**DAFTAR SIMBOL**

**Simbol Flowmap**

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Pengertian** | **Keterangan** |
|  | Dokumen (*Document*) | Menunjukan dokumen sebagai yang digunakan untuk merekam data terjadinya suatu transaksi |
|  | Operasional Manual | Menunjukan proses yang dikerjakan secara manual |
|  | Garis aliran (*flow line*) | Menunjukan arus data antar simbol/proses |
|  | *Decision* | Menunjukan pilihan yang akan dikerjakan atau keputusan yang harus dibuat dalam proses pengolahan data |
|  | *Conector (On-page connector)* | Digunakan untuk penghubung dalam satu halaman |
|  | *Conector (Off-page connector)* | Digunakan untuk penghubung berbeda halaman |
| A | *Off line storage* | Digunakan untuk menyimpan data secara manual dan sementara, jika “A” berarti disimpan menurut abjad, “N” berarti disimpan menurut nomor urut dan jika “T” berarti disimpan menurut kronologis atau menurut tanggal |
|  | Keterangan atau komentar | Deskripsi proses atau komentar, untuk memperjelas pesan yang disampaikan dalam bagan alir |
|  | Pertemuan garis alir | Menunjukan dua garis alir bertemu dan salah satu garis mengikuti arus lainnya |
|  | Persimpangan garis alir | Menunjukan arah masing-masing garis, salah satu garis dibuat sedikit melengkung tepat pada persimpangan kedua garis tersebut |
|  | Catatan | Digunakan untuk menggambarkan catatan akuntansi yang digunakan untuk mencatat data yang direkam sebelumnya didalam dokumen atau formulir |
|  | Penyimpanan atau *Storage* | Menunjukan akses langsung perangkat penyimpanan atau *storage* pada disket |

**Simbol ERD (Entity Relation Diagram)**

|  |  |
| --- | --- |
| SIMBOL | KETERANGAN |
|  | **ENTITAS**  Menunjukan pelaku yang terlibat dalam sistem informasi |
|  | **GARIS RELASI**  Garis yang menghubungkan antara entitas dengan relasi |
|  | **PENYIMPANAN KOMPUTERISASI**  Menunjukan hubungan antara entitas satu dengan entitas lain. |

**Simbol Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Actor* | Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
| 2 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent )*akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
| 3 |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 4 |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara *eksplisit*. |
| 5 |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| 6 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

**Simbol Class Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Generalization* | Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 2 |  | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | *Collaboration* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 5 |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 6 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* aka nmempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 7 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya |

**Simbol Sequence Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *LifeLine* | Objek *entity*, antarmuka yang saling berinteraksi. |
| 2 |  | *Message* | Spesifikasi dari komunikas iantar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |
| 3 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |

**Simbol State Chart Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *State* | Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek. |
| 2 |  | *Initial Pseudo State* | Bagaimana objek dibentuk atau diawali |
| 3 |  | *Final State* | Bagaimana objek dibentuk dan dihancurkan |
| 4 |  | *Transition* | Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya |
| 5 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 6 |  | *Node* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

**Simbol Activity Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| **1** |  | *Activity* | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| **2** |  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatuaksi |
| **3** |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| **4** |  | *Activity Final Node* | Bagaimana objek dibentuk dan dihancurkan |
| **5** |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |