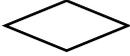
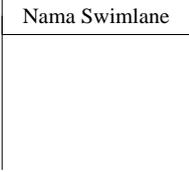


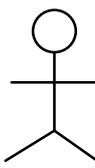
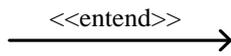
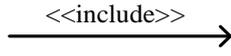
## DAFTAR SIMBOL

### **Simbol-Simbol *Activity Diagram***

Simbol	Deskripsi
Status awal 	Sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem biasanya diawali dengan kata kerja
Percabangan/ <i>decision</i> 	Percabangan terjadi jika ada pilihan lebih dari satu
Penggabungan/ <i>join</i> 	Ketika ada lebih dari satu aktivitas yang akan digabungkan
Status akhir 	Status akhir yang dilakukan sistem karena sebuah diagram aktivitas pasti memiliki status akhir
<i>Swimlane</i> 	Memisahkan organisasi yang bertanggung jawab terhadap aktivitas yang terjadi

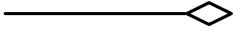
Sumber: Rosa dan M.Shalahuddin (2016:162)

## Simbol-Simbol *Use Case Diagram*

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal nama <i>use case</i> .
Aktor/actor 	Merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri.
Asosiasi 	Menunjukkan <i>use case</i> memiliki interaksi dengan aktor.
Extensi 	Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
Generalisasi 	Menunjukkan hubungan generalisasi-spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu lebih umum dari lainnya.
Include 	<i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.

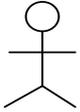
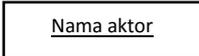
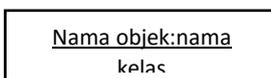
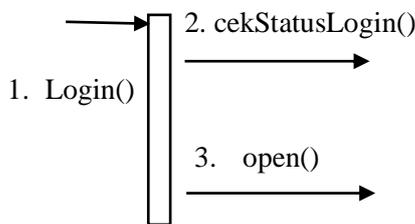
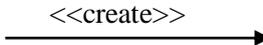
Sumber: Rosa dan M.Shalahuddin (2016:156)

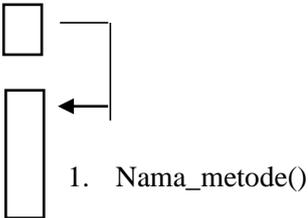
## Simbol-Simbol *Class Diagram*

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sebuah sistem
<p>Antarmuka/<i>interface</i></p>  <p><b>nama_interface</b></p>	Sama dengan konsep <i>interface</i> dalam pemrogramman berorientasi objek
<p>Asosiasi/<i>association</i></p> 	Menunjukkan relasi atau hubungan antarkelas dengan makna umum
<p>Asosiasi berarah/<i>directed association</i></p> 	Menunjukkan relasi kelas yang satu digunakan oleh kelas lain
<p>Generanlisasi</p> 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)
<p>Kebergantungan/<i>dependency</i></p> 	Relasi antarkelas dengan makna saling kebergantungan satu sama lainnya.
<p>Agregasi/<i>aggregation</i></p> 	Relasi antarkelas dengan makna semua bagian ( <i>whole-part</i> )

Sumber: Rosa dan M.Shalahuddin (2016:146)

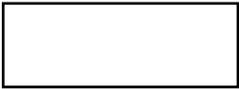
## Simbol-Simbol *Sequence Diagram*

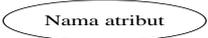
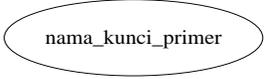
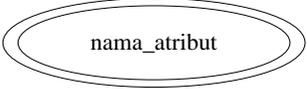
Simbol	Deskripsi
<p>Aktor</p>  <p>Nama aktor</p> <p>Atau</p>  <p>Tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang. Tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.</p>
<p>Garis hidup/<i>lifeline</i></p>	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinterkasi pesan.</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukandidalamnya, misalnya</p>  <p>Maka cekStatusLogin() dan open() dilakukan didalam metode login()</p> <p>Aktor tidak memiliki waktu aktif.</p>
<p>Pesan tipe create</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>

<p>Pesan tipe call</p> <p>1:nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.</p>  <p>1. Nama_metode()</p> <p>Arah panah menagarah pada objek yang memiliki operase/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
---	--

Sumber: Rosa dan M.Shalahuddin (2016:165)

**Simbol-Simbol *Entity Relationship Diagram (ERD)***

Simbol	Deskripsi
<p>Entitas/<i>entity</i></p> 	<p>Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama table</p>

<p>Atribut</p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas</p>
<p>Atribut Kunci Prime</p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)</p>
<p>Atribut Multinilai/<i>multivalue</i></p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu</p>
<p>Relasi</p> 	<p>Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja</p>

Sumber: Rosa dan M.Shalahuddin (2016:165)