

Aplikasi *Pattern Matching* pada Database Microsoft Excel

Fikra Hadi Ramadhan / 13518036
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13518036@std.stei.itb.ac.id

Abstract—Banyaknya bentuk keorganisasian mahasiswa Institut Teknologi Bandung tidak akan lepas dari database, baik database anggotanya, maupun database keperluan lainnya yang dimana banyak menggunakan Microsoft Excel dalam penyimpanan datanya. Oleh sebab itu, aplikasi *pattern matching* pada Microsoft Excel bisa setidaknya membantu mahasiswa dalam pengolahan database.

Keywords—Database; Microsoft Excel; Pattern Matching; ITB; Organisasi

I. PENDAHULUAN

Pada zaman serba ilmu pengetahuan ini, mahasiswa terus berkembang seiring berjalannya waktu. Banyak wadah untuk mahasiswa mengembangkan ilmu maupun bakat dan minatnya. Sesuai dengan Tri dharma perguruan tinggi yang pertama yaitu pendidikan dan pengajaran, mahasiswa dituntut untuk selalu mengembangkan ilmu pengetahuannya dibidang akademik. Mahasiswa juga diharapkan tidak hanya fokus pada akademiknya, tetapi juga harus menyeimbangkannya dengan kegiatan non akademik seperti organisasi.

ITB memiliki banyak wadah untuk berorganisasi. Seperti UKM (unit kegiatan mahasiswa), Himpunan, Keluarga Mahasiswa ITB dan masih banyak lagi. Setiap organisasi memiliki sistemnya sendiri, akan tetapi setiap organisasi pasti membutuhkan data-data anggotanya. Dalam pengambilan maupun pengolahan data anggota, organisator bisa memanfaatkan aplikasi-aplikasi yang dapat menunjang pengambilan dan pengolahan data. Oleh sebab itu sebagai mahasiswa yang sekaligus sebagai organisator harus mengetahui cara-cara mengolah data dengan baik.

II. DASAR TEORI

A. Organisasi

Secara umum, organisasi adalah sebuah wadah untuk sekumpulan orang yang bekerja sama secara rasional serta sistematis yang dipimpin atau terkendali untuk mencapai tujuan tertentu dengan memanfaatkan sumber daya yang ada di dalamnya. Sementara dalam dunia bisnis, organisasi adalah sebuah wadah untuk sekumpulan orang yang melakukan kolaborasi untuk mencapai tujuan secara komersial dengan struktur yang jelas serta memiliki budaya kerja khusus.

B. Database

Database merupakan sekumpulan informasi yang saling berkaitan pada suatu subjek tertentu pada tujuan tertentu pula. Database dapat diartikan sebagai kumpulan data yang disimpan secara sistematis di dalam computer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (*software*) program atau aplikasi untuk menghasilkan informasi. Perangkat lunak yang digunakan untuk mengelola dan memanggil query basis data disebut dengan system manajemen basis data (*database management system, DBMS*) dalam sistem basis data yang dapat dipelajari dalam ilmu informasi. Dengan kata lain, semua akses ke basis data akan ditangani oleh DBMS. DBMS ini menjadi lapisan yang menghubungkan basis data dengan program aplikasi untuk memastikan bahwa basis data tetap terorganisasi secara konsisten dan dapat diakses dengan mudah.

DBMS menangani beberapa fungsi seperti pendefinisian data, menangani permintaan pengguna untuk mengakses data, memeriksa sekuriti dan integrity data yang didefinisikan oleh DBA (*Database Administrator*), menangani kegagalan dalam pengaksesan data yang disebabkan oleh kerusakan system maupun media penyimpanan (*disk*) dan juga menangani unjuk kerja sama fungsi secara efisien.

Tujuan utama DBMS adalah untuk memberikan tinjauan abstrak data kepada pengguna. Jadi system menyembunyikan informasi tentang bagaimana data disimpan, dipelihara dan juga bisa diakses secara efisien. Pertimbangan efisien yang dimaksud adalah rancangan struktur data yang kompleks tetapi masih bisa digunakan oleh pengguna awam tanpa mengetahui kompleksitas strukturnya.

Dalam organisasi, tidak akan lepas dari database, baik database anggota maupun database lainnya yang diperlukan dalam suatu organisasi. Banyak aplikasi yang bisa digunakan untuk membantu dalam pembuatan database. Aplikasi yang digunakan tergantung keperluan dari organisasi tersebut, apakah yang memudahkan pekerjaan tapi keamanan kurang, atau ingin keamanan tinggi tetapi pekerjaannya sulit.

C. Microsoft Excel

Microsoft Excel adalah sebuah program atau aplikasi yang merupakan bagian dari paket instalasi Microsoft Office, berfungsi untuk mengolah data menggunakan spreadsheet yang terdiri dari baris dan kolom untuk mengeksekusi perintah. Microsoft Excel telah menjadi software pengolah data / angka terbaik di dunia, selain itu Microsoft Excel telah didistribusikan secara multi-platform. Microsoft Excel tidak hanya tersedia dalam platform Windows, Microsoft Excel juga tersedia di MacOS, Android dan Apple.

Microsoft Excel secara fundamental menggunakan spreadsheet untuk manajemen data serta melakukan fungsi-fungsi Excel yang lebih dikenal dengan formula Excel. Excel merupakan program spreadsheet elektronik. Spreadsheet adalah kumpulan dari Sel yang terdiri atas baris dan kolom tempat memasukkan data pada Microsoft Excel. Jumlah Sel Microsoft Excel 2016 terdiri dari 1.048.576 Baris dan 16.384 Kolom atau 17.179.869.184 Sel.

Sebagai program pengolah angka terpopuler, Microsoft Excel mempunyai banyak kelebihan dan beberapa kekurangan untuk penggunaan tertentu.

a) Kelebihan Microsoft Excel

1. *User interface* yang mudah untuk dipahami.
2. Kompatibilitas dengan berbagai platform/sistem operasi.
3. Mudah dipelajari untuk pengguna pemula.
4. Tersedia lisensi dalam versi grosir.
5. Mempunyai ekstensi (.xls) terpopuler untuk software spreadsheet.
6. Dapat membaca ekstensi standar spreadsheet (.csv).
7. Fitur pivot untuk mempermudah manajemen data.
8. Spreadsheet yang besar, dapat digunakan sebagai alternatif SQL untuk penggunaan sederhana.
9. Resource RAM dan memory kecil dibanding program sejenis.
10. Digunakan oleh berbagai industri, instansi dan pekerjaan.
11. Mendukung Visual Basic
12. Menyediakan fitur *Research Pane* untuk mempermudah mencari referensi analisis data dengan Microsoft Excel menurut para ahli terkait topik yang diteliti.

b) Kekurangan Microsoft Excel

1. Akses fungsi tertentu seperti fungsi statistik terbatas.
2. Jumlah sel terbatas.
3. Add-ins untuk disiplin ilmu tertentu seperti *neural network*, *fuzzy logic* tidak powerfull dibandingkan software sejenis contoh: MATLAB dan SAS .

Microsoft Excel digunakan di berbagai bidang pekerjaan, baik usaha kecil maupun perusahaan berskala internasional. Adapun beberapa fungsi dan kegunaan Microsoft Excel adalah sebagai berikut:

- Membuat, mengedit, mengurutkan, menganalisis, meringkas, dan memformat data serta grafik.
- Membuat catatan keuangan dan anggaran keuangan.
- Menghitung dan mengelola investasi, pinjaman, penjualan, inventaris.
- Melakukan analisis dan riset harga.
- Melakukan perhitungan statistika.
- Membantu berbagai sektor bisnis untuk mempermudah melakukan laporan keuangan.
- Membuat daftar nilai sekolah maupun universitas.
- Konversi mata uang.
- Membuat grafik persamaan matematika.
- Membuat program Excel dengan Visual Basic.
- Melakukan penelitian dengan berbagai metode penelitian.
- Sarana pembelajaran komputer dan logika.
- Dan lain-lain.

D. Pattern Matching

Pattern Matching adalah sebuah algoritma pencarian sebuah string dalam string lainnya. Didefinisikan ada sebuah string bernama Teks yang berukuran n karakter dan sebuah string lainnya yang bernama Pattern yang berukuran m karakter dengan asumsi $m < n$. Sebagai contoh:

Teks : "Terkonfirmasi sudah ada dua belas warga yang terkontaminasi COVID-19"

Pattern : "terkontaminasi"

Maka algoritma akan mencari di dalam string Teks yang bersesuaian dengan string Pattern.

Aplikasi dari *pattern matching* sangat banyak digunakan dalam keseharian, seperti pencarian dalam sebuah file (*find*) pencarian berbasis web seperti *Google*. Tidak hanya itu, *pattern matching* juga bisa digunakan dalam analisis citra (sidik jari) dimana nantinya titik-titik yang ditemukan akan diterjemahkan menjadi sebuah teks. Pada *bionformatics* juga bisa menggunakan *pattern matching* seperti mencari pencocokan rantai asam amino pada rantai DNA.

Pada konsepnya, algoritma akan memecah string menjadi huruf-huruf yang akan dimasukkan dalam sebuah array. Algoritma *pattern matching* juga menggunakan konsep-konsep seperti *prefix* yaitu substring yang dimulai dari awal string dan juga *suffix* yaitu substring yang dimulai dari akhir string. Ada banyak jenis algoritma *pattern matching*, seperti berikut ini:

a) Algoritma Brute Force

Dengan cara memeriksa satu persatu pattern dengan teks dari awal sampai akhir. Ini memakan waktu yang cukup lama dalam matchingnya. Pengaplikasian Algoritma Brute Force pada Java adalah sebagai berikut (*Sumber: Slide Materi Strategi Algoritma*):

```
public static int brute(String text,String pattern) {
    int n = text.length(); // n is length of text
    int m = pattern.length(); // m is length of pattern
    int j;
    for(int i=0; i <= (n-m); i++) {
        j = 0;
```

```

        while ((j < m) &&(text.charAt(i+j)==
pattern.charAt(j))) {
            j++;
        }
        if (j == m) {
            return i; // match at i
        }
    }
    return -1; // no match
} // end of brute()

public static void main(String args[]) {
    if (args.length != 2) {
        System.out.println("Usage: java BruteSearch
<text> <pattern>");
        System.exit(0);
    }
    System.out.println("Text: " + args[0]);
    System.out.println("Pattern: " + args[1]);
    int posn = brute(args[0], args[1]);
    if (posn == -1) {
        System.out.println("Pattern not found");
    }
    else {
        System.out.println("Pattern starts at posn "+
posn);
    }
}

```

```

        j++;
    }
    else if (j > 0) {
        j = fail[j-1];
    }
    else {
        i++;
    }
}
return -1; // no match
} // end of kmpMatch()

public static int[] computeFail(String pattern) {
    int fail[] = new int[pattern.length()];
    fail[0] = 0;
    int m = pattern.length();
    int j = 0;
    int i = 1;
    while (i < m) {
        if (pattern.charAt(j) == pattern.charAt(i)) {
//j+1 chars match
            fail[i] = j + 1;
            i++;
            j++;
        }
        else if (j > 0) { // j follows matching prefix
            j = fail[j-1];
        }
        else { // no match
            fail[i] = 0;
            i++;
        }
    }
}
return fail;
} // end of computeFail()

public static void main(String args[]) {
    if (args.length != 2) {
        System.out.println("Usage: java KmpSearch
<text> <pattern>");
        System.exit(0);
    }
    System.out.println("Text: " + args[0]);
    System.out.println("Pattern: " + args[1]);
    int posn = kmpMatch(args[0], args[1]);
    if (posn == -1) {
        System.out.println("Pattern not found");
    }
    else {
        System.out.println("Pattern starts at posn " +
posn);
    }
}

```

b) Algoritma Knuth-Morris-Pratt

Algoritma Knuth Morris Pratt (KMP) dikembangkan oleh D. E. Knuth, bersama dengan J. H. Morris dan V. R. Pratt. Untuk pencarian string dengan menggunakan algoritma Brute Force, setiap kali ditemukan ketidakcocokan pattern dengan teks, maka pattern akan digeser satu karakter ke kanan. Sedangkan Algoritma Knuth Morris Pratt bekerja dengan memanfaatkan pergeseran yang semaksimal mungkin dalam pencocokan string dalam teks. padanannya dalam bahasa Inggris. Aplikasi penerjemah haruslah melakukan pencocokan kata untuk mencari kata tersebut dalam bahasa asing. Haruslah diperhitungkan apakah kata tersebut merupakan bagian dari kata majemuk lainnya. Bentuk umum KMP adalah sebagai berikut (*Sumber: Slide Materi Strategi Algoritma*):

```

public static int kmpMatch(String text, String pattern) {
    int n = text.length();
    int m = pattern.length();
    int fail[] = computeFail(pattern);
    int i=0;
    int j=0;
    while (i < n) {
        if (pattern.charAt(j) == text.charAt(i)) {
            if (j == m - 1) {
                return i - m + 1; // match
            }
            i++;
        }
    }
}

```

c) Algoritma Boyer-Moore

Boyer-Moore secara rata-rata merupakan algoritma pencarian string yang paling baik jika dibandingkan dengan

algoritma pencarian string lainnya seperti Brute-Force ataupun Knuth-Morris-Pratt. Jika kita menggunakan fasilitas Find/Search pada berbagai aplikasi pengolah teks, web browser, dan aplikasi lainnya mungkin saja kita telah memanfaatkan algoritma Boyer-Moore dalam pencarian tersebut, karena algoritma ini paling banyak diimplementasikan dalam berbagai aplikasi untuk fasilitas pencarian teksnya.

Algoritma Boyer-Moore adalah salah satu algoritma untuk mencari suatu string di dalam teks, dibuat oleh R.M Boyer dan J.S Moore. Ide utama algoritma ini adalah mencari string dengan melakukan perbandingan karakter mulai dari karakter paling kanan dari string yang dicari. Bentuk umum BM adalah sebagai berikut (Sumber: Slide Materi Strategi Algoritma):

```
public static int bmMatch(String text, String pattern) {
    int last[] = buildLast(pattern);
    int n = text.length();
    int m = pattern.length();
    int i = m-1;
    if (i > n-1){
        return -1; // no match if pattern is
                // longer than text
    }
    int j = m-1;
    do {
        if (pattern.charAt(j) == text.charAt(i)){
            if (j == 0){
                return i; // match
            }
            else { // looking-glass technique
                i--;
                j--;
            }
        }
        else { // character jump technique
            int lo = last[text.charAt(i)]; //last
            occ
            i = i + m - Math.min(j, 1+lo);
            j = m - 1;
        }
    } while (i <= n-1);
    return -1; // no match
} // end of bmMatch()

public static int[] buildLast(String pattern) /* Return array
storing index of lastoccurrence of each ASCII char in
pattern. */{
    int last[] = new int[128]; // ASCII char set
    for(int i=0; i < 128; i++){
        last[i] = -1; // initialize array
    }
    for (int i = 0; i < pattern.length(); i++){
        last[pattern.charAt(i)] = i;
    }
    return last;
} // end of buildLast()
```

```
public static void main(String args[]){
    if (args.length != 2) {
        System.out.println("Usage: java BmSearch
<text> <pattern>");
        System.exit(0);
    }
    System.out.println("Text: " + args[0]);
    System.out.println("Pattern: " + args[1]);
    int posn = bmMatch(args[0], args[1]);
    if (posn == -1){
        System.out.println("Pattern not found");
    }
    else{
        System.out.println("Pattern starts at posn " +
posn);
    }
}
```

d) Regular Expression

Regular Expression, sering ditulis/disebut juga Regex / Regexp, adalah deretan karakter spesial yang mendefinisikan sebuah pola dalam pencarian teks. Bisa dibayangkan Regex itu seperti *wildcard* tetapi lebih pintar. Setiap bahasa pemrograman menyediakan *Regex engine* yang berbeda. Engine di Java tidak sama dengan PHP, beda juga dengan JavaScript. Karena beda engine, belum tentu ekspresi Regex yang disupport oleh satu bahasa juga disupport bahasa yang lain. Walaupun begitu, sintaks Regex tetap sama di mana-mana.

Bentuk paling dasar adalah mencari string literal. Misalnya regex "halo" akan cocok dengan kata "halo" (gambar bawah) karena kedua string tersebut identik.



Gambar 1 : Contoh 1 Penggunaan Regex (Sumber: Slide Materi Strategi Algoritma)

Jika terdapat lebih dari satu urutan string yang identik maka jumlah kecocokan akan menyesuaikan. Sebagai contoh, ada dua pola "halo" dalam "halohalo Bandung"



Gambar 2: Contoh 2 Penggunaan Regex
(Sumber: Slide Materi Strategi Algoritma)

Ada beberapa pembagian dalam regex yaitu sebagai berikut:

a) *Metacharacter*

Dalam regex, terdapat sejumlah karakter khusus yang mempengaruhi proses pencocokan string. Sebagai contoh, karakter titik “.” akan cocok dengan karakter apapun. Jadi regex .udi akan cocok dengan Budi, Rudi, yudi dan seterusnya.

Construct	Deskripsi
^...\$	Tanda awal dan akhir dar sebuah baris
{.....}	Untuk mendefinisikan range
[.....]	Untuk mendefinisikan character class
(.....)	Untuk mendefinisikan group
*	Untuk menyatakan perulangan minimal 0 kali sampai tak terhingga
+	Untuk menyatakan perulangan minimal satu kali sampai tak terhingga
?	Untuk menyatakan perulangan satu kali atau tidak sama sekali
	Operator logika untuk OR
\	Menjadikan karakter setelahnya tidak lagi dianggap sebagai metcharacter dan hanya akan dianggap sebagai karakter biasa
\t	Merepresentasikan karakter tab
\n	Merepresentasikan karakter newline
\r	Merepresentasikan carriage-return character
\f	Merepresentasikan form-feed character
\a	Merepresentasikan karakter bel

\e	Mereperesentasikan karakter escape
----	------------------------------------

b) *Character Classes*

Kelas Karakter yang dapat digunakan untuk mencari kesesuaian atau kecocokan karakter pada pola yang telah ditentukan.

Construct	Deskripsi
[abc]	a, b, atau c (simple class)
[^abc]	Semua karakter selain a,b,c (negasi)
[a-zA-Z]	a sampai z atau A sampai Z, inclusive (range)
[a-d[m-p]]	a sampai d atau m sampai p (gabungan)
[a-z&&[def]]	d, e atau f (irisan)
[a-z&&[^bc]]	a sampai z, kecuali b dan c (subtraksi)
[a-z&&[^m-p]]	a sampai z, dan bukan m sampai p (subtraksi)

c) *Predefined Character Class*

Kelas karakter yang sudah didefinisikan. Predefined class akan membuat regex lebih mudah dibaca dan mengurangi kesalahan.

Construct	Deskripsi
.	Semua karakter
\d	Digit [0-9]
\D	Non digit [^0-9] (hati-hati dengan huruf besar)
\s	Whitespace character [\t\n\x0B\f\r]
\S	Non whitespace character [^s]
\w	Word character [a-zA-Z_0-9]
\W	Non word character [^\w]

d) *Quantifier*

Quantifier digunakan untuk mendefinisikan jumlah perulangan pola. Quantifier punya peran sangat penting dalam regex.

Construct	Deskripsi
X?	X muncul satu atau tidak sama sekali
X*	X muncul nol atau banyak
X+	X muncul satu atau banyak
x{n}	X muncul tepat n kali
x{n,}	X muncul setidaknya n kali
x{n,m}	X muncul antara n sampai m kali

e) *Boundary Matchers*

Boundary matcher digunakan untuk mencari pola yang muncul di posisi tertentu. Misalnya di awal atau di akhir.

Construct	Deskripsi
^	Awal baris
\$	Akhir baris
\b	Batas kata
\B	Batas bukan kata
\G	Akhir match sebelumnya
\Z	Akhir dari input tapi untuk final terminator jika ada
\z	Akhir dari input

III. BATASAN MASALAH

Pemakaian *pattern matching* pada Microsoft Excel oleh penulis memiliki beberapa batasan, yaitu sebagai berikut:

- *Pattern Matching* dipakai dalam pencarian kata pada formula Microsoft Excel seperti formula COUNTIF().
- Contoh database yang dipakai adalah database yang dimiliki oleh penulis yang sudah disusun dengan sebaik mungkin agar mudah dipahami.

IV. IMPLEMENTASI DAN PENGUJIAN

Penulis akan memakai database mahasiswa baru SNMPTN ITB 2020 asal Sumatera Barat. Pada database ini penulis hanya menampilkan nama, asal daerah (kota / kabupaten), fakultas/sekolah, dan jenis kelamin. Juga penulis akan

menampilkan pengetahuan tentang ZOOM IN ITB 2020 yang merupakan acara pengabdian masyarakat mahasiswa minang ITB dengan pertanyaannya adalah tau dengan ZOOM IN ITB 2020, mendapatkan info dari mana, dan apakah mengikuti kegiatannya, kalau iya apa saja yang diikuti. Database ini diambil saat pengumuman penerimaan mahasiswa baru jalur SNMPTN 2020. Database berisi 65 responden dan penulis akan mengubah beberapa data seperti nama responden agar keamanan database terjaga.

Berikut beberapa pertanyaan yang penulis ajukan pada para responden:

a) Nama Lengkap

b) Gender

Template:

- Perempuan
- Laki-Laki

c) Asal Daerah

Template:

- Kabupaten Agam
- Kabupaten Dharmasraya
- Kabupaten Kepulauan Mentawai
- Kabupaten Lima Puluh Kota
- Kabupaten Padang Pariaman
- Kabupaten Pasaman
- Kabupaten Pasaman Barat
- Kabupaten Pesisir Selatan
- Kabupaten Sijunjung
- Kabupaten Solok
- Kabupaten Solok Selatan
- Kabupaten Tanah Datar
- Kota Bukittinggi
- Kota Padang
- Kota Padang Panjang
- Kota Pariaman
- Kota Payakumbuh
- Kota Sawahlunto
- Kota Solok

d) Masuk Fakultas/Sekolah apa?

Template:

- STEI
- SF
- SITH-R

- SITH-S
- SAPPK – G
- SAPPK – C
- SBM
- FMIPA
- FTI – G
- FTI – J
- FTI – C
- FTSL – G
- FTSL – J
- FTTM – G
- FTTM – C
- FTMD
- FITB – G
- FITB – C
- FSRD – G
- FSRD - C

e) Apakah tau tentang ZOOM IN ITB 2020?

Template:

- Tau
- Tidak Tau

f) Kalau tau, darimana mendapatkan informasinya?

Multiple Choice

Template:

- IG
- Twitter
- Youtube
- Line
- Roadshow ke sekolah
- Teman
- Orangtua
- Guru
- Tidak tau

g) Kalau tau, apakah kamu ikut kegiatannya? Apa saja?

Multiple Choice

Template:

- Roadshow
- Bedah Kampus
- Try Out se Nasional

- Tidak Ikut

Perlu diperhatikan bahwa template yang dimaksud adalah yang ada pada database, jika salah penulisan maka tidak menemukan kata yang diinginkan. Perhatikan tanda spasi dan tanda lainnya. Pada excel tidak memperdulikan huruf kecil atau huruf kapital. Penulis hanya akan menggunakan formula COUNTIF() untuk mencari banyaknya kata yang ada dalam suatu kondisi. Kondisi yang penulis ambil adalah sebuah kolom dalam sheet yang berbeda.

Beberapa percobaan yang akan penulis lakukan adalah sebagai berikut:

1) Menghitung banyaknya responden memiliki kata putri/putra pada nama lengkapnya

```
=COUNTIF(Makalah!A:A;
"*putri*")+COUNTIF(Makalah!A:A; "*putra*")
```

Hasilnya adalah 9. Makalah adalah nama sheet data, A:A adalah nama kolom yang berisi nama lengkap. Pemakaian "*" artinya muncul 0 atau banyak huruf selain putri/putra.

2) Menghitung banyaknya responden yang memiliki gender "Laki – Laki"

```
=COUNTIF(Makalah!C:C; "Laki - Laki")
```

Hasilnya adalah 0. Makalah adalah nama sheet data, C:C adalah nama kolom yang berisi gender. Hasilnya 0 karena penulisan "Laki – Laki" tidak sesuai dengan template yang harusnya tanpa spasi yaitu "Laki-Laki"

3) Menghitung banyaknya responden yang berasal dari "Kota Padang"

```
=COUNTIF(Makalah!D:D; "Kota Padang")
```

Hasilnya adalah 29. Makalah adalah nama sheet data, D:D adalah nama kolom yang berisi asal daerah. Jika menambahkan "*" seperti berikut

```
=COUNTIF(Makalah!D:D; "Kota Padang*")
```

Maka hasilnya adalah 30, karena ada template lain yang bernama "Kota Padang Panjang" yang dimana artinya responden yang berasal dari "Kota Padang Panjang" ada 1 orang.

4) Menghitung banyaknya responden yang masuk "FTI" baik ganesha, jatinangor, maupun Cirebon

```
=COUNTIF(Makalah!E:E; "FTI*")
```

Hasilnya adalah 5. Makalah adalah nama sheet data, E:E adalah nama kolom yang berisi fakultas/sekolah.

5) Menghitung banyaknya responden yang tau tentang acara ZOOM IN ITB 2020

```
=COUNTIF(Makalah!F:F; "Tau")
```

Hasilnya adalah 36. Makalah adalah nama sheet data, F:F adalah nama kolom yang berisi tentang pertanyaan "Apakah tau tentang ZOOM IN ITB 2020?".

- 6) Menghitung banyaknya responden yang tau info ZOOM IN ITB 2020 melalui media social media

```
=COUNTIF(Makalah!G:G;
"*IG*")+COUNTIF(Makalah!G:G;
"*Twitter*")+COUNTIF(Makalah!G:G;
"*Youtube*")+COUNTIF(Makalah!G:G; "*Line*")
```

Hasilnya adalah 50. Makalah adalah nama sheet data, G:G adalah nama kolom yang berisi tentang pertanyaan "Kalau tau, darimana mendapatkan infonya?". Karena pertanyaan ini merupakan multiple choice, maka menambahkan notasi "*" akan lebih teliti dalam mengambil datanya. Jika tidak menggunakan "*", maka sel yang berisi "IG, Twitter, Teman" tidak akan terhitung, sebaliknya jika menggunakan "*" maka akan terhitung. Akan tetapi kesalahannya disini adalah adanya penghitungan double dimana "IG, Twitter" akan terhitung 2 karena ada formula countif yang menemukan "*IG*" dan "*Twitter*" yang dimana dalam satu sel akan terhitung 2 kali, maka tidak dianjurkan untuk menggunakan cara ini, bisa menggunakan lainnya seperti formula regex yaitu membuat fungsi sendiri menggunakan VBA. Disini penulis tidak akan mengimplementasikannya, implementasinya akan diserahkan kepada pembaca.

- 7) Menghitung banyaknya responden yang mengikuti acara ZOOM IN ITB 2020

```
=65-COUNTIF(Makalah!H:H; "Tidak Ikut")
```

Hasilnya adalah 19. Makalah adalah nama sheet data, H:H adalah nama kolom yang berisi tentang pertanyaan "Kalau tau, apakah kamu ikut kegiatannya? Apa saja?".

V. KESIMPULAN

Pengaplikasian *pattern matching* pada Microsoft Excel tidak hanya bisa menggunakan formula COUNTIF(), tetapi juga bisa menggunakan formula sendiri dimana organisator akan membuat kodingannya sendiri menggunakan VBA dan disini akan menggunakan *regular expression*. Dengan menggunakan regex, maka akan lebih teliti lagi dalam pengolahan data. Penulis tidak mengimplementasikannya pada makalah kali ini, pengimplementasiannya akan diserahkan kepada para pembaca.

Dengan adanya aplikasi penggunaan *pattern matching* pada database Microsoft Excel, organisator lebih mudah dalam mengolah database. Diharapkan banyak organisator yang terbantu dalam pengolahan database organisasinya masing-masing dan mempermudah dalam pencarian kata pada database khususnya dalam penggunaan aplikasi Microsoft Excel walaupun tidak semua aplikasi *pattern matching* yang diimplementasikan oleh penulis. Masih banyak penerapan dari *pattern matching* pada Microsoft Excel seperti pemakaian formula dengan regex dan lainnya. Penggunaan *pattern matching* sangat disarankan bagi para organisator yang mengolah database organisasinya, apalagi dengan menggunakan regex.

VIDEO LINK AT YOUTUBE

<https://www.youtube.com/watch?v=Hz1BsO2EO4U>

"Aplikasi Patter Matching pada Database Microsoft Excel"

UCAPAN TERIMA KASIH

Penulis mengucapkan syukur kepada Tuhan Yang Maha Esa, berkat rahmat-Nya penulis bisa menyelesaikan penulisan makalah strategi algoritma ini. Tidak lupa juga penulis mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T., Ibu Dr. Masayu Leylia Khodra, dan Ibu Dr. Nur Ulfa Maulidevi, S.T. sebagai dosen mata kuliah strategi algoritma di program studi Teknik Informatika Institut Teknologi Bandung yang telah membimbing penulis baik dalam pengajaran di kelas maupun dalam membuat makalah strategi algoritma semester ini. Kemudian, penulis juga menyampaikan ucapan terima kasih kepada kedua orang tua penulis, keluarga, dan teman-teman yang telah memberikan motivasi kepada penulis sehingga penulis bisa semangat dalam menyelesaikan makalah strategi algoritma ini.

REFERENSI

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/stima19-20.htm>, diakses 1 Mei 2020 pukul 15.30.
- [2] <https://salamadian.com/pengertian-organisasi-adalah/>, diakses 3 Mei 2020 pukul 15.41 WIB.
- [3] <https://www.gurupendidikan.co.id/pengertian-database/>, diakses 3 Mei 2020 pukul 16.00 WIB.
- [4] <https://www.advernesia.com/blog/microsoft-excel/microsoft-excel-adalah/>, diakses 3 Mei 16.13 WIB.
- [5] <https://masputih.com/2015/11/belajar-regular-expression>, diakses 3 Mei 2020 pukul 20.33 WIB.
- [6] <https://febryharipurwanto.wordpress.com/2009/03/24/regular-expressions-ekspresi-reguler/>, diakses 4 Mei 2020 pukul 06.30 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, 4 Mei 2020



Fikra Hadi Ramadhan
13518036