

BAB II

TINJAUAN PUSTAKA

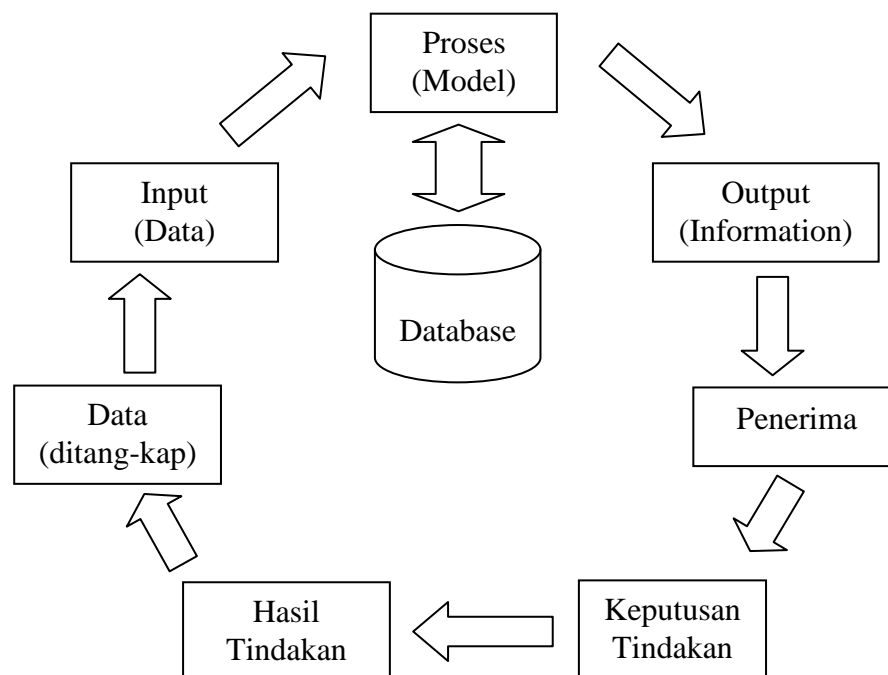
II.1. Pengertian Sistem

Menurut Riyanto, dkk (2009 : 22) Dengan berbagai pendekatan, beragam pula istilah “sistem” didefinisikan. Sistem adalah suatu pengorganisasian yang saling berinteraksi, saling bergantung dan terintegrasi dalam kesatuan variabel atau komponen. Terdapat dua kelompok pendekatan sistem, yaitu menekankan pada prosedur dan komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkelompok dan bekerjasama untuk melakukan kegiatan pencapaian sasaran tertentu. Makna dari prosedur sendiri, yaitu urutan yang tepat dari tahapan-tahapan instruksi. Sedangkan pendekatan yang menekankan pada komponen mendefinisikan sistem sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. “Serangkaian atau tatanan elemen-elemen yang diatur untuk mencapai tujuan yang ditentukan sebelumnya melalui pemrosesan informasi”.

II.2. Data Dan Informasi

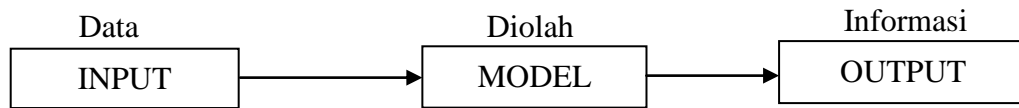
Menurut Riyanto, dkk. (2009 : 24), Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui model tertentu menjadi informasi yang dapat dimanfaatkan oleh penerima dalam membuat keputusan dan melakukan tindakan, yang berarti

melakukan suatu tindakan lain yang akan membuat sejumlah data kembali. Data yang masih belum diolah akan disimpan dalam bentuk *database*. Data yang disimpan ini nantinya dapat diambil kembali untuk diolah kembali menjadi informasi. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model tertentu dan seterusnya membentuk suatu siklus. Siklus ini oleh John Burch disebut dengan siklus informasi (*information cycle*).



Gambar II.1. Siklus Informasi
 Sumber : Riyanto, dkk. (2009 : 24)

Menurut Riyanto, dkk. (2009 : 23), Agar menjadi informasi yang berguna, data perlu diolah melalui sebuah siklus. Siklus ini disebut siklus pengolahan data (*data processing life cycle*).



Gambar II.2. Siklus Pengolahan Data
 Sumber : Riyanto, dkk. (2009 : 23)

II.3. Sistem Informasi

Menurut Tata Sutabri (2012:58) Tujuan sistem informasi adalah memenuhi kebutuhan informasi semua manajer dalam perusahaan atau dalam sub-unit organisasi perusahaan. Sistem informasi menyediakan informasi bagi pemakai dalam bentuk laporan dan output dari berbagai simulasi model matematika, dimana proses manajemen didefinisikan sebagai aktivitas-aktivitas :

1. Perencanaan, formulasi terinci untuk mencapai suatu tujuan akhir tertentu adalah aktivitas manajemen yang disebut perencanaan.
2. Pengendalian, perencanaan hanyalah setengah dari pertempuran.

II.4. Sistem Informasi Geografis

Menurut Riyanto, dkk. (2009 : 35-36), Sistem Informasi Geografis (SIG) adalah sistem informasi khusus yang mengelola data yang memiliki informasi spasial (bereferensi keruangan). Atau dalam arti sempit, adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis, misalnya data yang diidentifikasi menurut lokasinya, dalam sebuah *database*. Beberapa definisi dari SIG adalah sebagai berikut :

1. Aronoff menyatakan bahwa SIG sebagai suatu sistem berbasis komputer yang digunakan untuk menyimpan dan memanipulasi informasi-informasi geografis. SIG dirancang untuk mengumpulkan, menyimpan, dan menganalisis objek-objek dan fenomena dimana lokasi geografi merupakan karakteristik yang penting atau kritis untuk dianalisis.
2. Subaryono menyatakan bahwa SIG sebagai suatu himpunan terpadu dari *hardware, software, data, dan liveware* (orang-orang yang bertanggung jawab dalam mendesain, mengimplementasikan, dan menggunakan SIG).
3. ESRI (*Environmental System Research Institute*) menyatakan bahwa SIG adalah kumpulan yang terorganisir dari perangkat keras komputer, perangkat lunak, data geografis dan personil yang dirancang secara efisien untuk memperoleh, menyimpan, *update*, memanipulasi, menganalisis, dan menampilkan semua bentuk informasi yang bereferensi geografi.

II.5. ArcView

Menurut Eko Budiyanto (2010 : 177) Kemampuan *Arcview* GIS pada berbagai serinya tidaklah diragukan lagi. *Arcview* GIS adalah software yang dikeluarkan oleh ESRI (*Environmental Systems Research Institute*). Perangkat lunak ini memberikan fasilitas teknis yang berkaitan dengan pengolahan data spasial. Kemampuan grafis yang baik dan kemampuan teknis dalam pengolahan data spasial tersebut memberikan kekuatan secara nyata pada *Arcview* untuk

melakukan analisis spasial. Kekuatan analisis inilah yang pada akhirnya menjadikan *Arcview* banyak diterapkan dalam berbagai pekerjaan, seperti analisis pemasaran, perencanaan wilayah dan tata ruang, sistem informasi persis, pengendalian dampak lingkungan, bahkan untuk keperluan militer. Mengapa *Arcview* dapat memiliki keluwesan yang sedemikian hebat? Hal itu disebabkan oleh adanya dukungan dari skrip *Avenue*. Melalui *avenue* ini dapat dibentuk suatu “kemampuan baru” pada *Arcview*. Tentu saja hal ini membuat *Arcview* menjadi sangat luwes untuk diterapkan pada berbagai permasalahan spasial. *Avenue* dapat digunakan untuk “merombak” wajah *Arcview* sesuai kebutuhan penggunaanya.

Dialog designer diperlukan untuk membentuk antarmuka penampil data atribut yang menjadi dasar pemilihan objek. Untuk menghubungkan menu dan tombol dengan berbagai aksi yang diinginkan maka perlu dibentuk skrip atau program. Skrip atau program ini dibentuk menggunakan bahasa *Avenue*. Setiap aksi yang diperlukan diuraikan menjadi baris-baris perintah pada skrip *Avenue* dan selanjutnya dikaitkan ke masing-masing menu atau tombol yang bersangkutan.

II.6. Arsitektur GIS

Menurut Riyanto, dkk (2009 : 38) Beberapa Subsistem dalam sistem informasi geogafis antara lain adalah :

1. *Input*

Pada tahap *input* (pemasukan data) yang dilakukan adalah mengumpulkan dan mempersiapkan data spasial dan data atribut dari berbagai sumber

data. Data yang digunakan harus dikonversikan menjadi format *digital* yang sesuai. Proses konversi yang dilakukan dikenal dengan proses dijitalisasi (*digitizing*).

2. Manipulasi

Manipulasi data merupakan proses *editing* terhadap data yang telah masuk, hal ini dilakukan untuk menyesuaikan tipe dan jenis data agar sesuai dengan sistem yang akan dibuat, seperti : penyamaan skala, perubahan sistem, proyeksi, generalisasi dan sebagainya.

3. Manajemen data

Tahap ini meliputi seluruh aktifitas yang berhubungan dengan pengolahan data (menyimpan, mengorganisasi, mengelola, dan menganalisis data) ke dalam sistem penyimpanan permanen, seperti : sistem *file server* atau *database server* sesuai kebutuhan sistem. Jika menggunakan sistem *file server*, data disimpan dalam bentuk *file-file* seperti : *.txt, *.dat, dan lain-lain. Sedangkan jika menggunakan sistem *database server*, biasanya memanfaatkan *software Database Management System (DBMS)*, seperti : *MySQL, SQL Server, ORACLE*, dan *DBMS* sejenis lainnya.

4. Query

Suatu metode pencarian informasi untuk menjawab pertanyaan yang diajukan oleh pengguna SIG. Pada SIG dengan sistem *file server*, *query* dapat dimanfaatkan dengan bantuan *compiler* atau *interpreter* yang digunakan dalam mengembangkan sistem, sedangkan untuk SIG dengan sistem *database server*, dapat memanfaatkan *SQL (structured query*

language) yang terdapat pada *DBMS* yang digunakan. Penelusuran data menggunakan lebih dari satu *layer* dapat memberikan informasi untuk analisis data dan memperoleh data yang diinginkan.

5. Analisis

Terdapat dua jenis fungsi analisis dalam SIG, yaitu fungsi analisis spasial dan analisis atribut. Fungsi analisis spasial adalah operasi yang dilakukan pada data spasial. Sedangkan, Fungsi analisis atribut adalah fungsi pengolahan data atribut, yaitu data yang tidak berhubungan dengan ruang.

6. Visualisasi (*Data Output*)

Penyajian hasil berupa informasi baru atau *database* yang ada baik dalam bentuk *softcopy* maupun dalam bentuk *hardcopy* seperti dalam bentuk peta : peta (atribut peta dan atribut data), tabel, grafik, dan lain-lain.

II.7. MySQL

Menurut Riyanto, dkk (2009 : 306-308), Dengan *Database*, data atau informasi dapat disimpan secara permanen. Informasi yang tadinya ada didalam *variabel*, akan segera hilang bersamaan dengan selesainya skrip *PHP* yang dieksekusi. Untuk itu diperlukan *database* untuk menyimpan informasi yang ingin dipertahankan saat eksekusi selesai. Misalnya informasi nama, alamat, tanggal lahir, dan lain-lain.

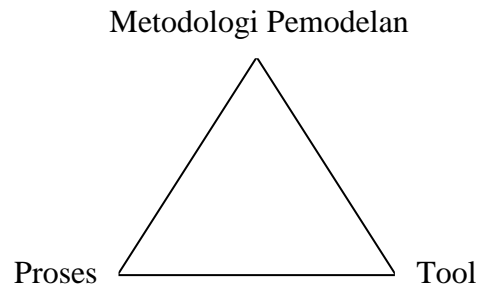
II.8. PHP

Menurut Budi Raharjo (2011 : 245) setelah anda merampungkan pembelajaran tentang MySQL pada bab-bab sebelumnya, pada bab 20 ini anda akan diajak untuk mengenal pembuatan aplikasi berbasis web yang menggunakan data dari database yang telah kita buat. Bahasa pemrograman yang akan kita gunakan untuk keperluan ini adalah PHP. Melalui bab ini, anda akan mengetahui cara mengakses data dari database kemudian menampilkannya ke dalam halaman web. Untuk dapat mengkawinkan database MySQL dengan aplikasi yang ditulis menggunakan PHP, terlebih dahulu kita perlu melakukan instalasi software-software berikut :

1. MySQL : software ini digunakan untuk menyampaikan database yang akan digunakan.
2. Apache: software ini digunakan sebagai web server aplikasi yang akan kita dibuat.
3. PHP : software ini digunakan untuk melakukan interpretasi dari kode PHP menjadi HTML sehingga hasilnya dapat ditampilkan di dalam web browser.

II.9. UML

Menurut Yuni Sugiarti (2013 : 34) *Unified Modelling Language (UML)* adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk *visualisasi*, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem.



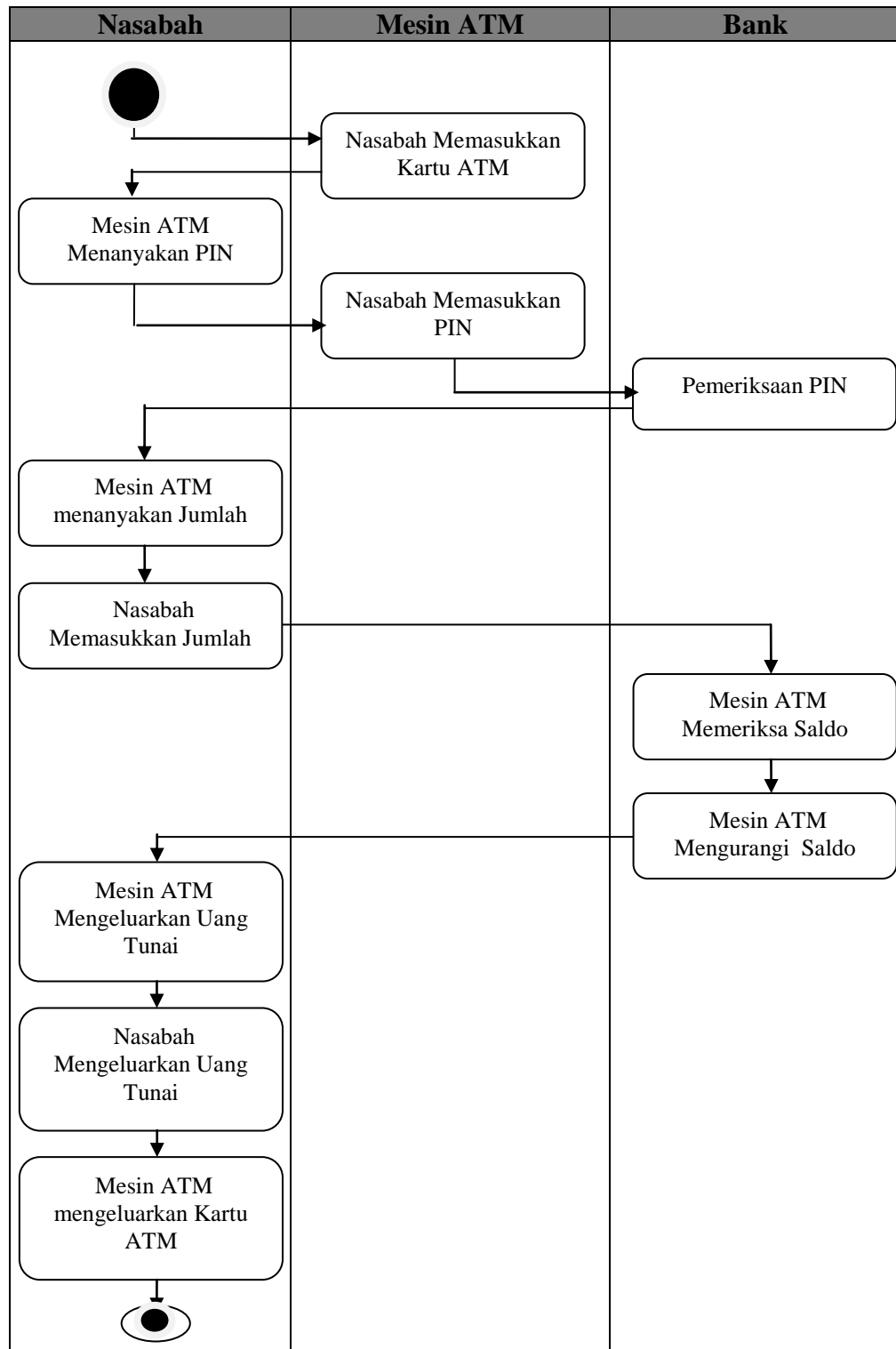
Gambar II.3. The Triangle For Success
Sumber : Yuni Sugiarti (2013 : 34)

Menurut Prabowo Pudjo Widodo dan Herlawati (2011 : 6-7), *UML* singkatan dari *Unified Modelling Language* yang berarti bahasa permodelan standar. UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

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

II.9.1. Activity Diagram

Menurut Adi Nugroho (2009 : 10) Apakah langkah yang harus kita lakukan selanjutnya setelah kita membuat use case diagram ? use case diagram merupakan gambaran menyeluruh dan pada umumnya sangatlah tidak terperinci. Apa perkakas (tool) yang bisa kita gunakan, jika kasus kita cukup sederhana, mungkin kita bisa menggunakan skenario seperti yang tercantum berikut, sementara jika kasusnya cukup kompleks, kita mungkin bisa menggunakan activity diagram agar bisa mendapatkan gambaran yang lebih menyeluruh.



Gambar II.4. Contoh Activity Diagram

Sumber : Adi Nugroho (2009 : 11)

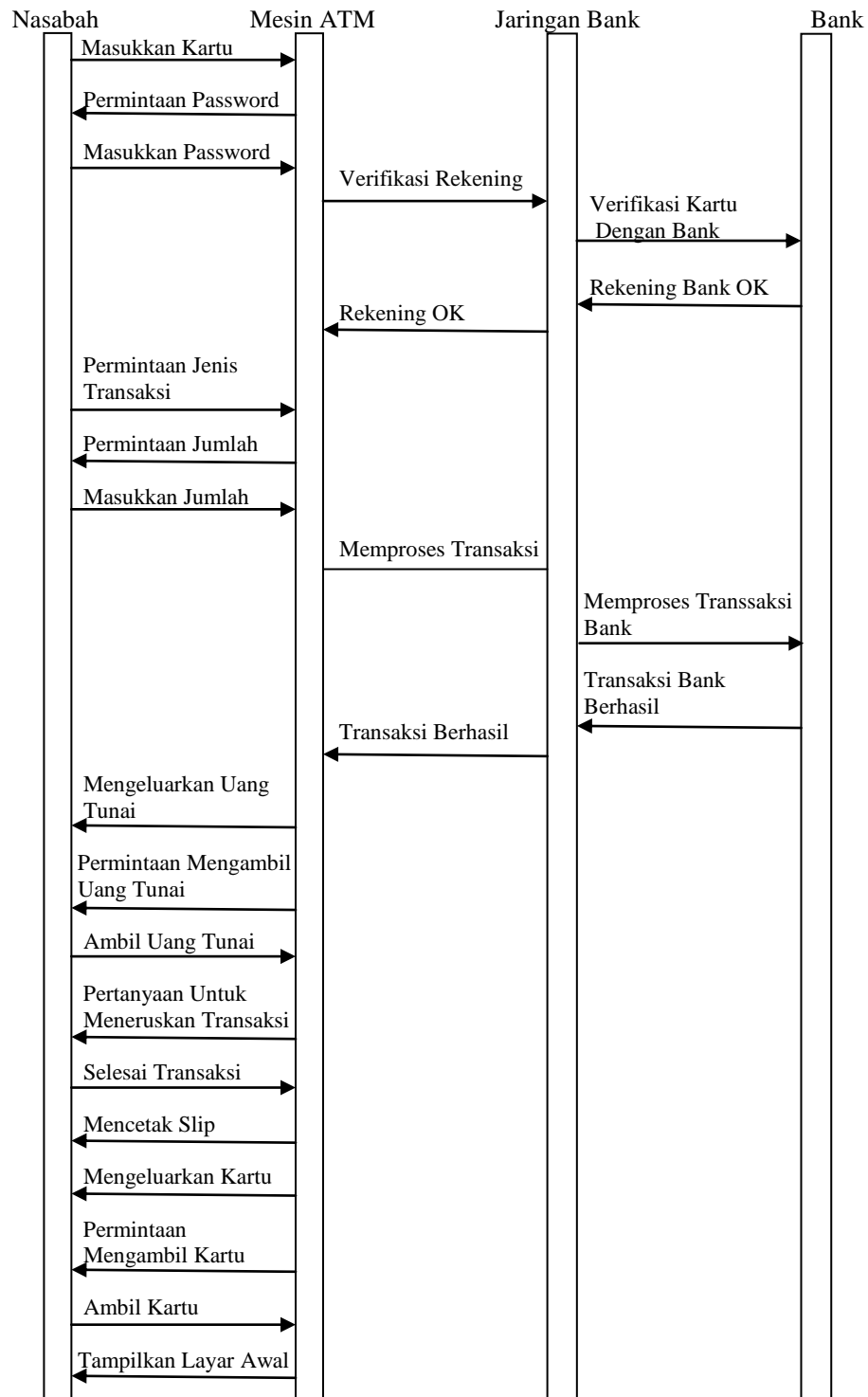
II.9.2. Sequence Diagram

Diagram sekuensial atau *sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Diagram sekuensial adalah diagram yang disusun berdasarkan urutan waktu. Kita membaca diagram sekuensial dari atas ke bawah. Setiap diagram sekuensial mempresentasikan suatu aliran dari beberapa aliran di dalam *use case*.

Jadi dengan kata lain sekuensial diagram menunjukkan aliran fungsionalitas berdasarkan urutan waktu serta kejadian yang nantinya akan menentukan metode/fungsi atribut masing-masing. Dimana fungsi-fungsi tersebut akan diterapkan pada suatu kelas/objek.

Perhatikan gambar II.5. dimana terlihat pengelompokkan *event-event* serta fungsi masing-masing atribut tersebut. Di dalam diagram terlihat jelas bagaimana aliran suatu proses kejadian dimana seorang nasabah yang akan melakukan transaksi dengan sebuah mesin ATM. Dari diagram tersebut kita mengetahui *event-event* yang terjadi, seperti : Nasabah memasukkan kartu ATM, Mesin ATM merespon dengan meminta *password* atau PIN, dan selanjutnya.

Kita dapat melihat setiap fungsi atribut dan *event-event* apa saja yang terjadi. Sehingga melalui diagram sekuensial ini kita dapat merancang suatu program aplikasi yang baik, sehingga dalam menghadapi sebuah kasus yang benar-benar kompleks diagram sekuensial ini sangat membantu.



Gambar II.5. Contoh Sequence Diagram

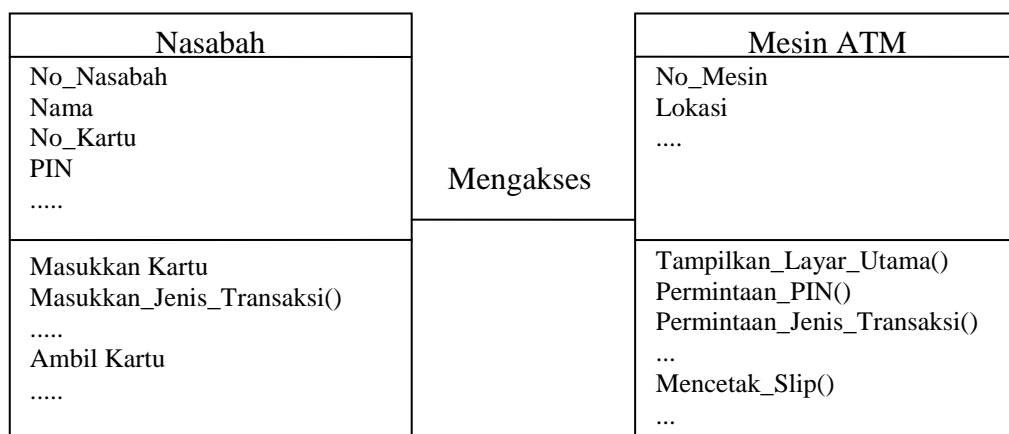
Sumber : Adi Nugroho (2009 : 36)

II.9.3. Class Diagram

Menurut Adi Nugroho (2009 : 18) *Class* didefinisikan sebagai kumpulan/himpunan objek yang memiliki kesamaan dalam atribut/properti, perilaku (operasi), serta cara berhubungan dengan objek lain.

Selain itu, kita juga mendefinisikan objek sebagai konsep, abstraksi dari sesuatu dengan batas nyata, sehingga kita dapat menggambarkan secara sistematis. Pemahaman objek memiliki dua fungsi, yaitu :

1. Memudahkan untuk mempelajari secara seksama hal-hal yang ada di dunia nyata.
2. Menyediakan suatu dasar yang kuat dalam implementasi ke dalam sistem terkomputerisasi.



Gambar II.6. Contoh Class Diagram

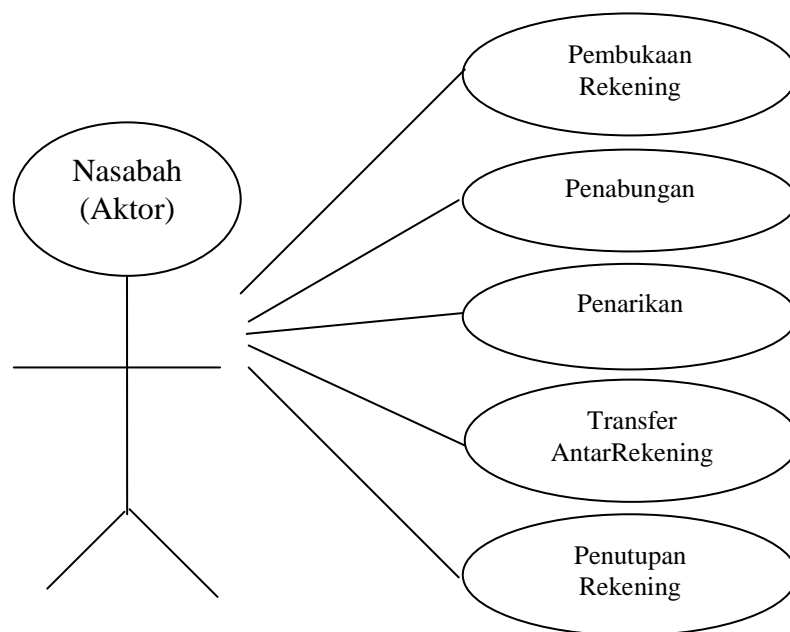
Sumber : Adi Nugroho (2009: 39)

II.9.4. Use Case Diagram

Menurut Adi Nugroho (2009 : 7) Segala sesuatu yang secara akademis dikembangkan pada umumnya berawal dari suatu konsep. Demikian juga halnya dengan pengembangan sistem pada umumnya dikembangkan berdasarkan analisis

kebutuhan. Analisis kebutuhan ini adalah tahap konseptualisasi, yaitu suatu tahap yang mengharuskan analis dan perancang sistem untuk berusaha tahu secara pasti mengenai hal yang menjadi kebutuhan dan harapan pengguna sehingga kelak aplikasi yang dibuat memang akan digunakan oleh pengguna (*user*) serta akan memuaskan kebutuhan dan harapannya.

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 di luar sistem yang sedang dikembangkan dan *actor* yang berinteraksi dengan sistem yang sedang dikembangkan.



Gambar II.7. Contoh Use Case Diagram

Sumber : Adi Nugroho (2009 : 8)

II.10. Database

Menurut Budi Raharjo (2011 : 245) dalam bekerja dengan *database*, anda perlu memiliki pengetahuan yang cukup tentang konsep *client/server* dan tipe-tipe arsitektur aplikasi *database* yang ada. Dengan demikian, anda akan mengetahui dengan pasti tujuan dan fungsi (untuk apa) anda membuat dan mengolah *database*.

II.11. Entity Relationship Diagram (ERD)

Menurut Budi Raharjo (2011 : 57) *entity relationship diagram (ERD)* merupakan salah satu alat bantu (berupa gambar) dalam *database relational* yang berguna untuk menjelaskan hubungan atau relasi antartabel yang terdapat di dalam *database*. Dalam *ERD* kita juga dapat melihat daftar kolom yang menyusun

masing-masing tabel. *ERD* inilah yang akan kita gunakan sebagai acuan untuk membahas materi dalam buku ini. Berikut keterangan tanda yang digunakan :

1. $1-\infty$ menunjukkan relasi 1-ke-banyak
2. $n-\infty$ menunjukkan relasi banyak-ke-banyak.