**STUDI KASUS PENCARIAN BUTA (BLIND SEARCH)**

**MENGGUNAKAN ALGORITMA DFS**

Oleh : Muhamad Ro’uf Firnanda – 1855201028

**Studi Kasus :**

Menemukan jalur tercepat dari desa Kras ke desa Pelas

**Deskripsi Masalah :**

Seorang kurir ingin mengirimkan sebuah paket barang pesanan pembeli yang berada di sebuah desa yakni desa Pelas yang berada di paling pelosok. Dalam satu kecamatan ini terdapat 12 desa. Kurir ingin segera mengirim paket barangnya sesegera mungkin, kira-kira jalur mana yang harus dipilih untuk segera sampai ke desa Pelas ?

**Jawaban :**

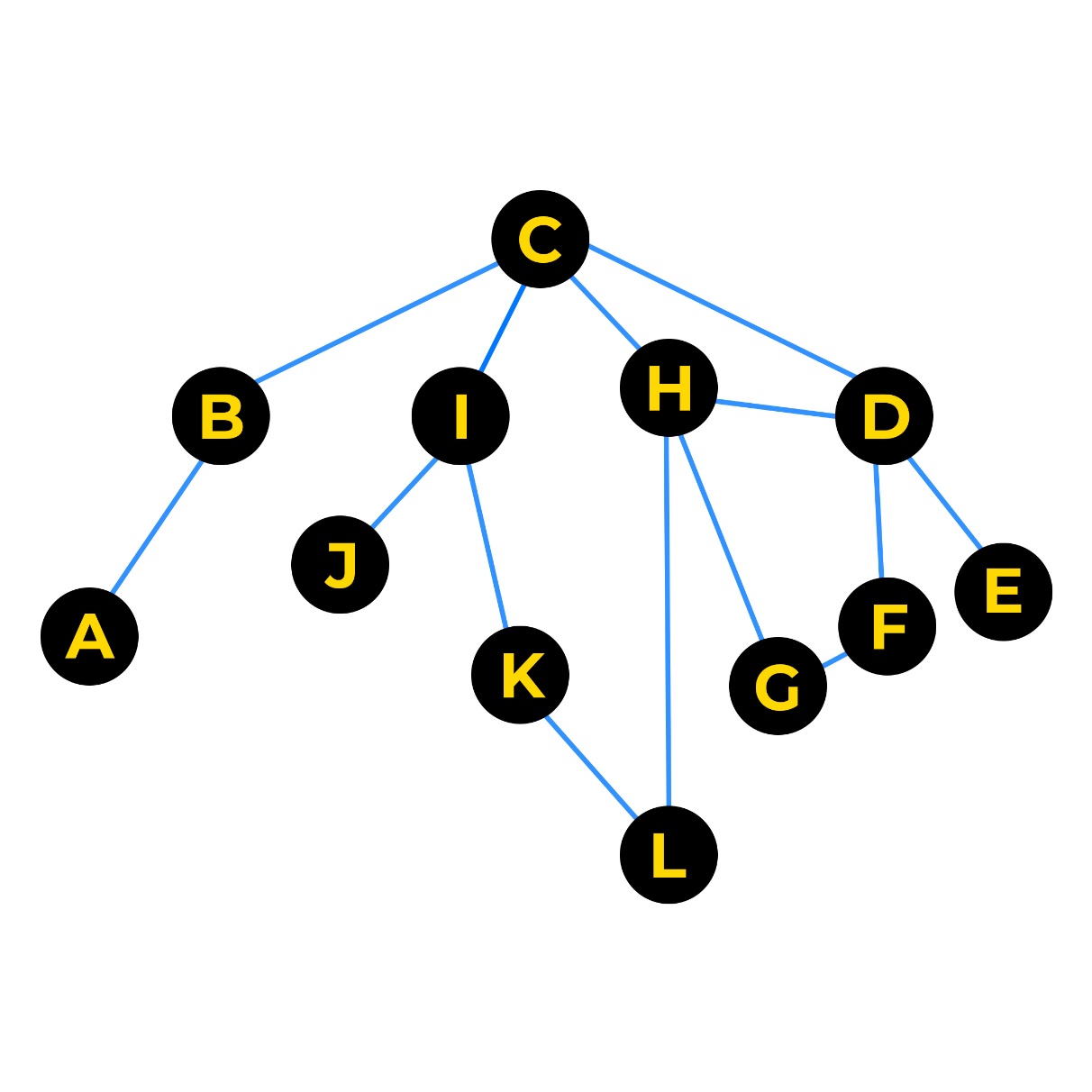
Start : Desa Kras (C)

Goal : Desa Pelas (K)

Stack : A, B, C, D, H, I, E, F, G, L, J, K

Result : A, B, C, D, H, I, E, F, G, L, J, K

**Impelementasi Ke Dalam Pohon Akar :**



Disini saya menggunakan Bahasa Phyton untuk pemrogamannya.

**Kode Phyton :**

peta={

'A':['B'],

'B':['A','C'],

'C':['D','H','I','B'],

'D':['E','F','G','H','C'],

'E':['D'],

'F':['G','H','D'],

'G':['H','D','F'],

'H':['D','G','L','C'],

'I':['C','K','J'],

'J':['I'],

'K':['L','I'],

'L':['K','H']

}

def dfs(peta, mulai, tujuan):

stack = [[mulai]]

visited = set()

while stack:

# Ambil jalur terakhir dari stack

jalur = stack.pop()

# Dapatkan node terakhir dari jalur

node = jalur[-1]

# Jika node belum dikunjungi

if node not in visited:

neighbours = peta[node]

# Telusuri semua neighbour

for neighbour in neighbours:

new\_jalur = list(jalur)

new\_jalur.append(neighbour)

stack.append(new\_jalur)

# Jika neighbour adalah tujuan, return jalur

if neighbour == tujuan:

return new\_jalur

# Tandai node sebagai sudah dikunjungi

visited.add(node)

# Jika tidak ada jalur yang ditemukan

return None

mulai = 'C'

tujuan = 'L'

jalur = dfs(peta, mulai, tujuan)

if jalur:

print(f"Jalur terpendek dari {mulai} ke {tujuan} adalah:", "->".join(jalur))

else:

print(f"Tidak ada jalur dari {mulai} ke {tujuan}")