PSEUDOCODE DJIKSTRA

```
function DIJKSTRA($s$)
    $dist \leftarrow$ new integer$[V+1]$
    $visited \leftarrow$ new boolean$[V+1]$
    $pred \leftarrow$ new integer$[V+1]$
    FILLARRAY($dist, \infty$)
    FILLARRAY($visited, false$)
    FILLARRAY($pred, -1$)

    $dist[s] \leftarrow 0$
    while $true$ do                                   ▷ Perulangan ini akan diakhiri dengan break
        $u \leftarrow -1$
        $minDist \leftarrow \infty$
        for $i \leftarrow 1, V$ do     ▷ Cari node yang belum dikunjungi dan memiliki $dist$ terkecil
            if (not $visited[i]$) $\wedge$ ($dist[i] < minDist$) then
                $u \leftarrow i$
                $minDist \leftarrow dist[i]$
            end if
        end for
        if $(u = -1) \vee$ ISINFINITE($dist[u]$) then
            break                                                   ▷ Akhiri perulangan while
        end if

        $visited[u] \leftarrow true$
        for $v \in adj(u)$ do                         ▷ Lakukan relax untuk semua tetangga $u$
            if $dist[v] > dist[u] + w[u][v]$ then
                $dist[v] = dist[u] + w[u][v]$
                $pred[v] = u$
            end if
        end for
    end while

    return $dist$                       ▷ Kembalikan tabel shortest path yang bermula dari $s$
end function
```

PSUEDOCODE KRUSKAL

```
function KRUSKAL($edgeList$)
    INITIALIZEDISJOINTSET()
    SORT($edgeList$)                                 ▷ Urutkan berdasarkan bobotnya
    for $\langle u, v \rangle \in edgeList$ do
        if not CHECK($u, v$) then
            $cost \leftarrow cost + w[u][v]$
            JOIN($u$, $v$)
        end if
    end for
    return $cost$
end function
```