# DAFTAR SIMBOL

***FlowMap***

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **Keterangan** |
| 1. |  | **Simbol arus/*flow***  menyatakan jalannya arus suatu proses. |
| 2. |  | **Simbol *process***  Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer. |
| 3. |  | **Simbol manual**  Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer. |
| 4. |  | **Simbol *decision***  Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya / tidak. |
| 5. |  | **Simbol terminal**  Menyatakan permulaan atau akhir sebuah program. |
| 6. |  | **Simbol *offline-storage***  Menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu. |
| 7. |  | **Simbol manual *input***  Memasukkan data secara manual dengan menggunakan *online keyboard.* |
| 8. |  | **Simbol *document***  Mencetak keluaran dalam bentuk dokumen (melalui printer). |

***Data Flow Diagram* (DFD)**

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **Keterangan** |
| 1. |  | **Proses**  Proses dapat mengolah data atau aliran data masuk menjadi aliran data keluar. Proses berfungsi mentransformasikan satu atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Setiap proses memiliki satu atau beberapa masukan serta menghasilkan satu atau beberapa data keluaran. |
| 2. |  | **Terminator**  Mewakili entitas luar dimana sistem berkomunikasi |
| 3. |  | ***Data Store***  Tempat penyimpanan data pengikat data yang ada dalam sistem. *Data store* disimbolkan dengan sepasang dua garis sejajar atau dua garis dengan salah satu sisi samping terbuka. Proses dapat mengambil data dati atau memberikan data ke *database.* |
| 4. |  | ***Data Flow***  Tempat mengalirnya informasi dna digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ditunjukan dengan arah panah dan garis diberi nama atas arus data yang mengali. |

***Use Case Diagram***

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **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 |

***Activity Diagram***

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **Keterangan** |
| 1. |  | ***Activity***  Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| 2. |  | ***Action***  State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 3. |  | ***Initial Node***  Bagaimana objek dibentuk atau diawali. |
| 4. |  | ***Actifity Final Node***  Bagaimana objek dibentuk dan dihancurkan |
| 5. |  | ***Fork Node***  Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran. |
| 6. |  | ***Decision***  Pilihan Untuk mengambil Keputusan. |

***Class Diagram***

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **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)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 7. |  | ***Association***  Apa yang menghubungkan antara objek satu dengan objek lainnya. |

***Sequence Diagram***

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **Keterangan** |
| 1. |  | **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 |
| 2. |  | **Actor**  Actor juga dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom. Simbol Actor sama dengan simbol pada Actor Use Case Diagram. |
| 3. |  | **Lifeline**  Lifeline mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk Lifeline adalah garis putus-putus vertikal yang ditarik dari sebuah obyek. |
| 4. |  | **Activation**  Activation dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah lifeline. Activation mengindikasikan sebuah obyek yang akan melakukan sebuah aksi. |
| 5. |  | **Message**  Message, digambarkan dengan anak panah horizontal antara Activation. Message mengindikasikan komunikasi antara object-object. |

***Entity Relationship Diagram* (ERD)**

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **Keterangan** |
| 1. |  | **Data Entitas**  Segala sesuatu baik yang nyata maupun abstrak yang datanya akan direkam |
| 2. |  | **Relasi**  Menunjukan adanya hubungan antar sejumlah entitas yang berasal dari himpunan entitas berbeda |
| 3. |  | **Aliran Data**  Menyatakan penghubung antar relasi dengan data entitas dan data entitas dengan atribut |
| 4. |  | **Atribut Data**  Atribut data yang dimiliki oleh tiap entitas yang nantinya akan menjadi field data |