

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. 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. (Asbon Hendra;2012:157)

##### **II.1.1. Informasi**

Informasi merupakan data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. (Asbon Hendra;2012:167)

##### **II.1.2. Sistem Informasi**

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat *manajerial*, dan kegiatan strategi dari suatu orgnaisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. (Asbon Hendra;2012:169)

#### **II.2. Sistem Pakar**

Sistem pakar (*expert system*) merupakan solusi *Artificial Intelligence* (AI) bagi masalah pemrograman pintar (*intelligent*). Dengan kata lain, sistem pakar

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

### **II.2.1. Kelebihan Sistem Pakar**

Sistem pakar memiliki beberapa fitur menarik yang merupakan kelebihannya, seperti :

- a. Meningkatkan ketersediaan (*increased availability*). Kepakaran atau keahlian menjadi tersedia dalam sistem komputer.
- b. Mengurangi biaya (*reduced cost*). Biaya yang diperlukan untuk menyediakan keahlian per satu orang *user* menjadi berkurang.
- c. Mengurangi bahaya (*reduced danger*). Sistem pakar dapat digunakan di lingkungan yang mungkin berbahaya bagi manusia.
- d. Permanen (*permanence*).
- e. Keahlian multipel (*multiple expertise*). Pengetahuan dari beberapa pakar dapat dimuat ke dalam sistem dan bekerja secara simultan dan kontinyu menyelesaikan suatu masalah setiap saat.
- f. Meningkatkan kehandalan (*increased reliability*). Sistem pakar meningkatkan kepercayaan dengan memberikan hasil yang benar sebagai alternatif pendapat dari seorang pakar atau sebagai penengah jika terjadi konflik antara beberapa pakar.

- g. Penjelasan (*explanation*). Sistem pakar dapat menjelaskan detail proses penalaran yang dilakukan hingga mencapai suatu kesimpulan.
- h. Respon yang cepat (*fast response*). Respon yang cepat atau *real time* diperlukan pada beberapa aplikasi.
- i. Stabil, tidak emosional, dan memberikan respon yang lengkap setiap saat.
- j. Pembimbing pintar (*intelligent tutor*).
- k. Basis data cerdas (*intelligent database*). (Rika Rosnelly;2012:5)

### II.2.2. Elemen manusia pada Sistem Pakar

Sistem pakar tidak lepas dari elemen manusia yang terkait di dalamnya.

Personil yang terkait dengan sistem pakar ada 4, yaitu :

1. Pakar (*expert*)
2. Pembangun pengetahuan (*knowledge engineer*)
3. Pembangun sistem (*system engineer*)
4. Pemakai (*user*). (Rika Rosnelly;2012:9-10)

### II.2.3. Struktur Sistem Pakar

Komponen yang terdapat dalam struktur sistem pakar ini adalah :

1. *Knowledge Base* (Basis Pengetahuan)

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Komponen sistem pakar disusun atas dua elemen dasar, yaitu fakta dan aturan.

## 2. *Inference Engine* (Mesin Inferensi)

Mesin inferensi merupakan otak dari sebuah sistem pakar dan dikenal juga dengan sebutan *control structure* (struktur control) atau *rule interpreter* (dalam sistem pakar berbasis kaidah).

## 3. *Working Memory*

Berguna untuk menyimpan fakta yang dihasilkan oleh *inference engine* dengan penambahan parameter berupa derajat kepercayaan atau dapat juga dikatakan sebagai global database dari fakta yang digunakan oleh rule-rule yang ada.

## 4. *Explanation facility*

Menyediakan kebenaran dari solusi yang dihasilkan kepada user (*reasoning chain*).

## 5. *Knowledge acquisition facility*

Meliputi proses pengumpulan, pemindahan dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi ke program komputer, yang bertujuan untuk memperbaiki atau mengembangkan basis pengetahuan.

## 6. *User Interface*

Mekanisme untuk memberi kesempatan kepada user dan sistem pakar untuk berkomunikasi. (Rika Rosnelly;2012:13)

#### II.2.4. Karakteristik Sistem Pakar

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

- a. Kinerja sangat baik (*high performance*). Sistem harus mampu memberikan respon berupa saran dengan tingkat kualitas yang sama dengan seorang pakar atau melebihinya.
- b. Waktu respon yang baik (*adequate respon time*). Sistem juga harus mampu bekerja dalam waktu yang sama baiknya atau lebih cepat dibandingkan dengan seorang pakar dalam menghasilkan keputusan.
- c. Dapat diandalkan (*good reliability*). Sistem harus dapat diandalkan dan tidak mudah rusak.
- d. Dapat dipahami (*understandable*). Sistem harus mampu menjelaskan langkah-langkah penalaran yang dilakukannya seperti seorang pakar.
- e. Fleksibel (*flexibility*). Sistem harus menyediakan mekanisme untuk menambah, mengubah, dan menghapus pengetahuan. (Rika Rosnelly;2012:20)

#### II.3. Metode *Ant Colony*

Algoritma *Ant Colony System* (ACS), merupakan bagian dari algoritma *Ant Colony Optimization* (ACO). ACO adalah kumpulan algoritma-algoritma yang meniru koloni semut (*ant*) dalam mencari makanan melalui rute yang dilaluinya. Yang termasuk ke dalam ACO yaitu *Ant system*, *Ant-Q System* dan ACS (Stutzle. and Dorigo, 1999). Sesuai dengan namanya, algoritma ini

terinspirasi oleh tingkah laku koloni semut, terutama tingkah laku dalam mencari makanan. Salah satu pokok pikiran dari algoritma ini adalah komunikasi tidak langsung dari agen koloninya (yang pada algoritma ini disebut semut-semut buatan) berdasarkan jejak *pheromone* (*pheromone* juga digunakan oleh semut asli untuk berkomunikasi). Jejak *pheromone* buatan tersebut merupakan informasi dalam bentuk angka yang dimodifikasi oleh semut-semut (buatan) untuk menunjukkan jalan yang telah mereka lalui dalam memecahkan masalah tertentu.

Menurut Gould (2001) *pheromone* ialah bau yang diproduksi oleh seekor binatang yang mempengaruhi tingkah laku dari binatang lain. Cara kerja *pheromone* dapat dianalogikan dengan cara kerja dari hormon-hormon di dalam tubuh yang mengirimkan sinyal kimia khusus dari sekelompok sel ke lainnya, yang menyebabkan mereka melakukan kegiatan tertentu. *Pheromone* ditemukan di seluruh dunia kehidupan dan merupakan bentuk komunikasi binatang yang paling kuno. Algoritma ACO yang pertama disebut *ant system* (AS), yang telah diaplikasikan pada TSP. Dimulai dari AS, kemudian dikemukakan beberapa perbaikan dari algoritma dasar tersebut. Secara khusus, algoritma-algoritma hasil perbaikan tersebut telah dites ulang dengan TSP. Semua versi perbaikan dari AS ini memiliki kesamaan yaitu ditemukannya suatu eksploitasi yang lebih kuat dari solusi yang terbaik, yang penemuan eksploitasi tersebut digunakan untuk mengarahkan proses pencarian oleh semut. Semua algoritma tersebut memiliki perbedaan utama yaitu dalam beberapa aspek yang mengontrol pencarian. Pada ACO, *performance* terbaik dalam memecahkan TSP, berguna untuk memperbaiki solusi yang dihasilkan oleh pencarian semut-semut sebelumnya, yaitu dengan

menggunakan algoritma *Local Search* ( Jurnal Techno Science Vol 3 No 1; Rindra Yusianto dan Budi Setyo Utomo; 2009: 325)

*Ant Colony Optimization (ACO)* diadopsi dari perilaku koloni semut yang dikenal sebagai system semut. Secara alamiah koloni semut mampu menemukan rute terpendek dalam perjalanan dari sarang menuju ke sumber makanan dan kembali lagi, pada saat semut berjalan, semut meninggalkan sebuah informasi yang disebut *pheromone*, di tempat yang dilaluinya dan menandai rute tersebut. *Pheromone* digunakan sebagai komunikasi antar semut pada saat membangun rute ( Jurnal IJCCS Vol 7 No 1; Yuliyani Siyamting Tyas dan Widodo Prijodiprodjo; 2013: 56)

#### **II.4. Visual Basic 2010**

Visual Basic 2010 merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh *Microsoft*, yaitu Microsoft Visual Studio 2010. Sebagai produk lingkungan pengembangan terintegrasi atau IDE andalan yang dikeluarkan oleh *Microsoft*, visual studio 2010 menambahkan perbaikan-perbaikan fitur dan fitur baru yang lebih lengkap versi Visual Studio pendahulunya, yaitu Microsoft Visual Studio 2008. (Wahana komputer;2010:2)

#### **II.5. UML (*Unified Modelling Language*)**

UML merupakan metodologi kolaborasi antara metoda-metoda Booch, *Object Modelling Technique (OMT)* dan *Object Oriented Software Engineering (OOSE)* dan beberapa metoda lainnya, merupakan metodologi yang paling sering

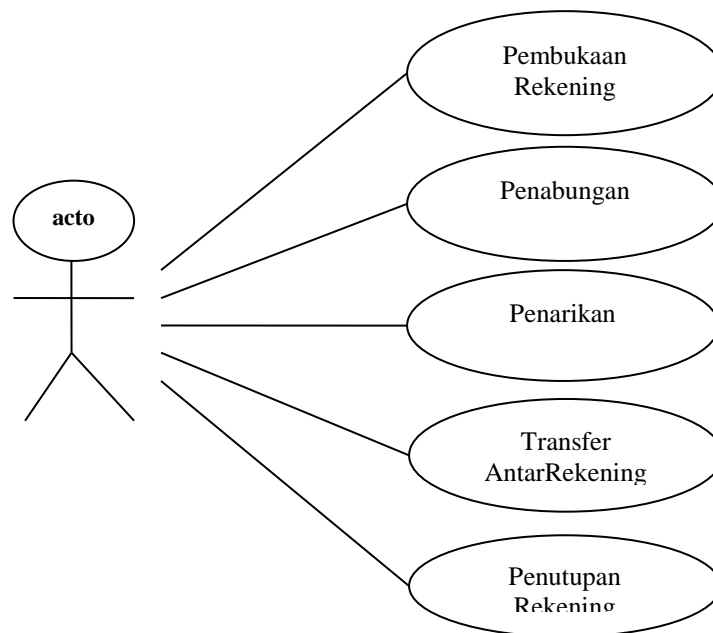
digunakan saat ini untuk mengadaptasi maraknya penggunaan bahasa “pemrograman berorientasi objek”. (Adi Nugroho;2009:4)

### **II.5.1. Komponen-Komponen UML**

#### *a. Use Case Diagram*

Dalam konteks UML, tahap konseptualisasi dilakukan dengan pembuatan *use case diagram* yang sesungguhnya merupakan deskripsi peringkat tinggi bagaimana perangkat lunak (aplikasi) akan digunakan oleh penggunanya. Selanjutnya, *use case diagram* tidak hanya sangat penting pada saat analisis, tetapi juga sangat penting dalam tahap perancangan (*design*), untuk mencari kelas-kelas yang terlibat dalam aplikasi, dan untuk melakukan pengujian (*testing*).

Saat akan mengembangkan *use case diagram*, hal yang pertama kali harus dilakukan adalah mengenali *actor* untuk sistem yang sedang dikembangkan. Dalam hal ini, ada beberapa karakteristik untuk para *actor*, yaitu *actor* yang ada diluar sistem yang sedang dikembangkan dan *actor* yang berinteraksi dengan sistem yang sedang dikembangkan. (Adi Nugroho;2009:7)

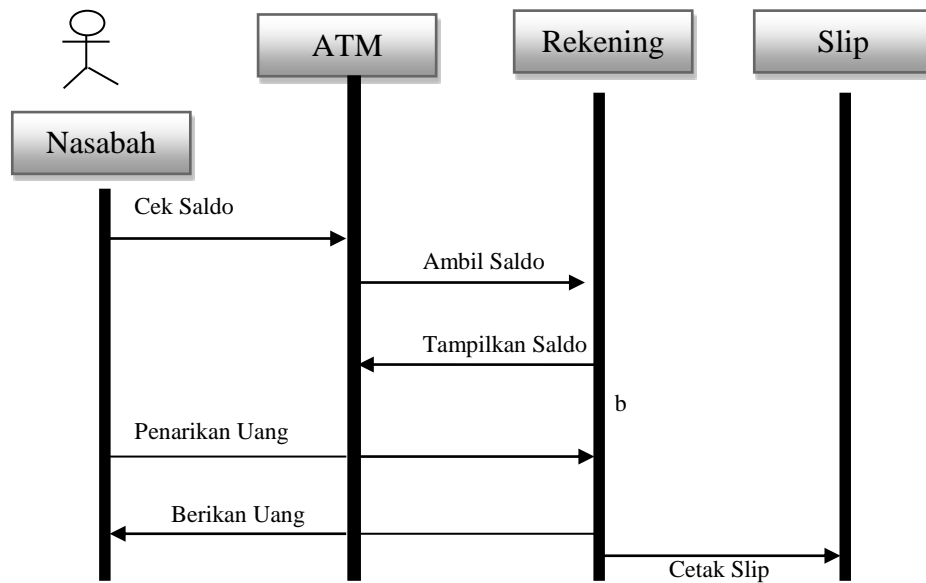


**Gambar II.1. Contoh Use Case Diagram**

(Sumber : Adi Nugroho;2009:8)

*b. Sequence Diagram*

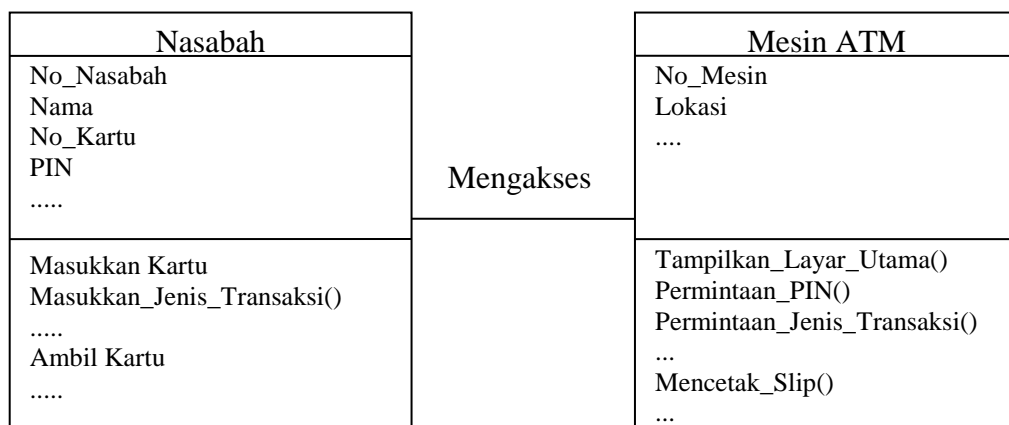
*Sequence diagram* juga menampilkan interaksi antar suatu kelas dengan kelas yang lainnya, bagaimana suatu *message* (pesan) dikirimkan dari suatu kelas ke kelas yang lainnya, dengan penekanan lebih pada urutan kejadian menurut waktu. Keunggulan dari *Sequence diagram* memperlihatkan dengan baik urutan interaksi yang terjadi antara suatu kelas dengan kelas lainnya, tetapi mengabaikan pengorganisasiannya. (Adi Nugroho;2009:101)



**Gambar II.2. Contoh Sequence Diagram**  
(Sumber : Adi Nugroho;2009:102)

*c. Class Diagram*

*Class diagram* menggambarkan struktur sistem dan segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem kelas memiliki apa yang disebut atribut dan metode atau operasi. (Rosa A.S dan M.Shalahuddin;2011:28)

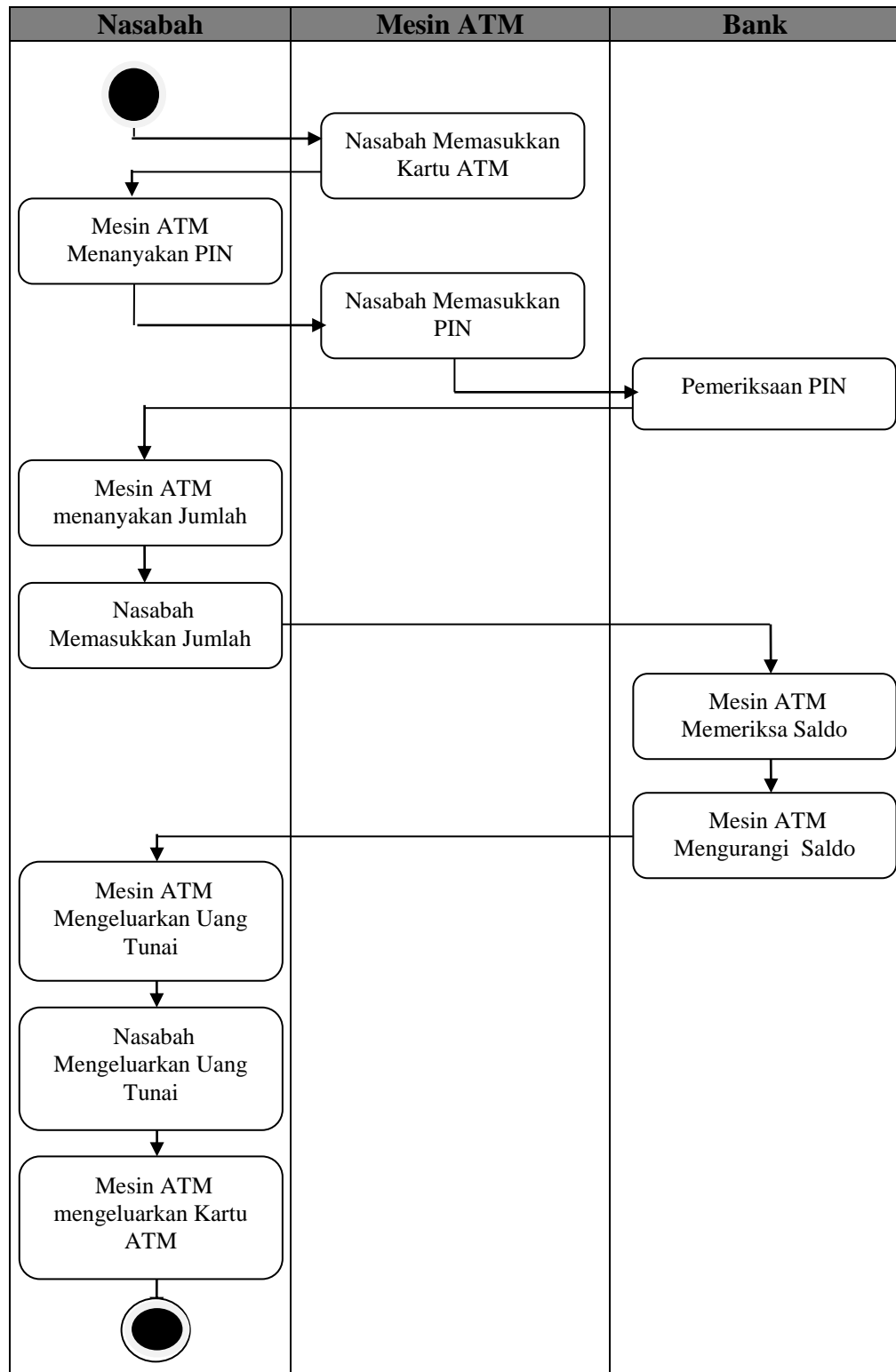


**Gambar II.3. Contoh Class Diagram**  
(Sumber : Adi Nugroho;2009: 39)

#### *d. Activity Diagram*

Keunggulan dari activity diagram adalah bahwa diagram tersebut lebih mudah dipahami dibandingkan skenario. Selain itu, dengan menggunakan *activity diagram*, kita juga bisa melihat di bagian manakah system dari suatu skenario akan berjalan. Misalnya, “Nasabah memasukkan kartu ATM” akan terjadi di bagian Mesin ATM, demikian juga dengan “Nasabah memasukkan PIN”. Selanjutnya, seperti halnya pada kasus penggunaan skenario, ada beberapa penyimpangan (*exception*) yang mungkin terjadi, yaitu (1) saat PIN yang dimasukkan Nasabah melebihi saldo yang dimilikinya. Dalam kasus sesungguhnya, penyimpangan tersebut juga perlu digambarkan dalam *activity diagram*.

Seperti pada gambar activity diagram dibawah ini, sesungguhnya setiap *state* (berbentuk empat persegi panjang) yang diperlihatkan dalam activity diagram menunjukkan hal yang serupa dengan apa yang tertulis dalam skenario. Hal itu memang benar. Dengan kata lain, *activity diagram* pada dasarnya menggambarkan skenario secara grafis. (Adi Nugroho;2009:13)



**Gambar II.4. Contoh Activity Diagram**

(Sumber : Adi Nugroho;2009 : 11)