

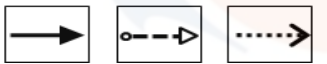
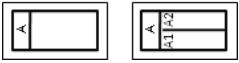


DAFTAR SIMBOL

1. Simbol BPMN





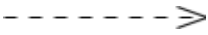


Daftar Simbol BPMN


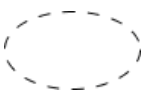

	<p><i>Event</i> Lingkaran dengan pusat terbuka untuk memungkinkan pembedaan <i>trigger</i> dan <i>result</i> yang berbeda. Terdapat 3 tipe <i>event</i> berdasarkan kapan mereka mempengaruhi aliran yaitu <i>Start</i>, <i>Intermediate</i>, dan <i>End</i></p>
<p style="text-align: center;">Activity Gateway</p> 	<p><i>Activity</i> Ditunjukkan dengan persegi panjang dengan ujung-ujung bulat dan merupakan bentuk umum untuk pekerjaan yang dilakukan oleh perusahaan.</p> <p><i>Gateway</i> digambarkan dengan bentuk seperti belah ketupat dan digunakan untuk mengontrol percabangan dan penggabungan <i>Sequence Flow</i>.</p>
<p style="text-align: center;">Sequence Flow Message Flow Association</p> 	<p><i>Sequence Flow</i> Digunakan untuk menunjukkan urutan yang kegiatan akan yang dilakukan dalam sebuah proses.</p> <p><i>Message Flow</i> Digunakan untuk menunjukkan aliran pesan antara dua entitas yang siap untuk mengirim dan menerima.</p> <p><i>Association</i> Digunakan untuk asosiasi data, informasi dan artefak dengan aliran benda.</p>
<p style="text-align: center;">Pool Lane</p> 	<p><i>Pool</i> Mewakili partisipan dalam sebuah proses.</p> <p><i>Lane</i> Sub-bagian dalam sebuah pool dan mengkategorikan aktivitas.</p>

2. Simbol UML

a. Use Case Diagram





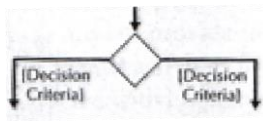

Daftar Simbol Use Case Diagram



No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem itu sendiri.
2.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3.		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.
4.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri.
5.		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.
6.		<i>Association</i>	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
7.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

8.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
9.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10.		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

b. *Activity Diagram*

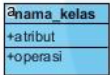






Daftar Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Activity</i> Merepresentasikan sekumpulan aktivitas.
	<i>Control flow</i> Menunjukkan rangkaian dari suatu eksekusi.
	<i>Initial node</i> Pertanda dari suatu awal aktivitas.
	<i>Final activity node</i> Untuk menunjukkan akhir dari suatu aktivitas.
	<i>Decision node</i> Merepresentasikan kondisi untuk memastikan bahwa alur objek hanya bergerak dalam satu jalur.
	<i>Merge node</i> Digunakan untuk mengembalikan berbagai <i>decision path</i> menjadi

	<p>satu.</p> <p><i>Fork node</i> Digunakan untuk membagi perilaku menjadi kumpulan aktivitas yang berjalan secara paralel atau bersamaan.</p>
	<p><i>Join node</i> Digunakan untuk menyatukan kembali kumpulan aktivitas yang berjalan secara paralel atau bersamaan.</p>


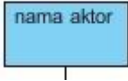






c. *Class Diagram*

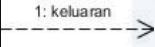
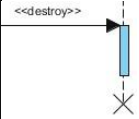
Daftar Simbol *Class Diagram*

Simbol	Deskripsi
<p>Kelas</p> 	<p>Kelas pada struktur system.</p>
<p>Antarmuka / <i>Interface</i></p>  <p>Nama_interface</p>	<p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.</p>
<p>Asosiasi / <i>Association</i></p> 	<p>Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Asosiasi berarah / <i>Directed Association</i></p> 	<p>Relasi antarkelas dengan makna kelas satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Generalisasi</p> 	<p>Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).</p>
<p>Kebergantungan / <i>Dependency</i></p> 	<p>Relasi antar kelas dengan makna kebergantungan antarkelas.</p>
<p>Agregasi / <i>Aggregation</i></p> 	<p>Relas antarkelas dengan makna semuabagian (<i>whole-part</i>).</p>

c. *Sequence Diagram*

Daftar Simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p>  <p>Atau</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah 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 dilakukan didalamnya.</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri. 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>
<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>

<p>Pesan tipe <i>return</i></p> 	<p>Menyatakan suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian keobjek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada <i>destroy</i>.</p>