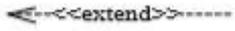
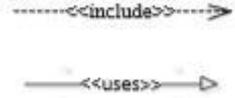


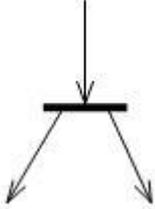
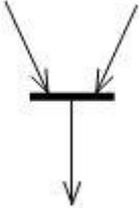
DAFTAR SIMBOL

1. Use Case Diagram

Gambar	Keterangan
<p><i>Use Case</i></p> 	<p>Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit dan actor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case</p>
<p><i>Aktor/actor</i></p> 	<p>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 dari orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p><i>Asosiasi/association</i></p> 	<p>Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor</p>
<p><i>Ekstensi/extend</i></p> 	<p>Relasi use case tambahan ke 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</p>
<p><i>Generalisasi/generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum khusus) antara dua buah use case diman fungsi yang satu adalah fungsi yang lebih umum dari lainnya</p>
<p><i>Menggunakan/include</i></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</p>

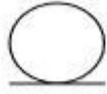
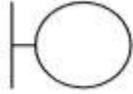
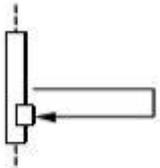
Sumber: (Sukamto &Salahudim, 2016)

2. Activity Diagram

Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	<i>End Point</i> , akhir aktivitas
	<i>Activities</i> , menggambar kan suatu proses atau kegiatan bisnis
	<i>Fork</i> (percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabung kan dua kegiatan paralel menjadi satu
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Points</i> , menggambar kan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa

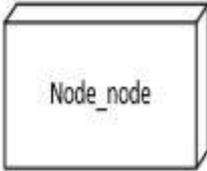
Sumber : Gata dan Grace Gata (Dalam Hendini, 2016)

3. Sequence Diagram

Gambar	Keterangan
	<p><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</p>
	<p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i></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>, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i></p>

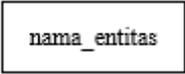
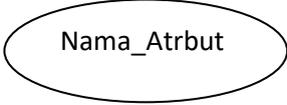
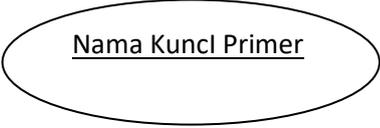
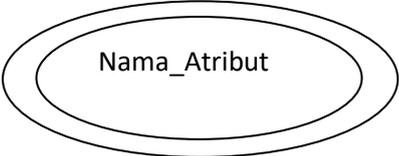
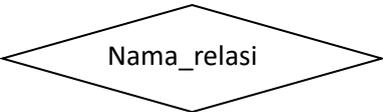
Sumber: Gata dan Grance Gata (Hendini, 2016)

4. Deployment Diagram

Gambar	Keterangan
<p><i>Package</i></p> 	<p><i>Package</i> merupakan sebuah bungkus dari satu atau lebih node</p>
<p><i>Node</i></p> 	<p>Biasanya mengacu pada perangkat keras (<i>hardware</i>), dan perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika didalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikut sertakan harus sesuai dengan komponen yang telah didefinisikan selumnya pada diagram komponen</p>
<p>Kebergantungan/<i>depedency</i></p> 	<p>Kebergantungan antar node, arah panah mengarah pada node yang dipakai</p>
<p>Link</p> 	<p>Relasi antar node</p>

Sumber: (Sukamto & Muslihuddin, 2016)

5. Entity Relationship Diagram (ERD)

Gambar	Keterangan
<p data-bbox="301 389 478 427"><i>Entitas/Entity</i></p> 	<p data-bbox="740 427 1359 622">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</p>
<p data-bbox="301 622 510 660"><i>Atribut/attribute</i></p> 	<p data-bbox="740 660 1359 831">Field atau kolom data yang butuh disimpan dalam suatu entitas.</p>
<p data-bbox="301 831 590 869">Atribut kunci primer</p> 	<p data-bbox="740 869 1359 1131">Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan, biasanya berupa id.</p>
<p data-bbox="301 1131 710 1169"><i>Atribut Multinilai/ Multivalue</i></p> 	<p data-bbox="740 1169 1359 1397">Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki lebih dari satu.</p>
<p data-bbox="301 1397 399 1435">Relasi</p> 	<p data-bbox="740 1435 1359 1606">Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.</p>
<p data-bbox="301 1606 606 1644"><i>Asosiasi / Association</i></p> 	<p data-bbox="740 1644 1359 1865">Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas</p>

Sumber: (Sukamto & M.Shalahuddin, 2015)