

BAB II

LANDASAN TEORI

2.1. Diabetes Melitus

Diabetes adalah suatu penyakit karena tubuh tidak mampu mengendalikan jumlah gula, atau glukosa dalam aliran darah. Ini menyebabkan hiperglikemia, suatu keadaan gula darah yang tingginya sudah membahayakan (Setiabudi, 2008). Sindrom ini ditandai oleh adanya hiperglikemia dan berkaitan dengan abnormalitas metabolisme karbohidrat, lemak dan protein. Beberapa faktor risiko terjadinya penyakit DM adalah sosiodemografi meliputi umur, jenis kelamin, pekerjaan, perilaku kesehatan serta sosio budaya masyarakat khususnya dalam perubahan pola makan mereka (Waspadji, 2007).

Diabetes mellitus didefinisikan sebagai suatu penyakit atau gangguan metabolisme kronis dengan multi etiologi yang ditandai dengan tingginya kadar gula darah disertai dengan gangguan metabolisme karbohidrat, lipid dan protein sebagai akibat insufisiensi fungsi insulin. Insufisiensi fungsi insulin dapat disebabkan oleh gangguan atau defisiensi produksi insulin oleh sel-sel beta *Langerhans* kelenjar pankreas, atau disebabkan oleh kurang responsifnya sel-sel tubuh terhadap insulin (WHO, 1999). Pathogenesis DM yaitu:

1). Diabetes Tipe I

Pada diabetes tipe I terdapat ketidak mampuan untuk menghasilkan insulin karena sel-sel beta pan-kreas telah dihancurkan oleh proses autoimun. Hiper-glikemia puasa terjadi akibat produksi glukosa yang tidak terukur oleh hati. Di samping itu, glukosa yang berasal dari makanan tidak dapat disimpan dalam hati meskipun tetap berada dalam darah dan menimbulkan hiperglikemia post prandial (sesudah makan).

Jika konsentrasi glukosa dalam darah cukup tinggi, ginjal tidak dapat menyerap kembali semua glukosa yang tersaring keluar : akibatnya, glukosa tersebut muncul dalam urin (glukosuria). Ketika glukosa yang berlebihan diekskresikan ke urin, ekskresi ini akan disertai pengeluaran cairan dan elektrolit yang berlebihan pula. Keadaan ini dinamakan diuresis osmotik. Sebagai akibat dari kehilangan cairan yang berlebihan, pasien akan mengalami peningkatan dalam berkemih (poliuria) dan rasa haus (polidipsia).

Defisiensi insulin juga mengganggu metabolisme protein dan lemak yang menyebabkan penurunan berat badan. Pasien dapat mengalami peningkatan selera makan (Polifagia), akibat menurunnya simpanan kalori, gejala lainnya mencakup kelelahan dan kelemahan.

2). Diabetes Tipe II

Pada diabetes tipe II terdapat dua masalah utama yaitu yang berhubungan dengan insulin, yaitu : resistensi insulin dan gangguan sekresi insulin. Normalnya insulin akan terikat dengan reseptor khusus pada permukaan sel sebagai akibat terikatnya insulin dengan reseptor tersebut, terjadi sel resistensi insulin pada diabetes tipe II disertai dengan penurunan reaksi intra sel ini. Dengan demikian insulin menjadi tidak efektif untuk menstimulasi pengambilan glukosa oleh jaringan.

Untuk mengatasi resistensi insulin dan mencegah terbentuknya glukosa dalam darah, harus terdapat peningkatan jumlah insulin yang disekresikan pada penderita toleransi glukosa terganggu, keadaan ini terjadi akibat sekresi insulin yang berlebihan, dan kadar glukosa akan dipertahankan pada tingkat yang normal atau sedikit meningkat. Namun untuk mengimbangi peningkatan kebutuhan akan insulin, maka kadar glukosa akan meningkat dan terjadi diabetes tipe II.

2.2. Menu Gizi

Menu adalah susunan makanan yang dimakan oleh seseorang untuk sekali makan atau untuk sehari-hari. Kata "menu" bias diartikan "hidangan". Menu seimbang adalah menu yang terdiri dari beraneka ragam makanan dalam jumlah dan proporsi yang sesuai, sehingga memenuhi kebutuhan gizi seseorang guna pemeliharaan dan perbaikan sel-sel tubuh dan proses kehidupan serta pertumbuhan dan perkembangan (Almatsier, 2005). Menu seimbang adalah konsumsi makanan untuk memenuhi kebutuhan tubuh akan zat gizi. Kekurangan gizi pada salah satu makanan dengan pemberian menu seimbang dapat dicukupi oleh makanan lain. Untuk itu pemberian menu seimbang dengan makanan yang beraneka ragam sangat dibutuhkan dalam memenuhi kecukupan gizi (Almatsier, 2005).

2.3. Penatalaksanaan Diabetes Mellitus

Tujuan penatalaksanaan diet secara umum pada penderita diabetes mellitus adalah mencapai dan mempertahankan kadar glukosa darah mendekati kadar normal, mencapai dan mempertahankan lemak mendekati kadar yang optimal, mencegah komplikasi akut/kronik dan meningkatkan kualitas hidup (Waspadji, 2007).

Menurut Waspadji (2007) mengutip pendapat Joslin (1952) dari *Medical Centre Institute*, dalam penatalaksanaan diet diabetes mellitus ada 3 (tiga) J yang harus diketahui dan dilaksanakan oleh penderita DM diabetes mellitus, yaitu jumlah makanan, jenis makanan dan jadwal makanan. Berikut ini uraian mengenai ketiga hal tersebut:

1. Jumlah makanan

Jumlah makanan yang diberikan disesuaikan dengan status gizi penderita DM, bukan berdasarkan tinggi rendahnya gula darah. Jumlah kalori yang disarankan berkisar antara 1100-2900 KKal.

2. Jenis makanan

Penderita diabetes mellitus harus mengetahui dan memahami jenis makanan apa yang boleh dimakan secara bebas, makanan yang mana harus dibatasi dan makanan apa yang harus dibatasi secara ketat. Makanan yang mengandung karbohidrat mudah diserap seperti sirup, gula, sari buah harus dihindari. Sayuran dengan kandungan karbohidrat tinggi seperti buncis, kacang panjang, wortel, kacang kapri, daun singkong, bit dan bayam harus dibatasi. Buah-buahan berkalori tinggi seperti pisang, pepaya, mangga, sawo, rambutan, apel, duku, durian, jeruk dan nanas juga dibatasi. Sayuran yang boleh dikonsumsi adalah sayuran dengan kandungan kalori rendah seperti oyong, ketimun, kol, labu air, labu siam, lobak, sawi, rebung, selada, toge, terong dan tomat (Waspadji, 2007).

3. Jadwal makan

Penderita diabetes mellitus harus membiasakan diri untuk makan tepat pada waktu yang telah ditentukan. Penderita diabetes mellitus makan sesuai jadwal, yaitu 3 kali makan utama, 3 kali makan selingan dengan interval waktu 3 jam. Ini dimaksudkan agar terjadi perubahan pada kandungan glukosa darah penderita DM, sehingga diharapkan dengan perbandingan jumlah makanan dan jadwal yang tepat maka kadar glukosa darah akan tetap stabil dan penderita DM tidak merasa lemas akibat kekurangan zat gizi. Jadwal makan standar yang digunakan oleh penderita DM diabetes mellitus (Waspadji, 2007) disajikan dalam tabel :

Waktu	Jadwal	Total kalori
Pukul 7.00	Makan pagi	20%
Pukul 10.00	Selingan	10%
Pukul 13.00	Makan siang	30%
Pukul 16.00	Selingan	10%
Pukul 19.00	Makan malam	20%
Pukul 21.00	Selingan	10%

Sumber : Suyono, 1996

Tabel 2. Jadwal Makan Penderita Diabetes Melitus

Kebutuhan kalori sesuai untuk mencapai dan mempertahankan berat badan ideal. komposisi energy adalah 45 – 65 % dari karbohidrat, 10 – 20 % dari protein dan 20 – 25 % dari lemak. Ada beberapa cara untuk menentukan jumlah kalori yang dibutuhkan orang dengan diabetes. Di antaranya adalah dengan memperhitungkan berdasarkan kebutuhan kalori basal yang besarnya 25– 30 kalori/kg BB ideal, ditambah dan dikurangi bergantung pada beberapa factor yaitu jenis kelamin, umur, aktifitas, kehamilan / laktasi, adanya komplikasi dan berat badan. Cara yang lebih gampang lagi adalah dengan cara pegangan kasar, yaitu untuk pasien kurus 2300– 2500 kalori, normal 1700–2100 kalori dan gemuk 1300–1500 kalori (Soegondo, dkk, 2009:54). Kebutuhan kalori penyandang diabetes dapat dilihat di tabel berikut :

Kalori / kg BB Ideal			
Status Gizi	Kerja Santai	Sedang	Berat
Gemuk	25	30	35
Normal	30	35	40
Kurus	35	40	40-50

Tabel 3. Kebutuhan Kalori Penyandang Diabetes

WHO menetapkan suatu pengukuran/ klasifikasi obesitas yang tidak bergantung pada bias-bias kebudayaan. Para ahli sedang memikirkan untuk membuat klasifikasi IMT tersendiri untuk penduduk Asia. Hasil studi di Singapura memperlihatkan bahwa orang Singapura dengan IMT 27 – 28 mempunyai lemak tubuh yang sama dengan orang-orang kulit putih dengan IMT 30. Pada Tabel 1 dapat dilihat klasifikasi Berat Badan yang diusulkan berdasarkan IMT pada orang Eropa menurut WHO pada tahun 1995 (1), orang Asia menurut International Obesity Task Force (IOTF) dan WHO (2), dan Indonesia menurut Depkes RI

Eropa		Asia		Indonesia	
Keadaan Gizi	IMT (Kg/m ²)	Keadaan Gizi	IMT (Kg/m ²)	Keadaan Gizi	IMT (Kg/m ²)
				Kurus sekali	< 17,0
Kurus	≤ 18,5	Kurus	≤ 18,5	Kurus	17,0 – 18,4
Normal	18,5 – 24,9	Normal	18,5 – 22,9	Normal	18,5 – 25,0
Kegemukan	≥ 25	Kegemukan	≥ 23	Gemuk	25,1 – 27,0
Pre obes	25,0 – 29,9	Pre obes	23,0 – 24,9	Gemuk sekali	>27,0
Obes I	30,0 – 34,9	Obes I	25,0 – 29,9		
Obes II	35,0 – 39,9	Obes II	≥ 30,0		
Obes III	≥ 40,0				

2.4. Penelitian Terkait

Penelitian yang dilakukan oleh Gideon Abram Filando Suwarso pada tahun 2014 berjudul Sistem Pakar untuk Penyakit Anak Menggunakan Metode Forward Chaining. Penerapan Metode Forward Chaining sebagai metode untuk pengambilan kesimpulan akhir sudah sesuai dengan hasil perhitungan manual dan hasil yang diberikan oleh sistem. Hasil perhitungan menyatakan bahwa sistem pakar mampu menghasilkan akurasi sebesar 90%.

Penulis mengacu pada jurnal ini untuk perhitungan hasil pengujian

2.5. Sistem Pakar

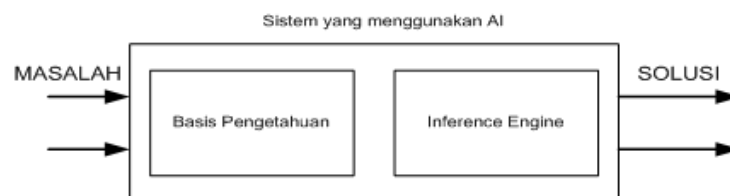
Sistem pakar merupakan cabang dari AI (*Artificial Intelligent*) yang membuat ekstensi khusus untuk spesialisasi pengetahuan guna memecahkan suatu permasalahan pada *Human Expert*. *Human Expert* merupakan seseorang yang ahli dalam suatu bidang ilmu atau *skill* khusus yang dimiliki oleh orang lain. Pengetahuan di dalam *Expert system* berasal dari orang atau *knowledge* yang berasal dari buku-buku referensi, surat kabar atau karya ilmiah orang lain. Sistem pakar mempunyai banyak definisi, tetapi pada dasarnya sistem pakar diterapkan untuk mendukung pemecahan masalah berikut ini beberapa definisi sistem pakar, antara lain:

- 1 Menurut Giarratano dan Riley (1994): Sistem pakar adalah suatu sistem yang bisa melayani atau meniru kemampuan seorang pakar.

- 2 Menurut Ignizio (1991): Sistem pakar merupakan suatu model dan prosedur yang berkaitan dalam suatu daerah tertentu, yang mana tingkat keahliannya dapat dibandingkan dengan keahlian seorang pakar.
- 3 Menurut Jhon Durkin (1994): Sistem pakar adalah program komputer yang didesain untuk meniru kemampuan memecahkan masalah dari seorang pakar, Pakar adalah orang yang memiliki kemampuan atau mengerti dalam menghadapi suatu masalah lewat pengalaman, seorang pakar mengembangkan kemampuan yang membuatnya dapat memecahkan permasalahan dengan hasil baik & efisien.
- 4 Menurut Marlin dan Ozman (1998): Sistem pakar merupakan sistem yang berbasis komputer yang menggunakan pengetahuan, fakta dan teknik penalaran dalam memecahkan suatu masalah, biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tertentu.

2.6. Inferensi (Inference Engine) dan Metode Forward Chaining Motor

Motor Inferensi (*Inference engine*) merupakan pusat pengambilan keputusan pada Sistem Pakar dengan penyesuaian fakta-fakta pada memori dengan basis pengetahuan untuk mendapatkan kesimpulan dan jawaban. *Inference engine* merupakan **otak dan pemikir dari suatu Sistem Pakar**. Dalam *inference engine* ini sistem melakukan suatu penalaran yang dilandasi oleh basis pengetahuan (didapat dari pakar) yang dimiliki sistem sehingga menghasilkan sebuah keputusan.



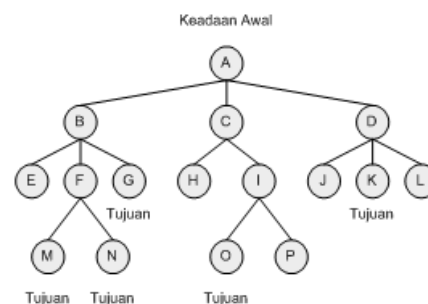
Gambar 1 Gambar Sistem menggunakan AI

Proses penalaran ada dua macam dan biasanya lebih disebut dengan proses *chaining* (*forward chaining* dan *backward chaining*). Kedua metode ini

mempunyai kelebihan tersendiri, semuanya itu tergantung terutama dari kondisi permasalahan yang dihadapi dan basis pengetahuan.

Forward chaining

Forward chaining merupakan suatu strategi pengambilan keputusan yang dimulai dari bagian sebelah kiri (*IF* lebih dahulu). Dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis.



Gambar 2 Gambar Penalaran Metode Forward Chaining

Sebagai contoh penalaran maju (*forward chaining*) adalah mengecek kerusakan mesin kendaraan bermotor akan dimulai dengan macam-macam kerusakan mesin yang akan ditelusuri kemudian dilanjutkan dengan jenis dari macam-macam kerusakan yang dipilih, dan seterusnya sampai pada diagnosis kerusakan dan hasil akhir kesimpulan kerusakan tersebut.

Forward Chaining merupakan suatu penalaran yang dimulai dari fakta untuk mendapatkan kesimpulan (conclusion) dari fakta tersebut. Forward chaining bisa dikatakan sebagai strategi inference yang bermula dari sejumlah fakta yang diketahui. Pencarian dilakukan dengan menggunakan rules yang premisnya cocok dengan fakta yang diketahui tersebut untuk memperoleh fakta baru dan melanjutkan proses hingga goal dicapai atau hingga sudah tidak ada rules lagi yang premisnya cocok dengan fakta yang diketahui maupun fakta yang diperoleh. Forward chaining bisa disebut juga runut maju atau pencarian yang dimotori data (data driven search).

Jadi pencarian dimulai dari premis-premis atau informasi masukan (if) dahulu kemudian menuju konklusi atau derived information (then). Forward Chaining berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan atau dengan menambahkan data ke memori kerja untuk diproses agar ditemukan suatu hasil. Forward Chaining digunakan jika :

- a. Banyak aturan berbeda yang dapat memberikan kesimpulan yang sama.
- b. Banyak cara untuk mendapatkan sedikit konklusi.
- c. Benar-benar sudah mendapatkan berbagai fakta, dan ingin mendapatkan konklusi dari fakta-fakta tersebut.

2.7. MySQL

Menurut Bunafit Nugroho (2004) MySQL merupakan database yang paling digemari dikalangan Programmer Web, dengan alasan bahwa program ini merupakan database yang sangat kuat dan cukup stabil digunakan sebagai media penyimpanan data. Sebagai sebuah database Server yang mampu untuk memajemen database dengan baik, MySQL terhitung merupakan database yang paling digemari dan paling banyak digunakan disbanding database lainnya.

Didalam dunia Internet, MySQL dijadikan sebagai sebuah database yang paling banyak digunakan selain database yang bersifat *share ware* seperti *Ms. Access*, penggunaan MySQL ini biasanya dipadukan dengan menggunakan program aplikasi PHP, karena dengan menggunakan kedua program tersebut di atas telah terbukti akan kehandalannya dalam menangani permintaan data.

2.8. PHP

Menurut Bunafit Nugroho (2004), PHP adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis. Maksud dari *server-side scripting* adalah sintaks dan perintah-perintah yang diberikan akan sepenuhnya dijalankan di server tetapi disertakan pada dokumen HTML. Pembuatan web ini merupakan kombinasi antara PHP sendiri sebagai bahasa pemrograman dan HTML sebagai pembangun halaman web.

Ketika seorang pengguna internet akan membuka suatu situs yang menggunakan fasilitas *server-side scripting* PHP, maka terlebih dahulu server yang bersangkutan akan memproses semua perintah PHP di server lalu mengirimkan hasilnya dalam format HTML ke web browser pengguna internet tadi. Dengan demikian seorang pengguna internet tidak dapat melihat kode program yang ditulis dalam PHP sehingga keamanan dari halaman web menjadi lebih terjamin.

PHP dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan di server.

Adapun kelebihan – kelebihan php adalah :

1. Mampu berkomunikasi dengan berbagai database yang umum dan sering digunakan. Sistem database yang didukung oleh php adalah : Oracle, MySQL, Sybase, PostgreSQL
2. PHP dapat dijalankan di berbagai sistem operasi seperti windows, linux/unix, solaris, maupun macintosh.
3. Php merupakan software open source yang dapat di download secara gratis.

4. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak. PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system

2.9. UML

Menurut Suhendar (2002), UML (*Unified Modeling Language*) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. *Artifact* dapat berupa model, deskripsi atau perangkat lunak) dari system perangkat lunak, seperti pada pemodelan bisnis dan *system* non perangkat lunak lainnya.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

2.7.1. *Static View*

Static View termasuk diagram yang memberikan gambaran dari unsur-unsur dari sistem tetapi tidak memberitahu bagaimana elemen akan berperilaku. Hal ini sangat mirip *Blueprint*. *Blueprint* itu komprehensif, tetapi mereka hanya menunjukkan apa yang tetap diam, maka disebut *Static View*. *Static View* dibentuk oleh dua diagram, yaitu *Class Diagram* dan *Object Diagram*.

2.7.2. *Dynamic View*

Pada *Dynamic View* meliputi diagram yang mengungkapkan bagaimana benda

berinteraksi dengan satu sama lain dalam respon terhadap lingkungan. Ini termasuk *Sequence Diagram* dan *Collaboration Diagram*, yang kolektif disebut sebagai diagram interaksi. Mereka secara khusus dirancang untuk menjelaskan bagaimana benda berbicara satu sama lain. Ini juga mencakup *Statechart Diagram*, yang menunjukkan bagaimana dan mengapa perubahan objek dari waktu ke waktu dalam menanggapi lingkungan.

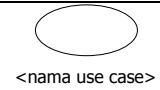
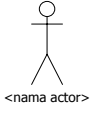
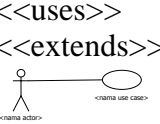
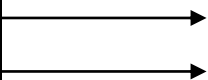
2.7.3. *Functional View*

Functional View terbentuk oleh *Use Case Diagram* dan *Activity Diagram*.

2.7.4. *Use Case Diagram*

Use case diagram menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (*actor*). Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia luar. *Use case* diagram dapat digunakan selama proses analisis untuk menangkap *requirements system* dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, *use case* diagram menetapkan perilaku sistem saat diimplementasikan. Sebuah model mungkin terdapat satu atau beberapa *use case* diagram.






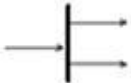
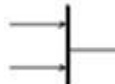


Tabel 1. Simbol *Use Case Diagram* (Suhendar, 2002)

Simbol	Nama	Deskripsi
	<i>Case</i>	Menggambarkan proses / kegiatan yang dapat dilakukan oleh aktor
	<i>Actor</i>	Menggambarkan entitas / subyek yang dapat melakukan suatu proses
	<i>Relation</i>	Relasi antara case dengan aktor ataupun case dengan case lain.
		

2.7.5. Activity Diagram

Activity diagram memodelkan alur kerja sebuah proses bisnisdan urutan aktivitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah *flowchart* karena kita dapat memodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas ke dalam keadaan sesaat (*state*). *Activity diagram* juga sangat berguna ketika kita ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai *use case* berinteraksi.

Tabel 2. Simbol *Activity Diagram* (Suhendar, 2002)

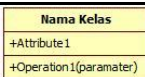





No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Mempelihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Start State</i>	Bagaimana objek dibentuk atau diawali
4		<i>End State</i>	Bagaimana objek dibentuk dihancurkan
5		<i>State Transition</i>	Menunjukkan kegiatan apa berikutnya setelah suatu kegiatan
6		<i>Fork</i>	Percabangan yang menunjukkan aliran pada <i>Activity Diagram</i>
7		<i>Join</i>	Penggabungan yang menjadi arah aliran pada <i>Activity Diagram</i>
8		<i>Decision</i>	Pilihan untuk mengambil keputusan
9		<i>Flow Final</i>	Aliran Akhir

2.7.6. *Class Diagram*

Class diagram membantu kita dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain (dalam *logical view*) dari suatu sistem.

Selama proses analisis, class diagram memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem, selama tahap desain *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. *Class diagram* juga merupakan fondasi untuk *component diagram* dan *deployment diagram*, dalam sebuah model mungkin terdapat beberapa diagram kelas dengan spesifikasi tersendiri.

Tabel 3. Simbol Class Diagram (Suhendar, 2002)

Simbol	Deskripsi
	Kelas pada struktur sistem
	Paket/ <i>package</i> merupakan sebuah bungkus dari satu atau lebih kelas (kumpulan kelas)
	Asosiasi merupakan hubungan antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
	Generalisasi merupakan hubungan generalisasi dan spesialisasi (umum khusus) antara dua kelas dimana fungsi yang satu adalah fungsi yang lebih umum dari fungsi yang lainnya
	Dependency merupakan hubungan antarkelas yang saling bergantung, membutuhkan satu sama lain.
	Agregasi merupakan hubungan antar kelas dimana satu kelas merupakan semua bagian dari kelas-kelas yang lain.

2.7.7. *Sequence Diagram*

Semua *Sequence diagram* lebih dimodelkan pada tingkat objek daripada tingkat kelas untuk memungkinkan skenario yang menggunakan lebih dari satu *instance* dari kelas yang sama dan bekerja pada tingkat fakta, data uji, dan contoh. *Sequence Diagram* menggunakan tiga elemen notasi mendasar: *object*, *message/stimuli*, and *object lifeline* (Wahono, 2003).