

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terkait**

Dalam Penelitian yang dilakukan oleh Dwi Haryanti, Helfi Nasution, Anggi Srimurdianti Sukamto, 2016, “Sistem Pendukung Keputusan Seleksi Penerimaan Mahasiswa Pengganti Beasiswa Penuh Bidikmisi Universitas Tanjung pura Dengan Menerapkan Metode SMARTER”. Beasiswa Bidikmisi merupakan bantuan biaya pendidikan yang hanya ditujukan untuk calon mahasiswa tidak mampu dan berprestasi. Seleksi penerimaan mahasiswa pengganti beasiswa penuh Bidikmisi menggunakan banyak kriteria pertimbangan, sehingga diperlukan suatu sistem yang dapat menyeleksi penerimaan mahasiswa pengganti beasiswa penuh Bidikmisi. Salah satu sistem tersebut yaitu sistem pendukung keputusan. Salah satu metode yang dapat digunakan yaitu metode SMARTER. Penelitian ini bertujuan untuk menyeleksi penerimaan mahasiswa pengganti beasiswa penuh Bidikmisi Universitas Tanjungpura dengan menerapkan metode SMARTER dan mengetahui tingkat persentase keakurasian sistem pendukung keputusan dengan metode SMARTER dalam menyeleksi mahasiswa pengganti beasiswa penuh Bidikmisi Universitas Tanjungpura terhadap nilai sesungguhnya.

#### **II.2. Landasan Teori**

##### **II.2.1. Sistem**

Sistem adalah kumpulan dari objek-objek seperti manusia, *resources*, konsep, dan prosedur yang ditujukan untuk melakukan fungsi tertentu atau

memenuhi suatu tujuan. Sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi diantara unsur-unsur tersebut dengan lingkungan. Sedangkan menurut Jogiyanto sistem adalah komponen-komponen yang saling berhubungan untuk mencapai satu tujuan tertentu. Suatu sistem dapat dirumuskan sebagai setiap kumpulan bagian-bagian atau subsistem-subsistem yang disatukan, yang dirancang untuk mencapai suatu tujuan. (Ma'ruf, 2016 : 290).

### **II.2.2. Keputusan**

Kata keputusan (*decision*) berarti pilihan (*choice*), yaitu pilihan dari dua atau lebih kemungkinan. Keputusan juga dapat berarti kegiatan memilih suatu strategi atau tindakan dalam pemecahan suatu masalah. Tindakan memilih strategi atau aksi yang diyakini manajer akan memberikan solusi terbaik atas sesuatu itu disebut pengambilan keputusan. (Ma'ruf, 2016 : 290)

### **II.2.3. Sistem Pendukung Keputusan**

Pengertian Sistem Pendukung Keputusan (SPK) SPK atau *Decision Support System* (DSS) hanyalah sebatas pada kegiatan membantu para manajer melakukan pembuatan keputusan dan tidak melakukan penilaian serta menggantikan posisi dan peran manajer. SPK merupakan pengembangan lebih lanjut dari sistem informasi manajemen terkomputerisasi yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakainya. Interaktif dengan tujuan untuk memudahkan integrasi antara berbagai komponen dalam proses pengambilan keputusan seperti prosedur, kebijakan, analisis, pengalaman

dan wawasan manajer untuk mengambil keputusan yang lebih baik. (Priyolistiyanto, 2013 : 127).

#### **II.2.4. Proses Pengambilan Keputusan**

Pengambilan keputusan merupakan pilihan antara benar dan salah, tetapi yang sering terjadi adalah pilihan yang “hamper benar” dan yang “mungkin salah”. Pengambilan keputusan di dalam suatu organisasi merupakan hasil proses komunikasi dan partisipasi yang terus menerus dari keseluruhan organisasi. Hasil keputusan tersebut dapat merupakan pernyataan yang disetujui antar alternative atau antar prosedur untuk mencapai tujuan tertentu. (Ma’ruf, 2016 : 291).

Menurut Ma’ruf (2016), Proses pengambilan keputusan melibatkan 4 tahapan, yaitu:

1. *Intelligence*, tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah.
2. *Design*, tahap ini merupakan proses menemukan, mengembangkan, dan menganalisis *alternative* tindakan yang bias dilakukan. Tahap ini meliputi proses untuk mengerti masalah, menurunkan solusi, dan menguji kelayakan solusi.
3. *Choice*, pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Dari tahap ini didapatkan keluaran berupa dokumen solusi dan rencana implementasinya.
4. *Implementation*, dalam tahap ini pengambilan keputusan menjalankan rangkaian aksi pemecahan yang dipilih di tahap *choice*. Implementasi yang

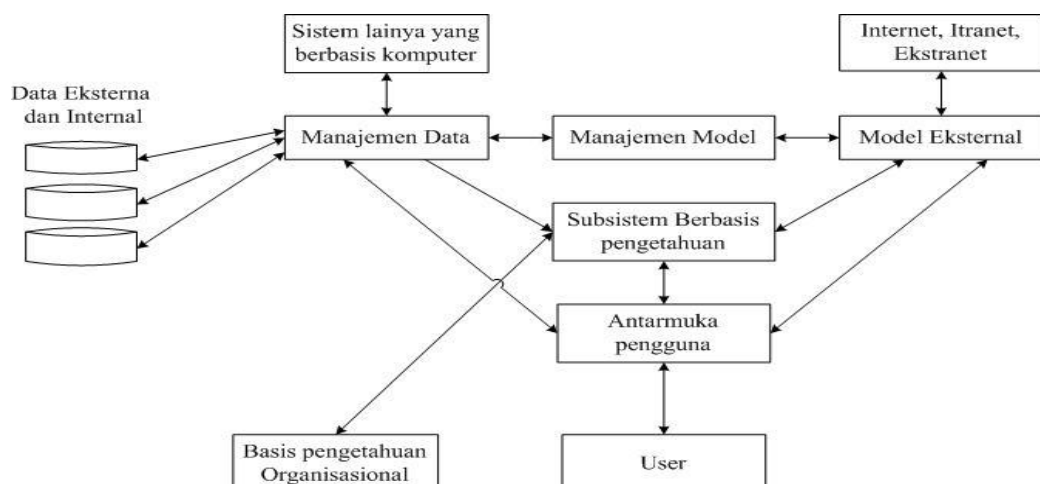
sukses ditandai dengan terjawabnya masalah yang dihadapi, sementara kegagalan ditandai dengan tahap adanya masalah yang sedang dicoba untuk diatasi.

## II.2.5. Komponen Sistem Pendukung Keputusan

Menurut Ma'ruf (2016 : 290), Komponen Sistem Pendukung Keputusan (SPK) terdiri dari:

1. Subsistem Manajemen Data (*Data-management Subsystem*).
2. Subsistem Manajemen Model (*Model Management Subsystem*).
3. Subsistem Antarmuka Pengguna (*User Interface Subsystem*).
4. Subsistem Manajemen Berbasis Pengetahuan (*Knowledge-based Management Sub system*).

Komponen ini dapat berperan untuk mendukung komponen yang lain, maupun berperan sebagai komponen yang independen. Komponen ini memperkaya kecerdasan pengguna dalam membuat keputusan. Komponen ini saling terhubung dengan basis data pengetahuan dari organisasi, yang biasa disebut dengan *Organizational Knowledge Base*.



**Gambar II.1. Komponen Sistem Pendukung Keputusan**  
(Sumber : Ma'ruf : 2016)

### II.3. Sepeda motor

Sepeda motor merupakan alat transportasi yang banyak diminati oleh masyarakat karena kepraktisan sepeda motor bila digunakan pada jalan raya yang padat, bisa dikendarai dengan cepat, dan harga yang relatif terjangkau bagi masyarakat dengan ekonomi menengah dan penggunaan bahan bakar yang lebih hemat dibandingkan dengan transportasi roda empat. (Agustian noor, 2016 : 7).

### II.4. Metode *SMARTER*

Metode pembobotan *Simple Multi Attribute Rating Technique Exploiting Ranks* (*SMARTER*) merupakan modifikasi yang diusulkan oleh Edwards dan Baron untuk metode pembobotan yang ditemukan sebelumnya, yaitu metode *SMART* (*Simple Multi Attribute Rating Technique*). Kedua metode ini digunakan untuk menentukan bobot dari setiap kriteria. Metode *SMARTER*, merupakan metode pembobotan yang menggunakan *range* antara 0 sampai 1, sehingga mempermudah perhitungan dan perbandingan nilai pada masing-masing alternatif. (Priyolistiyanto, 2013 : 129).

Pada metode *SMARTER*, bobot dihitung dengan menggunakan rumus pembobotan *Rank Order Centroid* (ROC). ROC didasarkan pada tingkat kepentingan atau prioritas dari kriteria. Teknik ROC memberikan bobot pada setiap kriteria sesuai dengan ranking yang dinilai berdasarkan tingkat prioritas. Misalkan dengan pernyataan dengan 2 atribut, A dan B, jika A ranking pertama, maka bobotnya harus berada diantara 0,5 dan 1 sehingga titik interval 0,75 diambil sebagai bobot perkiraan, yang merupakan dasar dari sebuah prinsip

komitmen minimum. Seperti bobot B akan menjadi 0,25 (merupakan titik tengah antara 0 dan 0,5). (Rezky Septian Razak, dkk, 2016 : 299).

Berikut merupakan Rumus dari perhitungan Metode *SMARTER*.

$$W1 \geq W2 \geq W3 \dots \geq Wk \quad (1)$$

$$W1 = \frac{(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{k})}{k} \quad (2)$$

$$W2 = \frac{(0 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{k})}{k} \quad (3)$$

$$W3 = \frac{(0 + 0 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{k})}{k} \quad (4)$$

Secara umum, jika K adalah jumlah kriteria, maka bobot dari kriteria ke K adalah:

$$Wk = \frac{1}{K} \sum_{i=k}^K \frac{1}{i} \quad (5)$$

Keterangan:

w = Nilai pembobotan kriteria,

k = Jumlah kriteria

i = Nilai alternative

Selanjutnya adalah perhitungan nilai *Utility* dengan menggunakan Persamaan :

$$V = \sum_{i=1}^n w_i v_i \quad (6)$$

Keterangan:

W<sub>i</sub> = Bobot yang mempengaruhi dari dimensi ke-*i* terhadap nilai keseluruhan evaluasi.

$V_i$  = Objek evaluasi pada dimensi ke  $i$

$n$  = Jumlah dimensi nilai yang berbeda

Selanjutnya menghitung nilai akhir menggunakan Persamaan :

$$n_i = \sum_{j=1}^n w_j u_{ij} \quad (7)$$

Keterangan:

$w_j$  = Bobot dari kriteria ke  $j$

$u_{ij}$  = Nilai Utility kriteria ke  $j$  untuk industry ke  $i$

$n_i$  = Nilai Akhir industri

Dimana nilai *utility* dikalikan dengan nilai bobot kriteria. Hasil akhir yang akan menentukan pilihan alternative.

## II.5. Basis Data

Basis data adalah kumpulan file-file yang saling berelasi, relasi tersebut biasa ditunjukkan dengan kunci dari tiap file yang ada. Satu basis data menunjukkan kumpulan data yang dipakai dalam satu lingkup informasi. Dalam satu file terdapat *record-record* yang sejenis, sama besar, sama bentuk, merupakan satu kumpulan *entity* yang seragam. Satu *record* terdiri dari *field field* yang saling berhubungan untuk menunjukkan bahwa *field* tersebut dalam satu pengertian yang lengkap dan direkam dalam satu *record*. Suatu *system* manajemen basis data berisi satu koleksi data yang saling berelasi dan satu set program untuk mengakses data tersebut. Jadi sistem manajemen basis data dan set

program pengelola untuk menambah data, menghapus data, mengambil data dan membaca data. (Mhd.Bustanur Rahmad,dkk , 2014 :1333).

## II.6. Normalisasi

Normalisasi merupakan parameter digunakan untuk menghindari duplikasi terhadap tabel dalam basis data dan juga merupakan proses mendekomposisikan sebuah tabel yang masih memiliki beberapa anomali sehingga menghasilkan tabel yang lebih sederhana dan struktur yang bagus.

### 1. *First Normal Form (1 NF)*

Sudah tidak ada *repeating group* yaitu pengulangan yang terjadi pada beberapa *atribut* atau kolom dalam sebuah tabel, dan juga setiap atribut harus bernilai tunggal.

### 2. *Second Normal Form (2 NF)*

Untuk menjadikan tabel normal tingkat ke 2 maka sudah 1NF dan setiap *atribut* yang bukan *primary key* sepenuhnya secara *funksional* tergantung pada semua atribut pembentuk *primary key*.

### 3. *Third Normal Form (3 NF)*

Tabel sudah 2NF dan tidak memiliki *transitive dependencies*, *Transitive dependency* adalah ketika ada *atribut* yang secara tidak langsung tergantung pada *primary key*.

### 4. *Fourth Normal Form (4 NF)*

Relasi berada pada bentuk normal keempat apabila memenuhi syarat *Boyce-Codd Normal Form (BCNF)* dan tidak mempunyai *multivalued dependency*.

### 5. *Fifth Normal Form (5 NF)*

Tabel bentuk normal kelima sering disebut PJNF (*Projection Join Normal Form*), penyebutan PJNF karena untuk suatu relasi akan berbentuk normal kelima jika tabel tersebut dapat dipecah atau diproyeksikan menjadi beberapa tabel dan dari proyeksi-proyeksi itu dapat disusun kembali (*join*) menjadi tabel yang sama dengan keadaan semula.(Gandung Triyono, 2012 : 19).

### **II.7. Macromedia Dreamweaver**

*Dreamweaver* merupakan editor web yang handal. *Macromedia Dreamweaver* tentunya dilengkapi dengan kemampuan manajemen situs,yang memudahkan kita mengelola keseluruhan elemen yang ada dalam sebuah situs. Kita juga dapat melakukan evaluasi situs dengan melakukan pengecekan *broken link*, kompatibilitas browser, termasuk validasi tag-tag HTML dan CSS yang tidak sesuai dengan pedoman secara otomatis serta perkiraan waktu *download* pada sebuah halaman web. (Sri Lestanti, dkk, 2016 : 73)

### **II.8. MySQL**

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang di distribusikan secara gratis. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan

pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. ( Sri Lestanti, dkk, 2016 : 72)

## **II.9. *Unified Modeling Language (UML)***


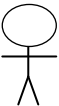


Menurut Windu Gata (2013) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Gellysa Urva,Dkk. 2015 : 93).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

### 1. *Use case* Diagram

*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini:

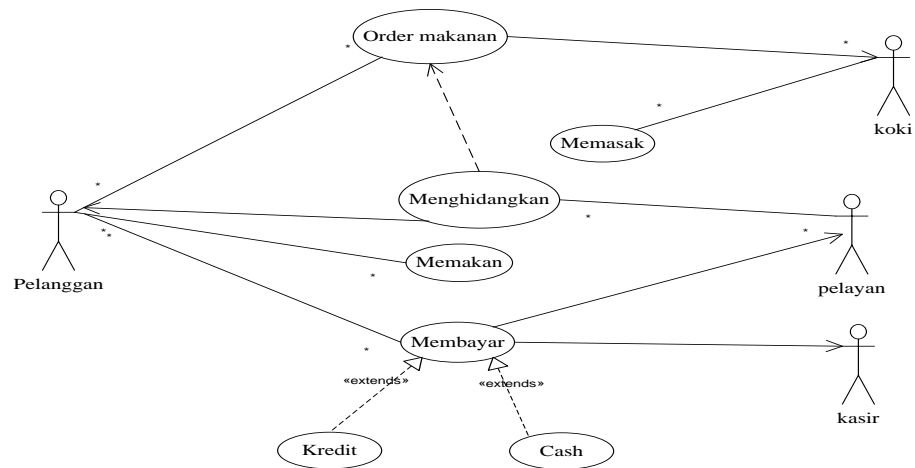
Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk</p>

Tabel II.1. Simbol *Use Case* ( lanjutan )

	mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
----->	<i>Include</i> , merupakan di dalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
<-----	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

Contoh *Use Case*:




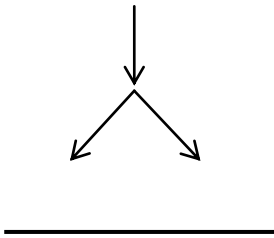
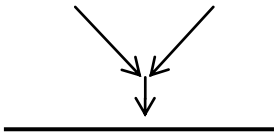
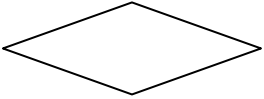
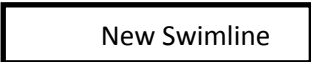


(Sumber : Gellysa Urva dkk; 2015, Hal : 94)

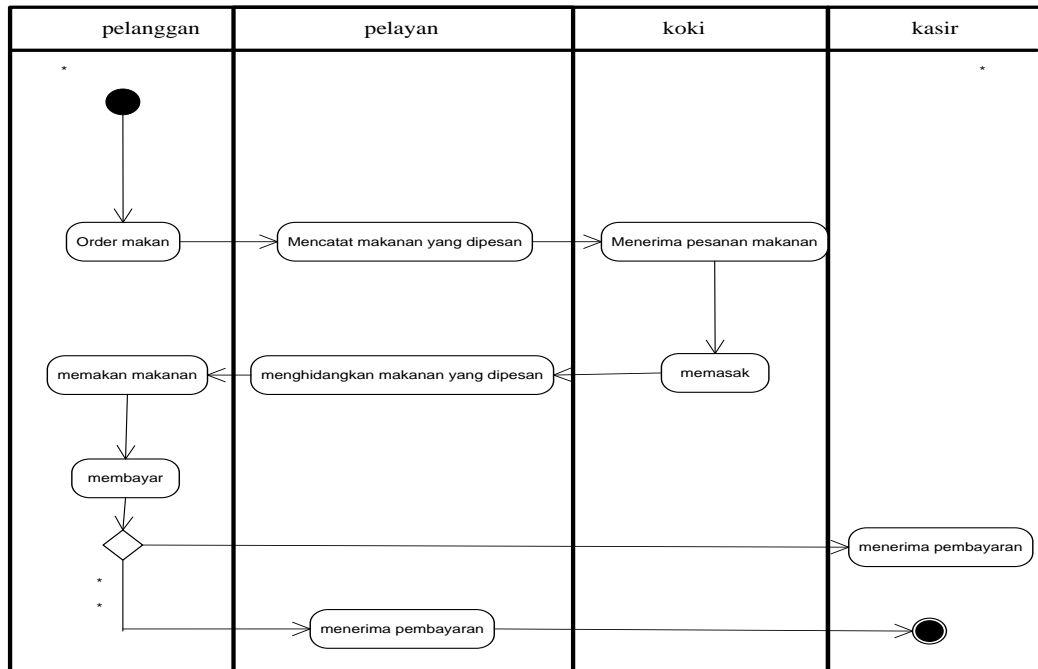
## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

**Tabel II.2. Simbol *Activity Diagram***

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true, false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

Contoh *Activity Diagram*:

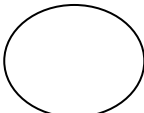


(Sumber : Gellysa Urva dkk; 2015, Hal : 94)

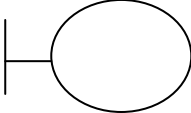
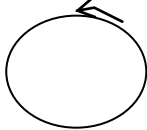

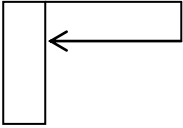


### 3. Diagram Urutan (*Sequence Diagram*)

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

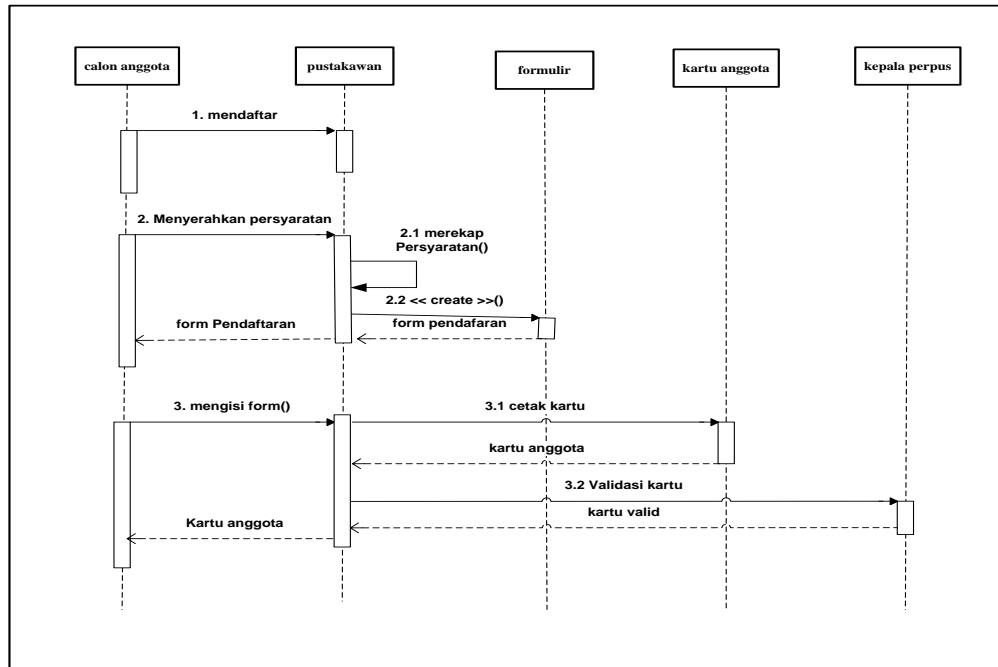
**Tabel II.3. Simbol *Sequence Diagram***

Gambar	Keterangan
	<p><i>EntityClass</i>, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.</p>

Tabel II.3. Simbol *Sequence Diagram* (lanjutan)

	<p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.</p>
	<p><i>Control class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.</p>
	<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>
	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Activation</i>, <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

Contoh *Sequence Diagram*:



(Sumber : Gellysa Urva dkk; 2015, Hal : 95)

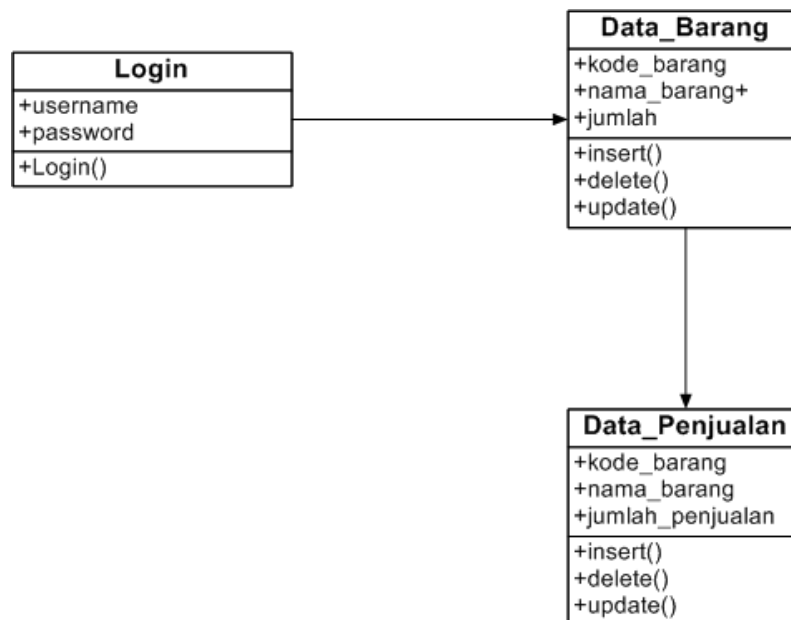
#### 4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.4 dibawah ini :

Tabel II.4. *Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

contoh:



(Sumber : Gellysa Urva dkk; 2015 : 95)