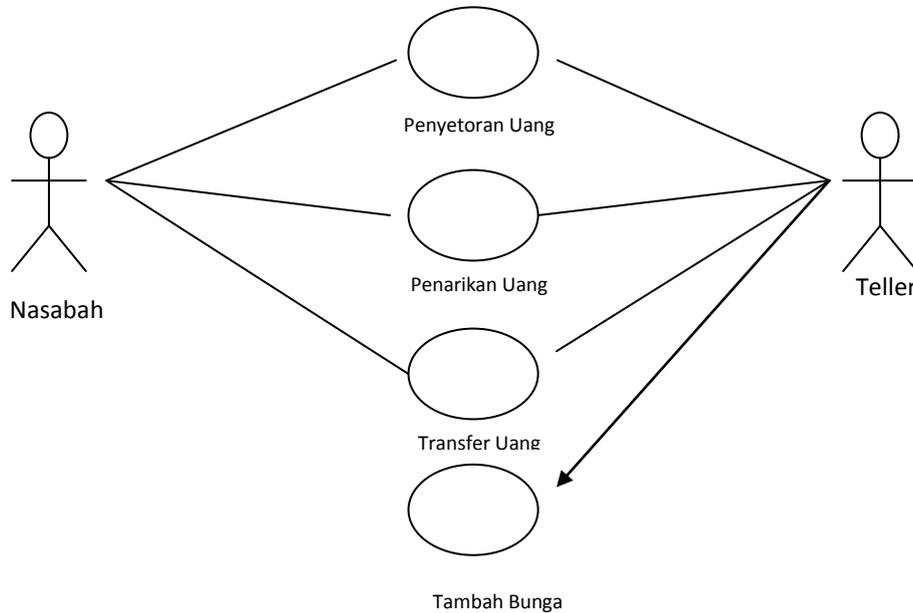
### ***Diagram Use Case (use case diagram)***

*Use Case* menggambarkan *external view* dari sistem yang akan kita buat modelnya. Menurut Pooley (2005:15) mengatakan bahwa model *use case* dapat dijabarkan dalam diagram, tetapi yang perlu diingat, diagram tidak identik dengan model karena model lebih luas dari diagram.

Komponen pembentuk diagram *use case* adalah :

- a. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
- b. *Use Case*, aktivitas/ sarana yang disiapkan oleh bisnis/sistem.
- c. Hubungan (*Link*), aktor mana saja yang terlibat dalam *use case* ini.

Gambar di bawah ini merupakan salah satu contoh bentuk diagram *use case*.

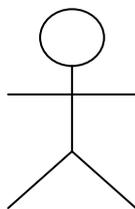


**Gambar II.4. Diagram *Use Case***

**Sumber : Probowo Pudjo Widodo (2011:17)**

### 1. Aktor

Menurut Chonoles (2003 :17) menyarankan sebelum membuat use case dan menentukan aktornya, agar mengidentifikasi siapa saja pihak yang terlibat dalam sistem kita. Pihak yang terlibat biasanya dinamakan *stakeholder*.

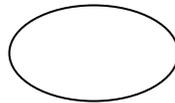


**Gambar II.5. Aktor**

**Sumber : Probowo Pudjo Widodo (2011:17)**

## 2. *Use Case*

Menurut Pilone (2005 : 21) *use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen kejadian atau kelas. Sedangkan menurut Whitten (2004 : 258) mengartikan *use case* sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario) baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Use case* digambarkan dalam bentuk *ellips/oval*



**Gambar II.6. Simbol *Use Case***

**Sumber : Probowo Pudjo Widodo (2011:22)**

*Use case* sangat menentukan karakteristik sistem yang kita buat, oleh karena itu Chonoles (2003:22-23) menawarkan cara untuk menghasilkan *use case* yang baik yakni :

### **a. Pilihlah nama yang baik**

*Use case* adalah sebuah *behaviour* (prilaku), jadi seharusnya dalam frase kata kerja. Untuk membuat namanya lebih detil tambahkan kata benda mengindikasikan dampak aksinya terhadap suatu kelas objek. Oleh karena itu diagram *use case* seharusnya berhubungan dengan diagram kelas.

### **b. Ilustrasikan perilaku dengan lengkap.**

*Use case* dimulai dari inisiasi oleh aktor primer dan berakhir pada aktor dan menghasilkan tujuan. Jangan membuat *use case* kecuali anda

mengetahui tujuannya. Sebagai contoh memilih tempat tidur (*King Size*, *Queen Size*, atau dobel) saat tamu memesan tidak dapat dijadikan *use case* karena merupakan bagian dari *use case* pemesanan kamar dan tidak dapat berdiri sendiri (tidak mungkin tamu memesan kamar tidur jenis king tapi tidak memesan kamar hotel).

**c. Identifikasi perilaku dengan lengkap.**

Untuk mencapai tujuan dan menghasilkan nilai tertentu dari aktor, *use case* harus lengkap. Ketika memberi nama pada *use case*, pilihlah frasa kata kerja yang implikasinya hingga selesai. Misalnya gunakan frasa *reserve a room* (pemesanan kamar) dan jangan *reserving a room* (memesan kamar) karena memesan menggambarkan perilaku yang belum selesai.

**d. Menyediakan *use case* lawan (*inverse*)**

Kita biasanya membutuhkan *use case* yang membatalkan tujuan, misalnya pada *use case* pemesanan kamar, dibutuhkan pula *use case* pembatalan pesanan kamar.

**e. Batasi *use case* hingga satu perilaku saja.**

Kadang kita cenderung membuat *use case* yang lebih dari satu tujuan aktivitas. Guna menghindari kerancuan, jagalah *use case* kita hanya fokus pada satu hal. Misalnya, penggunaan *use case* *check in* dan *check out* dalam satu *use case* menghasilkan ketidakfokusan, karena memiliki dua perilaku yang berbeda.

### 3. Diagram Kelas (*Class Diagram*)

Diagram kelas mempunyai dua jenis yaitu *domain class diagram* dan *design class diagram*. Fokus *domain class diagram* adalah pada sesuatu dalam lingkungan kerja pengguna, bukan pada *class* perangkat lunak yang nantinya akan anda rancang. Sedangkan *design class diagram* tujuannya adalah untuk mendokumentasikan dan menggambarkan kelas-kelas dalam pemrograman yang nantinya akan dibangun.



**Gambar II.7. Notasi *Domain Diagram Class***

**Sumber : E. Triandini dan G. Suardika (2012 : 49-50)**



**Gambar II.8. Notasi *Design Diagram Class***

**Sumber : E. Triandini dan G. Suardika (2012 : 49-50)**

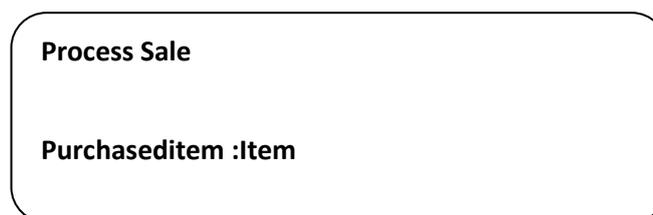
#### **4. Diagram Aktivitas (*Activity Diagram*)**

Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam kumpulan aksi-aksi. Ketika digunakan dalam pemodelan *software*, diagram aktivitas merepresentasikan pemanggilan suatu fungsi tertentu misalnya *call*. Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini menggambarkan aktivitas yang dipicu oleh kejadian-kejadian diluar seperti pemesanan atau kejadian-kejadian internal misalnya pengajian tiap jumat sore (Probowo Pudji Widodo ;2011 : 143-145).

Aktivitas merupakan kumpulan aksi-aksi. Aksi-aksi nelakukan langka sekali saja tidak boleh dipecah menjadi beberapa langkah-langkah lagi. Contoh aksinya yaitu :

- a. Fungsi Matematika
- b. Pemanggilan Perilaku
- c. Pemrosesan Data

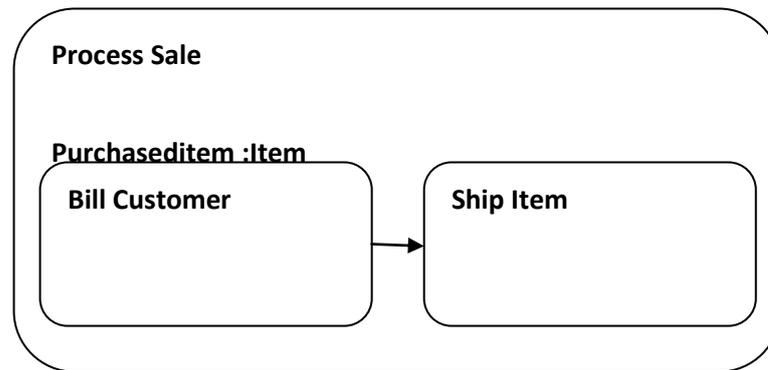
Ketika kita menggunakan diagram aktivitas untuk memodelkan perilaku suatu *classifier* dikatakan kontek dari aktivitas. Aktivitas dapat mengakses atribut dan operasi *classifier*, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku. Ketika digunakan dengan model proses bisnis, informasi itu biasanya disebut *process-relevant data*. Aktivitas diharapkan dapat digunakan ulang dalam suatu aplikasi, sedangkan aksi biasanya *specific* dan digunakan hanya untuk aktivitas tertentu.



**Gambar II.9. Aktivitas sederhana tanpa rincian**

**Sumber : Probowo Pudjo Widodo (2011:145)**

Detail aktivitas dapat dimasukan di dalam kotak. Aksi diperlihatkan dengan symbol yang sama dengan aktivitas dan namanya diletakkan didalam persegi panjang.

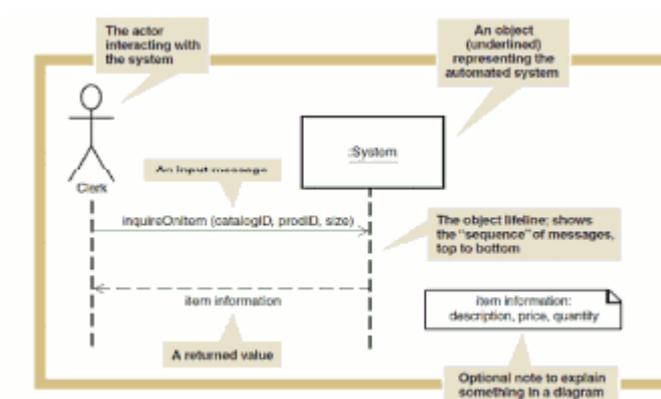


**Gambar II.10. Aktivitas dengan detail rincian**

**Sumber : Probowo Pudjo Widodo (2011:145)**

## 5. *Sequence Diagram*

Menurut John Satzinger, 2010, dalam buku *System Analysis and Design in a Changing World*, "System Sequence Diagram (SSD) adalah diagram yang digunakan untuk mendefinisikan input dan output serta urutan interaksi antara pengguna dan sistem untuk sebuah use case.



**Gambar II.11. Notasi Sequence Diagram**

**Sumber : Evi Triandini dan Gede Suardika (2012 : 71)**