

Final Project Struktur Data

Sistem *Tracing* dan *Tracking* Virus Berbasis Tempat dan Orang

Daanii Nabil Ghinannafsi Kusnanta 05111940000163

Struktur Data C, Teknik Informatika, ITS, Surabaya

1. LATAR BELAKANG

Pandemi CoViD-19 masih melanda dunia dan belum menunjukkan tanda akan usai. Sementara itu, jumlah kematian dan orang yang tertular terus bertambah. Di Indonesia jumlah orang terkonfirmasi tertular virus SARS-CoV-2 ini mencapai 22.750 orang dengan puncak pertambahan harian 973 kasus.

Pengembangan vaksin sedang dilakukan di berbagai negara. Namun, vaksin diperkirakan baru tersedia pada awal tahun 2021. Dengan begitu, untuk menangkai penyebaran virus digunakan 'imunitas buatan' yaitu dengan mengurangi mobilitas setiap orang.

Akan tetapi, masih saja ada orang yang tidak mematuhi aturan pembatasan ini. Orang-orang masih banyak yang abai dengan masih bepergian dan mengumpulkan massa. Sehingga, penelusuran dan pelacakan virus yang berbasis hubungan perseorangan dan riwayat tempat yang dikunjungi harus dilakukan dengan cepat dan tepat.

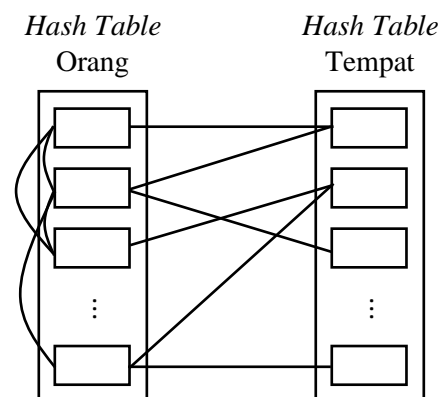
2. ANALISIS

Sistem pelacakan dan penelusuran ini menganalisis hubungan atau kontak seseorang dan juga riwayat tempat yang dikunjungi. Untuk itu struktur data yang tepat digunakan adalah graf.

Selain itu pencarian orang atau tempat membutuhkan waktu cepat dan tidak perlu melakukan penelusuran satu-persatu. Untuk menyiasati hal ini, daftar orang dan tempat dimasukkan dalam tabel *hash* yang

dapat diakses dengan waktu singkat menggunakan fungsi *hash*.

Struktur graf akan terdiri dari 2 *array* yang menyimpan tabel *hash* dari daftar orang dan tempat. *Input* yang dimasukkan pada sistem adalah 16 digit nomor KTP untuk orang dan nama tempat untuk tempat.



Gambar 1. Struktur Data

Dalam *hash function*, 16 digit diproses untuk diambil digit ke 7 sampai 10 yang merupakan tanggal lahir. Tanggal lahir lalu diproses untuk mendapat angka yang merupakan tanggal lahir yang dikonversi ke bentuk harian dengan asumsi 1 tahun adalah 366 hari untuk mengakomodasi tahun kabisat. Angka yang didapat kemudian dibagi 366. Sisa hasil baginya merupakan *hash value* dari *input* tersebut.

Tanggal lahir dipilih karena lebih muda (walau terdapat *birthday paradox*) daripada digit lainnya dari nomor KTP. Hal ini karena digit 6 digit pertama nomor KTP adalah identitas daerah. Menggunakan identitas daerah sebagai *hash value* memungkinkan terjadinya *clustering* karena penyebaran virus yang berbasis pada satu wilayah tertentu.

Untuk *input* nama tempat, digunakan rumus Horner pada setiap karakter pada nama tempat:

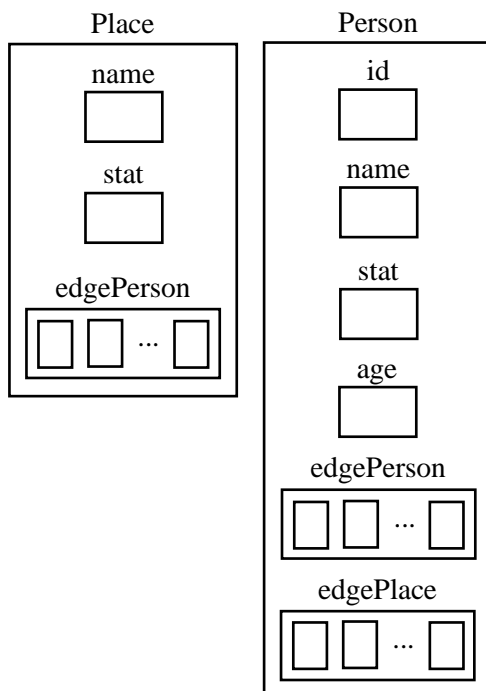
$$H(s) = \sum_{i=0}^n s[i] \cdot x^i$$

Kemudian hasil dari metode Horner tadi dibagi 265 yang merupakan ukuran tabel *hash* dan diambil sisa hasil baginya sebagai *hash value*.

Collision resolution pada *hash table* menggunakan *chaining* sehingga bentuk *hash table* adalah *array of list*.

Sebelum dikembalikan, hasil *hash value* ditambah 1 digit 0 atau 1 di belakangnya. Hal ini untuk membedakan hasil *hash value* dari tempat dan orang. Penambahan digit 0 untuk menandakan tempat dan penambahan digit 1 untuk menandakan orang.

Kembali pada struktur graf, untuk menyimpan *edge* dibuat objek *Place* dan *Person*. Objek *Place* akan memiliki *member* *edgePerson* yang menyimpan semua *edge* dengan objek *Person*.



Gambar 2. Struktur Objek

Objek *Person* akan memiliki *member* *edgePerson* yang menyimpan *edge* dengan objek *Person* lainnya dan *edgePlace* yang menyimpan *edge* dengan *Place*. Selain itu setiap objek juga memiliki *member* yang menyimpan identitas objek.

Status untuk objek *Person* adalah “NORMAL”, “OBSERVED”, atau “CONFIRMED”. Sementara status untuk objek *Place* adalah “CLEAN” atau “INFECTED”.

Selanjutnya agar bisa digunakan, diperlukan serangkaian perintah-perintah. Perintah-perintah tersebut adalah sebagai berikut:

add (id/place)	Menambahkan orang atau tempat
remove (id/place)	Menghapus orang atau tempat
info (id/place)	Menampilkan informasi orang atau tempat.
new (id) (id/place)	Menambahkan edge/koneksi.
delete (id) (id/place)	Menghapus edge/koneksi
show (‘person’/’place’)	Menampilkan daftar orang atau tempat
normal (id)	Mengubah status ke normal. Untuk orang sembuh
confirm (id)	Mengkonfirmasi positif
clean (place)	Menyatakan tempat telah didisinfeksi
help	Menampilkan bantuan
exit	Keluar dari program

Tabel 1. Daftar Perintah

Masing-masing perintah diimplementasi dengan fungsi-fungsi tertentu yang digunakan sesuai *input* pada fungsi utama. Setiap fungsi juga akan mengecek validitas dan jenis *input* untuk *error handling*.

3. DESAIN DAN ALGORITMA

3.1 FUNGSI add

Fungsi ini bertujuan menambah objek tempat atau orang pada graf. Algoritma:

1. Cek *input* melalui hasil *hash*
2. Masukkan *input* pada graf

3.2 FUNGSI remove

Fungsi ini bertujuan menghapus objek tempat atau orang dari graf. Fungsi ini juga akan otomatis menghapus semua *edge* yang terhubung dengan objek yang akan dihapus. Algoritma:

1. Cek *input* melalui hasil *hash*
- 2a. Jika *input* adalah tempat
 - a. Hapus *input* dari *edgePlace* semua objek *Person*
 - b. Hapus objek *input* dari graf
- 2b. Jika *input* adalah orang
 - a. Hapus *input* dari *edgePerson* objek *Person* lainnya
 - b. Hapus *input* dari *edgePlace* semua objek *Place*
 - c. Hapus objek *input* dari graf

3.3 FUNGSI info

Fungsi ini bertujuan menampilkan informasi objek tempat atau orang. Informasi dari objek adalah semua *member* objek tersebut termasuk daftar *edge*. Algoritma:

1. Cek *input* melalui hasil *hash*
2. Tampilkan semua *member* dari objek *input*

3.4 FUNGSI newEdge

Fungsi ini bertujuan menambah *edge/koneksi* objek orang dengan tempat atau orang. Fungsi memiliki 2 *input*. *Input* pertama adalah orang dan *input* kedua adalah orang atau tempat. Algoritma:

1. Cek *input* kedua melalui hasil *hash*
- 2a. Jika *input* kedua adalah tempat
 - a. Tambahkan *input* pertama pada *edgePerson input* kedua
 - b. Tambahkan *input* kedua pada *edgePlace input* pertama
 - c. Jika status *input* pertama “CONFIRMED”, status *input* kedua menjadi “INFECTED”
 - d. Jika status *input* kedua “INFECTED” dan status *input* pertama bukan “CONFIRMED”, status *input* pertama menjadi “OBSERVED”
- 2b. Jika *input* kedua adalah orang
 - a. Tambahkan *input* pertama pada *edgePerson input* kedua
 - b. Tambahkan *input* kedua pada *edgePerson input* pertama
 - c. Jika status *input* pertama “CONFIRMED” dan status *input* kedua bukan “CONFIRMED”, status *input* kedua menjadi “OBSERVED”
 - d. Jika status *input* kedua “CONFIRMED” dan status *input* pertama bukan “CONFIRMED”, status *input* pertama menjadi “OBSERVED”

3.5 FUNGSI deleteEdge

Fungsi ini bertujuan menghapus *edge/koneksi* objek orang dengan objek tempat atau orang dari graf. Fungsi memiliki 2 *input*. *Input* pertama adalah orang dan *input* kedua adalah orang atau tempat. Algoritma:

1. Cek *input* kedua melalui hasil *hash*
- 2a. Jika *input* kedua adalah tempat
 - a. Hapus *input* pertama dari *edgePerson input* kedua
 - b. Hapus *input* kedua dari *edgePlace input* pertama
- 2b. Jika *input* kedua adalah orang
 - a. Hapus *input* pertama dari *edgePerson input* kedua

- b. Hapus *input* kedua dari *edgePerson input* pertama

3.6 FUNGSI show

Fungsi ini bertujuan menampilkan daftar orang atau tempat yang ada pada graf. Fungsi ini menerima *input* “person” atau “place”. Algoritma:

1. Cek *input*
- 2a. Jika *input* adalah “place”, tampilkan semua objek Place dalam graf
- 2b. Jika *input* adalah “person”, tampilkan semua objek Person dalam graf

3.7 FUNGSI confirm

Fungsi ini bertujuan mengonfirmasi orang terkena virus. Lalu, fungsi ini akan mengubah status orang dan tempat yang terhubung dengan orang yang dikonfirmasi. Algoritma:

1. Mengubah status *input* menjadi “CONFIRMED”
2. Mengiterasi *edgePlace* pada *input*
 - a. Mengubah semua status orang pada *edgePerson* dari tempat menjadi “OBSERVED”
 - b. Mengubah status tempat menjadi “INFECTED”
3. Mengubah semua status orang pada *edgePerson* pada *input* menjadi “OBSERVED”

3.8 FUNGSI normal

Fungsi ini bertujuan mengubah status orang menjadi “NORMAL”. Fungsi ini digunakan jika orang sudah sembuh, *edge* dengan orang terkonfirmasi dihapus, atau terjadi kesalahan perintah. Algoritma:

1. Mengubah status *input* menjadi “NORMAL”

3.9 FUNGSI clean

Fungsi ini bertujuan mengubah status orang menjadi “NORMAL”. Fungsi ini digunakan jika orang sudah sembuh, *edge* dengan orang terkonfirmasi dihapus, atau terjadi kesalahan perintah. Algoritma:

1. Mengubah status *input* menjadi “NORMAL”

3.10 FUNGSI LAINNYA

Fungsi *showHelp* berfungsi untuk menampilkan bantuan berupa daftar perintah. Sementara fungsi *getCommand* yang berfungsi mengambil *line* perintah dan membaginya menjadi beberapa bagian *input* dan perintah.

Fungsi *digit* digunakan untuk mendapatkan jumlah digit dalam integer. Sementara fungsi *isAllDigit* berfungsi untuk menentukan apakah semua karakter *input* merupakan angka.

Fungsi *hashing* berfungsi untuk menghitung *hash value* dan mengembalikan nilai ID yang merupakan *hash value* yang ditambah 1 digit 0 atau 1 di belakang.

Fungsi *main* merupakan fungsi utama. Fungsi ini mengambil *input* dari pengguna lalu menjalankan *infinite loop* dan terus mengambil *input* perintah hingga mendapatkan perintah *exit*.

4. PENUTUP

Program ini merupakan masih merupakan model kecil dari sistem yang lebih besar dan belum siap pakai. Pengembangan model kecil ini juga akan dilanjutkan untuk menyempurnakan sistem.

Harapannya, program kecil ini bisa menginspirasi pemrogram lain untuk membuat program besar yang siap pakai sehingga dapat benar-benar membawa pengaruh dan membantu penanganan pandemi CoViD-19 di Indonesia.