BAB II

LANDASAN TEORI

2.1 Algoritma Genetika Dan Rapid Application Development (RAD)

2.1.1 Algoritma Genetika

Algoritma ini ditemukan di Universitas Michigan, Amerika Serikat oleh John Holland (1975) melalui sebuah penelitian dan dipopulerkan oleh salah satu muridnya, David Goldberg (1989). Dimana mendefenisikan algoritma genetika ini sebagai metode algoritma pencarian berdasarkan pada mekanisme seleksi alam dan genetik alam.

Algoritma genetika adalah algoritma yang berusaha menerapkan pemahaman mengenai evolusi alamiah pada tugas-tugas pemecahan-masalah (*problem solving*). Pendekatan yang diambil oleh algoritma ini adalah dengan menggabungkan secara acak berbagai pilihan solusi terbaik di dalam suatu kumpulan untuk mendapatkan generasi solusi terbaik berikutnya yaitu pada suatu kondisi yang memaksimalkan kecocokannya atau lazim disebut *fitness*. Generasi ini akan merepresentasikan perbaikan-perbaikan pada populasi awalnya. Dengan melakukan proses ini secara berulang, algoritma ini diharapkan dapat mensimulasikan proses evolusioner.

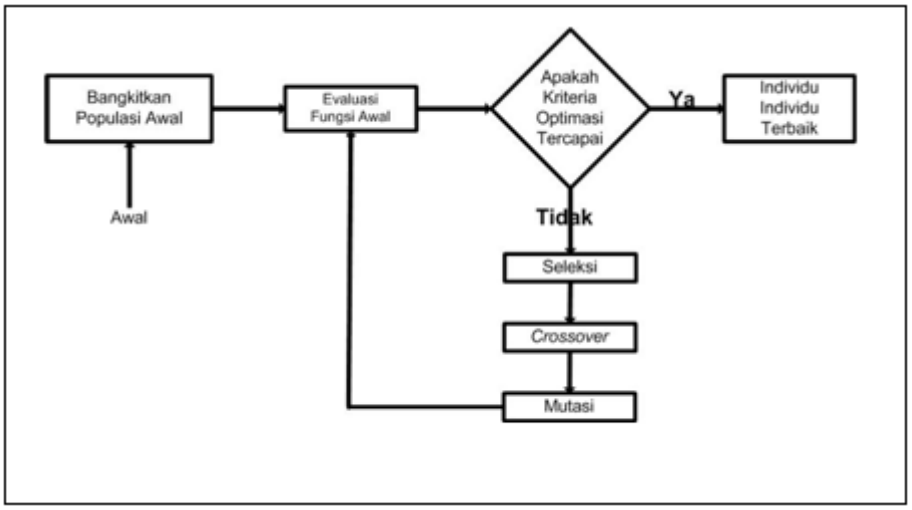
Pada akhirnya, akan didapatkan solusi-solusi yang paling tepat bagi permasalahan yang dihadapi. Untuk menggunakan algoritma genetik, solusi permasalahan direpresentasikan sebagai kromosom. Tiga aspek yang penting untuk enggunaan algoritma genetik:

1. Defenisi fungsi *fitness*
2. Defenisi dan implementasi representasi genetic
3. Defenisi dan implementasi operasi genetic

Jika ketiga aspek di atas telah didefinisikan, algoritma genetika akan bekerja dengan baik. Tentu saja, algoritma genetika bukanlah solusi terbaik untuk memecahkan segala masalah. Sebagai contoh, metode tradisional telah diatur untuk untuk mencari penyelesaian dari fungsi analitis *convex* yang "berperilaku baik" yang variabelnya sedikit. Pada kasus-kasus ini, metode berbasis kalkulus lebih unggul dari algoritma genetika karena metode ini dengan cepat menemukan solusi minimum ketika algoritma genetika masih menganalisa bobot dari populasi awal.

2.1.2 Struktur Umum Algoritma Genetika

Algoritma genetika memberikan suatu pilihan bagi penentuan nilai parameter dengan meniru cara reproduksi genetika, pembentukan kromosom baru serta seleksi alami seperti yang terjadi pada makhluk hidup. Algoritma Genetika secara umum dapat diilustrasikan dalam diagram alir berikut ini:



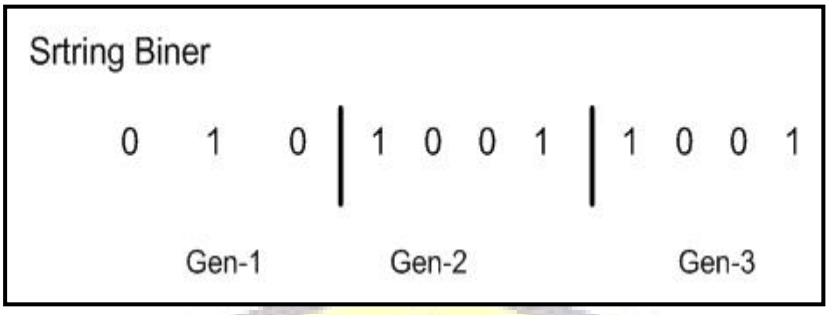
Gambar 2.1 Diagram Alir Algoritma Genetika (Kusumadewi, 2003)

(Kusumadewi, 2003) Pada algoritma ini, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi individu yang terdapat dalam satu populasi disebut dengan istilah **kromosom**, Charles L Karr (1999). Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasinya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan istilah **generasi**. Pada setiap generasi kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut **fungsi fitness**. Nilai fitness dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut.

2.1.3 Penyandian

Teknik penyandian disini meliputi penyandian gen dari kromosom. Gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu variable.

Gen dapat direpresentasikan dalam bentuk : *string bit, pohon, array* bilangan real, daftar aturan, elemen permutasi, elemen program, atau representasi lainnya yang dapat diimplementasikan untuk operator genetika.



Gambar 2.2 Penyandian Biner pada Operator Genetika

(Kusumadewi, 2003)

Demikian juga, kromosom dapat direpresentasikan dengan menggunakan :

- String bit : 011, 01101, 11101, dst.
- Bilangan Real : 65.65, -67.98. 562.88, dst.
- Elemen Program : pemrograman genetika
- Struktur lainnya

2.1.4 Operator Genetika

Algoritma genetik merupakan proses pencarian yang heuristik dan acak sehingga penekanan pemilihan operator yang digunakan sangat menentukan keberhasilan algoritma genetik dalam menemukan solusi optimum suatu masalah yang diberikan. Hal yang harus diperhatikan adalah menghindari terjadinya konvergensi *premature*, yaitu mencapai solusi optimum yang belum waktunya, dalam arti bahwa solusi yang diperoleh adalah hasil optimum lokal.

Operator genetika yang digunakan setelah proses evaluasi tahap pertama membentuk populasi baru dari generasi sekarang. Operator-operator tersebut adalah operator seleksi, *crossover* dan mutasi. (Kusumadewi, 2003). Berikut ini akan di jelaskan masing-masing operator pada Genetika.

1. Seleksi

Seleksi bertujuan memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling *fit*. Langkah pertama dalam seleksi ini adalah pencarian nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam wadah seleksi tersebut. Nilai *fitness* inilah yang nantinya akan digunakan pada tahap seleksi berikutnya (Kusumadewi, 2003).

Kemampuan algoritma genetik untuk memproduksi kromosom yang lebih baik secara progresif tergantung pada penekanan selektif (*selective pressure*) yang diterapkan ke populasi. Penekanan selektif dapat diterapkan dalam dua cara. Cara pertama adalah membuat lebih banyak kromosom anak yang dipelihara dalam populasi dan memilih hanya kromosom-kromosom terbaik bagi generasi berikut. Walaupun orang tua dipilih secara acak, metode ini akan terus menghasilkan kromosom yang lebih baik berhubungan dengan penekanan selektif yang diterapkan pada individu anak tersebut.

Cara lain menerapkan penekanan selektif adalah memilih orang tua yang lebih baik ketika membuat keturunan baru. Dengan metode ini, hanya kromosom sebanyak yang dipelihara dalam populasi yang perlu dibuat bagi generasi

berikutnya. Walaupun penekanan selektif tidak diterapkan ke level keturunan, metode ini akan terus menghasilkan kromosom yang lebih baik, karena adanya penekanan selektif yang diterapkan ke orangtua.

Ada beberapa metode untuk memilih kromosom yang sering digunakan antara lain adalah seleksi roda rolet (*roulette wheel selection*), seleksi ranking (*rank selection*) dam seleksi turnamen (*tournament selection*).

2. Crossover

Crossover (perkawinan silang) bertujuan menambah keanekaragaman string dalam populasi dengan penyilangan antar-string yang diperoleh dari sebelumnya. Beberapa jenis *crossover* tersebut adalah:

1. *Crossover* 1-titik

   Pada *crossover* dilakukan dengan memisahkan suatu string menjadi dua bagian dan selanjutnya salah satu bagian dipertukarkan dengan salah satu bagian dari string yang lain yang telah dipisahkan dengan cara yang sama. Proses yang demikian dinamakan operator *crossover* satu titik seperti diperlihatkan pada gambar berikut:

Tabel 2.1 Contoh Crossover 1 titik

Kromosom Orangtua 1	**11001**011
Kromosom Orangtua 2	11011**111**
Keturunan	**11001111**

2. Crossover 2 Titik

   Proses *crossover* ini dilakukan dengan memilih dua titik *crossover*. Kromosom keturunan kemudian dibentuk dengan barisan bit dari awal

kromosom sampai titik *crossover* pertama disalin dari orang tua pertama, bagian dari titik *crossover* pertama dan kedua disalin dari orang tua kedua, kemudian selebihnya disalin dari orang tua pertama lagi.

Tabel 2.2 Contoh Crossover 2 titik

Kromosom Orangtua 1	**11**001**11**
Kromosom Orangtua 2	11**011**11
Keturunan	**11011111**

3. Crossover Seragam

Crossover seragam manghasilkan kromosom keturunan dengan menyalin bit-bit secara acak dari kedua orangtuanya.

Table 2.3 Contoh Crossover Seragam

Kromosom Orangtua 1	**110**01**11**
Kromosom Orangtua 2	110**11**11
Keturunan	**11011111**

3. Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Operasi *crossover* yang dilakukan pada kromosom dengan tujuan untuk memperoleh kromosom-kromosom baru sebagai kandidat solusi pada generasi mendatang dengan *fitness* yang lebih baik, dan lama-kelamaan menuju solusi optimum yang diinginkan. Akan tetapi, untuk mencapai hal ini, penekanan selektif juga memegang peranan yang penting. Jika dalam proses pemilihan kromosom-kromosom cenderung pada kromosom yang

memiliki *fitness* yang tinggi saja, konvergensi *premature*, yaitu mencapai solusi yang optimal lokal sangat mudah terjadi.

Mutasi ini berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi.

- Mutasi bilangan real

  Pada mutasi bilangan real, ukuran langkah mutasi biasanya sangat sulit ditentukan. Ukuran yang kecil biasanya sering mengalami kesuksesan, namun adakalanya ukuran yang lebih besar akan berjalan lebih cepat.

- Mutasi bilangan real

  Cara sederhana untuk mendapatkan mutasi biner adalah dengan mengganti satu atau beberapa nilai gen dari kromosom.

2.1.5 Parameter Genetika

Pengoperasian algoritma genetik dibutuhkan 4 parameter (Juniawati, 2003) yaitu :

1. Probabilitas Persilangan (*Crossover Probability*)

   Menunjukkan kemungkinan *crossover* terjadi antara 2 kromosom. Jika tidak terjadi *crossover* maka keturunannya akan sama persis dengan kromosom orangtua, tetapi tidak berarti generasi yang baru akan sama persis dengan generasi yang lama. Jika probabilitas *crossover* 100% maka semua keturunannya dihasilkan dari *crossover*. *Crossover* dilakukan dengan harapan bahwa kromosom yang baru akan lebih baik.

2. Probabilitas Mutasi (*Mutation Probability*)

   Menunjukkan kemungkinan mutasi terjadi pada gen-gen yang menyusun sebuah kromosom. Jika tidak terjadi mutasi maka keturunan yang dihasilkan setelah *crossover* tidak berubah. Jika terjadi mutasi bagian kromosom akan berubah. Jika probabilitas 100%, semua kromosom dimutasi. Jika probabilitasnya 0%, tidak ada yang mengalami mutasi.

3. Jumlah Individu

   Menunjukkan jumlah kromosom yang terdapat dalam populasi (dalam satu generasi). Jika hanya sedikit kromosom dalam populasi maka algoritma genetik akan mempunyai sedikit variasi kemungkinan untuk melakukan *crossover* antara orangtua karena hanya sebagian kecil dari *search space* yang dipakai. Sebaliknya jika terlalu banyak maka algoritma genetik akan berjalan lambat.

4. Jumlah Populasi

   Menetukan jumlah populasi atau banyaknya generasi yang dihasilkan, digunakan sebagai batas akhir proses seleksi, persilangan dan mutasi.

2.1.6 Rapid Application Development (RAD)

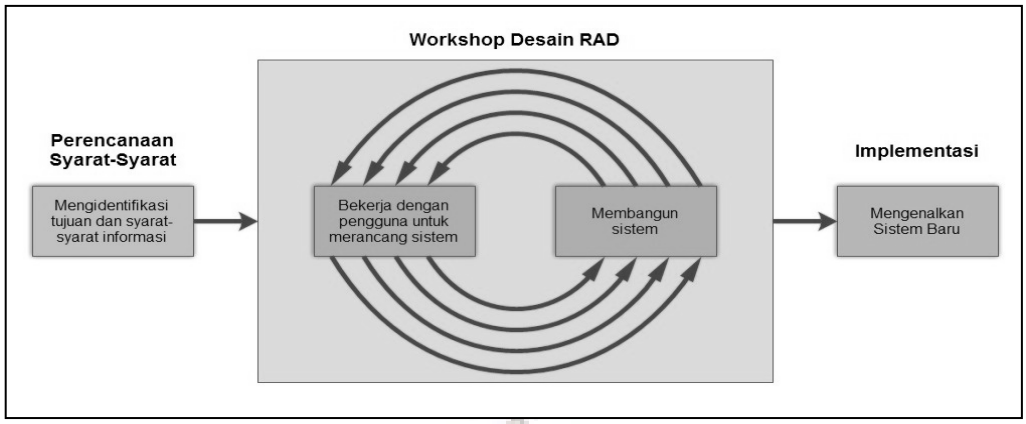
Rapid Application Development (RAD) adalah model pengembangan sistem yang melakukan beberapa penyesuaian terhadap *System Development Life Cycle (SDLC)* pada beberapa bagian sehingga lebih cepat untuk sampai ke tangan pengguna sistem. RAD adalah sebuah strategi pengembangan sistem yang menekankan kecepatan pengembangan melalui keterlibatan pengguna yang ekstensif dalam konstruksi, cepat, berulang dan bertambah ke serangkaian

prototype atau prototype yang bekerja ke sebuah sistem yang pada akhirnya berkembang kedalam sistem final.

Pemaparan konsep yang lebih spesifik lagi dijelaskan oleh Pressman (2005) dalam bukunya, *"Software Engineering: A Practition's Approach"*. Ia mengatakan bahwa RAD adalah proses model perangkat lunak inkremental yang menekankan siklus pengembangan yang singkat. Model RAD adalah sebuah adaptasi "kecepatan tinggi" dari model *waterfall*, di mana perkembangan pesat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen. Jika tiap-tiap kebutuhan dan batasan ruang lingkup projek telah diketahui dengan baik,

Proses RAD memungkinkan tim pengembang untuk menciptakan sebuah "sistem yang berfungsi penuh" dalam jangka waktu yang sangat singkat. Dari penjelasan Pressman (2012) ini, satu perhatian khusus mengenai metodologi RAD dapat diketahui, yakni implementasi metode RAD akan berjalan maksimal jika pengembang aplikasi telah merumuskan kebutuhan dan ruang lingkup pengembangan aplikasi dengan baik.

Sedangkan menurut Kendall (2010), RAD adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi. Pada akhirnya, RAD sama-sama berusaha memenuhi syarat-syarat bisnis yang berubah secara cepat.



Gambar 2.3 Siklus RAD (Kendall, 2010)

2.2 Proses Algoritma Genetika

2.2.1 Individu / Kromosom

Individu yang digunakan dalam proses pembuatan jadwal perkuliahan ini yaitu diambil dari kode mata kuliah dan kode dosen yang di jadikan gen dari data yang tela diinput, dan setiap gen mewakili mata kuliah tertentu dan dosen yang mengajar yang dinyatakan dalam suatu kode dalam bentuk integer.

2.2.2 Fitness

Fitness dari jadwal kuliah dilakukan dengan menghitung jumlah mata kuliah dan dosen yang mengajar dalam satu waktu. Terdapat tiga fungsi fitness yaitu fitness, fitnessInduk dan fitnessAnak. Dimana ketiganya mempunyai algoritma yang sama.

2.2.3 Membangkitkan Populasi Awal

Dari pendefinisian individu yang telah dilakukan diatas, selanjutnya proses pertama yang dilakukan adalah membangkitkan N individu sesuai dengan jumlah

individu (N) yang dimasukan oleh user dalam bentuk integer dari jadwal kuliah yang dibangkitkan secara acak.

2.2.4 Seleksi

Proses menyeleksi individu yang bisa dijadikan induk untuk proses selanjutnya dengan membangkitkan bilangan random sebanyak jumlah fitness dan memilih individu tersebut sebagai induk berdasarkan nilai fitness yang terpilih.

2.2.5 Crossover

Proses ini menyilangkan induk yang telah terseleksi dengan metode crossover langsung. Metodenya adalah mengacak 2 bilangan r1 dan r2 dengan r1 adalah batas awal dan r2 adalah batas akhir.

2.2.6 Mutasi

Proses mutasi dalam jadwal mencari 2 gen dalam satu record yang dilakuakn secara acak.kemudian menukar diantara kedua gen tersebut.

2.3 Tool Yang Digunakan

2.3.1 Microsoft Visual Basic 2010

Menurut (Budi Prayudi 2012) Visual Studio .NET adalah sebuah tools pengembangan perangkat lunak untuk membangun aplikasi ASP Web, layanan XML Web, aplikasi desktop, dan aplikasi mobile. Visual Basic .NET, Visual C++ .NET, Visual C# .NET, dan Visual J# .NET semuanya menggunakan Integrated Development Environment (IDE) atau lingkungan pengembangan terintegrasi yang sama, yang membolehkan mereka untuk saling berbagi tools dan fasilitas dalam pembuatan solusi yang memadukan beberapa bahasa (mixed-language solutions). Selain itu, bahasa-bahasa ini mempengaruhi fungsionalitas

dari .NET Framework, dan menyediakan pengaksesan ke kunci teknologi yang menyederhanakan proses pengembangan dari aplikasi ASP Web dan layanan XML Web. Ada banyak perubahan dalam VBNet 2010 ini dibandingkan VB6, antara lain:

1. Bahasa pemprograman sekarang benar-benar bahasa berbasis objek (Object Oriented Programming), sedangkan VB6 bukan bahasa berbasis objek.
2. Aplikasi dan komponen yang ditulis di VBNet 2010 mempunyai akses penuh ke Net Framework, sedangkan di VB6 tidak dikenal atau tidak digunakan Net Framework.
3. Semua aplikasi yang dibuat beroperasi dalam manajemen Common Language Runtime (CLR).

2.3.2 Sql Server

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft. Bahasa kueri utamanya adalah Transact-SQL yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan Sybase. Umumnya SQL Server digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya SQL Server pada basis data besar. (http://microsoft.com)

Menurut Wood, Leiter dan Turley (2007:1-5), SQL Server 2005 merupakan sebuah *Relational Database Management System (RDBMS)*. Pada SQL Server 2005 merupakan lanjutan atau pengembangan *feature-feature* dari produk SQL Server sebelumnya. Sebagai tamabahan SQL Server 2005 juga

mempunyai kemampuan pelaporan yang kaya, analisis data yang kuat, dan data mining.

Komponen utama dari SQL Server adalah engine basis data. Engine basis data ini merupakan Online Transaction Processing (OLTP) dari SQL Server. Engine basis data dari SQL Server terdapat beberapa pengembangan untuk mendukung scalability, availability, dan kemajuan (dan juga keamanan). Seperti :

- Partisi fisik tabel dan indeks
- Pemicu *Data Definition Language (DDL)*
- Lanjutan dari tipe data panjang variable
- Tipe data XML
- Multiple Active Result Set (MARS)
- Penanganan error terstruktur

Menurut Rizky Sotem dalam bukunya yang berjudul *Panduan Belajar SQL Sqerver 2005 Express Edition,* SQL Server memiliki keistimewaan, yaitu :

- Microsoft SQL Server lebih mudah digunakan dan memiliki lebih banyak fitur.
- Pemicunya antara lain adalah dukungan penuh dari Microsoft. Perangkat lunak yang ditawarkan oleh Microsoft juga menawarkan integrasi yang erat dengan .NET framework, dan ini tidak dimiliki oleh produk lain.
- Microsoft SQL Server memiliki sejumlah fitur dalam restorasi data dan pemulihan data.

2.4 Pemodelan UML

2.4.1 Pemodelan Sistem Dengan UML

Menurut Munawar (2005:17) UML (*Unified Modelling Language*) adalah Salah satu alat bantu yang sangat handal di dunia pengembangan system yang berorientasi objek. Hal ini di sebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat rancangan aplikasi yang mudah dimengerti.

Gambar 2.4 Pemodelan UML (Munawar, 2005)

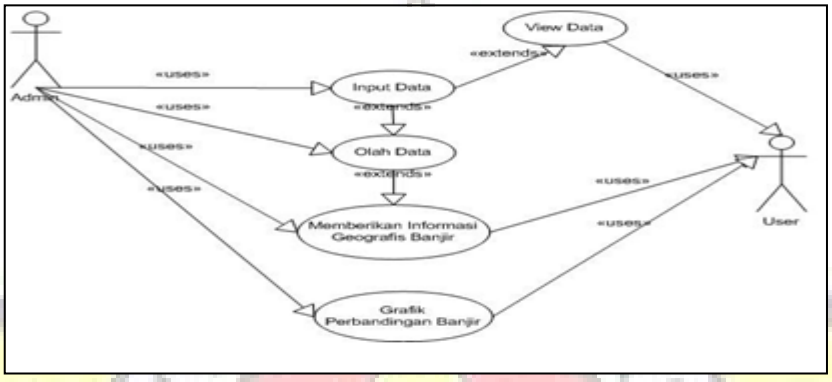
2.4.2 Use Case

Menurut Munawar (2005:64) *Use case* adalah deskripsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antar *user*(pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor

adalah sebuah *entitas* manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

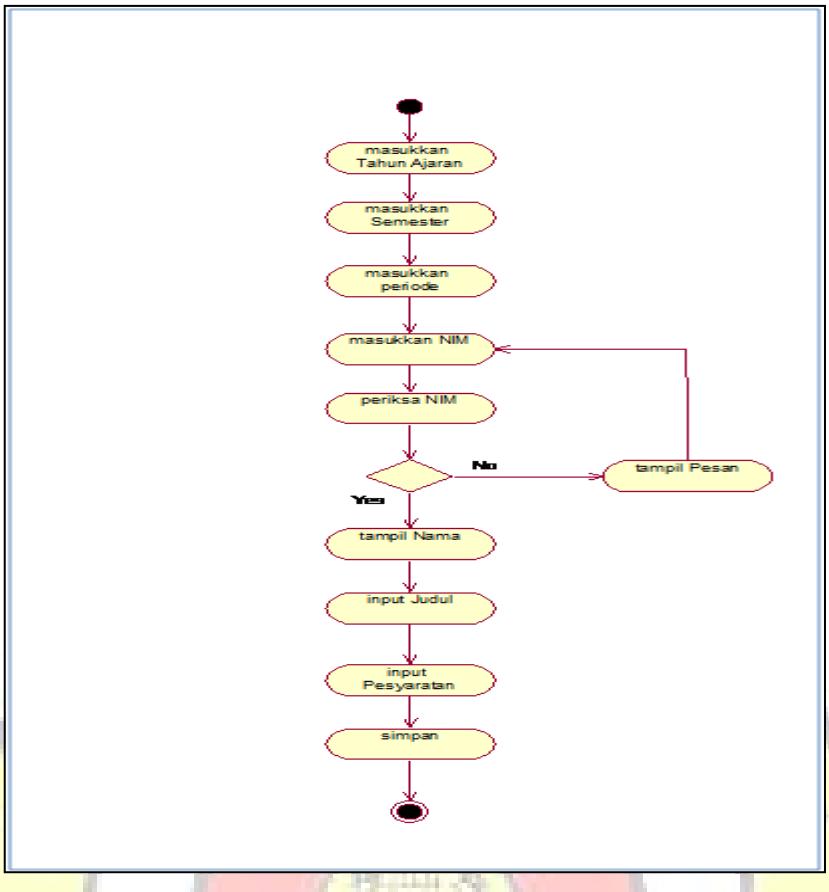
Use case diagram dapat sangat membantu apabila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan clien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.



Gambar 2.5 Use Case Diagram (Munawar, 2005)

2.4.3 Activity Diagram

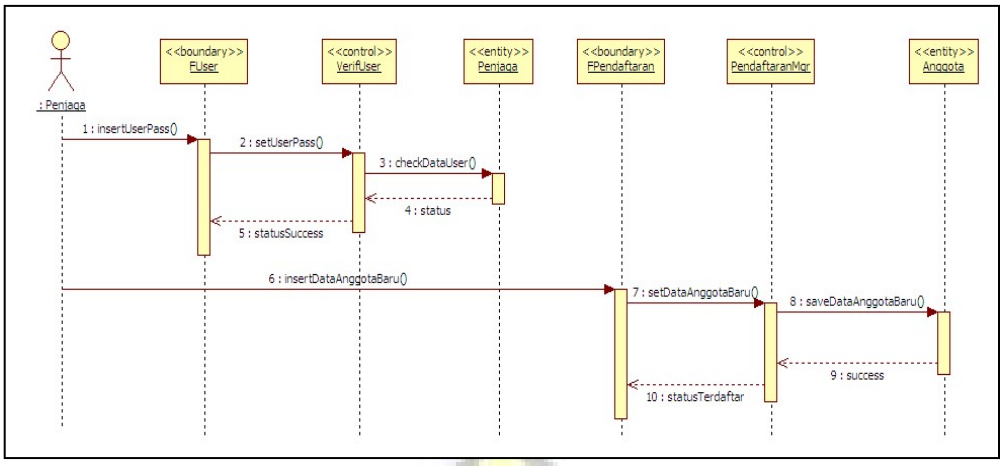
Menurut (Adi Nugroho 2005: 61) dalam buku Rational Rose Untuk Pemodelan Berorientasi Objek, *Activity diagram* adalah salah satu cara untuk memodelkan *event – event* yang terjadi dalam suatu *usecase*. Diagram ini juga bisa di gantikan dengan sejumlah teks. Namun, penggunaan teks terlalu sulit untuk dipahami, terutama jika aliran – aliran event yang memiliki alternatif sehingga *activity diagram* yang bersifat garis lebih mudah di mengerti. *Activity diagram* digunakan untuk memodelkan aspek dinamis dari sistem, yang memperlihatkan aliran kendali dari suatu aktifitas ke aktifitas lainnya.



Gambar 2.6 Activity Diagram (Adi Nugroho,2005)

2.4.4 Sequence Diagram

Menurut (Adi Nugroho 2005) dalam buku Rational Rose Untuk Pemodelan Berorientasi Objek, s*equence* diagram adalah iteraksi diagram yang memperlihatkan *event – event* yang berurutan sepanjang waktu menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.



Gambar 2.7 Sequence Diagram (Adi Nugroho, 2005)