Made Narodeon Handika Pramesta / 103032300101

```
function inDegree (G : graph, V : adrVertex) -> integer
kamus
    Ve : adrVertex
    E : adrEdge
    degree : integer
Algoritma
    degree = 0
    Ve = G.first
    while Ve ≠ NIL do
        E = Ve -> firstEdge
        while E ≠ NIL do
            if E -> idVertex == V -> id then
                degree = degree + 1
            endif
            E = E -> next
        endwhile
        Ve = Ve -> next
    endwhile
~~end function~~
    return degree
end function
```

```
function outDegree (G: graph, v: adrVertex) -> integer
kamus
    S : adrVertex
    E : adrEdge
    degree : integer
Algoritma
    degree = 0
    S = G.first
    while S ≠ NIL AND S->id ≠ V->id do
        S = S->next
    endwhile
    E = S->firstEdge
    while E ≠ NIL do
        degree = degree + 1
        E = E->next
    endwhile
    return degree
endfunction
```

```
function degree (G : graph, V : adrVertex) → integer
kamus
    inDegree (graph, adrVertex)
    outDegree (graph, adrVertex)
Algoritma
    return inDegree(G, V) + outDegree(G, V)
endfunction

function isSimpleGraph (G : graph) → boolean
kamus
    V : adrVertex
    E : adrEdge
Algoritma
    V = G.first
    while V ≠ NIL do
        E = V→firstEdge
        while E ≠ NIL do
            if V→id == E→idVertex then
                return false
            endif
            E = E→next
        endwhile
        V = V→next
    endwhile
    return true
endfunction
```