

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem**

Sistem merupakan suatu kumpulan dari komponen-komponen yang membentuk satu kesatuan. Sebuah organisasi dan sistem informasi adalah sistem fisik dan sosial yang ditata sedemikian rupa untuk mencapai tujuan tertentu. Seorang manajer yang mengorganisasikan pekerjaan unitnya adalah orang yang menciptakan sesuatu sistem pengembangan personal *computer* (PC) menciptakan suatu sistem yang terdiri dari perangkat keras dan lunak PC ini menjadi subsistem dari suatu perangkat konferensi elektronik. Sistem fisik dan sosial adalah sistem yang abstrak (*abstract system*) dari konsep dan ide. Contoh sistem yang abstrak ialah pengembangan daur hidup atau pengembangan sistem perangkat lunak (Tyoso ; 2016 : 1).

#### **II.2. Informasi**

Pengertian informasi masih bersifat kontradiktif dan belum mempunyai ciri-ciri khusus. Maka para pakar teori atau ilmu pengetahuan informasi belum dapat membuat satu definisi yang jelas dan lengkap mengenai apa yang disebut informasi.

Menurut para pakar informasi, informasi meliputi aspek abstrak dan khusus. Informasi dipandang sebagai ilmu pengetahuan tertulis atau yang disampaikan secara lisan dan sebagai hasil dari data yang diolah, biasanya diolah

secara formal. Ada pula yang menyebut informasi sebagai sarana pengurang ketidakpastian. Pengertian ini sama dengan arti informasi dalam istilah ekonomi, yakni informasi adalah penghalau ketidakpastian, misalnya di dalam pasar, *consumer preference*, dan harga (Tyoso ; 2016 : 31).

### **II.3. Akuntansi**

Akuntansi merupakan bahasa bisnis. Sebagai bahasa bisnis akuntansi menyediakan cara untuk menyajikan dan meringkas kejadian-kejadian bisnis dalam bentuk informasi keuangan kepada pemakainya. Informasi akuntansi merupakan bagian terpenting dari seluruh informasi yang diperlukan oleh manajemen. Informasi akuntansi yang dihasilkan oleh suatu sistem dibedakan menjadi dua, yaitu informasi akuntansi keuangan dan informasi akuntansi manajemen.

Pemakai informasi akuntansi pun terdiri dari dua kelompok, yaitu pemakai eksternal dan pemakai internal. Yang dimaksud dengan pemakai eksternal mencakup pemegang saham, investor, kreditor, pemerintah, pelanggan, pemasok, pesaing, serikat kerja dan masyarakat. Sedangkan pemakai internal adalah pihak manajer dari berbagai tingkatan dalam organisasi bersangkutan (Kusrini ; 2012 : 1).

### **II.4. Sistem Informasi Akuntansi**

Sistem informasi akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan

dengan saksi keuangan. Lingkup sistem informasi akuntansi dapat dijelaskan dari manfaat yang didapat dari informasi akuntansi. Manfaat atau tujuan sistem informasi akuntansi tersebut adalah sebagai berikut :

1. Mengamankan harta / kekayaan sekolah. Harta / kekayaan di sini meliputi kas sekolah, persediaan barang dagangan, termasuk aset tetap sekolah.
2. Menghasilkan beragam informasi untuk pengambilan keputusan. misal, pengelola toko swalayan memerlukan informasi mengenai barang apa saja yang diminati oleh konsumen. Membeli barang yang kurang laku berarti kas akan terjebak dalam persediaan dan berarti kehilangan kesempatan untuk membeli barang dagangan yang laku.
3. Menghasilkan informasi untuk pihak eksternal. Setiap pengelola usaha memiliki kewajiban untuk membayar pajak. Besarnya pajak yang dibayar tergantung pada omset penjualan (jika pengelola memilih menggunakan norma dalam perhitungan pajaknya) atau tergantung pada laba rugi usaha (jika pengelola memilih untuk tidak menggunakan norma dalam perhitungan pajaknya).
4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi. Sistem informasi dapat juga dimanfaatkan untuk penilaian kinerja karyawan atau divisi.
5. Menyediakan data masa lalu untuk kepentingan audit (pemeriksaan). Data yang tersimpan dengan baik sangat memudahkan proses audit (pemeriksaan).

6. Menghasilkan informasi untuk penyusunan dan evaluasi anggaran sekolah. Anggaran merupakan alat yang sering digunakan sekolah untuk mengendalikan pengeluaran kas.
7. Menghasilkan informasi yang diperlukan dalam kegiatan perencanaan dan pengendalian. Selain berguna untuk membandingkan informasi yang berkaitan dengan anggaran dan biaya standar dengan kenyataan seperti yang telah dikemukakan (Anastasia Diana ; 2011 : 6).

## **II.5. Penyusutan**

Seiring dengan waktu pemakaian sebuah aset tetap, maka pada saat yang sama aset tetap tersebut akan mulai berkurang kemampuannya atau mulai mengalami keusangan (*obsolescence*) untuk menciptakan barang dan jasa. Berkurangnya kemampuan aset tetap ini disebut sebagai penyusutan atau depresiasi (*depreciation*). Jumlah yang dapat disusutkan dialokasikan ke setiap periode akuntansi selama masa manfaat aktiva dengan berbagai metode yang sistematis. Metode manapun yang dipilih, konsistensi dalam penggunaannya adalah perlu, tanpa memandang tingkat profitabilitas perusahaan dan pertimbangan perpajakan, agar dapat menyediakan daya banding hasil operasi perusahaan dari periode ke periode. Aktiva tetap berwujud dapat disusutkan dalam beberapa metode, beberapa jenis metode penyusutan atas aset tetap menurut PSAK 16 yang dapat diterapkan di Indonesia adalah metode penyusutan garis lurus (*straight line method*), saldo menurun ganda (*double declining balance method*), dan metode unit produksi (*units of production method*). Serta tambahan metode penyusutan

lainnya yaitu penyusutan berdasarkan jumlah angka tahun (*sum of the years digits method*) (Samuel Mairuhu ; 2014 : 404).

## II.6. Metode Penyusutan

Menurut Winarto (2012) Metode Penyusutan atau Depresiasi yang digunakan untuk menentukan besarnya penyusutan aktiva tetap dalam tiap periode, maka diperlukan perhitungan. Besarnya perhitungan penyusutan aktiva tetap akan berbeda-beda, tergantung pendekatan mana yang diambil perusahaan untuk menentukan besarnya penyusutan. Penentuan besarnya penyusutan setiap periode akuntansi untuk berbagai jenis aktiva tetap dapat dilakukan dengan beberapa metode antara lain :

- a. Metode garis lurus (*Straight Line Method*)
- b. Metode saldo menurun (Depresiasi), yang meliputi :
  - Metode jumlah angka tahun (*Sum of The Years Digits Method*)
  - Metode Tarif Tetap atas nilai buku / Metode saldo menurun ganda (*Double Declining Balance Method*)
- c. Metode satuan produksi (*Unit of Production Method*)
  - Metode Hasil Produksi (*Productive Output Method*)
  - Metode jam kerja (*Service Hours Method*)
- d. Metode tarif kelompok / gabungan (*Composite Rate Depreciation Method*)

Penyusutan dengan saldo menurun ganda (*double declining balance method*) merupakan suatu metode dipercepat dimana untuk mengalokasikan harga pokok / harga perolehan aktiva tetap menjadi beban penyusutan untuk setiap

tahunnya semakin kecil atau menurun yang menggunakan tarif penyusutan dua kali dari yang digunakan metode garis lurus.

### **II.6.1. Metode Jumlah Angka Tahun**

Metode Jumlah Angka Tahun adalah metode ini menghitung beban penyusutan dengan cara membagi biaya perolehan dikurangi nilai residunya dengan jumlah keseluruhan dari umur manfaat aset tetap tersebut (Samuel Mairuhu ; 2014 : 407).

Dengan persamaan sebagai berikut :

$$\begin{aligned} \text{Dasar Penyusutan} &= \text{Tahun 1} + \text{Tahun 2} + \text{Tahun 3} + \text{Tahun 4} + \text{Tahun 5} \\ &= 15 \end{aligned}$$

### **II.6.2. Metode Menurun Ganda**

Penyusutan dengan saldo menurun ganda merupakan aset tetap yang dipercepat, dimana dasar penggunaan tarif garis lurus dikalikan dengan 2 (dua) dipakai untuk menentukan tarif penyusutan (Samuel Mairuhu ; 2014 : 407).

Dengan persamaan sebagai berikut :

Rumus tarif garis lurus :

$$\text{Beban Penyusutan} = \frac{\text{Biaya Perolehan}-\text{Nilai residu}}{\text{Umur Manfaat}}$$

Rumus menurun ganda :

$$\text{Tarif Penyusutan} = \text{Tarif Garis Lurus} \times 2$$

## **II.7. Aktiva Tetap**

Aktiva tetap adalah aktiva yang dimiliki oleh perusahaan yang digunakan untuk aktivitas/operasi perusahaan dan memiliki manfaat lebih dari satu tahun.

Aktiva tetap ini mempunyai sifat tetap atau permanen dibeli untuk digunakan dalam kegiatan normal perusahaan, tidak untuk dijual kembali dan nilainya cukup besar atau material.

Suatu aktiva dapat dikatakan termasuk dalam aktiva tetap apabila memenuhi kriteria sebagai berikut :

1. Digunakan dalam kegiatan normal perusahaan bukan untuk dijual kembali atau investasi.
2. Dapat dipakai atau digunakan secara berulang-ulang.
3. Masa manfaatnya lebih dari satu tahun atau satu siklus operasi normal perusahaan.
4. Mempunyai nilai yang cukup material artinya nilai atau harga aktiva tersebut cukup tinggi.

Berdasarkan sifat-sifatnya, aktiva tetap dibagi atas :

1. Aktiva tetap berwujud (*tangible fixed assets*) : Aktiva yang mempunyai bentuk fisik
2. Aktiva tetap tidak berwujud (*intangible fixed assets*) : adalah suatu hak tertentu untuk jangka panjang yang mempunyai nilai ekonomis dan yang tidak mempunyai bentuk fisik (Winarto ; 2012 : 6).

## **II.8. Visual Basic**

Aplikasi yang digunakan untuk melakukan perancangan sistem informasi akuntansi ini menggunakan aplikasi visual studio 2008. VB.NET adalah salah satu bahasa pemrograman tingkat tinggi yang mendekati bahasa manusia. Kemunculan

bahasa VB.NET ini sebagai jawaban untuk menyederhanakan bahasa pemrograman pada platform .NET yang diluncurkan tahun 2002 dan untuk menjembatani programmer Visual Basic. Bahasa VB.NET secara teknis mengadopsi sintak bahasa Visual Basic. Konsistensi API membuat bahasa VB.NET menjadi pilihan dalam membuat kode program diatas platform Windows. Fitur baru bahasa VB.NET dibandingkan Visual Basic bahwa bahasa VB.NET mendukung object-oriented dan juga dynamics programming. Ini menambah daftar kemudahan untuk belajar bahasa VB.NET (Agus Kurniawan ; 2013 : 10).

## **II.9. MySQL**

MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. penulis sendiri dalam menjelaskan buku ini menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada di bawah lisensi GNU/GPL (*general public license*) (Wahana Komputer; 2010 : 5)

## **II.10. Teknik Normalisasi**

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan



menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan.

### II.10.1. Bentuk-bentuk Normalisasi

#### 1. Bentuk normal tahap pertama (1<sup>st</sup> Normal Form)

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

Contoh normalisasi 1NF adalah seperti pada table berikut :

**Tabel II.1. Tabel Bentuk Normal Pertama (1NF)**

p#	status	kota	b#	Qty
p1	20	Yogyakarta	b1	300
p1	20	Yogyakarta	b2	200
p1	20	Yogyakarta	b3	400
p1	20	Yogyakarta	b4	200
p1	20	Yogyakarta	b5	100
p1	20	Yogyakarta	b6	100
p2	10	Medan	b1	300
p2	10	Medan	b2	400
p3	10	Medan	b2	200
p4	20	Yogyakarta	b2	200
p4	20	Yogyakarta	b4	300
p4	20	Yogyakarta	b5	400

## 2. Bentuk normal tahap kedua (2<sup>nd</sup> normal form)

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

**Tabel II.2. Tabel Bentuk Normal Kedua (2NF)**

Pemasok2			Barang		
p#	Status	Kota	p#	b#	qty
P1	20	Yogyakarta	p1	b1	300
P2	10	Medan	p1	b2	200
P3	10	Medan	p1	b3	400
P4	20	Yogyakarta	p1	b4	200
P5	30	Bandung	p1	b5	100
			p1	b6	100
			p2	b1	300
			p2	b2	400
			p3	b2	200
			p4	b2	200
			p4	b4	300
			p4	b5	400

## 3. Bentuk normal tahap ketiga (3<sup>rd</sup> normal form)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

**Tabel II.3. Tabel Bentuk Normal Ketiga (3NF)**

Pemasok Kota		Kota Status	
p#	Kota	Kota	status
P1	Yogyakarta	Yogyakarta	20
P2	Medan	Medan	10
P3	Medan	Yogyakarta	20
P4	Yogyakarta	Bandung	30
P5	Bandung		

#### 4. *Boyce Code Normal Form (BCNF)*

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

#### 5. **Bentuk Normal Keempat (4NF)**

Sebuah tabel rasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD). Sebuah ketergantungan multivalued tiga kolom, satu kolom mempunyai banyak baris bernilai sama, tetapi kolom lain bernilai berbeda.

**Tabel II.4. Tabel Bentuk Normal Keempat (4NF)**

Pegawai Proyek		Pegawai Ahli	
peg#	Pry#	Peg#	ahli
1211	P1	1211	Analisis
1211	P3	1211	Perancangan
		1211	Pemrograman

## 6. Bentuk Normal Kelima

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Janner Simarmata ; 2010 : 78).

**Tabel II.5. Tabel Bentuk Normal Kelima (5NF)**

Peg#	Pry#	Ahli
1211	11	Perancangan
1211	28	Pemrograman

### II.11. UML (*Unified Modeling Language*)

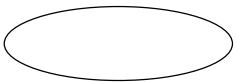
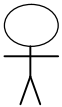


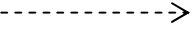
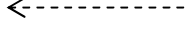
Menurut Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

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. 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, yaitu :

**Tabel II.6. Simbol Use Case**



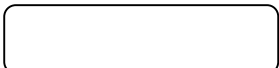
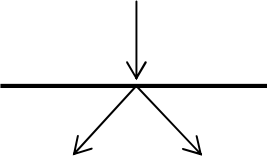
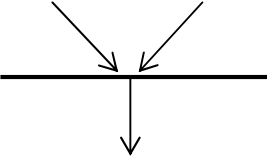
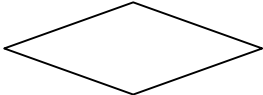
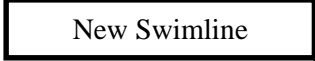
Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang 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> .
	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 mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk 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.

(Sumber : Gata, 2013 : 4)

## 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*, yaitu :

**Tabel II.7. Simbol *Activity Diagram***

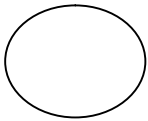
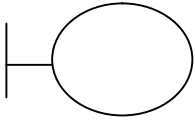
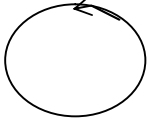

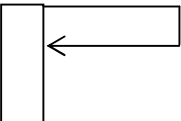


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau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

(Sumber : Gata, 2013 : 6)

## 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*, yaitu :

**Tabel II.8. Simbol *Sequence Diagram***

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<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.
	<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.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Gata, 2013 : 7)

#### 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.

**Tabel II.9. *Multiplicity Class Diagram***

<b>Multiplicity</b>	<b>Penjelasan</b>
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

(Sumber : Gata, 2013 : 9)