

***Pertemuan 11***

***State Machine Diagram***

# *State Machine Diagram*

State Machine Diagram adalah teknik yang umum digunakan untuk menggambarkan behaviour sebuah sistem. Hal ini digunakan untuk membantu analis, perancang dan pengembang untuk memahami perilaku obyek pada sistem.

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya statechart diagram menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statechart diagram).

# *State Machine Diagram*

Dalam UML, state digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar state umumnya memiliki kondisi guard yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. Action yang dilakukan sebagai akibat dari event tertentu dituliskan dengan diawali garis miring.

Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah

# *State Machine Diagram*

## State

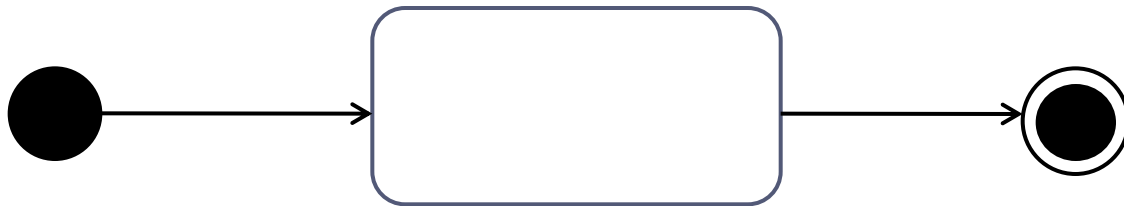
- Perilaku yang menspesifikasi unsur kedudukan suatu objek atau interaksi-interaksi sepanjang waktu dalam menanggapi event-event yang terjadi.
- Penggambaran suatu state memuat beberapa unsur yaitu state itu sendiri, transisi (perubahan dari suatu state ke state lainnya), event (suatu keadaan yang memicu sebuah transisi, serta aktivitas (tanggapan terhadap transisi)
- Digambarkan sebagai empat-persegi-panjang yang sudut-sudutnya melengkung, yang memuat namanya (serta substate didalamnya, jika ada)

## *State Machine Diagram*

Statechart Diagram, memperlihatkan state-state pada sistem; memuat state, transisi, event serta aktifitas.

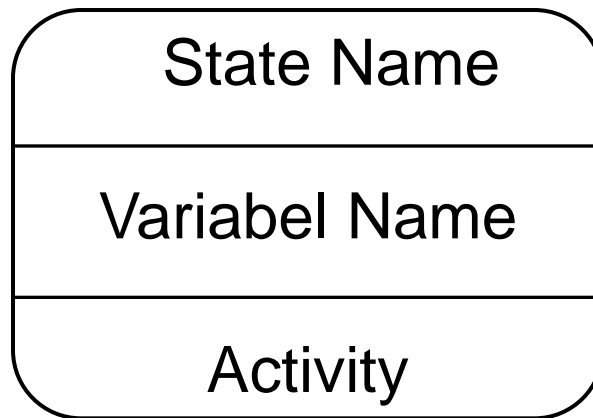
# Simbol

Simbol UML untuk state chart diagram adalah segiempat yang tiap pojoknya dibuat rounded. Titik awalnya menggunakan lingkaran solid yang diarsir dan diakhiri dengan mata.



## Simbol (lanjutan)

UML juga memberi pilihan untuk menambahkan detail ke dalam simbol tersebut dengan membagi tiga area yaitu nama state, variabel dan aktivitas.



State variabel seperti timer dan counter yang kadangkala sangat membantu.

# Simbol (lanjutan)

Activity terdiri atas events dan action, tiga hal yang sering digunakan dalam activity adalah:

## 1.Entry

Apa yang terjadi ketika sistem masuk ke state

## 2.Exit

Apa yang terjadi ketika sistem meninggalkan state

## 3.Do

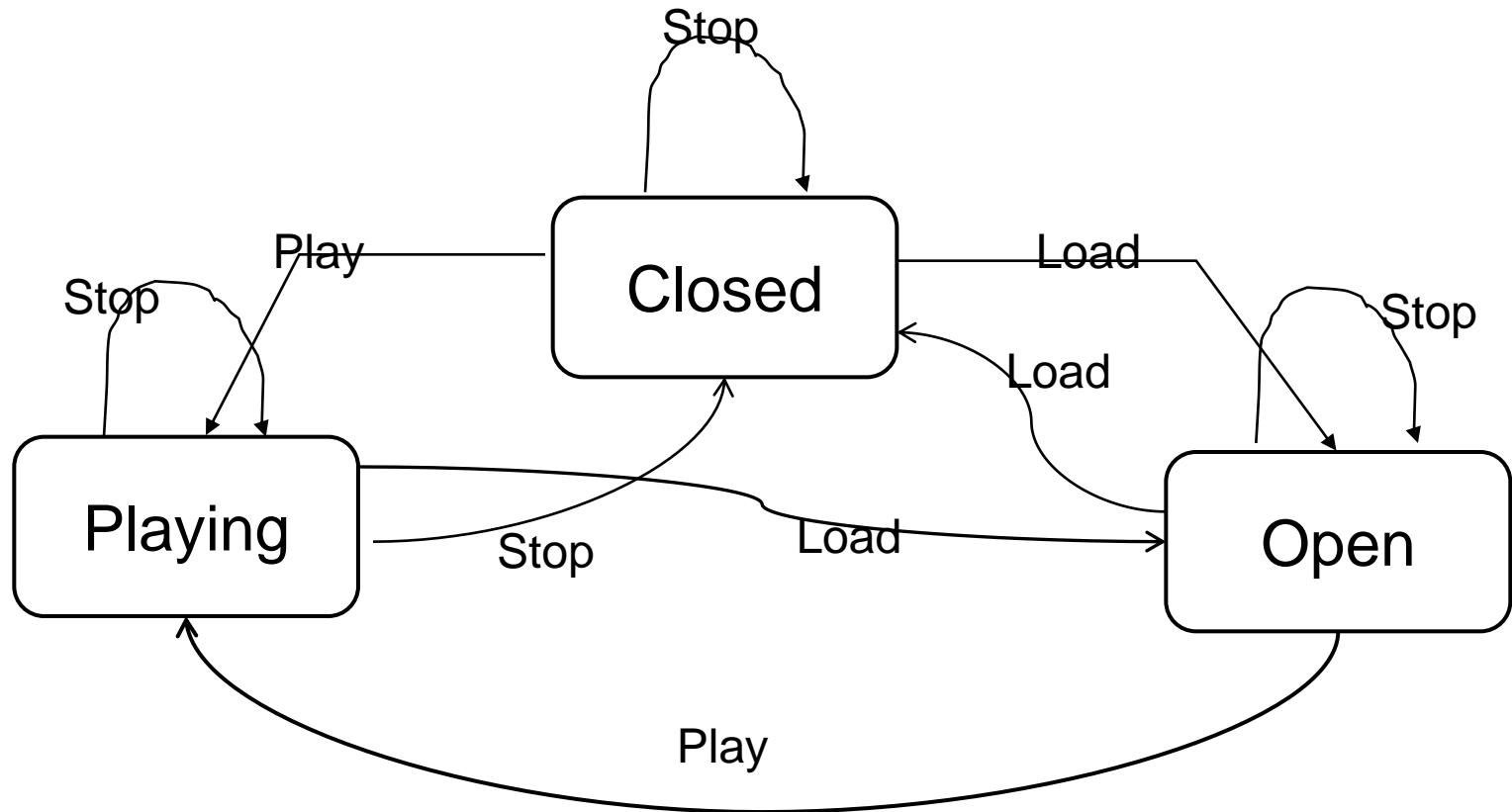
Apa yang terjadi ketika sistem ada di state



## ***State, Event dan Transition***

Statechart diagram menampilkan state-state yang mungkin dari sebuah obyek, event yang bisa dideteksi dan respon atas event-event tersebut. Secara umum, pendeteksian sebuah event dapat menyebabkan sebuah obyek bergerak dari satu state ke state yang lain yang disebut dengan transition. Sebagai contoh jika sebuah CD Player dalam keadaan terbuka, penekanan tombol load akan menyebabkan drawer CD Player berpindah ke state close, sebagaimana digambarkan sebagai berikut:

## *State, Event dan Transition*



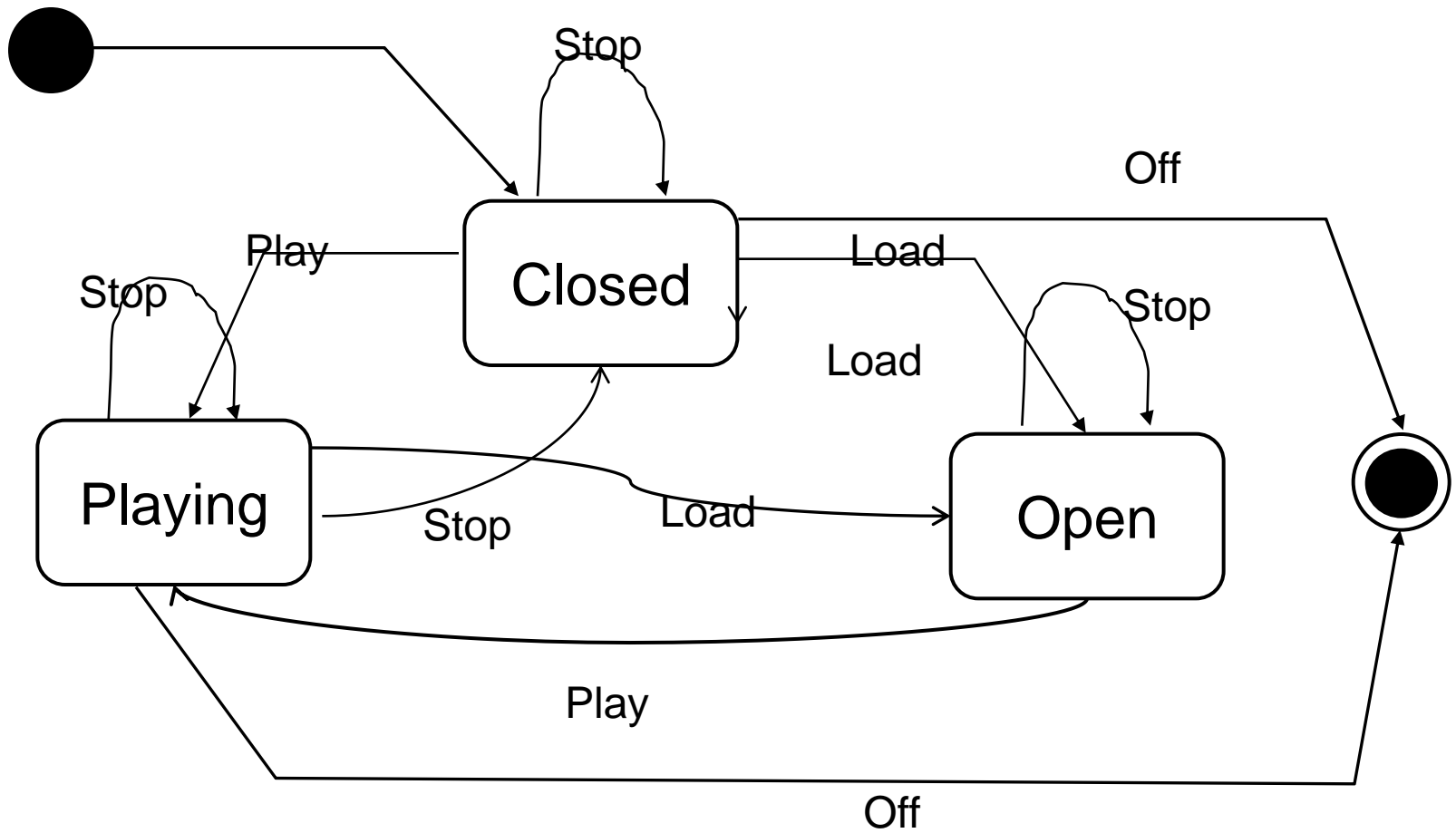
## Initial dan Final State

Gambar pada slide sebelumnya menjelaskan bagaimana ketika CD Player sedang dipakai, tetapi tidak menjelaskan apa yang terjadi ketika mesin di matikan atau dihidupkan.

Untuk menunjukkan perilaku ini kita bisa menambahkan initial state diagram. Gambar berikut ini menunjukkan bahwa initial state untuk CD Player selalu diposisi close setelah mesin dihidupkan. Tidak ada event yang harus dituliskan pada initial state.

Event yang menyebabkan final state bisa tercapai adalah saat CD Player dimatikan (event = off)

# Initial dan Final State

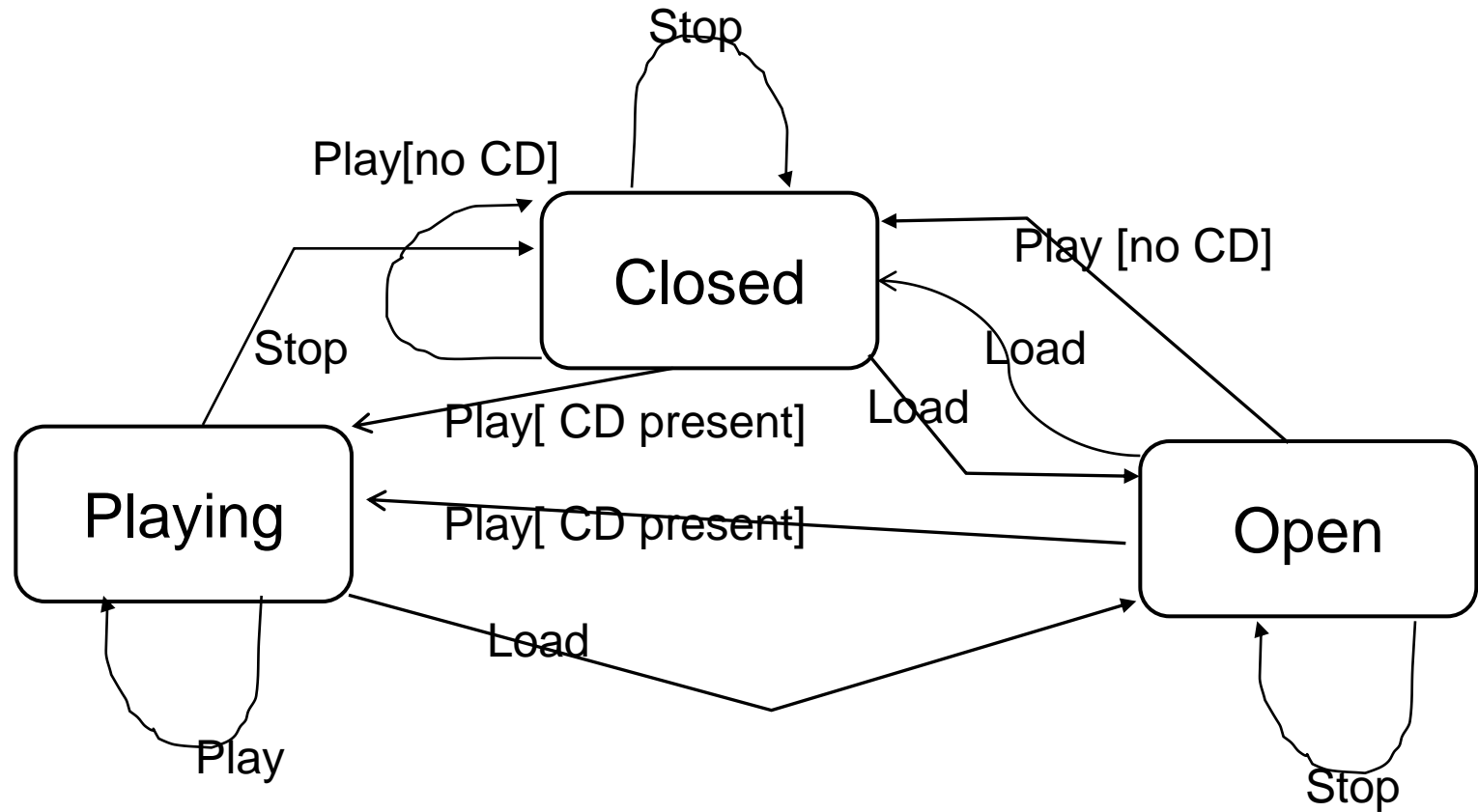


# Guard Condition

Dalam gambar sebelumnya salah satu masalah adalah CD player tidak selalu pada state playing ketika tombol play ditekan. Seharusnya state playing dilakukan jika ada CD di drawer atau tetap dalam kondisi close jika tidak ada CD di drawer.

Untuk menampilkan informasi tersebut dengan statechart bisa ditambahkan guard condition untuk transition play. Guard condition adalah bagian spesifikasi dari transition dan ditulis dengan sepasang kurung kotak/[] sesudah nama event yang memberi label transition

## Guard Condition (lanjutan)



## Guard Condition (lanjutan)

Jika ada lebih dari satu transition yang mempunyai hasil guard condition true, maka hanya ada satu saja yang akan dijalankan. Sebagai contoh anggaplah CD player pada posisi state open dan tombol play ditekan, yang pertama kali terjadi adalah drawer akan ditutup. Hal ini penting karena mesin akan bisa mendeteksi apakah ada CD atau tidak di drawer bila drawer dalam posisi tertutup. Dalam kondisi masih state open, guard condition dievaluasi pada transition play untuk melihat transition mana yang harus dijalankan. Jika CD ada, transition akan berubah dari open menjadi playing.

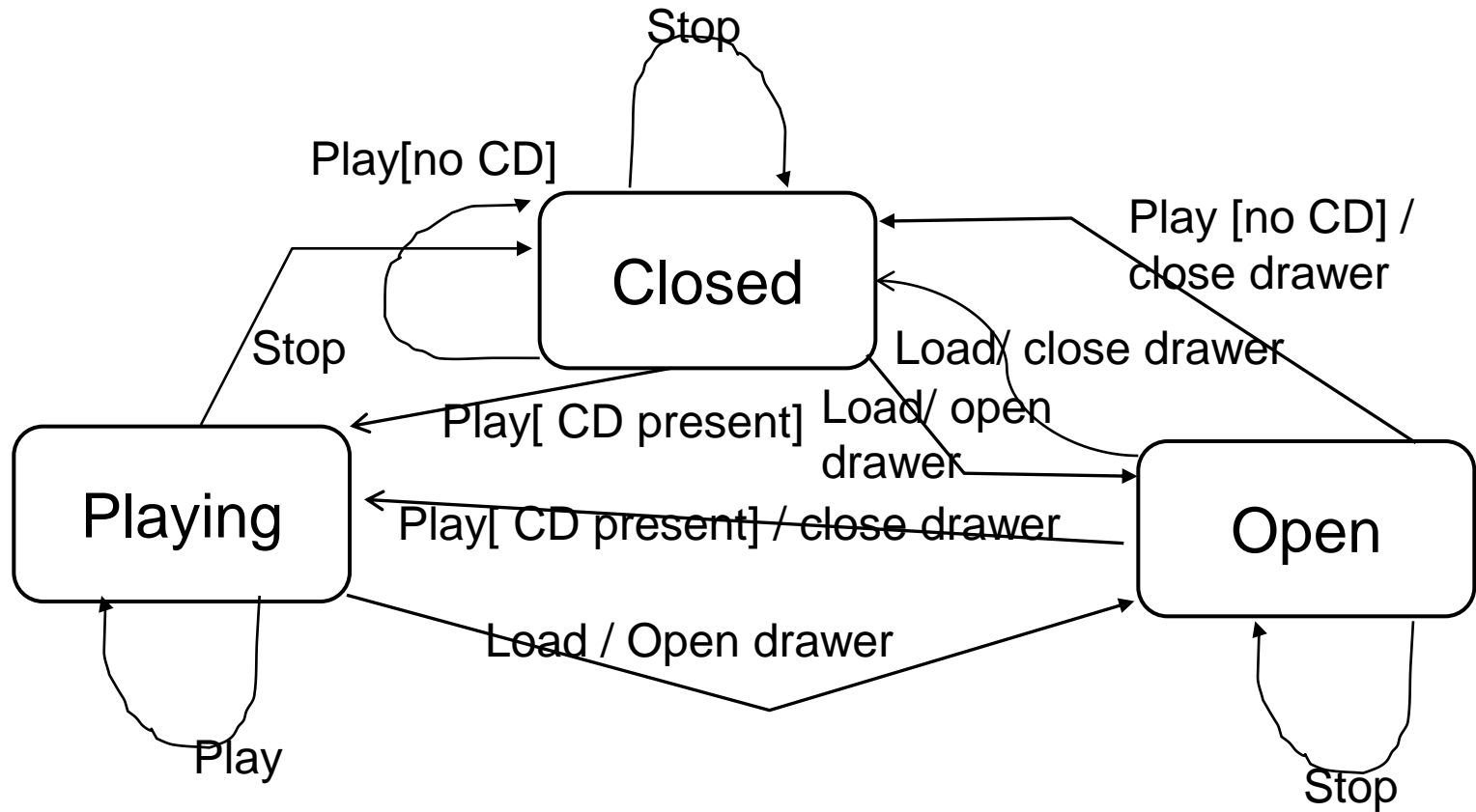
# Action

Action ditulis setelah nama event dengan diawali tanda slash (/). Action bisa dideskripsikan dalam pseudo-code atau dengan menggunakan notasi bahasa pemrograman yang akan dipakai. Transition dapat membawa condition dan action sekaligus.

Action dibuat dalam bentuk pendek, mengandung bagian kecil dari proses yang tidak butuh waktu lama. Karakteristik action harus bisa selesai sebelum transition mencapai state yang baru. Hal tersebut mengindikasikan bahwa action tidak dapat diinterupsi oleh event yang lain yang mungkin dideteksi oleh obyek, namun tetap harus bisa selesai dilakukan.

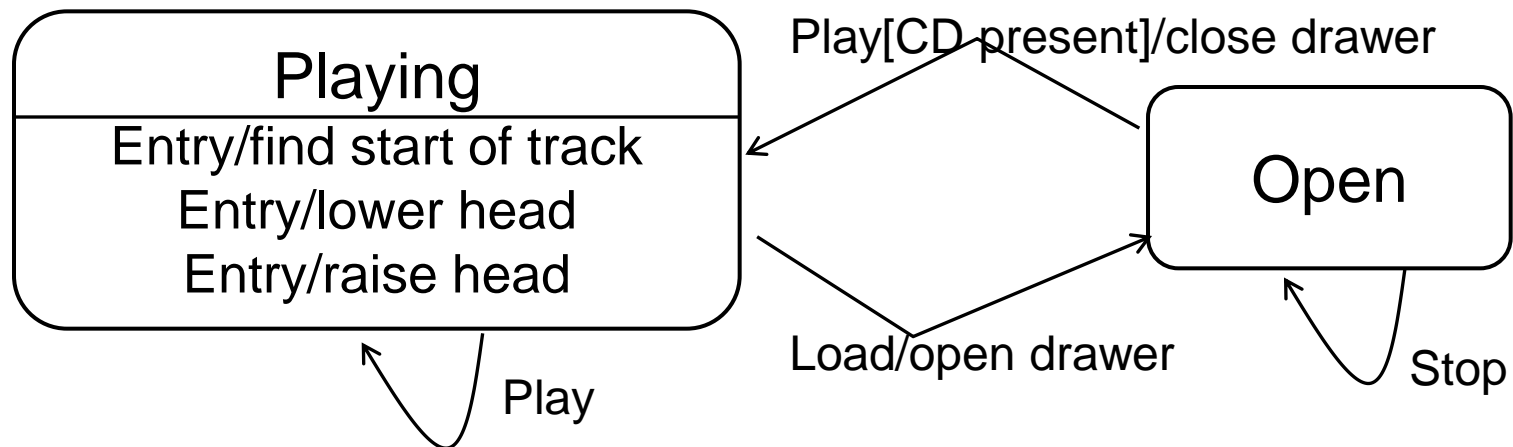


# Action (lanjutan)



# Entry dan Exit Action

Entry action dijalankan setiap saat jika state menjadi aktif, segera setelah action transition selesai. Sebagai contoh jika CD player dalam state open dan tombol play ditekan, drawer akan menutup dan transition play akan dijalankan. Hasilnya state playing menjadi aktif dan entry action pada state playing segera dilakukan. Exit action akan dijalankan kapanpun saat sebuah action ditinggalkan.



## Activity

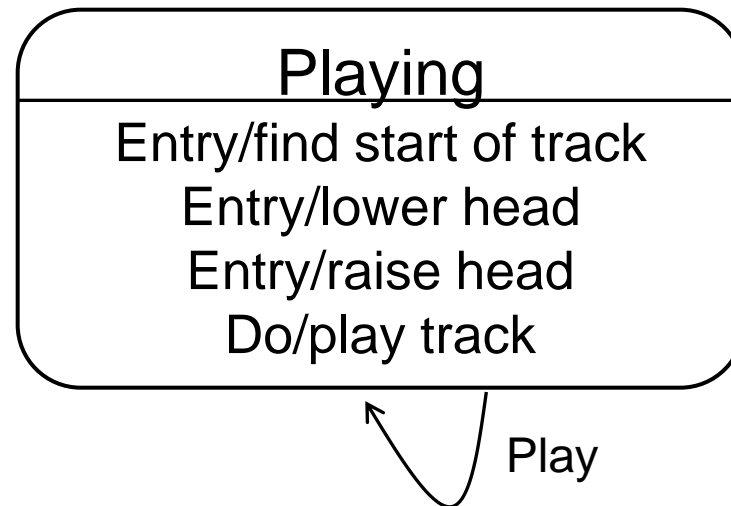
Saat posisi state playing, melakukan sesuatu yang disebut memainkan current track dari CD, operasi ini yang disebut activity. Seperti pada action maka activity ditulis dalam state yang diawali dengan **do**.

Beda antara action dan activity adalah action lebih merupakan instance sedangkan activity merupakan perluasan dari waktu.

Ketika state menjadi aktif, entry action dijalankan dan activitynya dimulai. Activity berjalan secara kontinyu sepanjang periode dimana state tersebut aktif.

## Activity (lanjutan)

Entry action harus dijalankan sampai selesai sebelum obyek dapat merespon event apapun. Akan tetapi activity bisa diinterupsi oleh event apapun yang mengakibatkan transition keluar.

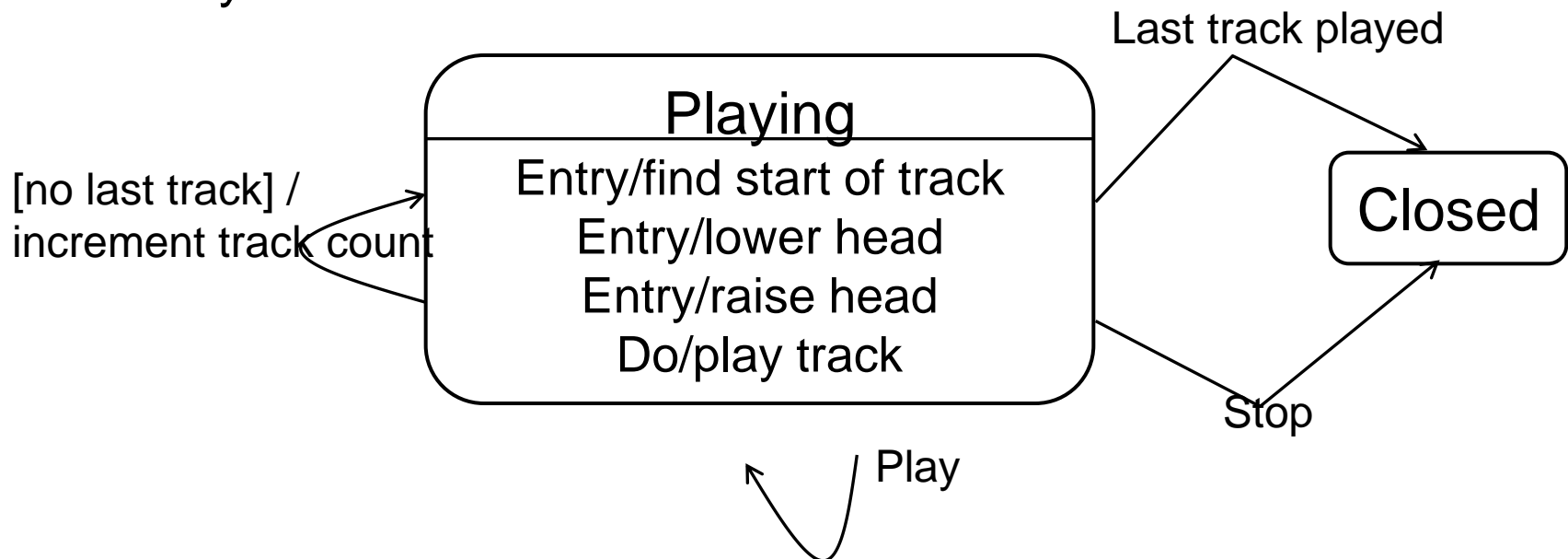


# Completion Transition

Completion transition adalah transition yang tidak mempunyai nama event. Ini bisa ditrigger ketika sebuah state activity internal berhenti secara normal tanpa interupsi oleh event dari luar. Ketika CD player dimainkan user bisa menekan tombol play atau stop untuk interupsi. Jika salah satu dari event ini terdeteksi maka track akan berakhir. Dengan demikian calon transition yang harus dijalankan tinggal completion transition. Yang terjadi berikutnya tergantung pada apakah track yang baru saja selesai adalah track terakhir atau bukan.

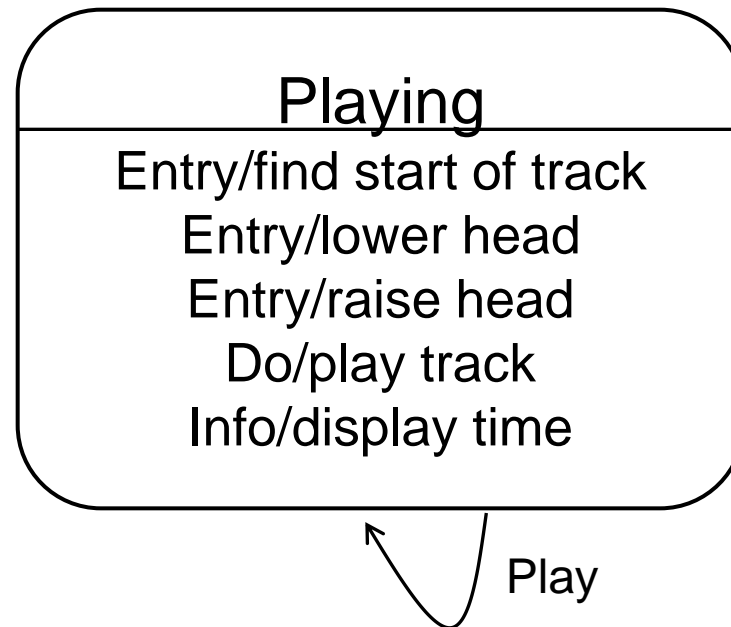
## Completion Transition (lanjutan)

Completion transition mempunyai guard condition untuk membedakan diantara dua kasus. Jika yang baru selesai adalah track terakhir, transition pada state closed akan dijalankan dan CD berhenti, tetapi jika tidak transition ke dirinya sendiri akan dijalankan dan penhitung track akan naik, state playing akan masuk kembali dan CD player akan menjalankan track berikutnya.



# Internal Transition

Internal transition ditulis didalam state yang diberi nama dengan event yang menyebabkannya. Sebagai contoh adalah informasi yang ditampilkan dalam CD player.



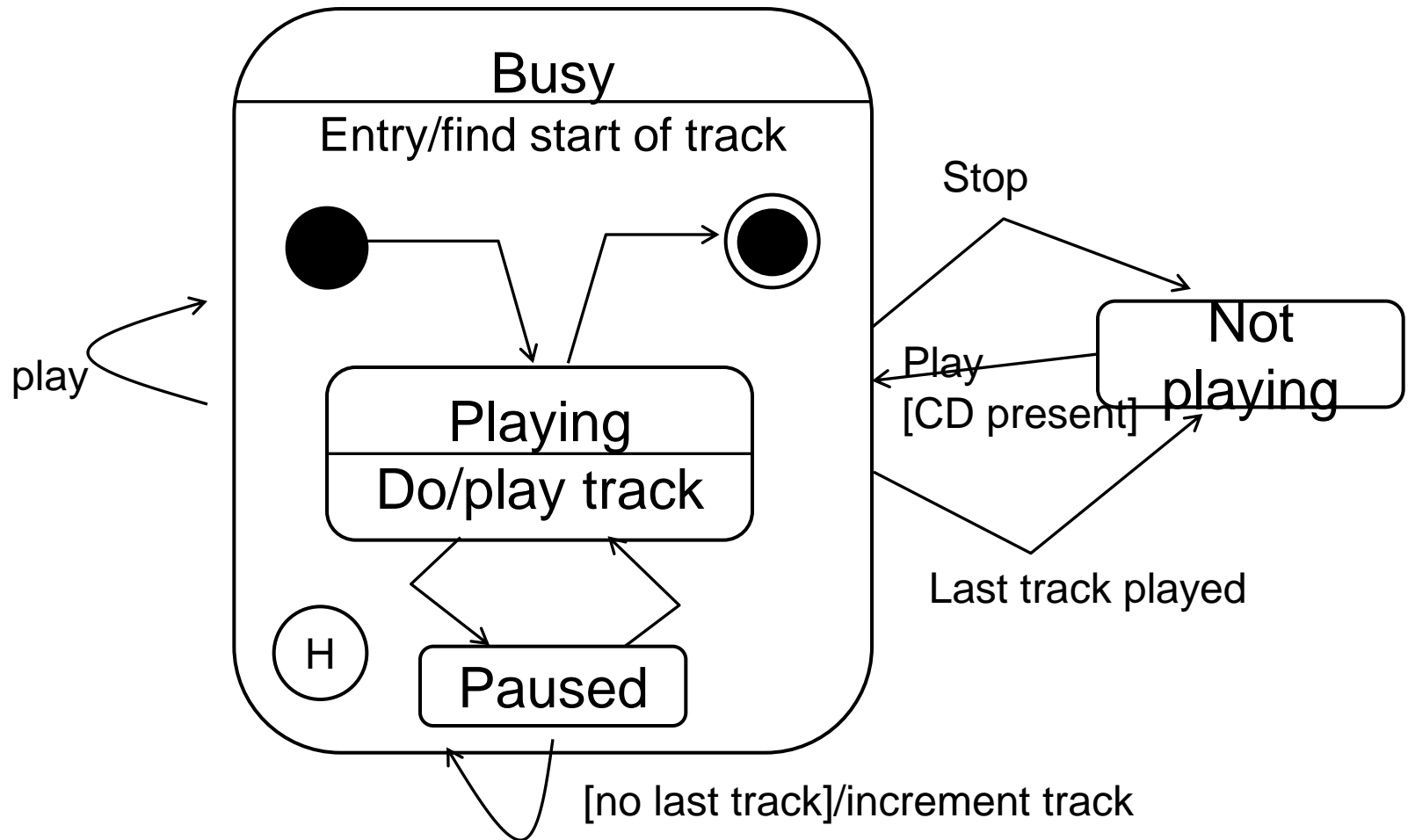
# History State

Ketika tombol pause ditekan, maka saat tombol play ditekan lagi CD player otomatis akan melanjutkan jalannya CD dari kondisi terakhir pada saat tombol pause ditekan dan tidak memulai dari awal lagi.

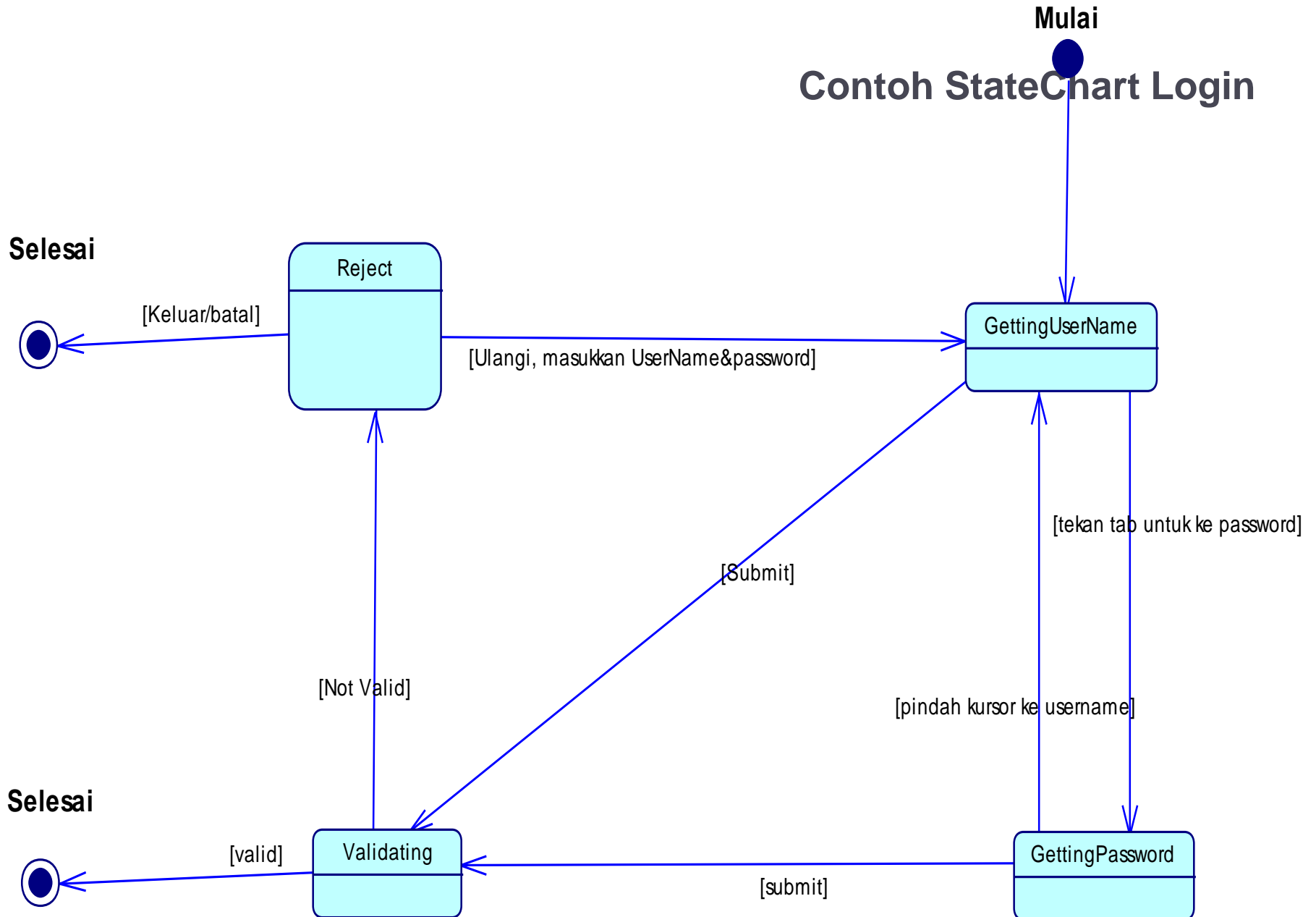
UML menangkap ide ini dengan menggunakan history state yang dinyatakan dengan huruf H dalam lingkaran. Cara kerja history state dilakukan dengan cara composite state mengingat sub state yang aktif saat obyek keluar dari transition composite state. Dengan adanya transition ke history state akan mengakibatkan substate yang saat ini aktif akan menjadi aktif lagi



# History State (lanjutan)



# Contoh StateChart Login

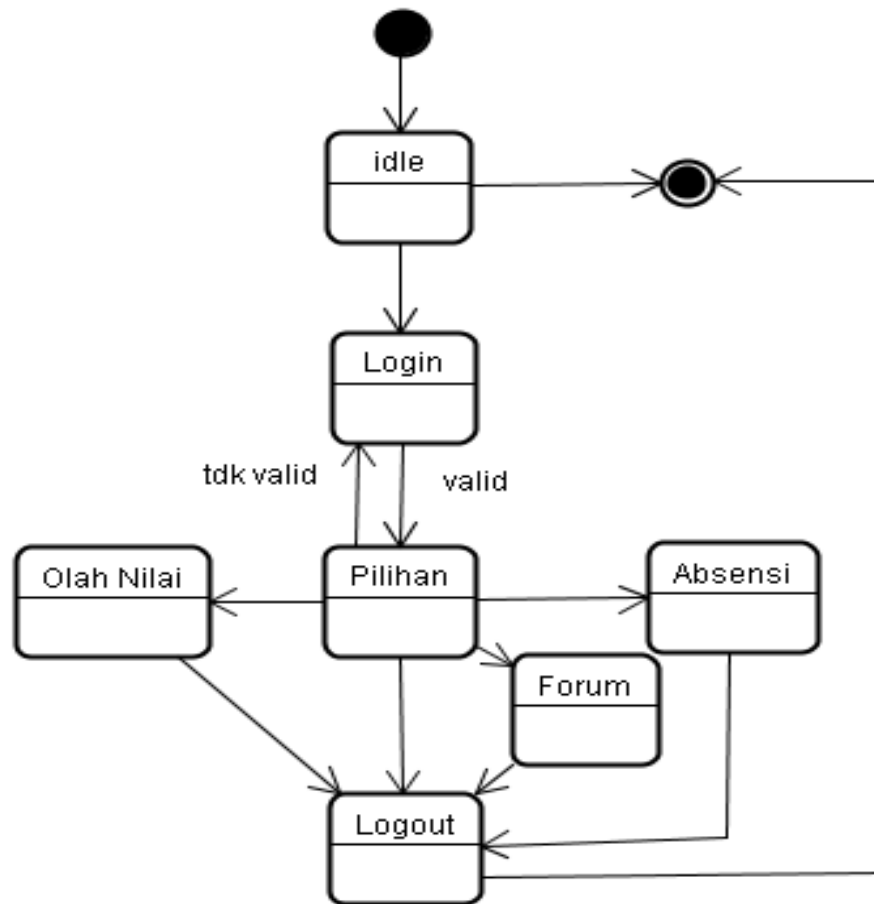


## Contoh soal

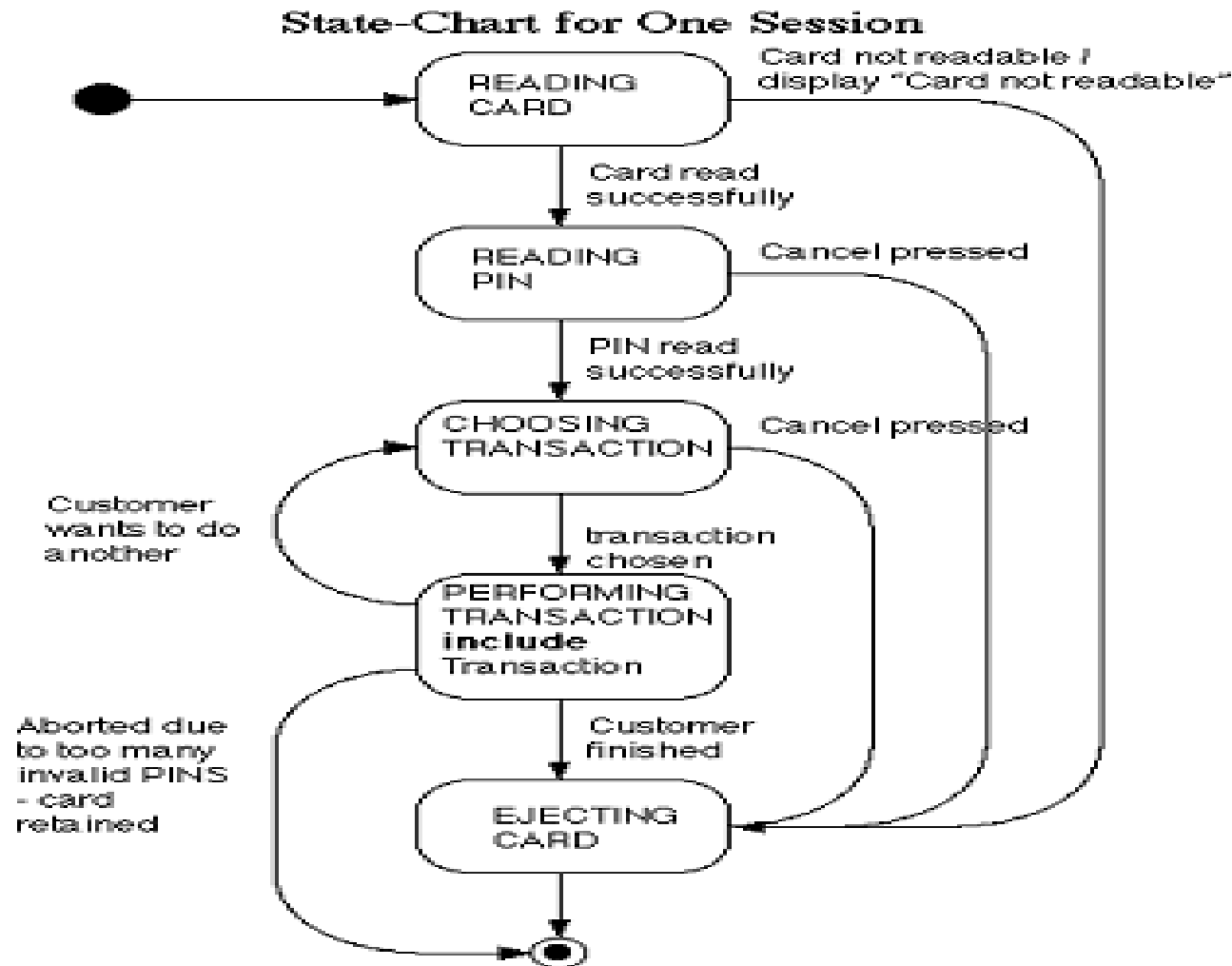
- ▶ Ketika dosen melakukan login ke sistem akan divalidasi user name dan password, jika valid akan muncul tampilan pilihan olah nilai, absensi, forum diskusi atau logout

# Jawaban

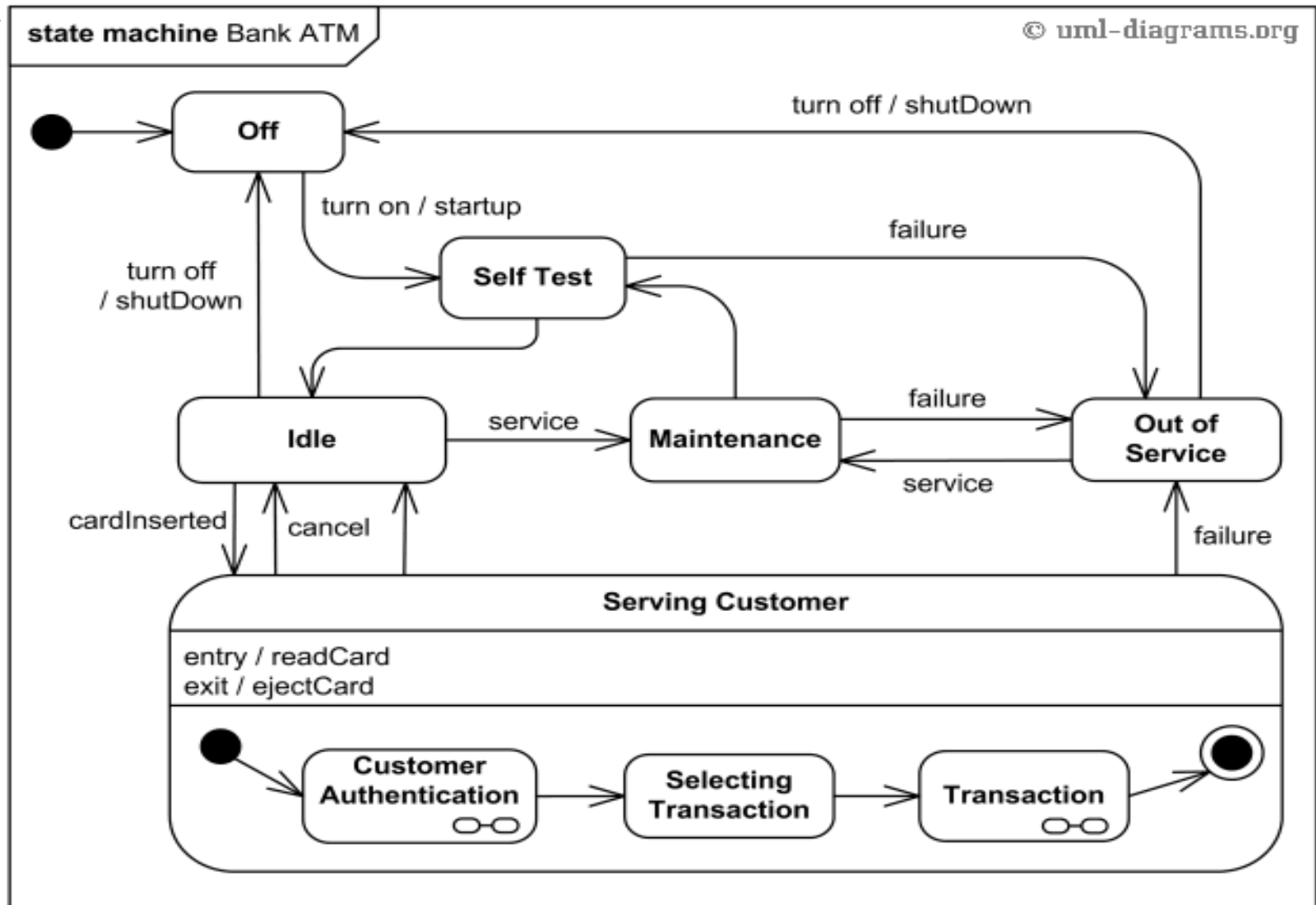
**stm** Statemachine Diagram Sisfo Poltek



# State chart diagram ATM Machine



# State chart diagram ATM Machine



# State chart diagram ATM Machine

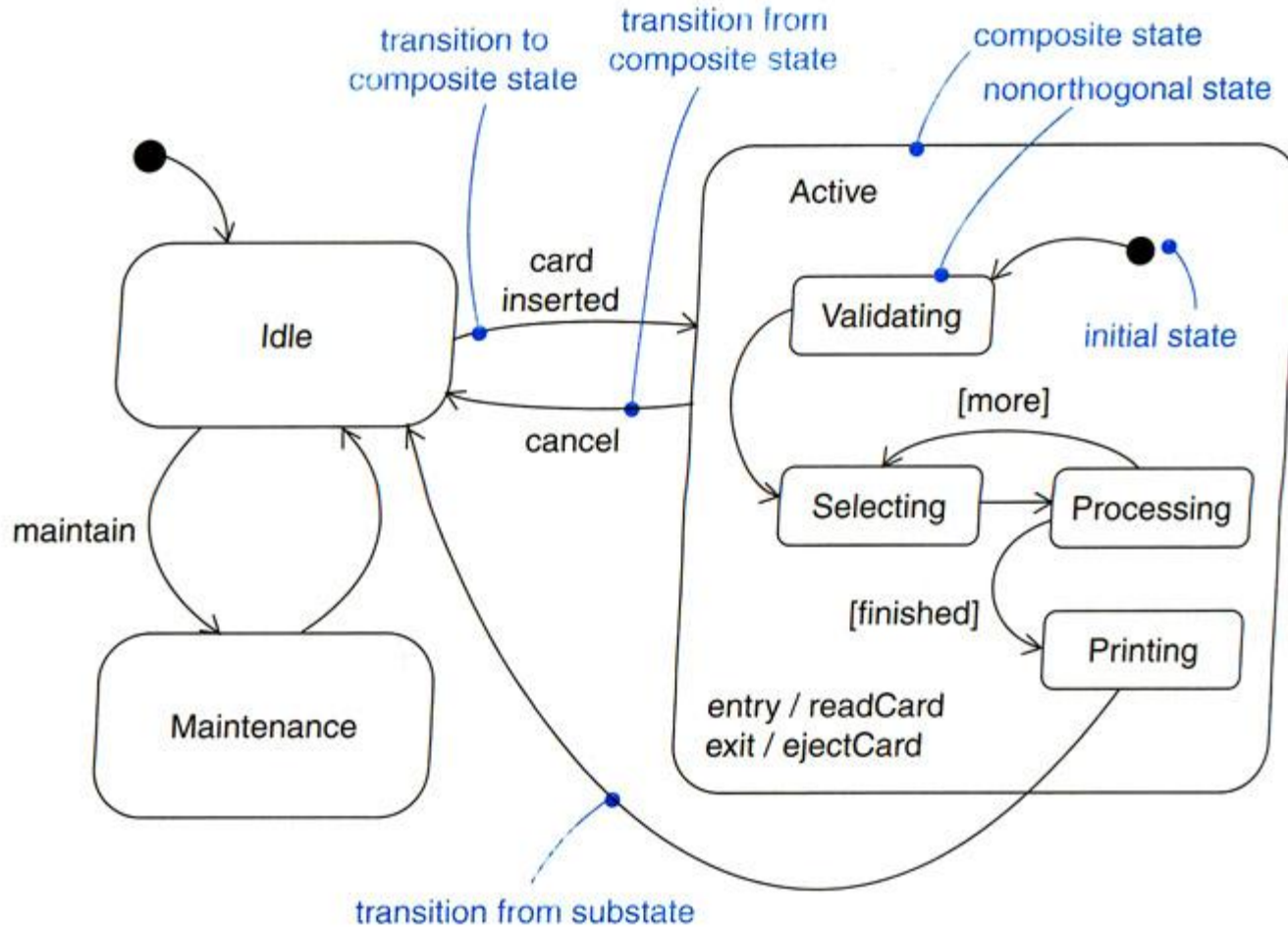
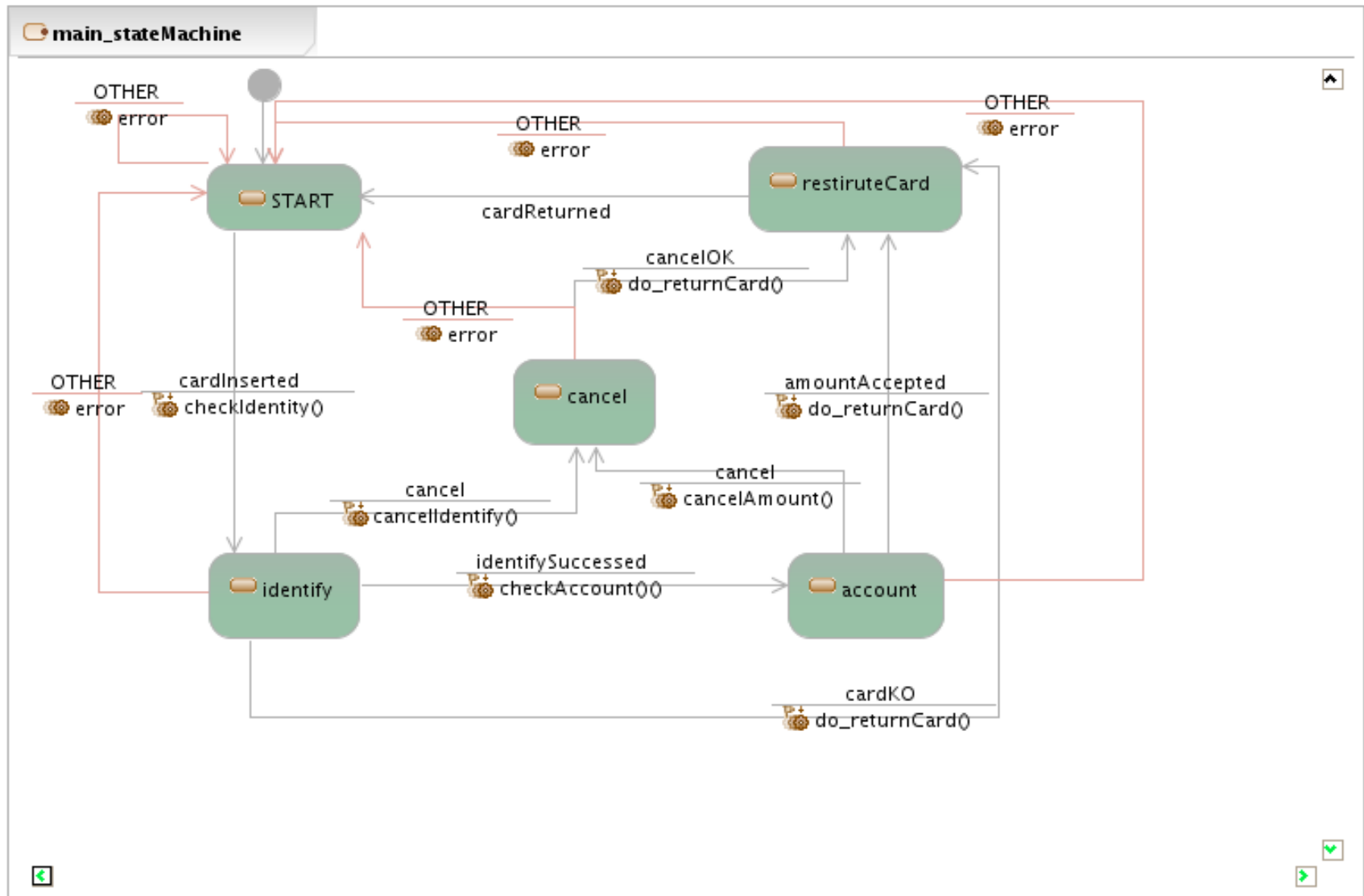


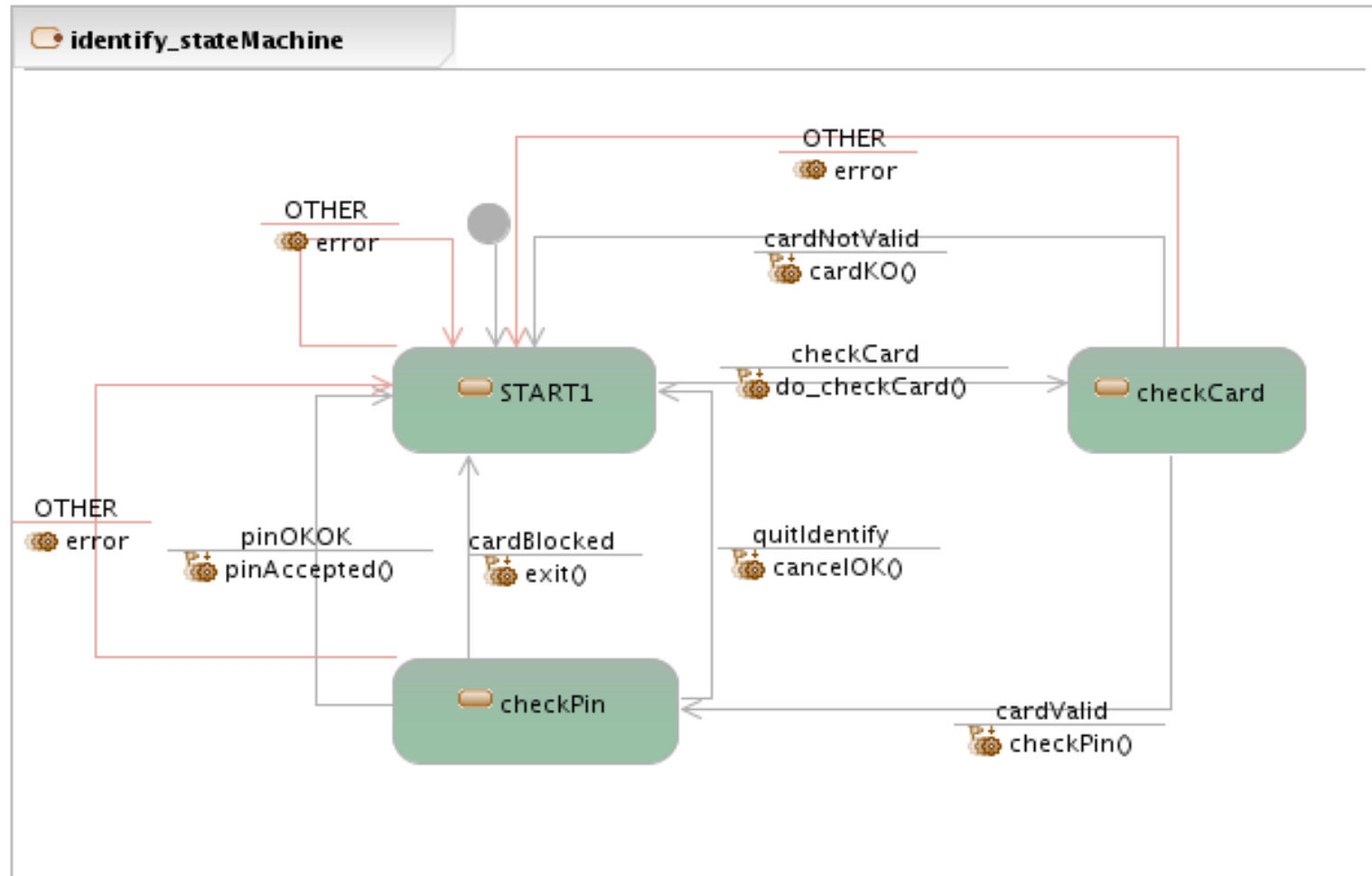
Figure 22-5: Sequential Substates

# State chart diagram ATM Machine

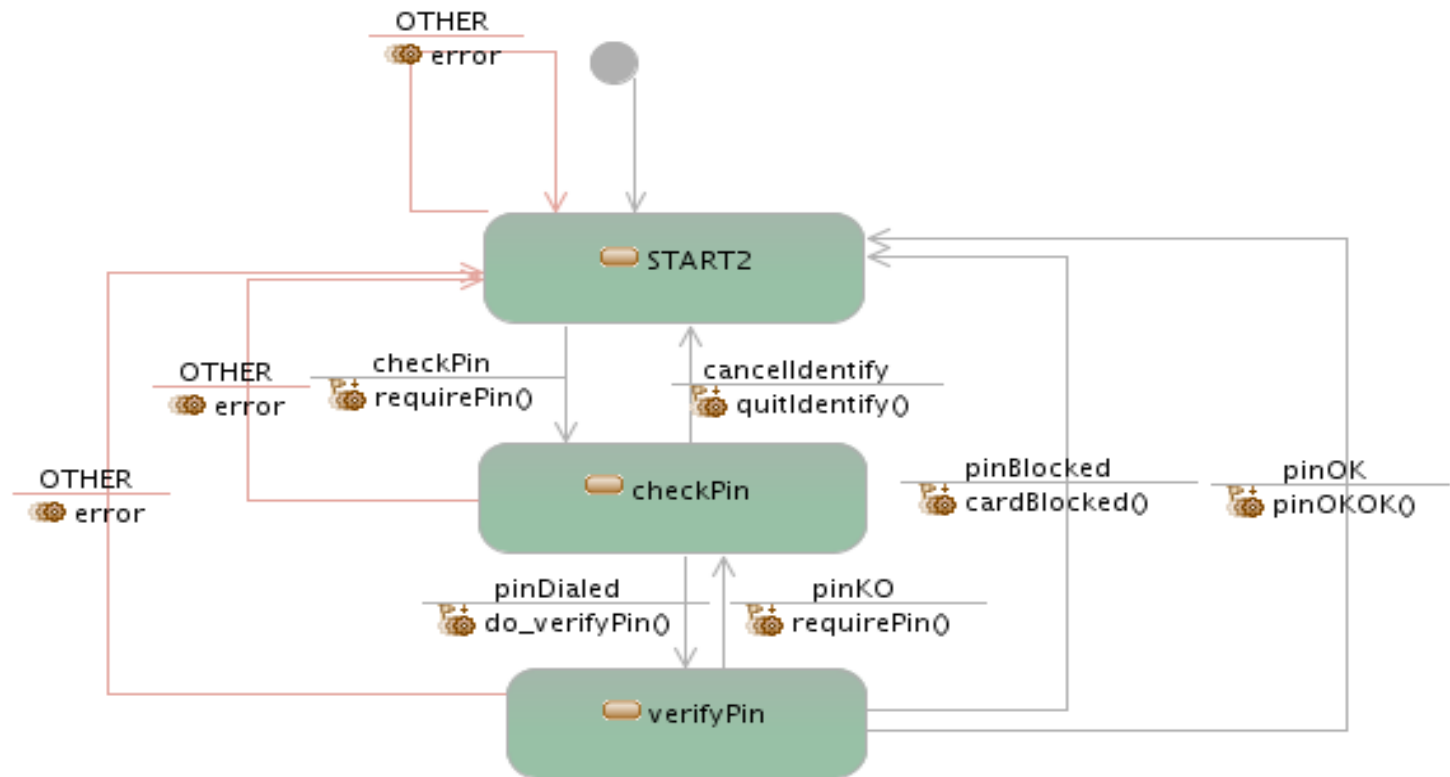


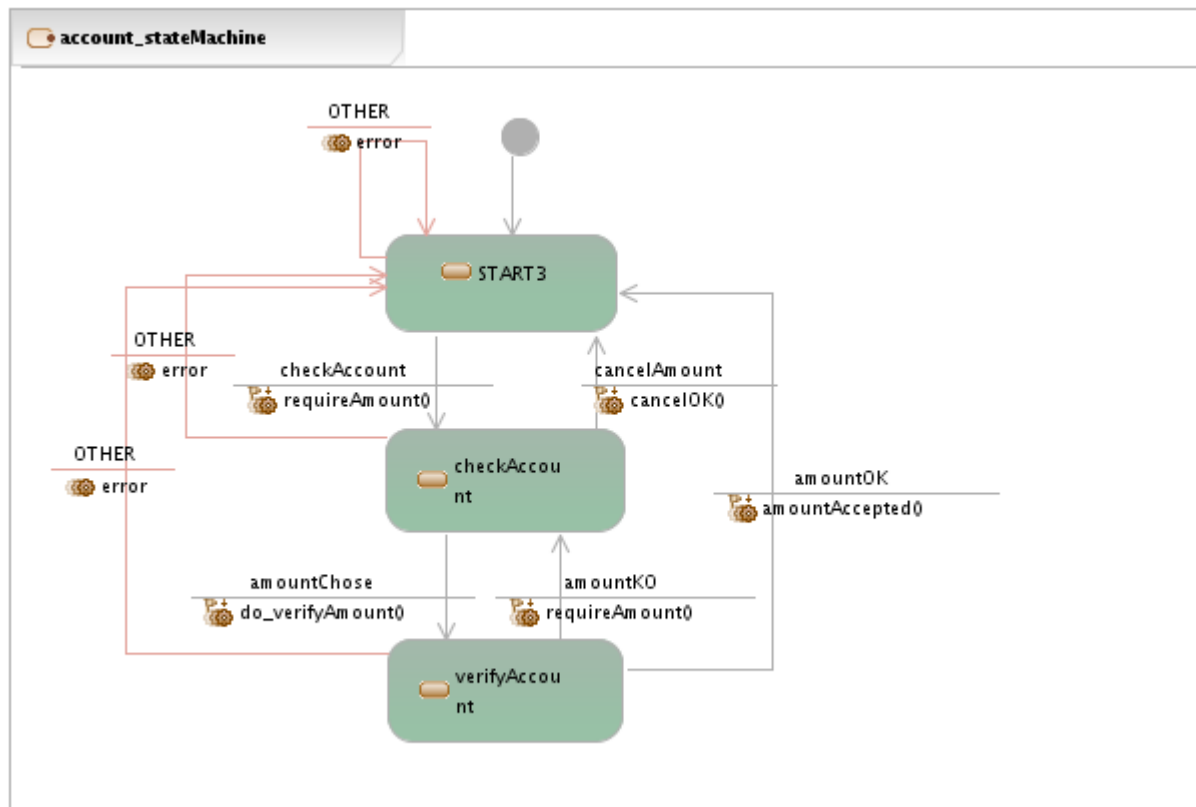


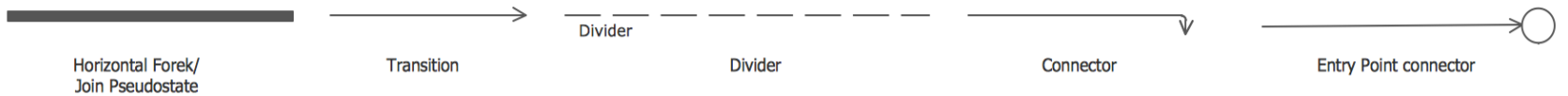
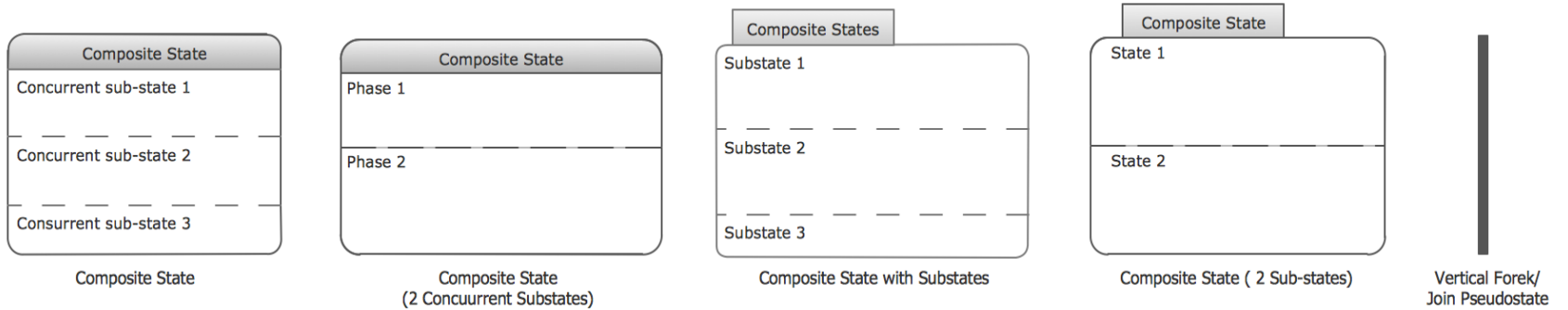
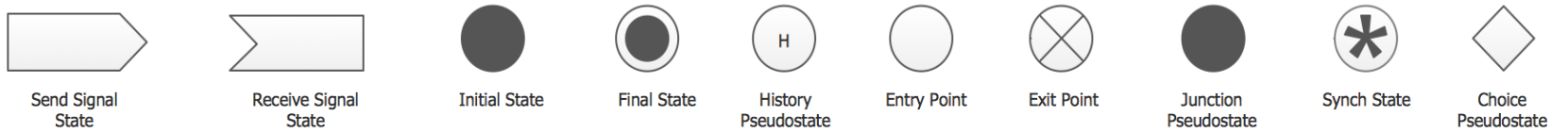
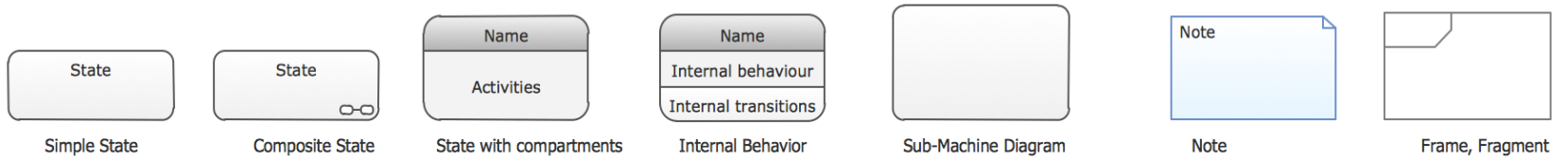
# State chart diagram ATM Machine

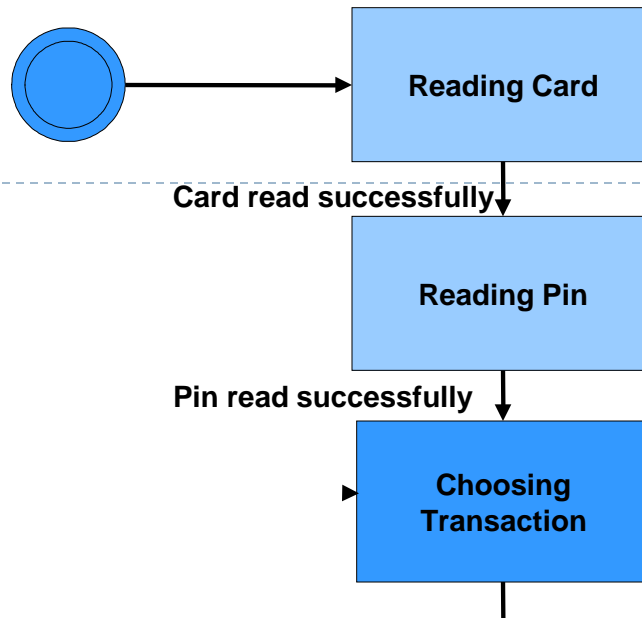


## pin\_stateMachine

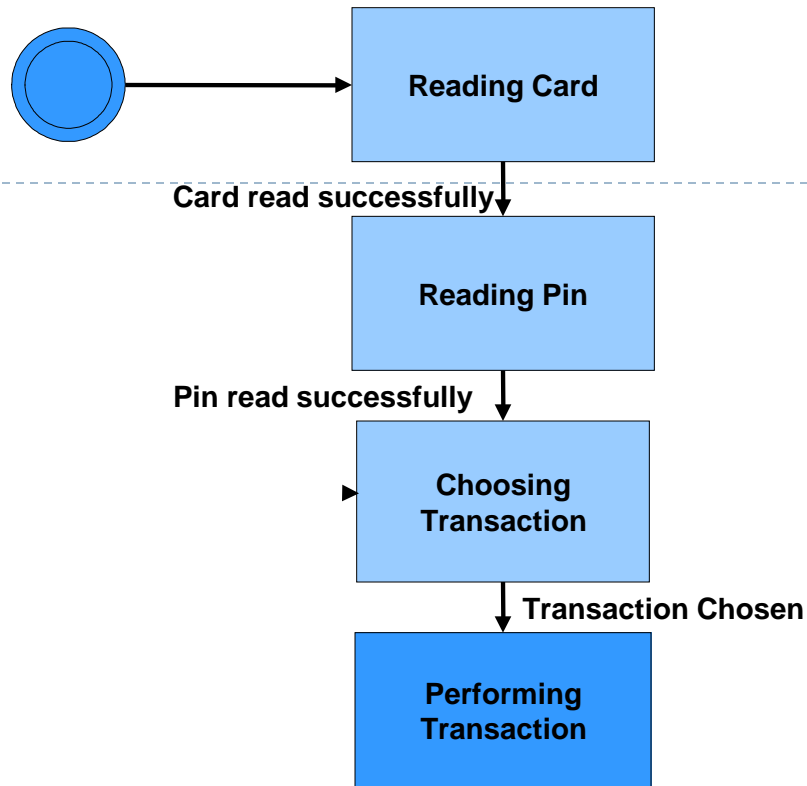








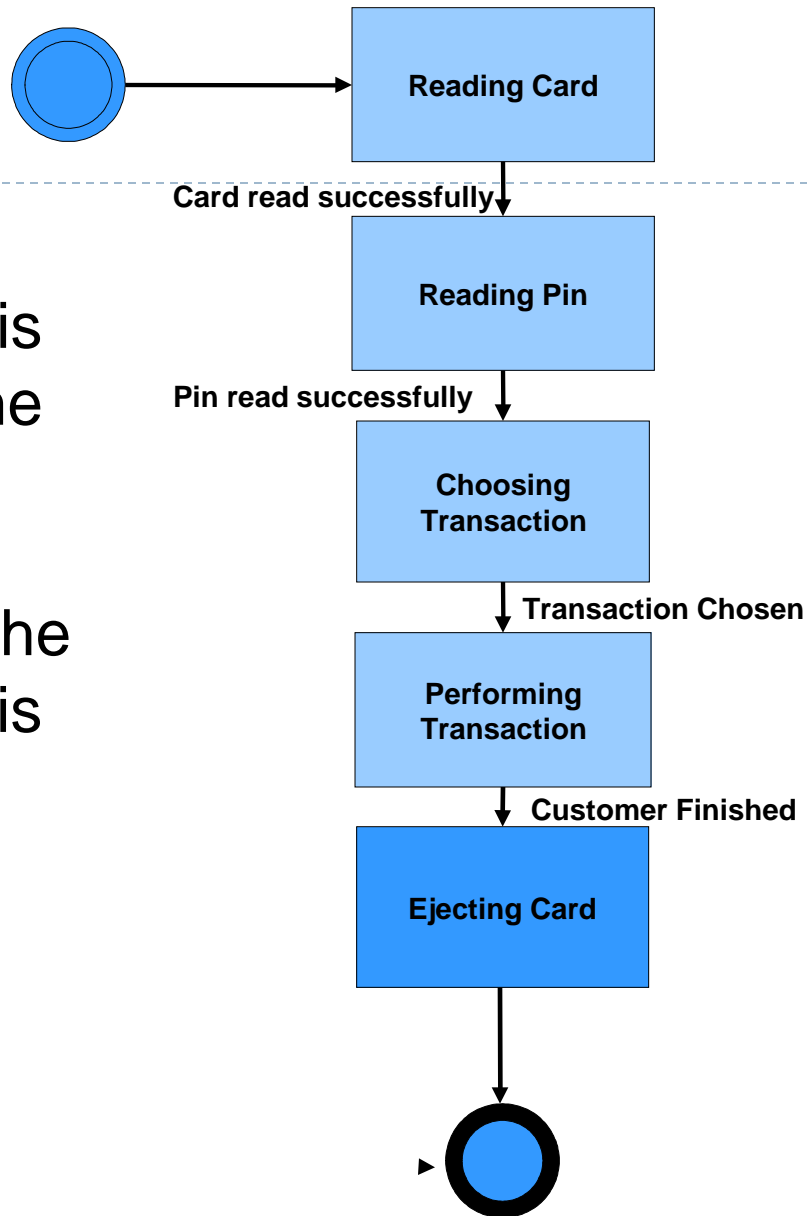
Then if the pin is read successfully then it the customers chooses transactions.

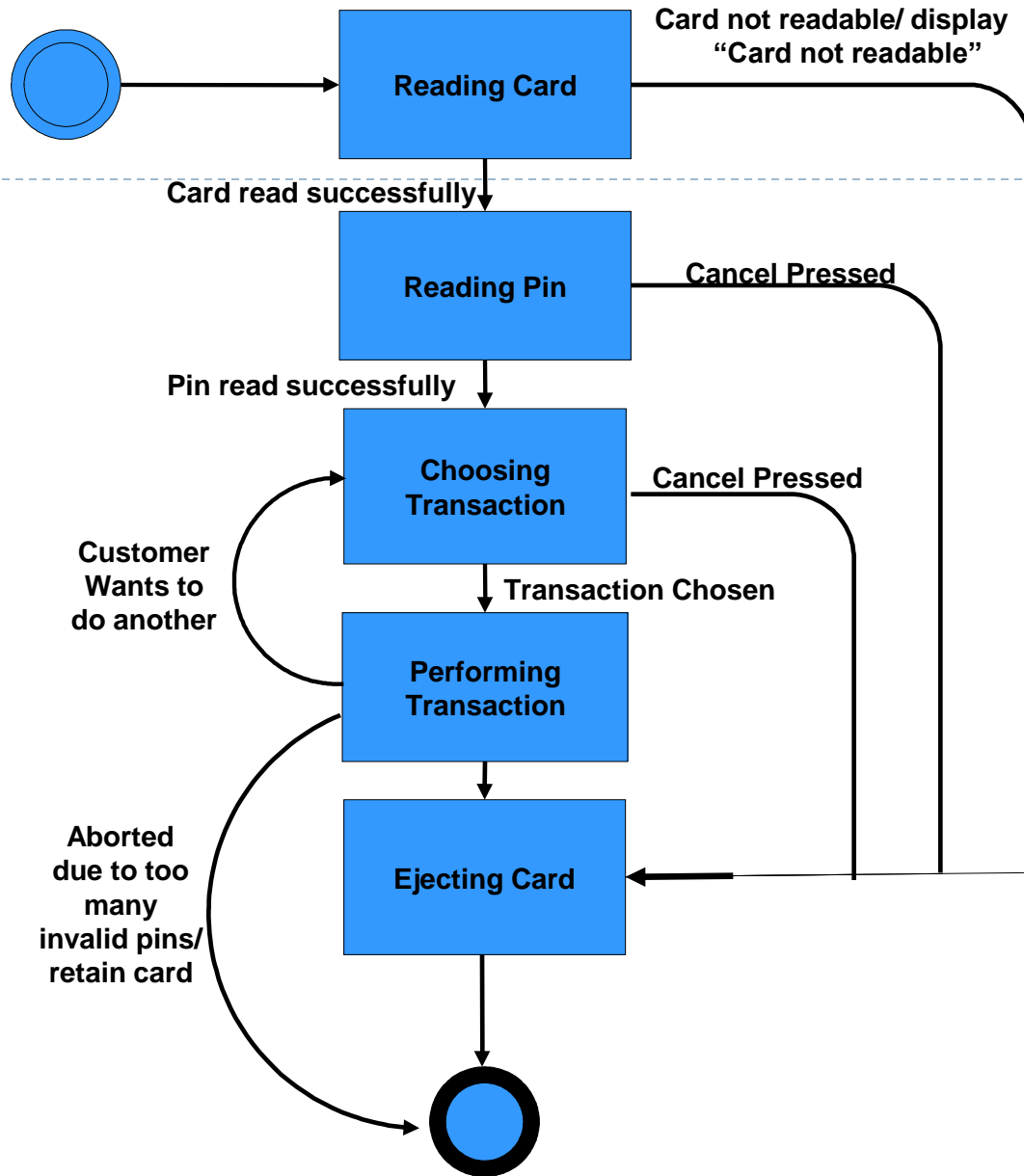


When the transaction is chosen then the machine performs the transaction.



If the customer is finished the machine ejects the card and the end state is achieved.





Other options provided if user does not complete transaction.

