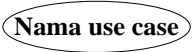
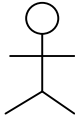

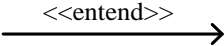

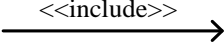



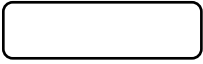
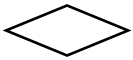


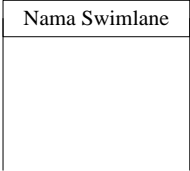
DAFTAR SIMBOL

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 (<i>Actor</i>)  | 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. |


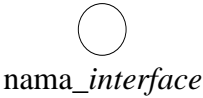





Sumber: Rosa dan Shalahuddin (2016:156)

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 |

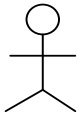
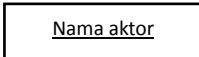
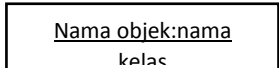

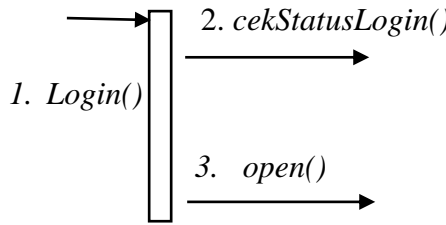
Sumber: Rosa dan M.Shalahuddin (2016:162)

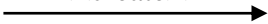

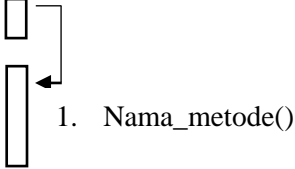
Simbol Diagram Kelas (*Class Diagram*)

| Simbol | Deskripsi |
|--|---|
| <p>Kelas</p>  | Kelas pada struktur sebuah sistem |
| <p>Antarmuka (<i>interface</i>)</p>  | Sama dengan konsep <i>interface</i> dalam pemrogramman berorientasi onjek |
| <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 antar kelas dengan makna generalisasi-spesialisasi (umum-khusus) |
| <p>Kebergantungan (<i>dependency</i>)</p>  | Relasi antar kelas dengan makna saling kebergantungan satu sama lainnya. |
| <p>Agregasi (<i>aggregation</i>)</p>  | Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>) |

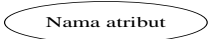
Sumber: Rosa dan Shalahuddin (2016:146).

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 <i>frase</i> nama aktor.</p> |
| <p>Garis hidup (<i>lifeline</i>)</p> | <p>Menyatakan kehidupan suatu objek.</p> |
| <p>Objek</p>  | <p>Menyatakan objek yang berinteraksi 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 <i>cekStatusLogin()</i> dan <i>open()</i> dilakukan didalam <i>metode login()</i></p> <p>Aktor tidak memiliki waktu aktif.</p> |
| <p>Pesan tipe create</p> | <p>Menyatakan suatu objek membuat objek yang lain,</p> |

| | |
|--|--|
| <p style="text-align: center;"> <<create>>  </p> | <p>arah panah mengarah pada objek yang dibuat.</p> |
| <p>Pesan tipe call</p> <p style="text-align: center;"> 1:nama_metode()  </p> | <p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.</p> <div style="text-align: center;">  </div> <p>Arah panah menagarah pada objek yang memiliki metode atau operasi, karena ini memanggil metode atau operasi maka metode atau operasi yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p> |

Simbol *Entity Relationship Diagram (ERD)*

| Simbol | Deskripsi |
|--|--|
| Entitas (<i>entity</i>)  | Entitas merupakan data inti yang akan disimpan, pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi computer, penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel |
| Atribut  | Kolom atau <i>Field</i> data yang akan disimpan dalam suatu entitas |
| Atribut Kunci Primer atau Kunci Utama (<i>Primary Key (PK)</i>)  | Kolom atau <i>Field</i> data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan, biasanya berupa <i>id</i> , kunci primer atau kunci utama dapat lebih dari satu kolom dengan kombinasi dari kolom tersebut dapat bersifat unik yaitu berbeda tanpa ada yang sama |
| Atribut Multinilai (<i>multivalue</i>)  | Kolom atau <i>Field</i> data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu |
| Relasi  | Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja |

Sumber: Rosa dan Shalahuddin (2016:165)