

Penerapan Regular Expression Terhadap Nama Anggota Keluarga Dalam Graf Relasi Keluarga

Hamdi Ahmad Zuhri, 13516150¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

¹13516150@std.stei.itb.ac.id

Abstrak — Makalah ini adalah hasil sedikit perombakan dari makalah penulis sebelumnya tentang graf dan penerapan regex di sini adalah dalam perkiraan (tidak direalisasikan). Regular Expression digunakan untuk melakukan perbandingan antara nama yang dicari dengan nama-nama yang ada dalam informasi relasi keluarga. Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Salah-satunya, graf dapat digunakan untuk merepresentasikan anggota-anggota (objek) dari suatu keluarga beserta relasi-relasi (hubungan) yang ada dilamnya. Dengan program yang menerapkan pendekatan teori graf berarah berbobot sederhana dan memanfaatkan data-data keluarga beserta relasi-relasinya, kita dapat menentukan seorang anggota keluarga berdasarkan suatu relasi dengan anggota keluarga.

Kata kunci — keluarga, data keluarga, relasi keluarga

I. PENDAHULUAN

Sebuah bahasa dinyatakan regular jika terdapat finite state automata yang dapat menerimanya. Bahasa-bahasa yang diterima oleh FSA bisa dinyatakan secara sederhana dengan ekspresi regular (regular expression).

Menurut Kamus Besar Bahasa Indonesia (KBBI) versi kbbi.web.id, salah-satu definisi keluarga adalah ibu dan bapak beserta anak-anaknya. Di samping itu, masih dengan sumber yang sama, pengolahan adalah proses, cara, perbuatan mengolah;- data proses, cara, perbuatan mengolah data. Kemudian, masih dari sumber yang sama, relasi adalah hubungan; perhubungan; pertalian.

Jadi, dari definisi beberapa kata di atas, dapat diambil pemaknaan bahwa pengolahan data keluarga adalah kegiatan memproses data keluarga (dalam hal ini, berupa ayah, ibu, dan anak-anaknya beserta gabungan beberapa ‘keluarga kecil’ tersebut) sehingga dihasilkan *output* yang diinginkan dari data tersebut. Salah-satu hal yang bisa dijadikan *output* atau hasil adalah info anggota keluarga dari seorang anggota keluarga berdasarkan suatu relasi.

Seiring berjalannya waktu, suatu keluarga besar akan mengandung semakin banyak anggota keluarga dan semakin banyak relasi di antara mereka (contoh: kakeknya kakek, nenek, ayah, istri, anak, adik, paman, sampai kepada anaknya anak atau cucu) sehingga semakin sulit untuk mengetahui atau mengolah info atau data secara manual.

II. DASAR-DASAR TEORI REGEX DAN GRAF

2.1. Teori Regular Expression

Regular expressions are patterns used to match character combinations in strings. In JavaScript, regular expressions are also objects. These patterns are used with the `exec` and `test` methods of `RegExp`, and with the `match`, `replace`, `search`, and `split` methods of `String`. This chapter describes JavaScript regular expressions.

2.2. Pengertian Teori Graf

Teori graf merupakan pokok bahasan yang sudah tua usianya namun memiliki banyak terapan sampai saat ini. Dalam matematika dan ilmu komputer, teori graf adalah cabang kajian yang mempelajari sifat-sifat graf. Secara informal, suatu graf adalah himpunan benda-benda yang disebut simpul (vertex atau node) yang terhubung oleh sisi (edge) atau busur (arc). Biasanya graf digambarkan sebagai kumpulan titik-titik (melambangkan simpul) yang dihubungkan oleh garis-garis (melambangkan sisi) atau garis berpanah (melambangkan busur).

2.3. Definisi Graf

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Definisi Graf dapat dijelaskan sebagai berikut:

Graf G didefinisikan sebagai pasangan himpunan (V, E) , yang dalam hal ini:

- V adalah himpunan tak-kosong dari simpul-simpul (vertices atau nodes).
- E adalah himpunan sisi (edges atau arcs) yang menghubungkan sepasang simpul.

2.4. Terminologi Dasar Dalam Graf

Ada beberapa istilah yang perlu diketahui sebelum lebih jauh membahas graf sebagai berikut:

- Gelang (*loop*) yaitu busur yang berawal dan berakhir pada simpul yang sama. Busur ganda (*multiple edge*) yaitu suatu busur yang menghubungkan simpul yang sama.
- Ketetanggaan (*adjacent*) : dua buah simpul dikatakan bertetangga, jika terdapat busur dengan ujung awal dan akhir adalah v_1 dan v_2 .
- Kehadiran (*incident*) : suatu busur dikatakan hadir pada suatu simpul, jika busur tersebut menghubungkan simpul

tersebut.

- Derajat (*degree*) yaitu banyaknya busur yang ada pada suatu simpul v .
- Simpul terminal adalah simpul yang berderajat 1.
- Simpul terpencil adalah simpul yang berderajat 0 (tidak bertetangga dengan simpul lain).
- $n = |V|$ = kardinalitas simpul
- $m = |E|$ = kardinalitas busur

2.5. Jenis-Jenis Graf

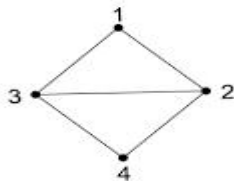
Graf memiliki banyak jenis, dalam tulisan ini akan dibahas beberapa jenis graf yang sering digunakan. Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf dan berdasarkan sisi pada graf yang mempunyai orientasi arah.

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf maka graf digolongkan menjadi dua jenis:

1. Graf sederhana (*simple graph*)

Graf yang tidak mengandung gelang maupun sisi ganda dinamakan graf sederhana.

Contoh graf sederhana (*simple graph*) adalah seperti gambar di bawah ini:

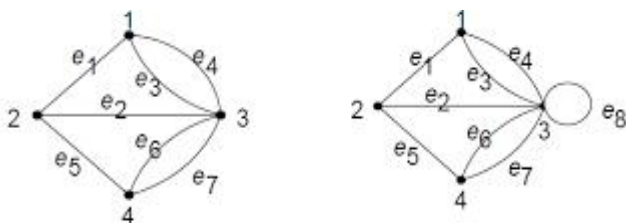


Sumber : <http://rabbitjeyek.blogspot.co.id/2011/12/teori-graf-3.html>

2. Graf tak-sederhana (*unsimple graph*)

Graf yang mengandung sisi ganda atau gelang dinamakan graf tak sederhana (*unsimple graph*). Ada dua macam graf tak sederhana, yaitu graf ganda (*multigraph*) atau graf semu (*pseudograph*). Graf ganda adalah graf yang mengandung sisi ganda. Graf semu adalah graf yang mengandung gelang (loop). Jumlah simpul pada graf disebut sebagai kardinalitas graf, dan dinyatakan dengan $n = |V|$, dan jumlah sisi kita nyatakan dengan $m = |E|$.

Contoh graf tak-sederhana (*unsimple graph*) adalah seperti gambar di bawah ini:



Sumber : <http://rabbitjeyek.blogspot.co.id/2011/12/teori-graf-3.html>

Berdasarkan orientasi arah pada sisi, maka secara umum, graf dibedakan atas 2 jenis:

1. Graf tak-berarah (undirected graph)

Graf yang sisinya tidak mempunyai orientasi arah disebut tak-berarah. Pada graf tak-berarah, urutan pasangan simpul

yang dihubungkan oleh sisi tidak diperhatikan. Jadi, $(u,v) = (v,u)$ adalah sisi yang sama.

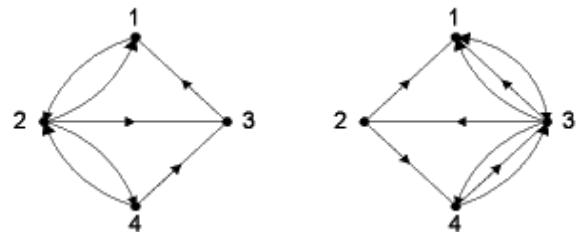
Contoh graf tak-berarah adalah seperti gambar yang menjadi contoh untuk graf sederhana dan tak-sederhana.

2. Graf berarah (directed graph atau digraph)

Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah. Pada graf berarah, (u, v) dan (v, u) menyatakan dua buah busur yang berbeda, dengan kata lain $(u, v) \neq (v, u)$. Untuk busur (u, v) simpul u dinamakan simpul asal (initial vertex) dan simpul v dinamakan simpul terminal (terminal vertex).

Definisi graf dapat diperluas sehingga mencakup graf-ganda berarah (directed multigraph). Pada graf-ganda berarah, gelang dan sisi ganda diperbolehkan ada.

Contoh graf berarah (directed graph atau digraph) adalah seperti gambar di bawah ini:



Sumber : <http://sha-essa.blogspot.co.id/2011/12/>

2.6. Graf Berarah Berbobot Sederhana

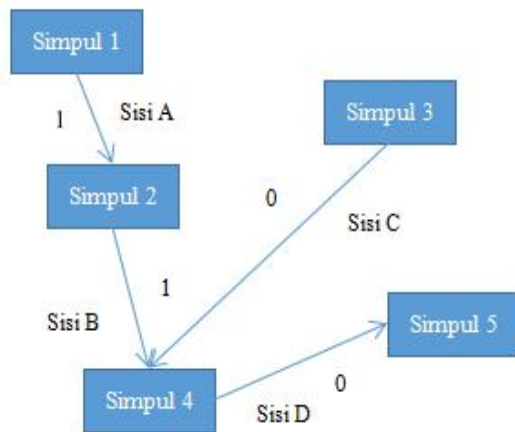
Graph berarah (*directed graph* atau *digraph*) adalah Graph yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah.

Sebuah struktur graf bisa dikembangkan dengan memberi bobot pada tiap sisi. Graf berbobot dapat digunakan untuk melambangkan banyak konsep berbeda. Sebagai contoh jika suatu graf melambangkan jaringan jalan maka bobotnya bisa berarti panjang jalan maupun batas kecepatan tertinggi pada jalan tertentu. Ekstensi lain pada graf adalah dengan membuat sisinya berarah, yang secara teknis disebut graf berarah atau digraf (*directed graph*).

Graf sederhana adalah graf yang tidak mengandung *loops* atau *multiple edges*. *Loops* adalah busur yang memiliki *endpoint* sama, sedangkan *multiple edges* adalah busur yang memiliki pasangan *endpoint* sama.

Jadi, jika ketiga sifat-sifat graf di atas disatukan terbentuklah graf berarah berbobot sederhana. Graf ini dapat merepresentasikan suatu kasus dimana setiap objek (simpul) terkait dengan beberapa sisi (busur) yang memiliki orientasi arah dan memiliki bobot yang minimal dapat menjadi tanda (*mark*).

Contoh graf berarah berbobot sederhana adalah seperti gambar di bawah ini (bobot bernilai 1 atau 0).



Gambar 1 (dibuat oleh penulis)

III. IMPLEMENTASI REGEX UNTUK NAMA DAN RELASI KELUARGA DALAM GRAF

3.2. Penerapan Regex

Regular Expression, dalam perkiraan penulis, dapat digunakan untuk melakukan perbandingan antara nama yang dicari dengan nama-nama yang ada dalam data relasi keluarga.

3.2. Pemilihan Graf untuk Pemodelan

Model adalah suatu deskripsi logis tentang bagaimana sistem bekerja atau komponen-komponen berinteraksi. Suatu sistem diharapkan dapat menjadi lebih mudah untuk proses analisis dengan pembuatan model. Hal ini merupakan prinsip pemodelan, yaitu bahwa pemodelan bertujuan untuk mempermudah analisis dan pengembangannya.

Kegiatan Pemodelan adalah suatu cara untuk mempelajari sistem dan model itu sendiri dan juga bermacam-macam perbedaan perilakunya. Model merupakan contoh sederhana dari sistem dan menyerupai sifat-sifat sistem yang dipertimbangkan, tetapi tidak sama dengan sistem. Penyederhanaan dari system sangat penting agar dapat dipelajari secara seksama. Model dikembangkan dengan tujuan untuk studi tingkah-laku sistem melalui analisis rinci akan komponen atau unsur dan proses utama yang menyusun sistem dan interaksinya antara satu dengan yang lain. Jadi, pengembangan model merupakan suatu pendekatan yang tersedia untuk mendapatkan pengetahuan yang layak akan sistem. Model berperan penting dalam pengembangan teori karena berfungsi sebagai konsep dasar yang menata rangkaian aturan yang digunakan untuk menggambarkan sistem.

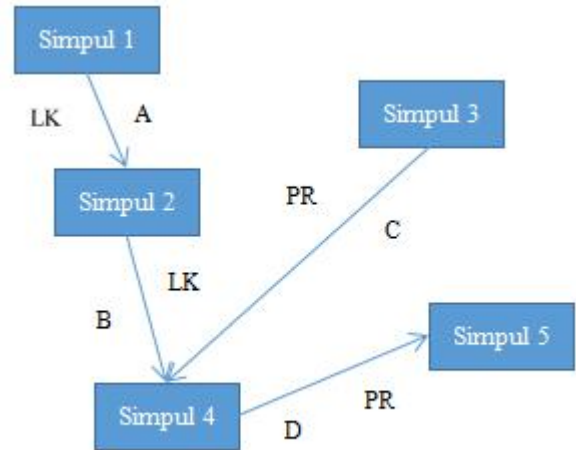
Jika graf berarah berbobot sederhana yang ada di bab sebelumnya dimodifikasi untuk memodelkan relasi keluarga, akan diperoleh graf berarah berbobot sederhana yang “baru” (berdasarkan gambar 1 di bab sebelumnya) dimana:

- Setiap simpul, dalam hal ini 1, 2, 3, 4 dan 5, merupakan suatu simpul “Relasi” yang menghubungkan beberapa

anggota keluarga untuk setiap unitnya (dibahas di subbab berikutnya).

- Nama setiap sisi (busur), dalam hal ini A, B, C, dan D, merupakan nama (atau info) setiap anggota keluarga.
- Bobot di setiap sisi (busur) adalah *gender* (dalam hal ini, 1 menjadi LK atau laki-laki dan 2 menjadi PR atau perempuan).

Contoh graf berarah berbobot sederhana yang termodifikasi adalah seperti gambar di bawah ini:



Gambar 2 (dibuat oleh penulis)

3.2. Konsep Unit dari Relasi Keluarga

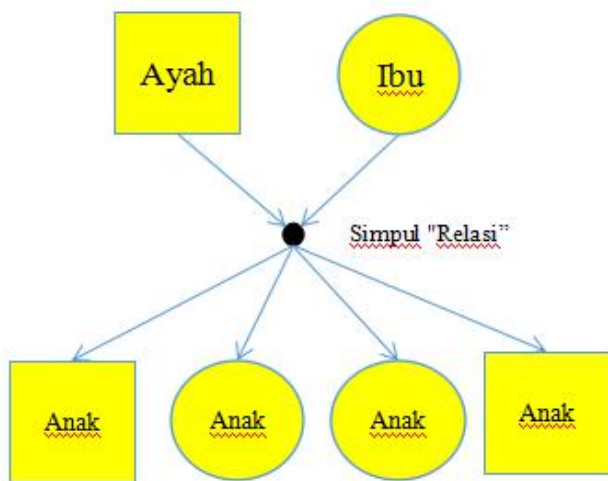
Menurut Kamus Besar Bahasa Indonesia (KBBI) versi kbbi.web.id, Pengertian unit adalah sebagai berikut:

- Unit adalah bagian terkecil dari sesuatu yang dapat berdiri sendiri; satuan: keluarga merupakan -- sosial yang paling kecil; desa sebagai -- kelompok hidup masyarakat atau
- Unit adalah kadar yang digunakan untuk mengukur (menilai dan sebagainya) sesuatu; dasar ukuran.

Pemodelan yang digambarkan pada subbab sebelumnya diperoleh setelah penulis mendapatkan suatu konsep dalam menyederhanakan persoalan relasi keluarga yang dibahas di bab pendahuluan.

Dasar dari pemodelan relasi keluarga ini adalah kegiatan menyederhanakan relasi keluarga yang besar dan rumit menjadi unit-unit (satuan terkecil) berupa ayah, ibu, dan anak-anaknya. Konsep pemecahan atau penyederhanaan ini sangat sesuai dengan definisi keluarga menurut Kamus Besar Bahasa Indonesia (KBBI) versi kbbi.web.id yang penulis sampaikan di bab pertama.

Visualisasi dari konsep unit terkecil berdasarkan definisi keluarga di atas adalah seperti gambar di bawah ini:



Gambar 3 (dibuat oleh penulis)

Jadi, dari unit relasi keluarga di atas, setiap simpul “Relasi” dalam graf berarah berbobot sederhana yang sudah dirancang ini, memiliki sifat-sifat sebagai berikut:

- Ada dua sisi (busur) yang menuju simpul “Relasi” tersebut.
 - ✓ Jika bobot sisi (busur) tersebut adalah 1 atau laki-laki, maka sisi (nama sisi) tersebut berperan sebagai **ayah** terhadap simpul “Relasi” tersebut.
 - ✓ Jika bobot sisi (busur) tersebut adalah 0 atau perempuan, maka sisi (nama sisi) tersebut berperan sebagai **ibu** terhadap simpul “Relasi” tersebut.
- Ada beberapa sisi (busur) yang mengarah menjauh dari simpul “relasi” tersebut. Sisi (nama sisi) tersebut berperan sebagai **anak** terhadap simpul “Relasi”. Bobot pada busur yang berperan sebagai anak digunakan untuk menentukan apakah sisi (nama sisi) adalah anak laki-laki atau anak perempuan terhadap simpul “Relasi” tersebut.

3.3. Penerapan Dalam Model Graf Relasi Keluarga

Setelah dibahas tentang pemodelan graf yang dipilih untuk memodelkan relasi keluarga dan unit (satuan) relasi keluarga yang digunakan di dalamnya, mari kita lihat bagaimana penjelasan dari model yang sudah dibuat.

Sebagai contoh, dari gambar 2 di halaman sebelumnya, diperoleh beberapa kesimpulan, yaitu:

- **A adalah ayah dari B** karena A berperan sebagai orang tua berbobot LK (laki-laki) atau dapat dikatakan sebagai ayah dan B berperan sebagai anak terhadap simpul “Relasi” 2.
- **B adalah ayah dari D** karena B berperan sebagai orang tua berbobot LK (laki-laki) atau dapat dikatakan sebagai ayah dan D berperan sebagai anak terhadap simpul “Relasi” 4.
- **C adalah ibu dari D** karena C berperan sebagai orang tua berbobot PR (perempuan) atau dapat dikatakan sebagai ibu dan D berperan sebagai anak terhadap simpul “Relasi” 4.
- **D adalah anak dari B** karena dari poin sebelumnya, B adalah orang tua (ayah) dari D.

- **D adalah anak dari C** karena dari poin sebelumnya, C adalah orang tua (ibu) dari D.
- **B adalah suami dari C** karena dari poin-poin sebelumnya, sudah didapat kesimpulan bahwa C adalah ayah D dan B adalah ibu dari D maka pastilah B adalah suami dari C.
- **C adalah istri dari B** karena dari poin-poin sebelumnya, sudah didapat kesimpulan bahwa C adalah ayah D dan B adalah ibu dari D maka pastilah C adalah istri dari B.
- **A adalah kakek dari D** karena dari poin-poin sebelumnya, sudah didapat kesimpulan bahwa A adalah ayah dari B dan B adalah orang tua D.
- **D adalah cucu dari A** karena dari poin sebelumnya diperoleh kesimpulan bahwa D adalah orang tua dari orang tua dari A.
- **A adalah mertua dari C** karena dari poin sebelumnya, A adalah orang tua (ayah) dari suami dari C.
- **C adalah menantu dari A** karena dari poin sebelumnya, A adalah mertua dari C.

Lebih jauh lagi, bisa jadi, diperoleh kesimpulan dari gambar 2 sebagai berikut:

- A adalah anak dari seorang anggota keluarga yang berperan sebagai ayah atau ibu terhadap simpul “Relasi” 1.
- D adalah ayah dari seorang anggota keluarga yang berperan sebagai anak terhadap simpul “Relasi” 5.

Dapat dilihat bahwa inti dari penentuan relasi-relasi di atas berasal dari konsep unit relasi keluarga yang sudah dikenalkan di subbab sebelumnya, yaitu ayah, ibu, dan anak. Demikian contoh-contoh pengambilan simpulan berdasarkan model graf berarah berbobot sederhana yang sudah dirancang untuk relasi keluarga ini. Dengan begitu, sudah jelas lah bagaimana menerapkan konsep pemodelan graf berarah berbobot sederhana yang dirancang ini.

IV IMPLEMENTASI REGEX DALAM PYTHON DAN MODEL GRAF RELASI KELUARGA DALAM BAHASA C

4.1. Perkiraan Kode Regex dalam Python

Regular Expression digunakan untuk melakukan perbandingan antara nama yang dicari dengan nama-nama yang ada dalam data relasi keluarga.

```
import re
def isSama (nama, X) :
    if (len(nama) == len(X)) :
        self.re_test = re.compile(nama, re.IGNORECASE)
        return self.re_test.match(X)
    else :
        return false
```

*belum dicompile

4.2. Tipe Bentuk dan Selektor

Dalam implementasi program yang dibuat oleh penulis, info yang disimpan untuk setiap anggota keluarga adalah gender (yang menjadi bobot sisi atau busur) dan nama (yang menjadi identitas dari suatu sisi atau busur).

Setiap nama dan simpul diwakilkan oleh suatu **bilangan bulat positif** sebagai indeks tempat data tersebut disimpan dalam array atau matriks, sedangkan bilangan bulat nol digunakan untuk menunjukkan ketidakterdisebutan data tersebut.

Ada dua tipe bentukan dan selektor yang menjadi dasar utama dari program ini yaitu:

- Tipe bentukan dan selektor dari nama anggota keluarga

```
#define BrsMax 100
#define KolMax 50

typedef struct {
    char Mem[BrsMax+1][KolMax+1];
    int IdxEffMax;
} MChar;

#define ElmtM(M,i,j) (M).Mem[(i)][(j)]
//untuk j=1, elemen menentukan gender (p atau l).

#define IEMM(M) (M).IdxEffMax
```

- Tipe bentukan dan selektor dari simpul “Relasi”

```
typedef struct {
    int Tab[BrsMax+1];
    int IdxEffMax;
} ArrInt;

#define Elmt(T,i) (T).Tab[(i)]
#define Neff(T) (T).IdxEffMax

typedef struct {
    int A; // Ayah
    int I; // Ibu
    ArrInt TAnak;
} ElType;

typedef struct {
    ElType Tab[BrsMax+1];
    int IdxEffMax;
} Arr;

#define Ayah(T,i) (T).Tab[(i)].A
//Idx orang yang berperan menjadi ayah di simpul ke-i

#define Ibu(T,i) (T).Tab[(i)].I
//Idx orang yang berperan menjadi ibu di simpul ke-i

#define Anak(T,i,j) (T).Tab[(i)].TAnak.Tab[(j)]
//Idx orang yang berperan menjadi anak ke-j di simpul ke-i

#define IEMT(T) (T).IdxEffMax
```

4.3. Bentuk File Eksternal

Dalam implementasi program yang dibuat oleh penulis, data-data mengenai sisi dan simpul disimpan dalam file eksternal (terpisah dalam dua file).

- Contoh file eksternal nama anggota keluarga

```
1|lAkdan.
2|pAzmaini.
3|lAzmil.
4|pHanidar.
5|lAkmal.
6|pSukmanidar.
7|lAhmad Zuhri.
8|pAuffah Yumni.
9|lZulhadi.
10|pKhairunnisa.
11|lRudianto.
12|pHusna Hayati.
13|lJon Hendri.
14|pFaza Khairani.
```

```
15|lJamal Irsyad.
16|pHernelly.
17|lRusdan.
18|pWidya Ningsih.
19|pShafiatul Asma.
20|lAzhar Fuadi.
21|pTri Setia Ningsih.
```

- ✓ Bilangan bulat pertama adalah indeks dari suatu info anggota keluarga (*gender* dan nama).
- ✓ Karakter | adalah pembatas sebagai alat bantu dalam membaca file eksternal.
- ✓ Karakter p atau l setelah | adalah *gender* dari seorang anggota keluarga (dalam model graf, diwakilkan oleh bobot sisi atau busur)
- ✓ Gabungan karakter setelah *gender* sampai sebelum titik adalah nama seorang anggota keluarga (dalam model graf, diwakilkan oleh sisi (nama sisi) atau busur).
- ✓ Karakter titik adalah *mark* sebagai alat bantu dalam membaca file.

- Contoh file eksternal simpul “Relasi”

```
1|1|2|4|5|7|9|12|14|15|17|.
2|3|4|20|22|23|.
3|5|6|24|25|26|27|.
4|7|8|28|29|30|31|.
5|9|10|32|33|.
6|11|12|34|35|36|.
7|13|14|37|.
8|15|16|38|.
9|17|18|39|40|.
10|15|19|41|.
11|20|21|.
```

- ✓ Bilangan bulat pertama adalah indeks dari suatu simpul “Relasi”.
- ✓ Setiap karakter | adalah pembatas sebagai alat bantu dalam membaca file eksternal.
- ✓ Bilangan bulat kedua adalah indeks dari seorang anggota keluarga yang berperan sebagai ayah terhadap simpul “Relasi” tertentu.
- ✓ Bilangan bulat ketiga adalah indeks dari seorang anggota keluarga yang berperan sebagai ibu terhadap simpul “Relasi” tertentu.
- ✓ Bilangan bulat ketiga dan seterusnya adalah indeks dari seorang anggota keluarga yang berperan sebagai anak terhadap simpul “Relasi” tertentu.
- ✓ Karakter titik adalah *mark* sebagai alat bantu dalam membaca file.

4.4. Kode Program Relasi Keluarga

Dalam hal ini, penulis mencantumkan contoh potongan-potongan program yang berupa fungsi/prosedur yang mengolah data-data simpul “Relasi” keluarga sehingga mengembalikan beberapa nilai indeks yang mengacu kepada info (nama dan *gender*) anggota keluarga.

- Fungsi untuk mendapatkan indeks ayah dari anggota keluarga yang berindeks IdxIn

```

int GetIdxAyah (Arr T, int IdxIn) {
    boolean Found = false;
    int i,j;

    if (IdxIn == 0) {
        return 0;
    }
    else {
        i = 0;
        do {
            i++;
            j = 1;
            while ((!Found) && (Anak(T,i,j) != 0)) {
                if (Anak(T,i,j) == IdxIn) {
                    Found = true;
                }
                else {
                    j++;
                }
            }
        } while ((!Found) && (i < IEMT(T)));
        if (Found) {
            return Ayah(T,i);
        }
        else {
            return 0;
        }
    }
}

```

- Fungsi untuk mendapatkan indeks Ibu dari anggota keluarga yang berindeks IdxIn

```

int GetIdxIbu (Arr T, int IdxIn) {
    boolean Found = false;
    int i,j;

    if (IdxIn == 0) {
        return 0;
    }
    else {
        i = 0;
        do {
            i++;
            j = 1;
            while ((!Found) && (Anak(T,i,j) != 0)) {
                if (Anak(T,i,j) == IdxIn) {
                    Found = true;
                }
                else {
                    j++;
                }
            }
        } while ((!Found) && (i < IEMT(T)));
        if (Found) {
            return Ibu(T,i);
        }
        else {
            return 0;
        }
    }
}

```

- Fungsi untuk mendapatkan indeks-indeks anak dari anggota keluarga yang berindeks IdxIn

```

ArrInt IdxIdxAnak (Arr T, int IdxIn) {
    ArrInt TAnak;
    int i,j;

    MakeEmptyArrInt (&TAnak);
    if (IdxIn != 0) {
        for (i = 1; i < IEMT(T); i++) {
            if (ElmtM(M,IdxIn,1) == '1') {
                if (Ayah(T,i) == IdxIn) {
                    j = 1;
                    while (Anak(T,i,j) != 0) {
                        Neff(TAnak)++;
                        Elmt(TAnak,Neff(TAnak)) = Anak(T,i,j);
                        j++;
                    }
                }
                else {
                    if (Ibu(T,i) == IdxIn) {
                        j = 1;
                        while (Anak(T,i,j) != 0) {
                            Neff(TAnak)++;
                            Elmt(TAnak,Neff(TAnak)) = Anak(T,i,j);
                            j++;
                        }
                    }
                }
            }
        }
    }

    return TAnak;
}

```

- Fungsi untuk mendapatkan indeks-indeks anak dari anggota keluarga yang berindeks IdxIn

```

ArrInt IdxIdxAdik (Arr T, int IdxIn) {
    ArrInt TAdik;
    int i,j,k;

    MakeEmptyArrInt (&TAdik);
    if (IdxIn != 0) {
        i = 1;
        boolean Found = false;
        while ((!Found) && (i < IEMT(T))) {
            j = 1;
            while ((Anak(T,i,j) != 0) && (!Found)) {
                if (Anak(T,i,j) == IdxIn) {
                    Found = true;
                }
                else {
                    j++;
                }
            }
            if (!Found) {
                i++;
            }
        }

        if (Found) {
            k = j+1;
            while (Anak(T,i,k) != 0) {
                Neff(TAdik)++;
                Elmt(TAdik,Neff(TAdik)) = Anak(T,i,k);
                k++;
            }
        }
    }

    return TAdik;
}

```

*Penentuan apakah dia adik dapat dilakukan dalam program ini karena di dalam tipe data simpul "Relasi" digunakan asumsi: "Array of anak terurut dari paling tua sampai paling muda."

- Fungsi untuk mendapatkan indeks-indeks suami dari anggota keluarga yang berindeks IdxIn

```

ArrInt IdxIdxSuami (Arr T, int IdxIn) {
    ArrInt TSuami;
    int i,j,k;

    MakeEmptyArrInt (&TSuami);
    if (IdxIn != 0) {
        if (ElmtM(M,IdxIn,1) == 'p') {
            for (i = 1; i < IEMT(T); i++) {
                if (Ibu(T,i) == IdxIn) {
                    Neff(TSuami)++;
                    Elmt(TSuami,Neff(TSuami)) = Ayah(T,i);
                }
            }
        }
    }

    return TSuami;
}

```

*Suami dari seorang perempuan dalam satu waktu pastilah satu, tetapi asumsi dalam program, bisa jadi, seorang perempuan cerai dan menikah lagi.

- Fungsi untuk mendapatkan indeks-indeks istri dari anggota keluarga yang berindeks IdxIn

```

ArrInt IdxIdxIstri (Arr T, int IdxIn) {
    ArrInt TIstri;
    int i,j,k;

    MakeEmptyArrInt (&TIstri);
    if (IdxIn != 0) {
        if (ElmtM(M,IdxIn,1) == 'l') {
            for (i = 1; i < IEMT(T); i++) {
                if (Ayah(T,i) == IdxIn) {
                    Neff(TIstri)++;
                    Elmt(TIstri,Neff(TIstri)) = Ibu(T,i);
                }
            }
        }
    }

    return TIstri;
}

```

*Istri dari seorang laki-laki dalam satu waktu pastilah tidak lebih dari empat (jika tidak sanggup adil, maka satu saja), tetapi asumsi dalam program, bisa jadi, seorang laki-laki cerai dan menikah lagi.

- Prosedur untuk menuliskan nama dari indeks yang sudah diperoleh

```

void TulisNama (MChar M, int IdxIn) {
    if (IsNamaAda(M,IdxIn)) {
        int j = 2;
        while (ElmtM(M,IdxIn,j) != '.') {
            printf("%c",ElmtM(M,IdxIn,j));
            j++;
        }
    }
    else {
        printf("Tidak diketahui");
    }
}

```

*Jika nama tidak ada di dalam data (file eksternal info / nama dan gender), maka akan dituliskan string "Tidak diketahui".

4.4. Hasil Eksekusi Program

Berikut ini adalah contoh-contoh eksekusi program (perlu diketahui bahwa ">>" menandakan ada input dari user).

- Contoh masukan satu relasi/hubungan yang memberikan satu hasil

```

>> Berapa banyak relasi yang ingin diinput? 1
> Silahkan Pilih pilihan yang ada di bawah ini:
1. Ayah dari ...
2. Ibu dari ...
3. Abang dari ...
4. Kakak dari ...
5. Adik dari ...
6. Suami dari ...
7. Istri dari ...
8. Anak dari ...

>> Masukkan relasi ke-1 : 1

>> Masukkan nama orangnya : Hamdi Ahmad Zuhri

> Ayah dari Hamdi Ahmad Zuhri adalah :
> Ahmad Zuhri

```

- Contoh masukan dua relasi/hubungan yang memberikan satu hasil

```

>> Berapa banyak relasi yang ingin diinput? 2
> Silahkan Pilih pilihan yang ada di bawah ini:
1. Ayah dari ...
2. Ibu dari ...
3. Abang dari ...
4. Kakak dari ...
5. Adik dari ...
6. Suami dari ...
7. Istri dari ...
8. Anak dari ...

>> Masukkan relasi ke-1 : 1

>> Masukkan relasi ke-2 : 6

>> Masukkan nama orangnya : Auffah Yumni

> Ayah dari Suami dari Auffah Yumni adalah :
> Akdan

```

- Contoh masukan dua relasi/hubungan yang memberikan lebih dari satu hasil (masih ada *bug* atau kesalahan)

```

>> Berapa banyak relasi yang ingin diinput? 2
> Silahkan Pilih pilihan yang ada di bawah ini:
1. Ayah dari ...
2. Ibu dari ...
3. Abang dari ...
4. Kakak dari ...
5. Adik dari ...
6. Suami dari ...
7. Istri dari ...
8. Anak dari ...

>> Masukkan relasi ke-1 : 8

>> Masukkan relasi ke-2 : 5

>> Masukkan nama orangnya : Faza Khairani

> Anak dari Adik dari Faza Khairani adalah :
1. Tidak diketahui
2. Rusdan
3. Widya Ningsih
4. Muhammad Fauzil Hasan
5. Afra
6. Maisa Febiana

```


V. KENDALA UMUM

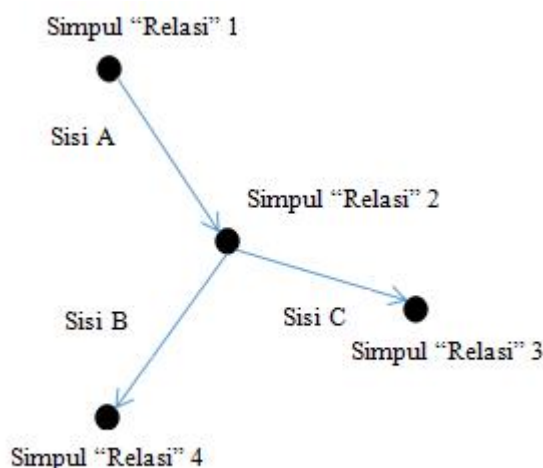
Kendala umum terhadap proses “pengolahan relasi keluarga dengan pendekatan teori graf berarah berbobot sederhana” adalah sebagai berikut:

- Penrapan regex dan relasi keluarga masih terpisah dan berbeda bahasa.
- Dapat terjadi kesalahan data di file eksternal (berupa konten ataupun strukturnya).
- Masih ada kesalahan dalam program (terlihat di contoh hasil ekskusi ketiga di bab sebelumnya).

VI. SIMPULAN DAN SARAN

Dari pembahasan penulis dari pendahuluan, pemodelan graf, sampai kepada implementasi program (dalam hal ini, penulis menggunakan bahasa Python untuk Regular Expression dan bahasa C untuk relasi keluarga), diperoleh beberapa simpulan sebagai berikut:

- Regular expressions are patterns used to match character combinations in strings. In JavaScript, regular expressions are also objects. These patterns are used with the exec and test methods of RegExp, and with the match, replace, search, and split methods of String. This chapter describes JavaScript regular expressions.
- Untuk mempermudah manusia mengetahui info seseorang anggota keluarga berdasarkan suatu relasi, diperlukan alat bantu yang dapat mengolah setiap dokumentasi relasi-relasi keluarga tersebut.
- Graf didefinisikan sebagai himpunan pasangan himpunan (V,E) dimana V adalah himpunan tak-kosong dari simpul-simpul dan E adalah himpunan dari busur yang menghubungkan sepasang simpul.
- Relasi keluarga dimodelkan dengan graf berarah berbobot sederhana dimana setiap sisi (A, B, C, dan D) merupakan nama dari anggota keluarga dan setiap sisi memiliki bobot (laki-laki atau perempuan) yang mewakili *gender*. Contoh sederhana dari model tersebut adalah gambar 4 di bawah ini.



Gambar 4 (dibuat oleh penulis)

- Dasar dari pemodelan relasi keluarga adalah kegiatan menyederhanakan relasi keluarga yang besar dan rumit menjadi unit-unit (satuan terkecil) berupa ayah, ibu, dan anak-anaknya (divisualisasikan oleh gambar 3) sehingga

setiap simpul “Relasi” merupakan pusat (simpul “Relasi” 1, 2, 3, dan 4 sebagai contoh di gambar 4) dari masing-masing unit relasi keluarga.

- Program “Pengolahan Relasi Keluarga” terdiri atas data info dan simpul (file eksternal), tipe bentukan beserta selektor,, dan kode program keseluruhan yang dibuat berdasarkan pemodelan graf berarah berbobot sederhana dan konsep unit relasi keluarga.

Di samping poin-poin simpulan, penulis memberikan saran terhadap “Pengolahan Relasi Keluarga” ini. Pembahasan ini masih sangat sederhana dan sangat mungkin untuk dikembangkan lebih jauh. Salah-satu kesempatan pengembangannya adalah menambah info dari anggota keluarga (selain nama dan *gender*), sebagai contoh: tempat tanggal lahir, nomor kontak, dan profesi.

REFERENSI

- [1] https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions 14 Mei 2018, pk 10.15 WIB
- [2] Rinaldi Munir, Diktat Kuliah Matematika Diskrit, 2003
- [3] <https://docs.python.org/2/library/re.html> 14 Mei 2018, pk 10.15 WIB
- [4] <http://danysatriokintoko.blogspot.co.id>, 3 Desember 2017, pk. 11.25 WIB
- [5] <http://ranindrar5.blogspot.co.id/2016/04/contoh-soal-graf-berarah.html>, 3 Desember 2017, pk. 11.25 WIB
- [6] <http://danysatriokintoko.blogspot.co.id/2013/02/jenis-jenis-graf.html>, 3 Desember 2017, pk. 11.45 WIB
- [7] <https://kbbi.web.id>, 3 Desember 2017, pk. 12.30 WIB
- [8] <http://matematikajeng.blogspot.co.id/2012/08/sejarah-lahirnya-teori-graf.html>, 3 Desember 2017, pk. 16.54 WIB
- [9] <http://whandany.blogspot.co.id/2011/07/langkah-pembuatan-model-simulasi.html>, 3 Desember 2017, pk. 18.47 WIB
- [10] <https://guruinformatika.blogspot.co.id/2015/04/materi-ekspresi-reguler-teori-bahasa.html> 14 Mei 2018, pk 10.15 WIB

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, 14 Mei 2018

Hamdi Ahmad Zuhri
13516150