**Login**  
Precondition 1 : User memasukkan Username

Precondition 2 :Username memasukkan Password .

Post- Condition: Login berhasil dan masukke halaman Awal.  
Algoritma:

def current\_user

return @current\_user if defined? @current\_user

@current\_user = current\_user\_session && current\_user\_session.user

End

**Register**Precondition 1: User Menginputkan Username,Email,Password,Password Confirmation(Mencocokan Dengan Password yang sebelum nya di inputkan agar mengurangi kesalahan dalam penulisan password atau typo),Alamat,Nomor Telfone.  
  
Post-Condition : Membuat akun di Database.  
  
Algoritma: def create

@user = User.new(user\_params)

respond\_to do |format|

if @user.save

format.html { redirect\_to dashboard\_path, notice: "User was successfully created." }

format.json { render :show, status: :created, location: @user }

else

format.html { render :new }

format.json { render json: @user.errors, status: :unprocessable\_entity }

end

end

end

Input(Username,email,password,passwordConfirmation,Alamat,NomorTelfon)

**Create Transaction**Pre Condition: Customer Memilih Barang barang dan submit barang yang akan di beli.

Post Condition: Customer Telah memilih barang yang akan di beli dan masuk ke database transaction

Algoritma: def create

customer = User.find(transaksi\_params[:user\_id])

require\_specific(customer)

@transaksi = Transaksi.new(transaksi\_params)

create\_transaction\_details if saved = @transaksi.save

respond\_to do |format|

if saved

format.html { redirect\_to @transaksi, notice: "Transaksi was successfully created." }

format.json { render :show, status: :created, location: @transaksi }

else

format.html { redirect\_to Produsen.find(transaksi\_params[:produsen\_id]) }

format.json { render json: @transaksi.errors, status: :unprocessable\_entity }

end

end

end