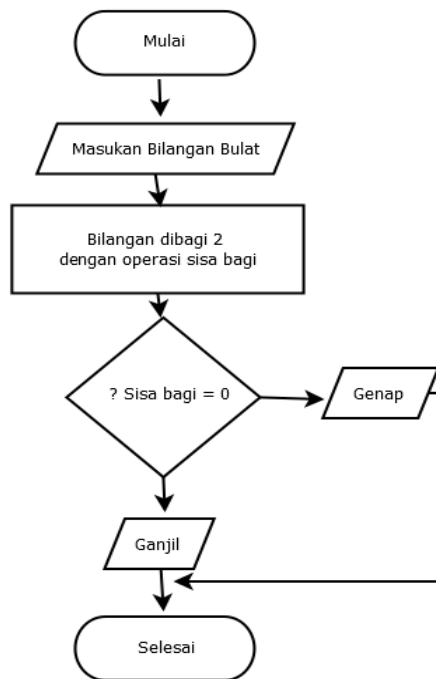


Nama : Abdul Kholik  
Nim : 200510011  
Kelas : D3 Teknik Informatika  
Fakultas : Teknik  
Matkul : Analisis dan Strategi Algoritma

## JAWABAN UTS ANALISIS & STRATEGI ALGORITMA

### 1. Diagram Alur



### 2. Jenis-Jenis Strategi Algoritma adalah :

#### 1) Strategi Solusi langsung

- ✓ Algoritma Brute Force
- ✓ Algoritma greedy

**Brute Force** adalah sebuah pendekatan yang langsung (straightforward) untuk memecahkan masalah, biasanya didasarkan pada pernyataan masalah (problem statenent) dan definisi konsep yang dilibatkan.

Algoritma Brute Force memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (obvious way).

**Algoritma greedy** merupakan jenis algoritma yang menggunakan pendekatan penyelesaian masalah dengan mencari nilai maksimum sementara pada setiap langkahnya. Nilai maksimum sementara ini dikenal dengan istilah local maximum. Pada kebanyakan kasus, algoritma greedy tidak akan menghasilkan solusi paling optimal, begitupun algoritma greedy biasanya memberikan solusi yang mendekati nilai optimum dalam waktu yang cukup cepat.

## 2) Strategi Berbasis Pencarian pada ruang status

- ✓ Teknik DFS (Depth First Search)
- ✓ Teknik BFS (Breadth First Search)
- ✓ Algoritma Backtracking
- ✓ Algoritma Brach and Bound

**DFS (Depth-First-Search)** adalah salah satu algoritma penelusuran struktur graf / pohon berdasarkan kedalaman. Simpul ditelusuri dari root kemudian ke salah satu simpul anaknya ( misalnya prioritas penelusuran berdasarkan anak pertama [simpul sebelah kiri] ), maka penelusuran dilakukan terus melalui simpul anak pertama dari simpul anak pertama level sebelumnya hingga mencapai level terdalam.

Setelah sampai di level terdalam, penelusuran akan kembali ke 1 level sebelumnya untuk menelusuri simpul anak kedua pada pohon biner [simpul sebelah kanan] lalu kembali ke langkah sebelumnya dengan menelusuri simpul anak pertama lagi sampai level terdalam dan seterusnya.

**Breadth-first search** adalah algoritma yang melakukan pencarian secara melebar yang mengunjungi simpul secara preorder yaitu mengunjungi suatu simpul kemudian mengunjungi semua simpul yang bertetangga dengan simpul tersebut terlebih dahulu. Selanjutnya, simpul yang belum dikunjungi dan bertetangga dengan simpulsimpul yang tadi dikunjungi , demikian seterusnya.

Jika graf berbentuk pohon berakar, maka semua simpul pada aras  $d$  dikunjungi lebih dahulu sebelum simpul-simpul pada aras  $d+1$ .

Algoritma ini memerlukan sebuah antrian  $q$  untuk menyimpan simpul yang telah dikunjungi. Simpul-simpul ini diperlukan sebagai acuan untuk mengunjungi simpul-simpul yang bertetangga dengannya. Tiap simpul yang telah dikunjungi masuk ke dalam antrian hanya satu kali. Algoritma ini juga membutuhkan table Boolean untuk menyimpan simpul yang telah dikunjungi sehingga tidak ada simpul yang dikunjungi lebih dari satu kali.

**Algoritma Backtracking** merupakan salah satu bentuk algoritma yang banyak digunakan oleh para programmer ataupun pengguna komputer ahli untuk menyelesaikan suatu permasalahan komputasional pada perangkat komputer yang mereka gunakan. Dalam programming algoritma backtracking, rekursi adalah kunci dari programming backtracking. Rekursi sendiri merupakan proses pengulangan suatu hal yang mencakup kesamaan-diri. Penggunaan yang paling umum dari rekursi terdapat dalam kajian ilmu matematika dan ilmu komputer.

**Algoritma B&B (Branch and Bound)** adalah salah satu algoritma yang digunakan untuk pencarian jalur. Contoh yang dibahas kali ini adalah mengenai pencarian jalur yang melalui semua titik dengan biaya terendah.

Algoritma ini memiliki 2 prinsip, yaitu:

Algoritma ini akan melakukan perhitungan secara rekursif, akan memecah masalah kedalam masalah-masalah kecil, sambil tetap menghitung nilai terendah / terbaik. Proses ini dinamakan branching

Jika branching diterapkan secara sendirian, maka hasilnya akan tetap mencari setiap kemungkinan yang ada. Untuk meningkatkan performa, algoritma ini akan melakukan pencatatan biaya minimum sebagai bound dalam setiap perhitungan, sehingga untuk calon hasil jawaban yang diperkirakan akan

melebihi bound akan dibuang karena tidak mungkin akan mencapai nilai terbaik.

### 3) Strategi solusi atas – bawah

✓ Algoritma Divide and Conquer

**Algoritma Divide and Conquer** merupakan algoritma yang sangat populer di dunia Ilmu Komputer. Divide and Conquer merupakan algoritma yang berprinsip memecah-mecah permasalahan yang terlalu besar menjadi beberapa bagian kecil sehingga lebih mudah untuk diselesaikan. Langkah-langkah umum algoritma Divide and Conquer :

Divide : Membagi masalah menjadi beberapa upa-masalah yang memiliki kemiripan dengan masalah semula namun berukuran lebih kecil ( idealnya berukuran hampir sama ).

Conquer : Memecahkan ( menyelesaikan ) masing-masing upa-masalah ( secara rekursif ).

Combine : Menggabungkan solusi masing-masing upa-masalah sehingga membentuk solusi masalah semula.

Objek masalah yang di bagi adalah masukan (input) atau instances yang berukuran n: tabel (larik), matriks, dan sebagainya, bergantung pada masalahnya. Tiap-tiap upa-masalah mempunyai karakteristik yang sama (the same type) dengan karakteristik masalah asal, sehingga metode Divide and Conquer lebih natural diungkapkan dalam skema rekursif. Sesuai dengan karakteristik pembagian dan pemecahan masalah tersebut, maka algoritma ini dapat berjalan baik pada persoalan yang bertipe rekursif (perulangan dengan memanggil dirinya sendiri). Dengan demikian, algoritma ini dapat diimplementasikan dengan cara iteratif ( perulangan biasa ), karena pada prinsipnya iteratif hampir sama dengan rekursif. Salah satu penggunaan algoritma ini yang paling populer adalah dalam hal pengolahan data yang bertipe array ( elemen larik ). Mengapa ? Karena pengolahan array pada

umumnya selalu menggunakan prinsip rekursif atau iteratif. Penggunaan secara spesifik adalah untuk mencari nilai minimal dan maksimal serta untuk mengurutkan elemen array. Dalam hal pengurutan ini ada empat macam algoritma pengurutan yang berdasar pada algoritma Divide and Conquer, yaitu merge sort, insert sort, quick sort, dan selection sort. Merge sort dan Quick sort mempunyai kompleksitas algoritma  $O(n^2 \log n)$ . Hal ini lebih baik jika dibandingkan dengan pengurutan biasa dengan menggunakan algoritma brute force.

#### 4) Strategi bawah – atas

- ✓ Algoritma Dinamic programming

**Dynamic Programming** dapat didefinisikan sebagai suatu pendekatan matematik yang memiliki prosedur sistematis yang dirancang sedemikian rupa dengan tujuan untuk mengoptimalkan penyelesaian suatu masalah tertentu yang diuraikan menjadi sub-sub masalah yang lebih kecil yang terkait satu sama lain dengan tetap memperhatikan kondisi dan batasan permasalahan tersebut.

#### 5) Strategi Shortest Path

- ✓ Algoritma Djikstra
- ✓ Algoritma Floyd-Warshall

**Algoritma Dijkstra** merupakan sebuah algoritma yang dipakai untuk memecahkan permasalahan jarak terpendek (shortest path problem) dalam sebuah graf berarah (directed graph) dengan bobot-bobot sisi (edge weights) yang bernilai tak-negatif.

**Algoritma Floyd-Warshall** adalah sebuah algoritma analisis graf untuk mencari bobot minimum dari graf berarah. Algoritma Floyd-Warshall adalah matriks hubung graf berarah berlabel, dan keluarannya adalah path terpendek dari semua titik ke titik yang lain.

Dalam usaha untuk mencari path terpendek, Algoritma Warshall memulai iterasi dari titik awalnya kemudian memperpanjang path dengan mengevaluasi titik demi titik hingga mencapai titik tujuan dengan jumlah bobot yang seminimum mungkin

3. Permasalahan, Permasalahan dari gambar tersebut adalah knapsack yang tidak dapat menampung semua barang tersebut karena bila dijumlahkan bobot semua barangnya adalah 27 kg yang akan melebihi kapasitas dari knapsack tersebut yang hanya bisa menampung bobot sebesar 17 kg.

Diketahui, Dari gambar tersebut diketahui ada satu knapsack yang bisa menampung 17 kg, lalu ada 5 barang yang masing2 memiliki bobot dan harga yang berbeda, yaitu barang abu 7kg dengan harga \$5, barang biru 5kg dengan harga \$2, barang hijau 2kg dengan harga \$2, barang hijau 1kg dengan harga \$1, dan barang abu 12kg dengan harga \$10

Solusi hingga mencapai keuntungan yang maksimal, untuk mendapatkeuntungan lebih maka dengan memilih barang abu 12kg dengan harga \$10 dan barang biru 5kg dengan harga \$2. Dengan memilih kedua barang tersebut kita bisa memilih barang dengan harga paling sedikit dan bobot yang tidak melebihi batas maksimal dari knapsack tersebut

#### 4. Penyelesaian TSP

Diketahui :

Simpul Awal = A

Simpul Akhir = A

Ditanyakan :

Rute Terpendek dan Terjauh ?

Penyelesaian :

NO	Rute(tour)	Bobot	Hasil
1	a-b-c-f-e-d-a	12+16+8+16+14+10	76
2	a-b-f-c-e-d-a	12+17+8+18+14+10	79
3	a-e-f-c-b-d-a	21+16+8+16+19+10	90
4	a-e-c-f-b-d-a	21+18+8+17+19+10	93
5	a-d-e-f-c-b-a	10+14+16+8+16+12	76
6	a-d-b-c-f-e-a	10+19+16+8+16+21	90
7	a-d-b-f-c-e-a	10+19+17+8+18+21	93

Hasilnya sebagai Berikut :

Rute terpendek

a-b-c-f-e-d-a = 76

a-d-e-f-c-b-a = 76

**Rute terpanjang**

a-e-c-f-b-d-a = 93

a-d-b-f-c-e-a = 93

5. Jurnal ini mempunyai latar belakang berdasarkan Pengelolaan perangkat jaringan seperti switch, router, dan link yang menjadi salah satu tantangan dalam memperbaiki performansi Quality of Service (QoS) suatu jaringan yang berskala besar. Untuk memudahkan pengelolaan perangkat tersebut, dibutuhkan ekosistem yang berbasis virtual. Tujuannya agar pengelola jaringan dapat fokus dalam memperbaiki performansi QoS suatu jaringan. Sebagai solusi dari permasalahan tersebut, dapat digunakan konsep arsitektur Software Defined Networking (SDN).

penelitian ini membandingkan dua algoritma yaitu algoritma dijkstra dengan algoritma BFS dalam routing saat packet forwarding

Permasalahan, Menganalisis performansi QoS pada packet forwarding menggunakan IPF-Dijkstra dan IPF-BFS di suatu arsitektur SDN

Metode Penelitian Metode penelitian yang dilakukan dalam mengerjakan penelitian ini adalah metode penelitian dengan flowchart yaitu metode penyelesaian masalah di mana simbol-simbol yang digunakan untuk mewakili operasi, data, aliran, peralatan, objek dan panah digunakan untuk menunjuk aliran data berurutan dari satu ke yang lainnya.

Proses pertama ialah merancang topologi jaringan yang digunakan dalam penelitian pada arsitektur SDN. Setelah itu membuat aplikasi penerapan algortima routing dijkstra dan BFS yang dinamakan IPFBFS dan IPF-Dijkstra. Aplikasi tersebut digunakan dalam

menerapkan algoritma routing dijkstra dan BFS saat packet forwarding. Kemudian perlu dilakukan perancangan skenario pengujian saat menguji QoS pada packet forwarding. Proses yang terakhir yaitu menentukan data-data performansi yang diujikan pada hasil pengiriman paket yang dilakukan

Hasil Penelitian Pada pengukuran performansi QoS saat packet forwarding menggunakan algoritma BFS dan dijkstra dapat disimpulkan bahwa pada topologi jaringan ring, algoritma routing yang tepat dalam packet forwarding yaitu algoritma Dijkstra