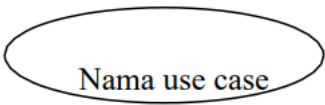
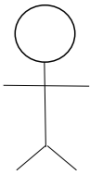

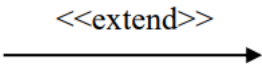
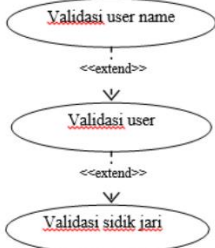

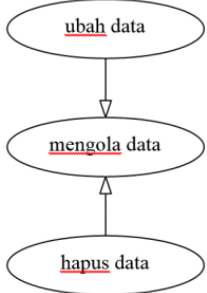






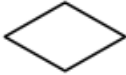


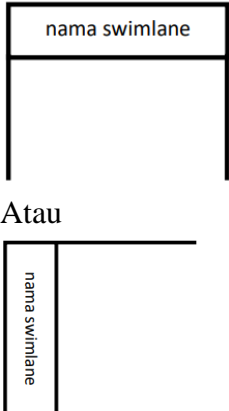
DAFTAR SIMBOL

A. Use Case Diagram








No	Simbol	Deskripsi
1.	Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama use case.
2.	Aktor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar 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 di awal frase nama aktor.
3.	Asosiasi 	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
4.	Ekstensi 	Relasi use case tambahan sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu, mirip dengan prinsip inheritance pada pemrograman berorientasi objek, biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, misal

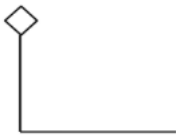
		 <p>Arah panah mengarah pada use case yang ditambahkan, biasanya use case yang menjadi extend-nya merupakan jenis yang sama dengan use case yang menjadi induknya.</p>
5.	<p>Generalisasi</p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :</p>  <p>arah panah mengarah pada use case yang menjadi generalisasinya (umum).</p>
6.	<p>Menggunakan</p> <p><<include>></p>  <p><<uses>></p> 	<p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini Ada dua sudut pandang yang cukup besar mengenai include di use case.</p>

B. Activity Diagram

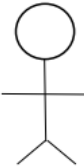

No	Simbol	Deskripsi
1.	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Pecabangan/ decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan/ Join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

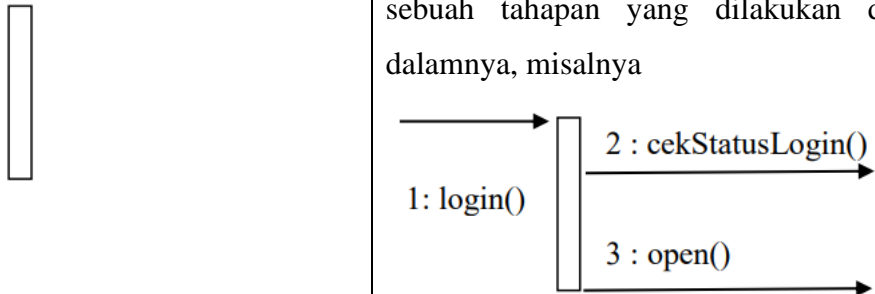
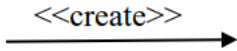

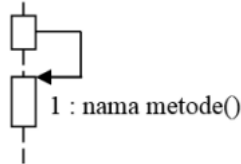

C. Class Diagram


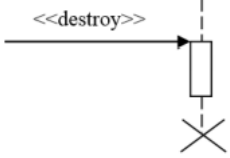
No	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Antar Muka 	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	Pecabangan/ decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Asosiasi 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
5.	Asosiasi Berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
6.	Generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum - khusus).
7.	Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas

8.	Agregasi 	Relasi antar kelas dengan makna semua-bagian (whole-part).
----	---	--

D. Sequence Diagram

No	Simbol	Deskripsi
1.	Aktor  Atau <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <u>Nama aktor</u> </div> Tanpa waktu aktif	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 dalam menggunakan kata benda diawal frase nama aktor.
2.	Garis Hidup 	Menyatakan kehidupan suatu objek.
3.	Objek <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <u>Nama objek : nama kelas</u> </div>	Menyatakan objek yang berinteraksi pesan
4.	Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah

		<p>sebuah tahapan yang dilakukan di dalamnya, misalnya</p> <p>Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif.</p>
5.	<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
6.	<p>Pesan tipe <i>call</i></p> <p>1 : nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
7.	<p>Pesan tipe <i>send</i></p> <p>1 : masukkan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
8.	<p>Pesan tipe <i>return</i></p>	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau</p>

	<p>1 : keluaran</p> 	<p>metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
9.	<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>Destroy</i>.</p>