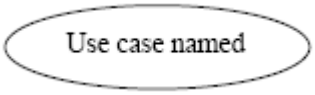
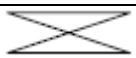
## DAFTAR SIMBOL

Tabel Notasi *Use Case Diagram*

| SIMBOL | NAMA | KETERANGAN |
|---|---|---|
|  | **Actor** | Actor adalah pengguna sistem. Actor tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan input atau memberikan output, maka aplikasi tersebut juga bisa dianggap sebagai actor. |
|  | **Use Case** | Use case digambarkan sebagai lingkaran elips dengan nama use case dituliskan didalam elips tersebut. |
|  | **Association** | Asosiasi digunakan untuk menghubungkan actor dengan use case. Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara Actor dengan Use Case. |

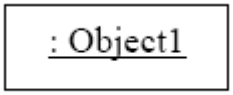
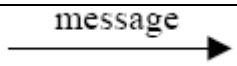
(Sumber : Bambang Heriyanto, *Rekayasa Sistem Berorientasi Objek* 2004, Informatika : Bandung)

Tabel Notasi *Activity Diagram*

| SIMBOL | KETERANGAN |
|---|---|
|  | Titik Awal |
|  | Titik Akhir |
|  | Activity |
|  | Pilihan Untuk mengambil Keputusan |
|  | Fork; Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan peralel menjadi satu. |
|  | Rake; Menunjukkan adanya dekomposisi |
|  | Tanda Waktu |
|  | Tanda pengiriman |
|  | Tanda penerimaan |
|  | Aliran akhir (Flow Final) |

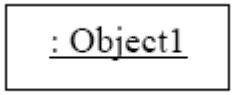
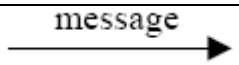
(Sumber : Bambang Heriyanto, *Rekayasa Sistem Berorientasi Objek* 2004, Informatika : Bandung)

Tabel Notasi *Sequence Diagram*

| SIMBOL | NAMA | KETERANGAN |
|---|---|---|
|  | **Object** | Object merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama obyek didalamnya yang diawali dengan sebuah titik koma |
|  | **Actor** | Actor juga dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom. Simbol Actor sama dengan simbol pada Actor Use Case Diagram. |
|  | **Lifeline** | Lifeline mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk Lifeline adalah garis putus-putus vertikal yang ditarik dari sebuah obyek. |
|  | **Activation** | Activation dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah lifeline. Activation mengindikasikan sebuah obyek yang akan melakukan sebuah aksi. |
|  | **Message** | Message, digambarkan dengan anak panah horizontal antara Activation. Message mengindikasikan komunikasi antara object-object. |

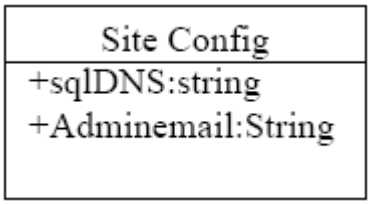
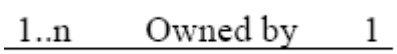
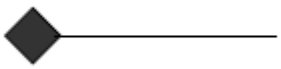
(Sumber : Bambang Heriyanto, *Rekayasa Sistem Berorientasi Objek* 2004, Informatika : Bandung)

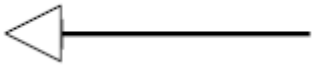
Tabel Notasi *Collaboration Diagram*

| SIMBOL | NAMA | KETERANGAN |
|---|---|---|
|  | **Object** | Object merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama obyek didalamnya yang diawali dengan sebuah titik koma |
|  | **Actor** | Actor juga dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom. Simbol Actor sama dengan simbol pada Actor Use Case Diagram. |
|  | **Message** | Message, digambarkan dengan anak panah yang mengarah antar obyek dan diberi label urutan nomor yang mengindikasikan urutan komunikasi yang terjadi antar obyek. |

(Sumber : Bambang Heriyanto, *Rekayasa Sistem Berorientasi Objek* 2004, Informatika : Bandung)

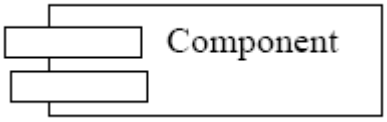
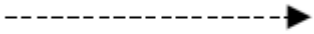
Tabel Notasi pada *Class Diagram*

| SIMBOL | NAMA | KETERANGAN |
|---|---|---|
|  | **Class** | *Class* adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah class digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari *class*. Bagian tengah mendefinisikan property/atribut *class*. Bagian akhir mendefinisikan method method dari sebuah *class*. |
|  | **Assosiation** | Sebuah asosiasi merupakan sebuah *relationship* paling umum antara 2 class, dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 *class*. Garis ini bisa melambangkan tipe-tipe *relationship* dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah *relationship* (Contoh: One-to-one, one-to-many, many-to-many). |
|  | **Composition** | Jika sebuah *class* tidak bisa berdiri sendiri dan harus merupakan bagian dari *class* yang lain, maka *class* tersebut memiliki relasi *Composition* terhadap *class* tempat dia bergantung tersebut. Sebuah *relationship composition* digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid. |

| | | |
|---|---|---|
|  | **Dependency** | Kadangkala sebuah *class* menggunakan *class* yang lain. Hal ini disebut *dependency*. Umumnya penggunaan *dependency* digunakan untuk menunjukkan operasi pada suatu *class* yang menggunakan *class* yang lain. Sebuah *dependency* dilambangkan sebagai sebuah panah bertitik-titik. |
|  | **Aggregation** | *Aggregation* mengindikasikan keseluruhan bagian *relationship* dan biasanya disebut sebagai relasi "mempunyai sebuah" atau "bagian dari". Sebuah *aggregation* digambarkan sebagai sebuah garis dengan sebuah jajaran genjang yang tidak berisi/tidak solid. |
|  | **Generalization** | Sebuah relasi *generalization* sepadan dengan sebuah relasi *inheritance* pada konsep berorientasi obyek. Sebuah *generalization* dilambangkan dengan sebuah panah dengan kepala panah yang tidak solid yang mengarah ke kelas "*paren*t"-nya/induknya. |

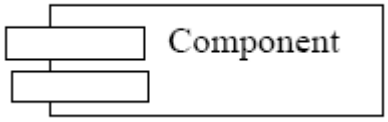
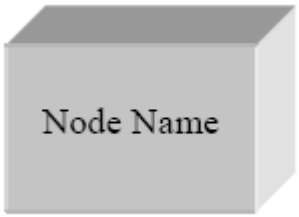
(Sumber : Bambang Heriyanto, *Rekayasa Sistem Berorientasi Objek* 2004, Informatika : Bandung)

Tabel Notasi *Component Diagram*

| SIMBOL | NAMA | KETERANGAN |
|---|---|---|
|  | Component | Sebuah komponen melambangkan sebuah entitas software dalam sebuah sistem. Sebuah komponen dinotasikan sebagai sebuah kotak segiempat dengan dua kotak kecil tambahan yang menempel disebelah kirinya. |
|  | Depedency | Sebuah Dependency digunakan untuk menotasikan relasi antara dua komponen. Notasinya adalah tanda panah putus-putus yang diarahkan kepada komponen tempat sebuah komponen itu bergantung. |

(Sumber : Bambang Heriyanto, *Rekayasa Sistem Berorientasi Objek* 2004, Informatika : Bandung)

Tabel Notasi *Deployment Diagram*

| SIMBOL | NAMA | KETERANGAN |
|---|---|---|
|  | **Componen** | Pada deployment diagram, komponenkomponen yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka |
|  | **Node** | Node menggambarkan bagian-bagian hardware dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi. |
|  | **Association** | Sebuah association digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara element-elemen hardware. |

(Sumber : Bambang Heriyanto, *Rekayasa Sistem Berorientasi Objek* 2004, Informatika : Bandung)