

Aplikasi Algoritma Bron-Kerbosch dalam Sistem Rekomendasi Konten YouTube

Ignatius Jhon Hezkiel Chan- 13522029¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13522029@std.stei.itb.ac.id

Abstraksi—Youtube merupakan platform mainstream untuk mengupload dan menikmati konten video. Konten-konten video tersebut memiliki ragam variasi dan jenisnya. Tiap pengguna memiliki preferensi sendiri terhadap jenis konten yang dinikmati. Hubungan antar pengguna dapat menjadi indikator tentang preferensi konten, dan dapat digunakan sebagai kunci untuk menentukan rekomendasi konten yang sesuai, dengan bantuan algoritma Bron-Kerbosch. Dengan merepresentasikan pengguna sebagai simpul dan hubungan preferensi sebagai sisi dalam *graph*, kita dapat mengidentifikasi kelompok pengguna dengan preferensi serupa. Makalah ini akan membahas bagaimana aplikasi algoritma Bron-Kerbosch dalam memberikan rekomendasi konten yang sesuai kepada pengguna.

Keywords—Bron-Kerbosch, Rekomendasi, Preferensi, Video

I. PENDAHULUAN

Menonton konten-konten video merupakan kegemaran banyak orang dalam mengisi waktu luangnya. Saat sedang senggang, ataupun saat makan, menonton video menjadi salah satu kegiatan sambilan yang sering dilakukan orang. Menonton video ini menjadi sarana kita untuk *refreshing*, serta sebagai sarana entertainment yang “murah” dan mudah. Kita hanya perlu mempunyai device dan internet, kita sudah bisa menonton banyak video dengan berbagai variasi dan jenis. Apalagi sekarang hampir semua orang sudah memiliki *smartphone*, membuat konten video online menjadi sangat terjangkau.

Salah satu platform untuk berbagi dan menonton video yang paling populer saat ini adalah YouTube. Dengan jutaan pembuat konten yang menyajikan beragam topik mulai dari hiburan, pendidikan, hingga informasi, membuat YouTube menjadi tempat terdepan bagi pengguna untuk mengeksplorasi konten yang sesuai dengan minat dan preferensi mereka.

Tiap pengguna memiliki minat dan preferensi yang berbeda. Contohnya ada orang yang menyukai lagu dengan genre *rock*, tetapi ada orang lain yang tidak suka dengan genre *rock* melainkan suka genre *pop*. Penentuan rekomendasi konten sesuai preferensi menjadi factor utama keberhasilan platform untuk menarik pengguna agar terus menonton di YouTube. Jika kita ketika membuka YouTube dan meng-*scroll* laman beranda terus dipenuhi oleh jenis konten yang kita tidak cocok, tentu kita akan merasa tidak betah dan akan cepat keluar dari aplikasi.



Gambar 1 Tampilan Laman Rekomendasi Youtube (sumber: <https://nfinity8.com/wp-content/uploads/2019/07/youtube.jpg>)

Algoritma Bron-Kerbosch disini dapat diimplementasikan pada sistem rekomendasi konten YouTube, sehingga pengguna akan merasa nyaman dan akan lebih lama menghabiskan waktu menonton konten video di YouTube.

II. LANDASAN TEORI

A. Graf

1) Definisi Graf

Graf adalah struktur data yang digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Graf terdiri atas dua komponen, yakni

a) Simpul (vertex)

Merepresentasikan suatu objek diskrit, seperti wilayah dan orang

b) Sisi (edge)

Menyatakan hubungan antara objek diskrit, seperti jalan antara wilayah dan hubungan pertemanan

2) Jenis Graf

a) Graf tak-berarah (*undirected graph*).

Graf tak-berarah adalah graf yang sisinya tidak mempunyai orientasi arah. Jika simpul A terhubung ke simpul B, maka relasi ini tidak memiliki arah spesifik dan hubungan tersebut dianggap saling menghubungkan.

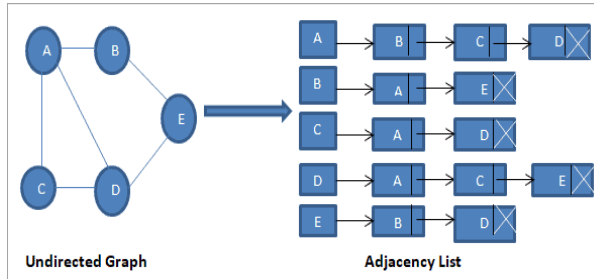
b) Graf lengkap (*complete graph*)

Graf lengkap adalah graf tak-berarah sederhana yang tiap pasangan simpul yang berbeda

dihubungkan oleh sisi yang unik

3) Representasi Graf

Salah satu bentuk representasi graf berupa bentuk Adjacency List. Pada bentuk ini, tiap simpul dalam graf menyimpan list simpul-simpul tetangganya.



Gambar 2 Representasi Graf dengan Adjacency List (sumber: <https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/08/7.undirected-graph-and-its-adjacency-list.png>)

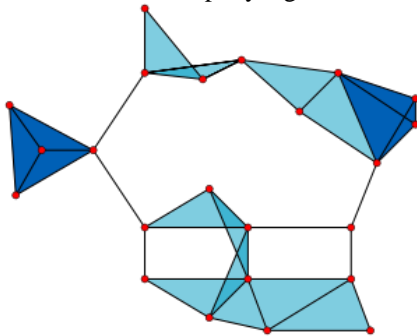
4) Terminologi Graf

a) Ketetanggaan (*Adjacent*)

Dua buah simpul dikatakan *bertetangga* bila keduanya terhubung langsung oleh suatu sisi

b) Klik (*clique*)

Klik adalah subset dari simpul dalam graf tak-berarah yang dimana setiap dua simpul berbeda di klik terhubung oleh suatu sisi (bertetangga). Klik dalam suatu graf adalah subgraph yang merupakan bentuk graf lengkap. Klik maksimal adalah klik yang tidak bisa diperluas dengan menambahkan simpul yang lain.



Gambar 3 Klik di dalam graf (sumber: https://upload.wikimedia.org/wikipedia/commons/thumb/d/d0/VR_complex.svg/300px/VR_complex.svg.png)

B. Algoritma Bron-Kerbosch

Algoritma Bron-Kerbosch adalah algoritma enumerasi untuk mencari semua klik maksimal dalam graf tak-berarah. Bentuk-bentuk implementasi algoritma Bron-Kerbosch adalah sebagai berikut.

a) Algoritma sederhana

Bentuk dasar dari algoritma Bron-Kerbosch adalah algoritma *backtracking* rekursif yang mencari semua klik maksimal dalam graf tertentu. Rekursi menerima tiga parameter, yakni R, P, dan X. R adalah himpunan simpul klik sekarang, P adalah himpunan kandidat simpul, dan X adalah himpunan eksklusi. Berikut algoritma dalam bentuk

pseudocode

```

algoritma BronKerbosch(R, P, X) adalah
jika P dan X keduanya kosong maka
    laporkan R sebagai klik maksimal
untuk setiap simpul v di P do
    BronKerbosch(R ∪ {v}, P ∩ N(v), X ∩
N(v))
    P := P \ {v}
    X := X ∪ {v}
    
```

Gambar 4 Pseudocode Algoritma simple Bron-Kerbosch (sumber: arsip penulis, referensi: https://en.wikipedia.org/wiki/Bron%E2%80%9993Kerbosch_algorithm)

Rekursi dimulai dengan menetapkan R dan X sebagai himpunan kosong dan P sebagai himpunan titik dari grafik. Dalam tiap pemanggilan rekursi, algoritma mempertimbangkan simpul di P secara bergantian, Jika himpunan P dan X kosong, maka R ditetapkan sebagai klik maksimal. Tetapi jika P kosong dan X tidak kosong, maka akan dilakukan backtracking. Setiap simpul v dalam p, akan dilakukan rekursi dengan v ditambahkan ke R, dan P dan X dibatasi pada himpunan tetangga N(v) dari v. Kemudian, v dihapus dari P dan ditambahkan pada X.

b) Algoritma dengan Pivoting

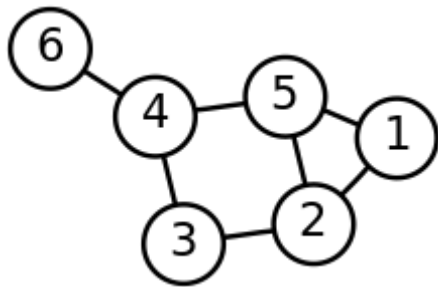
Varian algoritma Bron-Kerbosch sederhana yang lebih efisien, dimana dilakukan dengan menambahkan pivot vertex u, yang dipilih dari P. Iterasi bukan lagi dilakukan terhadap anggota P, melainkan P \ N(u). Penambahan ini bertujuan untuk menghemat waktu dan memungkinkan algoritma untuk melakukan backtrack lebih cepat pada cabang pencarian yang tidak mengandung klik maksimal. Setiap klik maksimal harus menyertakan u atau salah satu dari non-tetangganya. Hanya u dan non-tetangganya yang perlu diuji sebagai pilihan simpul v yang ditambahkan ke R pada setiap pemanggilan rekursif. Berikut adalah pseudocode Bron-Kerbosch dengan pivoting.

```

algoritma BronKerbosch(R, P, X) adalah
jika P dan X keduanya kosong maka
    laporkan R sebagai klik maksimal
    pilih titik pivot u di P ∪ X
untuk setiap titik v di P \ N(u) do
    BronKerbosch(R ∪ {v}, P ∩ N(v), X ∩
N(v))
    P := P \ {v}
    X := X ∪ {v}
    
```

Gambar 5 Pseudocode Algoritma Bron-Kerbosch Pivoting (sumber: arsip penulis, referensi: https://en.wikipedia.org/wiki/Bron%E2%80%9993Kerbosch_algorithm)

Contoh implementasi algoritma Bron-Kerbosch pivoting pada graf adalah sebagai berikut



Gambar 6 Graf sederhana dengan lima klik maksimal (sumber: <https://upload.wikimedia.org/wikipedia/commons/thumb/5/5b/6n-graf.svg/220px-6n-graf.svg.png>)

Misalkan terdapat suatu graf seperti di atas dengan 6 simpul. Kita mencari klik-klik maksimal graf dengan menggunakan algoritma Bron-Kerbosch pivoting. Pohon panggilan untuk algoritma Bron-Kerbosch dengan Pivoting untuk graf tersebut adalah sebagai berikut

```

BronKerbosch( $\emptyset$ , {1,2,3,4,5,6},  $\emptyset$ )
BronKerbosch({2}, {1,3,5},  $\emptyset$ )
BronKerbosch({2,3},  $\emptyset$ ,  $\emptyset$ ): Klik{2,3}
BronKerbosch({2,5}, {1},  $\emptyset$ )
BronKerbosch({1,2,5},  $\emptyset$ ,  $\emptyset$ ):Klik{1,2,5}
BronKerbosch({4}, {3,5,6},  $\emptyset$ )
BronKerbosch({3,4},  $\emptyset$ ,  $\emptyset$ ): Klik{3,4}
BronKerbosch({4,5},  $\emptyset$ ,  $\emptyset$ ): Klik {4,5}
BronKerbosch({4,6},  $\emptyset$ ,  $\emptyset$ ) Klik {4,6}
BronKerbosch({6},  $\emptyset$ , {4}): tidak ada keluaran

```

Gambar 7 Tahapan rekursif algoritma Bron-Kerbosch untuk graf pada Gambar 6 (sumber: arsip penulis, referensi: https://en.wikipedia.org/wiki/Bron%E2%80%93Kerbosch_algorithm)

- c) Algoritma dengan konsep Degenerasi
 Metode alternatif berikutnya untuk meningkatkan bentuk dasar algoritma Bron-Kerbosch melibatkan tidak melakukan pivot pada Tingkat rekursi terluar, dan sebagai gantinya memilih urutan panggilan rekursif dengan hati-hati untuk meminimalkan ukuran himpunan P dari kandidat simpul dalam setiap panggilan rekursi.

Degenerasi suatu graf G adalah bilangan terkecil d sehingga setiap subgraph dari G mempunyai titik sudut yang berderajat d atau kurang. Setiap graf mempunyai urutan degenerasi, suatu urutan simpul sedemikian sehingga setiap simpul mempunyai d atau lebih sedikit tetangga yang datang nantinya dalam pengurutan tersebut. Tatanan degenerasi dapat ditemukan dalam waktu linear dengan berulang kali memilih simpul dengan derajat minimum di antara simpul-simpul yang tersisa.

Jika urutan simpul v yang dilalui algoritma Bron-Kerbosch adalah urutan degenerasi, maka himpunan P calon simpul dalam setiap pemanggilan (tetangga dari v yang kemudian diurutkan) akan dijamin memiliki ukuran paling banyak D. Himpunan X dari simpul yang dikecualikan akan terdiri dari semua tetangga sebelumnya dari v, dan mungkin jauh lebih besar dari d. Dalam pemanggilan rekursif ke algoritma di bawah level rekursi paling atas, versi pivot masih dapat digunakan.

Berikut adalah pseudocode algoritma Bron-Kerbosch dengan degenerasi graf.

```

Algoritma BronKerbosch(G) adalah
P = V(G)
R = X = kosong
Untuk setiap simpul v dalam urutan degenerasi G do
BronKerbosch({v}, P ∩ N(v), X ∩ N(v))
P := P \ {v}
X := X ∪ {v}

```

Gambar 8 Pseudocode Algoritma Bron-Kerbosch dengan degenerasi graf (sumber: arsip penulis, referensi: https://en.wikipedia.org/wiki/Bron%E2%80%93Kerbosch_algorithm)

III. IMPLEMENTASI

A. Representasi Pengguna pada Graf

Graf merepresentasikan hubungan preferensi antara pengguna-pengguna YouTube. Pada graf ini, simpul merepresentasikan pengguna YouTube. Antar simpul akan dihubungkan oleh sisi yang menggambarkan hubungan atau interaksi antara dua pengguna. Sisi ini dapat diasosiasikan dengan berbagai tindakan atau hubungan antar pengguna yang mencerminkan preferensi mereka terhadap video. Berikut adalah beberapa contoh bagaimana sisi dapat menghubungkan pengguna YouTube:

1. Menonton video yang sama
 Jika dua pengguna A dan B menonton video yang sama atau serupa, maka terdapat sisi antara simpul A dan simpul B. Ini mencerminkan adanya kesamaan dalam konten yang dinikmati oleh keduanya.
2. Menyukai video yang sama
 Sisi dapat mewakili interaksi positif seperti “menyukai” atau “mengomentari” video yang sama. Jika pengguna A dan B melakukan interaksi positif terhadap video yang sama, maka sisi dapat dihubungkan antara simpul A dan B.
3. Hubungan sosial
 Jika pengguna A dan B saling mengikuti atau terlibat dalam aktivitas sosial lainnya di platform, sisi dapat mewakili hubungan sosial di luar preferensi konten

Situasi dimana suatu sisi antara dua simpul pengguna bisa dihilangkan adalah ketika terjadi interaksi yang bertolak belakang antar dua pengguna. Contohnya ketika pengguna A menyukai suatu video, tetapi pengguna B memberikan dislike terhadap video tersebut.

Sisi antara dua pengguna bisa dihilangkan ketika untuk suatu rekomendasi hasil tontonan A yang diberikan kepada B, diabaikan selama beberapa kali. Hal ini bisa disimpulkan bahwa untuk sekarang rekomendasi tontonan A dan B tidak cocok. Sisi ini bisa dimunculkan kembali ke depannya jika didapatkan lagi antara A dan B melakukan interaksi yang sama terhadap suatu video.

B. Sistem Rekomendasi

Ketika suatu pengguna telah menonton suatu video, video tersebut akan masuk dalam list rekomendasi video untuk pengguna-pengguna lain yang berada pada klik yang sama. Penentuan target pengguna rekomendasi akan diiterasi

terhadap tiap klik yang dimana pengguna menjadi salah satu anggotanya.

Penentuan klik-klik ini akan dilakukan dengan menggunakan Algoritma Bron-Kerbosch pivoting. Klik-klik ini kemudian akan disimpan pada suatu array. Array ini akan digunakan di kemudian saat ketika salah satu penonton sudah menonton suatu video. Array ini akan diiterasi untuk mencari pengguna mana saja yang berada pada satu klik yang sama dengan pengguna awal. Pengguna-pengguna yang telah dilist ini yang akan menjadi target rekomendasi video tersebut.

Berikut adalah implementasi algoritma Bron-Kerbosch pivoting dalam Bahasa Python, dimana pivot yang dipilih adalah elemen pertama dalam P.

```
def bronKerbosch(R, P, X, graf, kliks):
    if not P and not X:
        kliks.append(R.copy())
        return

    pivot = next(iter(P.union(X)))

    for v in P.difference(graf[pivot]):
        bronKerbosch(R.union({v}),
        P.intersection(graf[v]), X.intersection(graf[v]),
        graf, kliks)
        P.remove(v)
        X.add(v)
```

Gambar 9 Implementasi Algoritma Bron-Kerbosch Pivoting dalam Python (sumber: arsip penulis)

Fungsi bronKerbosch akan menerima lima parameter, yakni R,P,X,graf, dan kliks. Pada awalnya, R dan X berupa set kosong, P berisi semua simpul dalam graf, variable graf berupa graf hubungan preferensi dengan representasi adjacency list, dan kliks berupa array kosong yang akan menampung klik-klik maksimal dalam graf. Fungsi ini akan menghasilkan variable kliks yang berisi klik-klik maksimal yang ditemukan dalam graf. Ketika didapatkan array of klik, maka kemudian dilakukan iterasi tiap klik untuk mencari pengguna mana saja yang termasuk dalam satu klik dengan pengguna awal.

```
def findNeighborKliks(kliks, pengguna):
    neighbor = set()
    for klik in kliks:
        if pengguna in klik:
            klik.remove(pengguna)
            for user in klik:
                neighbor.add(user)
    return neighbor
```

Gambar 10 Implementasi Fungsi findNeighborKliks dalam Python (sumber: arsip penulis)

Fungsi findNeighborKliks akan me-return sebuah set yang berisi pengguna-pengguna yang berada pada satu klik dengan "pengguna". Isi set inilah yang menjadi target pengguna untuk direkomendasikan videonya.

Berikut merupakan program sederhana dalam Python, yang mensimulasikan sistem rekomendasi dengan meminta nama pengguna dan judul video yang ditonton, serta menampilkan target rekomendasi dari videonya.

```
def recommendVideo(graf):
    pengguna = input("Nama pengguna: ")
    namaVid = input("Title video: ")
    simpuls = set(graf.keys())
    allKliks = []
    bronKerbosch(set(), simpuls, set(), graf, allKliks)
    targetUsers = findNeighborKliks(allKliks, pengguna)
    if targetUsers:
        print("Video ber-title", namaVid, "akan khusus direkomendasikan kepada ", end='')
        for user in targetUsers:
            print(user, end=', ')
    else:
        print("Video ber-title", namaVid, "akan direkomendasikan secara random kepada user lain")
```

Gambar 11 Implementasi Fungsi recommendVideo dalam python (sumber: arsip penulis)

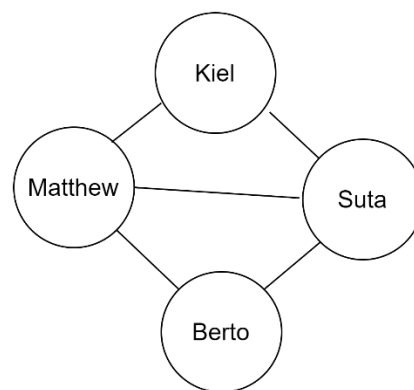
Setelah meminta nama pengguna dan video yang telah ditonton, variable simpuls akan diisi dengan nama-nama pengguna. Variable allKliks akan diinisialisasi kosong, dan fungsi bronKerbosch kemudian dipanggil dengan argumentnya R berupa set kosong, P berupa simpuls, X berupa set kosong, graf berupa graf hubungan preferensi, dan kliks berupa allKliks yang merupakan array kosong. Program kemudian akan menampilkan target-target pengguna untuk direkomendasikan video dengan title namaVid.

IV. ANALISIS KASUS

Untuk menunjukkan bagaimana aplikasi algoritma Bron-Kerbosch menentukan target rekomendasi suatu video, makalah ini akan meninjau beberapa contoh kasus.

A. Contoh Kasus Pertama

Misalkan terdapat 4 pengguna YouTube, yakni Kiel, Matthew, Suta, dan Berto. Kiel, Matthew, dan Suta sama-sama menyukai konten video bertipe *prank*. Matthew, Suta, dan Berto sama-sama menyukai konten video bertipe Edukasi. Berto tidak suka menonton video *prank*. Maka, bentuk graf hubungan preferensinya adalah sebagai berikut.



Gambar 12 Graf untuk contoh kasus pertama (sumber: arsip penulis)

Bentuk representasi adjacency list dari graf tersebut adalah sebagai berikut.

```
graf = {
    "Kiel": {"Matthew", "Suta"},
    "Matthew": {"kiel", "Suta", "Berto"},
    "Suta": {"Kiel", "Matthew", "Berto"},
    "Berto": {"Matthew", "suta"}
}
```

Gambar 13 Representasi graf pada Gambar 12 dengan bentuk Adjacency List dalam python (sumber: arsip penulis)

Suatu saat, Kiel menonton video *mukbang* yang berjudul “MUKBANG SAMYANG”. YouTube kemudian akan merekomendasikan video tersebut kepada pengguna lain yang memiliki preferensi yang sama dengan Kiel sesuai dengan graf preferensi. Simulasi sederhana-nya dapat kita lihat dengan fungsi `recommendVideo` sebelumnya.

```
PS C:\Users\user\Documents\Bron-Kerbosch> python BronKerbosch.py
Nama pengguna: Kiel
Title video: MUKBANG SAMYANG
Video ber-title MUKBANG SAMYANG akan khusus direkomendasikan kepada Suta, Matthew,
```

Gambar 14 Hasil program I kasus pertama (sumber: arsip penulis)

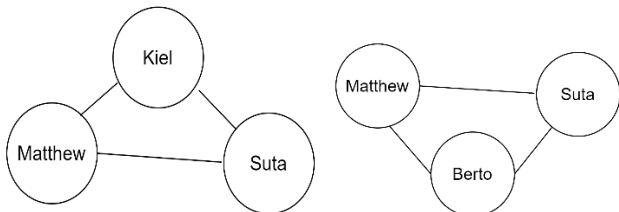
Terlihat bahwa video berjudul “MUKBANG SAMYANG” akan direkomendasikan khususnya pada Suta dan Matthew. Pemilihan Suta dan Matthew karena mereka berada pada klik yang sama dengan Kiel, sedangkan Berto tidak.

Dalam menentukan klik-klik maksimal yang terdapat pada graf tersebut, kita gunakan fungsi `bronKerbosch`. Berikut iterasi masing-masing pemanggilan fungsi rekursif.

```
BronKerbosch(∅, {Kiel, Matthew, Suta, Berto}, ∅)
{Pivot: kiel}
BronKerbosch({Kiel}, {Matthew, Suta}, ∅)
{Pivot: Matthew}
BronKerbosch({Kiel, Matthew}, {Suta}, ∅)
{Pivot: Suta}
BronKerbosch({Kiel, Matthew, Suta}, ∅, ∅)
BronKerbosch({Berto}, {Matthew, Suta}, ∅)
{Pivot: Matthew}
BronKerbosch({Berto, Matthew}, {Suta}, ∅)
{Pivot: Suta}
BronKerbosch({Berto, Matthew, Suta}, ∅, ∅)
```

Gambar 15 Tahapan rekursif algoritma Bron-Kerbosch untuk graf pada kasus pertama (sumber: arsip penulis)

Pemanggilan fungsi yang *dihighlight* kuning menandakan klik maksimal yang ada pada graf. Hasil pemanggilan fungsi menunjukkan bahwa Kiel, Matthew, dan Suta berada pada satu klik, dan Berto, Sutha dan Matthew berada pada satu klik.



Gambar 16 Klik-klik graf pada contoh kasus pertama (sumber: arsip penulis)

Setelah didapatkan list dari klik-klik maksimal, kemudian dilist pengguna-pengguna yang berada pada satu klik dengan Kiel. Penyataan pengguna ini dilakukan pada fungsi

`findNeighborKliks`, yang meng-return array berisi target-target pengguna. Ketika dipanggil fungsi `findNeighborKliks` dengan argument pertama array klik dan argument kedua “Kiel”, didapatkan hasil array yang berisi [“Matthew”, “Suta”]. Matthew dan Suta lah yang kemudian direkomendasikan video berjudul “MUKBANG SAMYANG”, oleh YouTube.

Jika berikutnya Matthew yang menonton video berjudul “Jalan-Jalan di ITB”, maka hasil list target pengguna pada fungsi `findNeighborKliks`, akan me-return array yang berisi [“Berto”, “Kiel”, “Suta”]. Mereka bertiga yang kemudian akan direkomendasikan video “Jalan-Jalan di ITB”.

```
PS C:\Users\user\Documents\Bron-Kerbosch> python BronKerbosch.py
Nama pengguna: Matthew
Title video: Jalan-Jalan di ITB
Video ber-title Jalan-Jalan di ITB akan khusus direkomendasikan kepada Berto, Kiel, Suta,
```

Gambar 17 Hasil program II contoh kasus pertama (sumber: arsip penulis)

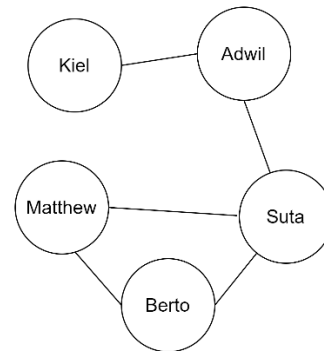
Jika kemudian Berto menonton video berjudul “Planet di Tata Surya”, maka hasil list target pengguna pada fungsi `findNeighborKliks`, akan me-return array yang berisi [“Suta”, “Matthew”]. Suta dan Matthew kemudian akan direkomendasikan video “Planet di Tata Surya”.

```
PS C:\Users\user\Documents\Bron-Kerbosch> python BronKerbosch.py
Nama pengguna: Berto
Title video: Planet di Tata Surya
Video ber-title Planet di Tata Surya akan khusus direkomendasikan kepada Suta, Matthew,
```

Gambar 18 Hasil program III contoh kasus pertama (sumber: arsip penulis)

B. Contoh Kasus Kedua

Misalkan terdapat 5 pengguna YouTube dengan graf hubungan preferensi sebagai berikut.



Gambar 19 Graf untuk contoh kasus kedua (sumber: arsip penulis)

```
graf = {
    "Kiel": {"Adwil"},
    "Matthew": {"Suta", "Berto"},
    "Suta": {"Adwil", "Matthew", "Berto"},
    "Berto": {"Matthew", "Suta"},
    "Adwil": {"Kiel", "Suta"}
}
```

Gambar 20 Representasi graf pada Gambar 18 dengan bentuk Adjacency List dalam python (sumber: arsip penulis)

Klik-klik maksimal yang terdapat pada graf tersebut dicari dengan algoritma Bron-Kerbosch. Berikut iterasi

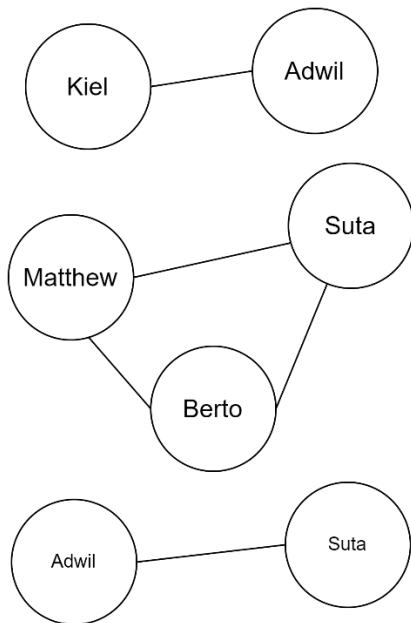
pemanggilan fungsi rekursif.

```

BronKerbosch( $\emptyset$ , {Kiel, Matthew, Suta, Berto, Adwil},  $\emptyset$ )
{Pivot: Kiel}
BronKerbosch({Kiel}, {Adwil},  $\emptyset$ )
{Pivot: Adwil}
BronKerbosch({Kiel, Adwil},  $\emptyset$ ,  $\emptyset$ )
BronKerbosch({Berto}, {Matthew, Suta},  $\emptyset$ )
{Pivot: Matthew}
BronKerbosch({Berto, Matthew}, {Suta},  $\emptyset$ )
{Pivot: Suta}
BronKerbosch({Berto, Matthew, Suta},  $\emptyset$ ,  $\emptyset$ )
BronKerbosch({Matthew}, {Suta}, {Berto})
{Pivot: Suta}
BronKerbosch({Matthew, Suta},  $\emptyset$ , {Berto})
BronKerbosch({Suta}, {Adwil}, {Berto, Matthew})
{Pivot: Adwil}
BronKerbosch({Adwil, Suta},  $\emptyset$ ,  $\emptyset$ )
    
```

Gambar 21 Tahapan rekursif algoritma Bron-Kerbosch untuk graf pada contoh kasus kedua (sumber: arsip penulis)

Didapatkan tiga klik pada graf preferensi, yakni klik pertama berisi Kiel dan Adwil, klik kedua berisi Berto, Matthew, Suta, dan klik ketiga berisi Adwil dan Suta.



Gambar 22 Klik-klik graf pada contoh kasus kedua (sumber: arsip penulis)

Jika Suta menonton video berjudul “Mobile Legends”, maka hasil list target pengguna pada fungsi findNeighborKliks, akan me-return array yang berisi [“Matthew”, “Berto”, “Adwil”]. Mereka bertiga yang kemudian akan direkomendasikan video “Mobile Legends”.

```

PS C:\Users\user\Documents\Bron-Kerbosch> python BronKerbosch.py
Nama pengguna: Suta
Title video: Mobile Legends
Video ber-title Mobile Legends akan khusus direkomendasikan kepada Adwil, Berto, Matthew,
    
```

Gambar 23 Hasil program I contoh kasus pertama (sumber: arsip penulis)

Jika Adwil menonton video “Tutorial Matdis”, maka hasil list target pengguna pada fungsi findNeighborKliks, akan me-return array yang berisi

[“Kiel”, “Suta”]. Mereka berdua akan direkomendasikan video “Tutorial Matdis”.

```

PS C:\Users\user\Documents\Bron-Kerbosch> python BronKerbosch.py
Nama pengguna: Adwil
Title video: Tutorial Matdis
Video ber-title Tutorial Matdis akan khusus direkomendasikan kepada Kiel, Suta,
    
```

Gambar 24 Hasil program II contoh kasus kedua (sumber: arsip penulis)

V. KESIMPULAN

Menentukan rekomendasi konten video yang sesuai dengan penonton YouTube menjadi factor penting dalam keberhasilan platform YouTube. Melalui makalah ini, kita dapatkan bahwa algoritma Bron-Kerbosch tentang klik di dalam graf, bisa diaplikasikan untuk sistem rekomendasi video. Sistem rekomendasi ini kita representasikan dalam bentuk graf, dimana simpul mewakili para pengguna YouTube, dan sisi antara simpul merepresentasi hubungan preferensi pengguna. Konten video akan direkomendasikan pada pengguna-pengguna lain yang memiliki preferensi yang sama, yang dimana dalam konteks graf berarti berada dalam suatu klik yang sama. Penerapan ini diharapkan dapat memberikan gambaran sederhana tentang sistem rekomendasi video melalui algoritma Bron-Kerbosch pada graf.

VI. UCAPAN TERIMA KASIH

Penulis ingin mengucapkan terima kasih kepada semua dosen IF2120 Matematika Diskrit, khususnya kepada Dr. Nur Ulfa Maulidevi, S.T., M.Sc sebagai dosen pengajar di Kelas 1 IF2120 untuk Matematika Diskrit, atas pengajarannya dan dukungannya pada para mahasiswa/i untuk menulis makalah ini. Melalui makalah ini, saya semakin mengerti dan paham tentang graf dan aplikasinya. Saya juga ingin berterima kasih kepada Dr. Ir. Rinaldi, M.T, yang telah menyediakan media pembelajaran bagi para mahasiswa/i melalui website beliau.

REFERENSI

- [1] Wikipedia contributors, "Bron-Kerbosch algorithm," Wikipedia, December 2, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Bron%E2%80%93Kerbosch_algorithm. Diakses pada December 10, 2023.
- [2] Wikipedia contributors, "Clique (graph theory)," Wikipedia. [Online]. Available: [https://en.wikipedia.org/wiki/Clique_\(graph_theory\)](https://en.wikipedia.org/wiki/Clique_(graph_theory)). Diakses pada December 10, 2023.
- [3] A. W. Lumbantobing, "Permasalahan Clique dalam Graf," Tidak diterbitkan. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2010-2011/Makalah2010/MakalahStima2010-051.pdf>. Diakses pada December 11, 2023.
- [4] [7] Cstutorials, "Maximal Clique Enumeration: Bron-Kerbosch Algorithm," YouTube. [Online]. Available: https://www.youtube.com/watch?v=j_uQChgo72I. Diakses pada December 10, 2023.
- [5] R. Munir, "Matematika Diskrit - ITB, 2023-2024," Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung. [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/matdis23-24.htm>. Diakses pada December 10, 2023.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Desember 2023

A handwritten signature in black ink, consisting of several overlapping loops and a long horizontal stroke extending to the right.

Ignatius Jhon Hezekiel Chan 13522029