R Course DQ Lab Academy

Modul 5

Data Exploration In Data Science Using R

Review Data Wrangling Part 1

Pada course Data Wrangling Part 1 – kita telah melakukan beberapa tahapan awal data wrangling di R yang mencakup topik berikut:

- **Missing Value**: Bagaimana kita mengenal missing value atau data kosong atau tidak terisi yang direpresentasikan oleh NA (Not Available) di R. Termasuk pada bab ini, operasi matematika yang tidak menghasilkan angka yang bisa diolah (Not a Number).
- Struktur data kategori bernama **Factor**: Melengkapi bab awal – diperkenalkan juga tipe data yang juga banyak dijumpai, yaitu data kategori.
- **Membaca file-file teks dan Excel** – file Excel adalah file yang paling banyak ditemui sehari-hari.
- Melakukan **perubahan struktur data** seperti merubah nama, menambah dan membuang kolom, dan normalisasi struktur data (pivot) sehingga cocok digunakan lebih lanjut.

Apa yang dipelajari di Data Wrangling Part 2?

Melanjutkan bagian 1, fokus Data Wrangling Part 2 adalah pembacaan sistem database, data cleansing, dan data enrichment dengan detail berikut:

- **Contoh Dataset "Kotor":** Perkenalan contoh dataset master pelanggan yang sengaja dirancang dengan "kotor" atau mengandung isi yang tidak standar – menyerupai kondisi riil yang banyak ditemukan oleh tim DQLab selama terlibat dalam proyek-proyek pengolahan data di Indonesia.
- **Profiling**: Bagaimana mengidentifikasi pola dataset kita sebelum tau apa yang perlu dibersihkan atau dirapikan.
- **Membaca Database Relasional**: Bagaimana mengakses dari sistem database dengan memperkenalkan objek-objek database dan bahasa SQL (Structured Query Language).
- **Data Cleansing – Standarisasi**: Bagaimana melakukan perapian isi berbagai tipe data dengan menggunakan fungsi-fungsi transformasi data.
- **Data Cleansing – Missing Value**: Bagaimana mengisi *missing value* pada kolom numerik.
- **Data Cleansing – Deduplikasi**: Menemukan data yang duplikat dan melakukan grouping terhadap data-data tersebut.
- **Data Enrichment**: Bagaimana melengkapi data kosong dengan melakukan lookup dari internal data.

Walaupun cukup padat materinya, seperti biasa DQLab akan memecah topik-topik ini ke bab-bab yang cukup ringkas dan setahap demi setahap sehingga mudah diikuti.

Dua bab pertama akan berisi teori dan pengenalan dataset, setelah itu bab berikutnya akan penuh praktek latihan.

Klik tombol Next untuk melanjutkan.

Perkenalan Dataset

Sepanjang course ini, kita akan bekerja dengan dataset pelanggan (*customer*) yang kotor dalam dua format:

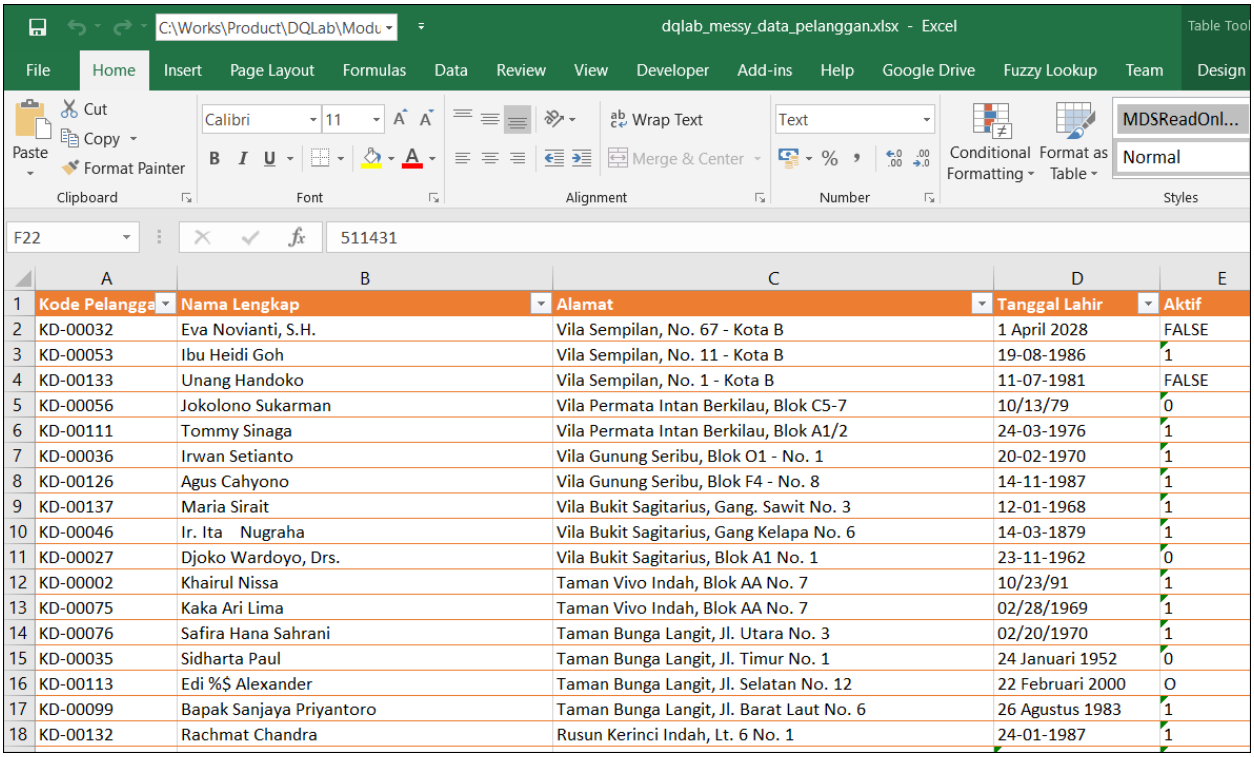
- File Excel bernama **xlsx**. File ini dapat didownload pada url https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx**.**
- Table **dqlab_pelanggan_messy** di sistem database MySQL – salah satu sistem database open source terpopuler saat ini.

Dataset ini sengaja dirancang agar "kotor" atau semrawut dimana terdapat data dengan format yang berbeda untuk kolom yang sama, data yang hilang, dan kolom dengan lebih dari satu informasi.

Selengkapnya kolom-kolom yang terdapat pada dataset ini adalah sebagai berikut:

- **Kode Pelanggan**: Merupakan data kode dari tiap pelanggan yang bersifat unik.
- **Nama Lengkap**: Nama lengkap dari pelanggan.
- **Alamat**: Merupakan data kode dari tiap pelanggan yang bersifat unik.
- **Tanggal Lahir**: Merupakan tanggal lahir dari pelanggan.
- **Aktif**: Berisi informasi aktif tidaknya pelanggan tersebut saat ini.
- **Kode Pos**: Nomor kode pos dari alamat pelanggan.
- **No Telepon**: Nomor telepon telepon yang dapat dihubungi.
- **Nilai Belanja Setahun**: Nilai total belanja dalam waktu setahun terakhir.

Berikut adalah tampilan sebagian dataset tersebut pada aplikasi Excel.



	A	B	C	D	E
1	Kode Pelangga	Nama Lengkap	Alamat	Tanggal Lahir	Aktif
2	KD-00032	Eva Novianti, S.H.	Vila Sempilan, No. 67 - Kota B	1 April 2028	FALSE
3	KD-00053	Ibu Heidi Goh	Vila Sempilan, No. 11 - Kota B	19-08-1986	1
4	KD-00133	Unang Handoko	Vila Sempilan, No. 1 - Kota B	11-07-1981	FALSE
5	KD-00056	Jokolono Sukarman	Vila Permata Intan Berkilau, Blok C5-7	10/13/79	0
6	KD-00111	Tommy Sinaga	Vila Permata Intan Berkilau, Blok A1/2	24-03-1976	1
7	KD-00036	Irwan Setianto	Vila Gunung Seribu, Blok O1 - No. 1	20-02-1970	1
8	KD-00126	Agus Cahyono	Vila Gunung Seribu, Blok F4 - No. 8	14-11-1987	1
9	KD-00137	Maria Sirait	Vila Bukit Sagitarius, Gang. Sawit No. 3	12-01-1968	1
10	KD-00046	Ir. Ita Nugraha	Vila Bukit Sagitarius, Gang Kelapa No. 6	14-03-1879	1
11	KD-00027	Djoko Wardoyo, Drs.	Vila Bukit Sagitarius, Blok A1 No. 1	23-11-1962	0
12	KD-00002	Khairul Nissa	Taman Vivo Indah, Blok AA No. 7	10/23/91	1
13	KD-00075	Kaka Ari Lima	Taman Vivo Indah, Blok AA No. 7	02/28/1969	1
14	KD-00076	Safira Hana Sahrani	Taman Bunga Langit, Jl. Utara No. 3	02/20/1970	1
15	KD-00035	Sidharta Paul	Taman Bunga Langit, Jl. Timur No. 1	24 Januari 1952	0
16	KD-00113	Edi %$ Alexander	Taman Bunga Langit, Jl. Selatan No. 12	22 Februari 2000	O
17	KD-00099	Bapak Sanjaya Priyantoro	Taman Bunga Langit, Jl. Barat Laut No. 6	26 Agustus 1983	1
18	KD-00132	Rachmat Chandra	Rusun Kerinci Indah, Lt. 6 No. 1	24-01-1987	1

Klik tombol Next untuk melanjutkan ke deskripsi permasalahan dari beberapa kolom yang ada pada dataset ini sebelum bab profiling untuk menganalisa pola data secara sistematis dengan bantuan R.

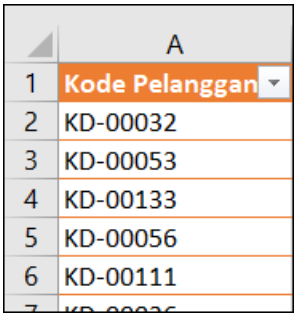
Kolom Kode Pelanggan

Kolom kode pelanggan adalah kolom identifikasi – yaitu kolom yang menjadi kunci pembeda antara baris data ini dengan baris data lainnya – di dalam dataset pelanggan.

Kolom identifikasi biasanya memiliki pola yang teratur, untuk dataset kita polanya adalah sebagai berikut.

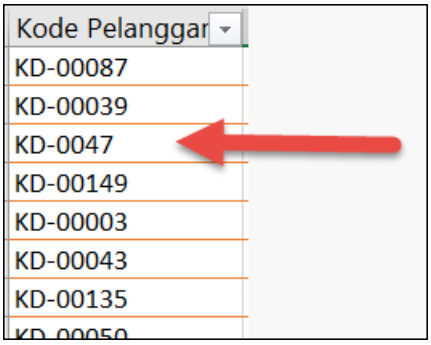
- Memiliki prefix atau awalan teks yang fix bernilai "**KD-**"
- Memiliki suffix atau akhiran angka – dengan format lima digit angka.
- Karena pola yang fix tersebut, panjang total kolom tersebut adalah 8 karakter/digit.

Berikut adalah sebagian contoh data kode pelanggan.



	A
1	Kode Pelanggan
2	KD-00032
3	KD-00053
4	KD-00133
5	KD-00056
6	KD-00111

Namun pada baris tertentu ada pola yang tidak sesuai, dimana jumlah angka digit di belakang "KD-" hanya empat seperti terlihat pada screenshot berikut.



Kode Pelanggar
KD-00087
KD-00039
KD-0047
KD-00149
KD-00003
KD-00043
KD-00135

Dengan demikian, ada permasalahan inkonsitensi pola dengan panjang yang berbeda.

Klik tombol Next untuk melanjutkan.

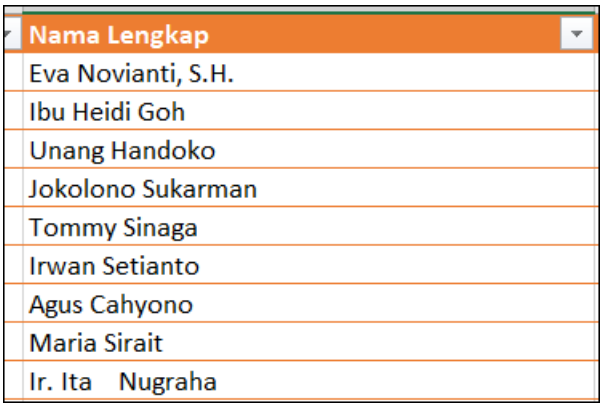
Apa yang menjadi permasalahan pada kolom Kode Pelanggan?

Apa yang sejauh ini menjadi permasalahan pada kolom Kode Pelanggan berdasarkan deskripsi subbab sebelumnya?

- [x] Ada data dengan panjang total kolom yang tidak sama.
- [x] Pola yang tidak konsisten.
- [ ] Pola yang konsisten
- [ ] Pola prefix yang berbeda.
- [x] Pola suffix yang berbeda.

Kolom Nama Lengkap

Kolom Nama Lengkap adalah kolom kedua pada dataset dengan sebagian tampilan isinya adalah sebagai berikut.



Nama Lengkap
Eva Novianti, S.H.
Ibu Heidi Goh
Unang Handoko
Jokolono Sukarman
Tommy Sinaga
Irwan Setianto
Agus Cahyono
Maria Sirait
Ir. Ita Nugraha

Disini terlihat ada contoh penulisan panggilan untuk data "Ibu Heidi Goh". Ini pada sebagian perusahaan tidak menjadi masalah, namun untuk industri perbankan yang mengharuskan standarisasi nama berdasarkan regulasi OJK (Otoritas Jasa Keuangan) – maka nama panggilan Ibu ini harus dihilangkan.

Kemudian terdapat spasi berlebih pada data dengan nama "Ir. Ita Nugraha". Ini tentunya tidak standar secara umum.

Akan ada banyak permasalahan lain pada penulisan nama lengkap ini. Kita akan melakukan identifikasi lebih lanjut pada bab profiling.

Klik tombol Next untuk melanjutkan.

Kolom Tanggal Lahir

Kolom Tanggal Lahir adalah kolom penting lainnya yang biasanya dipasangkan dengan nama untuk identifikasi individu. Sebagian tampilan isinya adalah sebagai berikut.

Tanggal Lahir
1 April 2028
19-08-1986
11-07-1981
10/13/79
24-03-1976
20-02-1970
14-11-1987
12-01-1968
14-03-1879
23-11-1962
10/23/91
02/28/1969
02/20/1970
24 Januari 1952
22 Februari 2000
26 Agustus 1983

Disini sudah langsung terlihat masalahnya, yaitu ada beberapa pola yang penulisannya berbeda. Ada yang memiliki pemisah tanda minus (-) dan garis miring (/) dan penulisan nama bulan dan bukan angka pada sebagian data.

Selain itu ada tahun lahir pelanggan di 1879. Walaupun secara isi, data tersebut bisa dianggap tanggal yang valid. Namun secara bisnis, data tanggal lahir ini mungkin tidak logis dan butuh perbaiki.

Penulisan seperti ini sudah pasti perlu distandarisasi dan diperbaiki agar dapat diolah lebih lanjut untuk analisa.

Klik tombol Next untuk melanjutkan.

Apa yang menjadi permasalahan pada kolom Nama Lengkap?

Apa yang sejauh ini menjadi permasalahan pada kolom Nama Lengkap berdasarkan deskripsi subbab sebelumnya?

- [x] Isinya tidak standar sesuai dengan ketentuan lembaga tertentu seperti OJK (Otoritas Jasa Keuangan)
- [x] Isinya tidak sesuai dengan standar umum penulisan nama
- [ ] Isinya terlalu rapi
- [ ] Semua benar
- [ ] Penulisan gelar yang tidak standar

Apa yang menjadi permasalahan pada kolom Tanggal Lahir?

Apa yang sejauh ini menjadi permasalahan pada kolom Tanggal Lahir berdasarkan deskripsi subbab sebelumnya?

- [ ] Tanda pemisah antara tanggal, bulan dan tahun tidak sama
- [ ] Informasi bulan diisi dengan angka dan nama bulan.
- [ ] Variasi posisi tanggal, bulan dan tahun.
- [ ] Tanggal lahir pelanggan yang tidak logis.
- [x] Semua benar.

Data Pelanggan yang Duplikat

Selain isi data yang tidak standar, ternyata dataset ini juga memiliki duplikat untuk pelanggan yang sama.

Kode Pelanggan	Nama Lengkap	Alamat	Tanggal Lahir
KD-00012	Cahyono, Agus	Pulo Bambu No. 15, Kota Tenggara Lama	02/08/1967
KD-00001	Agus Cahyono's	Jl. Pulo Bambu No. 15, Kota Tenggara Lama	8 Februari 1967
KD-00778	Cahyono Agus H.	Jalan. Pulau Bambu No. 15 - Kota Tenggara Lama	08-02-1967

Terlihat tiga baris data dengan nama Agus Cahyono ini sebenarnya sama terlihat dari isi data Nama Lengkap, Alamat dan Tanggal Lahir. Hanya saja format penulisan semuanya berbeda.

Ini akan memiliki konsekusensi atau *impact* besar terhadap bisnis. Jika setiap pelanggan ini telah memiliki transaksi, maka kode-kode pelanggannya akan berbeda semua. Dan pada saat analisa data, maka seluruh data transaksi tersebut akan terpisah tiga dan nilai total tidak pernah didapatkan.

Dengan demikian, seluruh laporan transaksi untuk seorang Agus Cahyono akan lebih rendah dari seharusnya.

Dan jika ada program loyalty yang harusnya menyasar pelanggan dengan jumlah transaksi tertentu sebagai bentuk apresiasi dan menjual lebih, maka Agus Cahyono kemungkinan tidak akan terkena *reach* dan *lost opportunity* (kehilangan kesempatan) bagi bisnis.

Dengan demikian, akurasi laporan akan sangat rendah dan bisnis bisa mengambil keputusan yang salah.

Ini tentunya adalah tantangan besar dari sisi komputasi yang akan coba kita pecahkan dengan framework dan pengalaman dari DQLab pada enam bab ke depan.

Klik tombol Next untuk melanjutkan.

Membaca dan review isi data file pelanggan

Dataset pelanggan berupa file Excel ini dapat dibaca dengan function **read.xlsx** seperti telah diperkenalkan pada course "Introduction to R", "Data Visualization with GGPlot2", dan "Data Wrangling Part 1".

Tugas Praktek

Pada code editor telah dimasukkan code untuk membaca file Excel contoh kita dan dimasukkan ke dalam variable. Ganti bagian […] dengan perintah untuk menampilkan hasil pembacaan file tersebut.

Jika berjalan dengan lancar maka akan tampil potongan hasil berikut.

```
    Kode.Pelanggan                          Nama.Lengkap
1          KD-00032                    Eva Novianti, S.H.
2          KD-00053                         Ibu Heidi Goh
…
155        KD-00492                  dr. Yati Octavianus

                                                 Alamat     Tanggal.Lahir Aktif
1                     Vila Sempilan, No. 67 - Kota B      1 April 2028 FALSE
2                     Vila Sempilan, No. 11 - Kota B        19-08-1986     1
…
155                  Kompleks Pelaut Tangguh, No. 5A       21 Mei 1980     1

    Kode.Pos        No.Telepon Nilai.Belanja.Setahun
1     567130   085419651438216               1275600
2     567130  6282189517223455                317800
…
155   321321 +6285879131063825                904900
```

Terlihat ada 155 baris data dengan kolom-kolom data seperti yang telah dijelaskan sejauh ini.

Code Editor

```
library(openxlsx)

library(bpa)

#Membaca dataset pelanggan

data.pelanggan <-
read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx",sheet="Pelanggan")


#Menampilkan variable data.pelanggan

data.pelanggan #[...]
```

Console

(LANGSUNG PRAKTEK)

Profil sederhana dataset dengan function str

Pembacaan dataset secara keseluruhan biasanya tidak selalu diperlukan. Kita lebih banyak kepentingan melihat ringkasan informasi dari data tersebut, misalkan berapa jumlah baris data yang ada.

Dan function str cukup untuk memenuhi kepentingan tersebut. Disiplinkan diri untuk selalu menggunakan function str ini pada saat mengolah data dengan R.

Tugas Praktek

Gunakan function str dengan input berupa nama variable yang menyimpan hasil pembacaan file Excel untuk menggantikan bagian […] pada code editor.

Jika berjalan dengan lancar, maka akan muncul hasil sebagai berikut.

```
'data.frame':    155 obs. of  8 variables:
 $ Kode.Pelanggan       : chr  "KD-00032" "KD-00053" "KD-00133" "KD-00056" ...
 $ Nama.Lengkap         : chr  "Eva Novianti, S.H." "Ibu Heidi Goh" "Unang Handoko" "
Jokolono Sukarman" ...
 $ Alamat               : chr  "Vila Sempilan, No. 67 - Kota B" "Vila Sempilan, No. 1
1 - Kota B" "Vila Sempilan, No. 1 - Kota B" "Vila Permata Intan Berkilau, Blok C5-7"
...
 $ Tanggal.Lahir        : chr  "1 April 2028" "19-08-1986" "11-07-1981" "10/13/79" ..
.
 $ Aktif                : chr  "FALSE" "1" "FALSE" "0" ...
 $ Kode.Pos             : chr  "567130" "567130" "567130" "876551" ...
 $ No.Telepon           : chr  "085419651438216" "6282189517223455" "+628295295558697
9" "6289278629437370" ...
 $ Nilai.Belanja.Setahun: num  1275600 317800 1537200 1524700 655400 ...
```

Dimana pada baris awal terdapat informasi bahwa dataset ini bertipe data.frame dengan jumlah 155 baris data dan 8 kolom.

Code Editor

```
library(openxlsx)

library(bpa)
```

#Membaca dataset pelanggan

data.pelanggan <-
read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx",sheet="Pelanggan")

#Menampilkan struktur variable data.pelanggan

str(data.pelanggan) #[...]

Console

```
> library(openxlsx)

> library(bpa)

> #Membaca dataset pelanggan
> data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pela
nggan.xlsx",sheet="Pelanggan")

> #Menampilkan struktur variable data.pelanggan
> str(data.pelanggan) #[...]
'data.frame':    155 obs. of  8 variables:
 $ Kode.Pelanggan       : chr  "KD-00032" "KD-00053" "KD-00133" "KD-00056" ...
 $ Nama.Lengkap         : chr  "Eva Novianti, S.H." "Ibu Heidi Goh" "Unang Handoko" "
Jokolono Sukarman" ...
 $ Alamat               : chr  "Vila Sempilan, No. 67 - Kota B" "Vila Sempilan, No. 1
1 - Kota B" "Vila Sempilan, No. 1 - Kota B" "Vila Permata Intan Berkilau, Blok C5-7"
...
 $ Tanggal.Lahir        : chr  "1 April 2028" "19-08-1986" "11-07-1981" "10/13/79" ..
.
 $ Aktif                : chr  "FALSE" "1" "FALSE" "0" ...
 $ Kode.Pos             : chr  "567130" "567130" "567130" "876551" ...
 $ No.Telepon           : chr  "085419651438216" "6282189517223455" "+628295295558697
9" "6289278629437370" ...
 $ Nilai.Belanja.Setahun: num  1275600 317800 1537200 1524700 655400 ...
```

Kesimpulan

Sepanjang course ini, kita akan banyak bekerja dengan data cleansing dan memerlukan contoh dataset yang komprehensif.

DQLab membuat dataset pelanggan yang cukup berantakan. Beberapa permasalahan sudah terlihat dari screenshot dan deskripsi yang diberikan pada bab ini. Walaupun yang dibahas adalah kolom Kode Pelanggan, Nama Lengkap dan Tanggal Lahir dan duplikat data, namun kolom lainnya juga tidak terhindar dari masalah yang memerlukan data cleansing.

Permasalahan data seperti ini hampir sulit dihindari, walaupun sudah dicoba dengan pengembangan sistem entri yang baik – karena dari pengalaman kami dinamika bisnis lebih cepat dibandingkan dinamika pengembangan sistem entri terkomputerisasi.

Selain itu, menyangkut data pelanggan – ini biasanya perlu integrasi dari beberapa sistem seperti ERP (Enterprise Resource Planning), core banking, CRM (Customer Relationship Management) yang kemungkinan besar memiliki standar penulisan yang berbeda.

Klik tombol Next untuk melanjutkan ke bab berikutnya – dimana kita akan melakukan identifikasi pola data untuk menemukan potensi permasalahan pada dataset kita.

Apa itu Data Profiling?

Data profiling adalah tahap awal untuk melakukan data cleansing. Di dalam proses ini kita melakukan aktifitas yang sederhana tapi penting:

- Kita akan mengidentifikasi pola-pola yang terdapat pada suatu kolom data.
- Dan membandingkannya dengan ekspektasi atau ukuran scientific yang wajar, untuk menemukan data yang perlu diperbaiki.

Teknik profiling bisa dilakukan dengan banyak cara, namun yang pasti secara umum akan menelusuri keseluruhan data.

Karena pola bisa banyak macam, kita akan memfokuskan profiling pada isi data dengan pola teks sederhana namun cukup efektif.

Agar dapat dipraktekkan dengan riil, secara spesifik kita akan menggunakan function dan operator berikut sepanjang bab ini.

- Function summary dari paket bawaan R.
- Function basic_pattern_analysis dari library bpa di R.
- Menggunakan operator == dan function grepl untuk menarik data untuk pola hasil temuan.

Klik tombol Next untuk melanjutkan.

Mana pernyataan berikut yang benar mengenai data profiling?

Mana pernyataan berikut yang benar mengenai data profiling?

- [ ] Profiling dilakukan dengan menyatukan semua kolom menjadi satu kolom, dan baru dilakukan profiling.
- [x] Profiling dilakukan untuk tiap kolom data.
- [x] Profiling dilakukan untuk seluruh baris data.
- [x] Profiling dapat menemukan data anomali atau yang tidak wajar.
- [x] Hasil profiling dapat digunakan sebagai dasar untuk memperbaiki data.

Menggunakan function summary

Jika pada Data Wrangling part 1 kita menggunakan function str untuk melihat struktur dan isi data, pada bab ini kita memperkenalkan function lain yaitu **summary**.

Function **summary** adalah function yang akan memberikan ringkasan singkat data dengan menganalisa isi data.

Penggunaan function summary cukup sederhana, cukup satu objek yang ingin dianalisa.

summary(objek)

Berbeda dengan output str, output dari summary ini akan berbeda untuk tiap tipe dari objek.

- Untuk tipe data numerik, maka summary akan memberikan nilai minimum, maksimum, median, mean, dan lain-lain.
- Untuk tipe data character akan melaporkan tipe data dan panjang saja.
- Untuk tipe data factor akan berisi mengenai nilai-nilai factor dan jumlah kemunculan data tersebut (frekuensi).

Untuk lebih jelasnya mari kita coba jalankan tugas praktek berikut.

Tugas Praktek

Pada code editor sudah terdapat code untuk membaca file dataset pelanggan berformat Excel yang kemudian disimpan ke dalam variable data.pelanggan.

Gunakan function summary terhadap variable tersebut untuk menggantikan bagian […] pada code editor.

Jika semua berjalan dengan lancar maka hasilnya akan terlihat seperti berikut.

```
Kode.Pelanggan     Nama.Lengkap          Alamat          Tanggal.Lahir
 Length:155         Length:155         Length:155         Length:155
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character




   Aktif              Kode.Pos          No.Telepon        Nilai.Belanja.Setahun
 Length:155         Length:155         Length:155         Min.   : 237400
 Class :character   Class :character   Class :character   1st Qu.: 504800
```

```
 Mode  :character   Mode  :character   Mode  :character   Median : 851600
                                                         Mean   : 857226
                                                         3rd Qu.:1179800
                                                         Max.   :1537200
                                                         NA's   :4
```

Penjelasan hasil:

- Pelanggan, Nama.Lengkap, Alamat, Tanggal.Lahir, Aktif, Kode.Pos, No.Telepon dan Nilai.Belanja.Setahun adalah nama kolom-kolom dari dataset kita.
- Di bawah dari tiap kolom tersebut adalah hasil summary. Selain kolom Nilai.Belanja.Setahun, semua hasil summary sama sebagai berikut.

  ```
  Length:155
  Class :character
  Mode  :character
  ```

  Ini artinya seluruh isi dari kolom-kolom tersebut semuanya bertipe karakter dengan panjang 155 karakter.

- Untuk kolom Nilai.Belanja.Setahun, hasilnya sebagai berikut.

  ```
  Min.    : 237400
  1st Qu.: 504800
  Median : 851600
  Mean   : 857226
  3rd Qu.:1179800
  Max.   :1537200
  NA's   :4
  ```

- Ini artinya kolom Nilai.Belanja.Setahun dikenal sebagai numerik dengan urutan profil data sebagai berikut: nilai minimum, kuartil pertama, median, rata-rata, kuartil ketiga, maksium, dan jumlah missing value.

Catatan: Penjelasan arti dari tiap-tiap nilai tersebut di luar cakupan praktek ini agar topik tidak melebar.

Code Editor

```
#Load library openxlsx

library(openxlsx)

#Membaca dataset pelanggan

data.pelanggan <-
read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx",sheet=
"Pelanggan")

#Menggunakan function summary

summary(data.pelanggan)
```

Console

```
> #Load library openxlsx
> library(openxlsx)

> #Membaca dataset pelanggan
> data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pela
nggan.xlsx",sheet="Pelanggan")

> #Menggunakan function summary
> summary(data.pelanggan)
 Kode.Pelanggan     Nama.Lengkap          Alamat          Tanggal.Lahir
 Length:155         Length:155         Length:155         Length:155
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character



    Aktif              Kode.Pos          No.Telepon        Nilai.Belanja.Setahun
 Length:155         Length:155         Length:155         Min.   : 237400
 Class :character   Class :character   Class :character   1st Qu.: 504800
 Mode  :character   Mode  :character   Mode  :character   Median : 851600
                                                         Mean   : 857226
                                                         3rd Qu.:1179800
                                                         Max.   :1537200
                                                         NA's   :4
```

Konversi factor dan hasil summary untuk kolom Aktif

Output summary sejauh ini tidak menceritakan banyak hal mengenai kondisi data kita, seluruhnya dibaca sebagai character dan tambahan informasi hanya length.

Kembali ke "Data Wrangling with R – Part 1", factor adalah tipe data yang dapat membantu. Pada praktek kali ini kita coba konversi kolom Aktif menjadi factor dan kita jalankan kembali fungsi summary.

Tugas Praktek

Gantilah bagian […] pada code editor dengan function as.factor.

Jika semua berjalan dengan lancar maka akan muncul hasil sebagai berikut.

```
Kode.Pelanggan     Nama.Lengkap          Alamat          Tanggal.Lahir
 Length:155         Length:155         Length:155         Length:155
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character




  Aktif      Kode.Pos          No.Telepon        Nilai.Belanja.Setahun
 -    : 1   Length:155         Length:155         Min.   : 237400
 0    :23   Class :character   Class :character   1st Qu.: 504800
 1    :98   Mode  :character   Mode  :character   Median : 851600
 FALSE:13                                         Mean   : 857226
 I    : 1                                         3rd Qu.:1179800
 O    : 2                                         Max.   :1537200
 TRUE :17                                         NA's   :4
```

Terlihat ada perbedaan untuk kolom Aktif, yang pada praktek sebelumnya masih bertipe character mengeluarkan hasil berikut.

```
   Aktif
   Length:155
   Class :character
   Mode  :character
```

Ketika telah dikonversi menjadi factor maka mengeluarkan hasil berikut.

```
     Aktif
   -    : 1
   0    :23
   1    :98
   FALSE:13
   I    : 1
   O    : 2
   TRUE :17
```

Hasil ini merupakan daftar nilai dan jumlah frekuensi dari nilai tersebut. Dari hasil terlihat terdapat tanda minus (-) sebanyak 1 data, 0 sebanyak 23 dat, 1 sebanyak 98 data, FALSE sebanyak 13 data, huruf I sebanyak 1 data, huruf O sebanyak 1 data, dan nilai TRUE sebanyak 17 data.

Code Editor

```
library(openxlsx)

#Membaca dataset pelanggan

data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx",sheet= "Pelanggan")


#Merubah data.pelanggan$Aktif menjadi factor

data.pelanggan$Aktif <- as.factor(data.pelanggan$Aktif)

#Menggunakan function summary

summary(data.pelanggan)
```

Console

```
> library(openxlsx)

> #Membaca dataset pelanggan
> data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pela
nggan.xlsx",sheet="Pelanggan")

> #Merubah data.pelanggan$Aktif menjadi factor
> data.pelanggan$Aktif <- as.factor(data.pelanggan$Aktif)

> #Menggunakan function summary
> summary(data.pelanggan)
 Kode.Pelanggan     Nama.Lengkap          Alamat          Tanggal.Lahir
 Length:155         Length:155         Length:155         Length:155
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character



   Aktif      Kode.Pos          No.Telepon        Nilai.Belanja.Setahun
 -    : 1   Length:155         Length:155         Min.   : 237400
 0    :23   Class :character   Class :character   1st Qu.: 504800
 1    :98   Mode  :character   Mode  :character   Median : 851600
 FALSE:13                                         Mean   : 857226
 I    : 1                                         3rd Qu.:1179800
 O    : 2                                         Max.   :1537200
 TRUE :17                                         NA's   :4
```

Summary untuk factor kolom lain

Pada praktek kali ini – dengan summary factor yang lebih bisa menceritakan kondisi data – kita akan melakukan konversi sisa kolom character lain ke dalam factor.

Tugas Praktek

Dengan cara yang sama pada praktek sebelumnya, lakukan konversi semua kolom character pada dataset pelanggan kita dengan mengganti bagian […1…] sampai dengan […7…]

Jika semua berjalan dengan lancar maka akan muncul hasil sebagai berikut.

```
 Kode.Pelanggan                    Nama.Lengkap
KD-00001:  1    Abdul Kadir              :  2
KD-00002:  1    Bapak Sanjaya Priyantoro:  2
KD-00003:  1    Budi Setiawan            :  2
KD-00004:  1    Budi Yahya               :  2
KD-00005:  1    Rachmat Chandra          :  2
KD-00006:  1    Risma Sihombing          :  2
(Other) :149    (Other)                  :143
                                  Alamat             Tanggal.Lahir   Aktif
Bukit Vivo Indah, Blok C 2/4         :  2   02/28/1969      :  5   -    : 1
Jl. Bintang Supernova, No. 78        :  2   01/01/01        :  4   0    :23
Jl. Pulau Sentosa No. 133            :  2   01/31/01        :  3   1    :98
Jl. Puri Arteri Raya, No. 88 - Kota T:  2   13-11-1962      :  3   FALSE:13
Kompleks Pelaut Tangguh, No. 5A      :  2   19 Maret 1950   :  3   I    : 1
Perum Bimasakti Raya, Blok A No. 10  :  2   30 November 1954:  3   O    : 2
(Other)                              :143   (Other)         :134   TRUE :17
   Kode.Pos               No.Telepon  Nilai.Belanja.Setahun
696193 :  9   +6281729600654645:  2   Min.   : 237400
321321 :  8   +6285879131063825:  2   1st Qu.: 504800
896549 :  8   +6286815308308264:  2   Median : 851600
986455 :  8   082989111122220  :  2   Mean   : 857226
896555 :  7   087642929298977  :  2   3rd Qu.:1179800
712984 :  6   +6281693345459608:  1   Max.   :1537200
(Other):109   (Other)          :144   NA's   :4
```

Terlihat semua kolom sekarang diolah sebagai factor dengan tampilan summary nilai dan frekuensi sehingga lebih jelas distribusi nilanya. Coba perhatikan untuk kolom Tanggal.Lahir, terlihat sekali ada penulisan nilai yang berbeda dengan jumlah kemunculan nilainya.

Code Editor

```
library(openxlsx)

#Membaca dataset pelanggan

data.pelanggan <-
read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx",sheet=
"Pelanggan")


#Merubah kolom data selain Nilai.Belanja.Setahun menjadi factor

data.pelanggan$Kode.Pelanggan <- as.factor(data.pelanggan$Kode.Pelanggan)

data.pelanggan$Nama.Lengkap <- as.factor(data.pelanggan$Nama.Lengkap)

data.pelanggan$Alamat <- as.factor(data.pelanggan$Alamat)

data.pelanggan$Tanggal.Lahir <- as.factor(data.pelanggan$Tanggal.Lahir)

data.pelanggan$Aktif <- as.factor(data.pelanggan$Aktif)

data.pelanggan$Kode.Pos <- as.factor(data.pelanggan$Kode.Pos)

data.pelanggan$No.Telepon <- as.factor(data.pelanggan$No.Telepon)


#Menggunakan function summary

summary(data.pelanggan)
```

Console

```
> library(openxlsx)

> #Membaca dataset pelanggan
> data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pela
nggan.xlsx",sheet="Pelanggan")

> #Merubah kolom data selain Nilai.Belanja.Setahun menjadi factor
> data.pelanggan$Kode.Pelanggan <- as.factor(data.pelanggan$Kode.Pelanggan)

> data.pelanggan$Nama.Lengkap <- as.factor(data.pelanggan$Nama.Lengkap)

> data.pelanggan$Alamat <- as.factor(data.pelanggan$Alamat)

> data.pelanggan$Tanggal.Lahir <- as.factor(data.pelanggan$Tanggal.Lahir)

> data.pelanggan$Aktif <- as.factor(data.pelanggan$Aktif)

> data.pelanggan$Kode.Pos <- as.factor(data.pelanggan$Kode.Pos)

> data.pelanggan$No.Telepon <- as.factor(data.pelanggan$No.Telepon)

> #Menggunakan function summary
> summary(data.pelanggan)
  Kode.Pelanggan                    Nama.Lengkap
 KD-00001:  1    Abdul Kadir              :  2
 KD-00002:  1    Bapak Sanjaya Priyantoro:  2
 KD-00003:  1    Budi Setiawan            :  2
 KD-00004:  1    Budi Yahya               :  2
 KD-00005:  1    Rachmat Chandra          :  2
 KD-00006:  1    Risma Sihombing          :  2
 (Other) :149    (Other)                  :143
                                 Alamat             Tanggal.Lahir   Aktif
 Bukit Vivo Indah, Blok C 2/4         :  2   02/28/1969      :  5   -    : 1
 Jl. Bintang Supernova, No. 78        :  2   01/01/01        :  4   0    :23
 Jl. Pulau Sentosa No. 133            :  2   01/31/01        :  3   1    :98
 Jl. Puri Arteri Raya, No. 88 - Kota T:  2   13-11-1962      :  3   FALSE:13
 Kompleks Pelaut Tangguh, No. 5A      :  2   19 Maret 1950   :  3   I    : 1
 Perum Bimasakti Raya, Blok A No. 10  :  2   30 November 1954:  3   O    : 2
 (Other)                              :143   (Other)         :134   TRUE :17
    Kode.Pos              No.Telepon  Nilai.Belanja.Setahun
 696193 :  9   +6281729600654645:  2   Min.    : 237400
 321321 :  8   +6285879131063825:  2   1st Qu.: 504800
 896549 :  8   +6286815308308264:  2   Median : 851600
 986455 :  8   082989111122220  :  2   Mean   : 857226
 896555 :  7   087642929298977  :  2   3rd Qu.:1179800
 712984 :  6   +6281693345459608:  1   Max.   :1537200
 (Other):109   (Other)          :144   NA's   :4
```

Menggunakan library 'bpa'

Profiling dengan function summary terlihat cukup berguna untuk mengidentifikasi data numerik dan sebaran nilai di factor.

Namun untuk mengidentifikasi pola teks yang benar seperti keharusan prefix dua alfabet, diikuti tanda – dan terakhir dengan 5 angka digit pada kolom Kode Pelanggan, summary tidak dapat mengeluarkan hal tersebut.

Untuk menganalisa pola seperti ini kita dapat menggunakan library bpa (Basic Pattern Analysis).

Klik tombol Next untuk melanjutkan.

Kenapa menggunakan library 'bpa'?

Kenapa menggunakan library 'bpa"?

- [x] Karena summary tidak bisa memberikan pola teks.
- [ ] Karena summary terlalu banyak fitur.
- [x] Karena kita memerlukan identifikasi lebih lanjut untuk posisi huruf ataupun angka yang tidak dapat diberikan oleh summary.
- [ ] Karena library bpa bisa menjawab segala hal.
- [ ] Semua jawaban salah.

Menggunakan function basic_pattern_analysis

Function untuk mengidentifikasi pola yang akan kita gunakan adalah basic_pattern_analysis dengan syntax yang akan kita gunakan sebagai berikut.

```
basic_pattern_analysis(x= objek)
```

Dimana x adalah berupa objek angka, character, vector angka, vector character atau data frame. Untuk kasus kita, maka x adalah variable dari hasil pembacaan dataset pelanggan.

Output dari function ini adalah pengenalan karakter per karakter menjadi simbol berikut:

- Tiap huruf besar A s/d Z akan direpresentasikan oleh huruf A.
- Tiap huruf kecil a s/d z akan direpresentasikan oleh huruf a.
- Tiap angka 0 s/d 9 akan direpresentasikan oleh angka 9.
- Spasi dan tab akan direpresentasikan oleh huruf w.
- Semua simbol akan direpresentasikan oleh dirinya sendiri. Contoh: tanda minus (-) akan tetap direpresentasikan dengan tanda minus (-).
- Missing value NA akan direpresentasikan oleh NA.
- NaN (Not a Number) akan direpresentasikan sebagai "AaA".

Sebagai contoh, jika kita identifikasi pola teks "DQLab" dengan fungsi basic_pattern_analysis sebagi beriku:

```
basic_pattern_analysis(x="DQLab")
```

akan menghasilkan output sebagai berikut:

```
[1] "AAAaa"
```

Dimana [1] adalah tampilan index, sedangkan teks "AAAaa" adalah identifikasi pola tiga huruf besar diikuti dua huruf kecil.

Contoh lain, jika kita masukkan

basic_pattern_analysis(x="17 Agustus 1945")

akan menghasilkan output sebagai berikut:

```
[1] "99wAaaaaaaw9999"
```

Dimana [1] adalah tampilan index, sedangkan teks "99wAaaaaaaw9999" adalah identifikasi pola dua angka, satu spasi, satu huruf besar, enam huruf kecil, satu spasi, dan empat angka.

Tugas Praktek

Berdasarkan penjelasan dari lesson di atas, ganti […1…] dengan pemanggilan function basic_pattern_analysis untuk teks "DQLab". Kemudian ganti bagian […2…] pada code editor dengan pemanggilan function basic_pattern_analysis untuk teks "17 Agustus 1945".

Dan terakhir ganti bagian […3…] pada code editor dengan pemanggilan function basic_pattern_analysis untuk angka 3.14 - untuk angka tidak perlu ditulis dengan tanda kutip.

Code Editor

```
library(bpa)

#Membaca dataset pelanggan

data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx",sheet="Pelanggan")

#Menggunakan function basic_pattern_analysis

basic_pattern_analysis(x="DQLab")

basic_pattern_analysis(x="17 Agustus 1945")

basic_pattern_analysis(x=3.14)
```

Console

```
> library(openxlsx)

> library(bpa)

> #Membaca dataset pelanggan
> data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx",sheet="Pelanggan")

> #Menggunakan function basic_pattern_analysis
> basic_pattern_analysis(x="DQLab")
[1] "AAAaa"

> basic_pattern_analysis(x="17 Agustus 1945")
[1] "99wAaaaaaaw9999"

> basic_pattern_analysis(x=3.14)
[1] "9.99"
```

Profiling terhadap vector

Selain satu teks, function basic_pattern_analysis juga bisa digunakan untuk vector seperti pada contoh berikut.

```
basic_pattern_analysis(c("KD-001", "DQLab", "KD-002"))
```

Parameter x pada praktek sebelumnya tidak perlu dimasukkan lagi dalam hal ini. Dan output dari perintah di atas adalah sesuai urutan vector seperti berikut.

```
[1] "AA-999" "AAAaa" "AA-999"
```

Terlihat teks pertama dan ketiga polanya sama, sedangkan teks kedua berbeda sendiri.

Tugas Praktek

Gantilah bagian […] pada code editor dengan function basic_pattern_analysis dengan input berupa vector yang terdiri dari teks "KD-008", "012345", "KD-010".

Code Editor

```
library(openxlsx)

library(bpa)

#Membaca dataset pelanggan

data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx",sheet= "Pelanggan")

#Menggunakan function basic_pattern_analysis

basic_pattern_analysis(c("KD-008","012345","KD-010"))
```

Console

```
> library(openxlsx)

> library(bpa)

> #Membaca dataset pelanggan
> data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pela
nggan.xlsx",sheet="Pelanggan")

> #Menggunakan function basic_pattern_analysis
> basic_pattern_analysis(c("KD-008","012345","KD-010"))
[1] "AA-999" "999999" "AA-999"
```

Menggunakan parameter unique_only=TRUE

Kembali pada contoh pada lesson sebelumnya sebagai berikut.

```
basic_pattern_analysis(c("KD-001", "DQLab", "KD-002"))
```

Yang menghasilkan output berikut.

```
[1] "AA-999" "AAAaa" "AA-999"
```

Dimana terdapat dua pola yang sama. Tampilan pola dengan data satu per satu seperti ini masih bisa kita identifikasi karena kebetulan cuma tiga data.

Bagaimana jika datanya berjumlah puluhan bahkan ribuan? Tentunya akan lebih sulit proses identifikasinya mana pola yang sama atau berulang. Akan lebih bagus jika ada ringkasan informasi seperti summary di atas […]

Beruntung function ini juga memiliki parameter unique_only yang jika diberikan nilai TRUE akan memberikan pola yang unik saja dan jumlah dari masing-masing pola yang teridentifikasi.

Contoh pengunaannya dengan modifikasi perintah di atas jadinya adalah sebagai berikut:

```
basic_pattern_analysis(c("KD-001", "DQLab", "KD-002"),
unique_only=TRUE)
```

Kali ini perintahnya akan menghasilkan output sebagai berikut.

```
AA-999  AAAaa      2      1
```

Dengan pola yang teridentifikasi adalah sebagai berikut:

- AA-999 muncul sebanyak 2 kali.
- AAAaa muncul sebanyak 1 kali.

Dengan informasi dari frekuensi ini, kita bisa mengidentifikasi distribusi pola yang tidak umum atau anomali.

Tugas Praktek

Gantilah bagian […] pada code editor dengan function basic_pattern_analysis dengan input berupa vector yang terdiri dari teks "KD-008", "012345", "KD-010" sehingga mendapatkan informasi pola unik dan jumlah kemunculan pada vector tersebut.

Code Editor

```
library(openxlsx)

library(bpa)


#Membaca dataset pelanggan

data.pelanggan <-
read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx",sheet=
"Pelanggan")


#Menggunakan function basic_pattern_analysis

basic_pattern_analysis(c("KD-008","012345","KD-010"),unique_only=TRUE)
```

Console

```
> library(openxlsx)

> library(bpa)

> #Membaca dataset pelanggan
> data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pela
nggan.xlsx",sheet="Pelanggan")

> #Menggunakan function basic_pattern_analysis
> basic_pattern_analysis(c("KD-008","012345","KD-010"),unique_only=TRUE)

999999 AA-999
     1      2
```

Profiling terhadap kolom Kode Pelanggan

Pada bab pengenalan dataset, telah diinformasikan bahwa kolom Kode Pelanggan memiliki pola yang berbeda – atau pola tidak standar atau anomali yang harus diperbaiki.

Pertanyaannya, bagaimana kita mencarinya dan berapa banyak jumlah anomali ini?

Jawabannya adalah dengan function basic_pattern_analysis yang telah kita gunakan di dua praktek sebelum ini, namun kali ini kita menggunakan input berupa kolom Kode.Pelanggan dari data.frame.

Berikut adalah contoh penggunaannya:

```
basic_pattern_analysis(data.pelanggan$Kode.Pelanggan,
unique_only = TRUE)
```

dimana:

- **pelanggan** adalah variable bertipe data.frame dari hasil pembacaan file pelanggan.
- **pelanggan$Kode.Pelanggan** adalah kolom Kode.Pelanggan dari variable data.pelanggan**.**
- unique_only = TRUE adalah parameter

Untuk mencoba function ini dan melihat hasil apa yang akan diperoleh, kita coba lakukan tugas praktek berikut.

Catatan: Kolom kode.pelanggan tidak perlu dikonversi menjadi factor untuk menggunakan function basic_pattern_analysis.

Tugas Praktek

Gunakan function basic_pattern_analysis untuk mengidentifikasi pola pada Kode.Pelanggan untuk menggantikan bagian […].

Jika berjalan dengan lancar maka akan diperoleh hasil berikut.

```
 AA-9999 AA-99999
       1      154
```

Terlihat ada dua pola yang teridentifikasi yaitu "AA-9999" dengan jumlah data hanya 1, dan pola "AA-99999" dengan jumlah sebanyak 154 data.

Dengan melihat fungsi kolom Kode Pelanggan yang merupakan kolom identifikasi dan harusnya memiliki pola yang konsisten, satu diantara 154 data ini tentunya adalah anomali atau *outlier*.

Pada praktek berikutnya, kita akan melakukan filter terhadap dataset telah kita identifikasi pola *outlier*-nya.

Code Editor

```
library(openxlsx)

library(bpa)


#Membaca dataset pelanggan

data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx",sheet= "Pelanggan")


#Menggunakan function basic_pattern_analysis pada kolom Kode.Pelanggan

basic_pattern_analysis(data.pelanggan$Kode.Pelanggan, unique_only=TRUE)
```

Console

```
> library(openxlsx)

> library(bpa)

> #Membaca dataset pelanggan
> data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pela
nggan.xlsx",sheet="Pelanggan")

> #Menggunakan function basic_pattern_analysis pada kolom Kode.Pelanggan
> basic_pattern_analysis(data.pelanggan$Kode.Pelanggan, unique_only=TRUE)

 AA-9999 AA-99999
       1       154
```

Filter Data dengan pola anomali

Pada praktek praktek "Profiling terhadap kolom Kode Pelanggan", kita telah mendapatkan pola anomali yaitu "AA-9999". Tahap berikutnya adalah bagaimana mengambil porsi dari dataset pelanggan dengan pola ini.

Ada dua proses, yaitu pertama membandingkan seluruh pola dengan teks anomali menggunakan operator == (tanda sama dengan ganda).

Bentuk penggunaannya sebagai berikut.

```
basic_pattern_analysis(data.pelanggan$Kode.Pelanggan)=="AA-9999"
```

Perintah ini akan menghasilkan daftar nilai TRUE/FALSE – dimana hasil akan TRUE jika operator == menemukan teks yang sama, FALSE jika sebaliknya – sebagai berikut.

```
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[49] FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

Terlihat hanya ada satu nilai TRUE dari seluruh data yang ditelusuri oleh operator ==.

Catatan: DQLab memberi tanda merah tebal untuk membedakan hasil yang mayoritas bernilai FALSE sehingga Anda cepat melakukan identifikasi. Hasil eksekusi aslinya tentu tidak memiliki pembeda warna ini.

Proses selanjutnya adalah menggunakan daftar nilai TRUE/FALSE ini untuk melakukan filter dengan konstruksi berikut.

```
data.pelanggan[ daftar_nilai_true, ]
```

dimana data.pelanggan diikuti dengan indeks hasil scan diikuti tanda koma (,). Sehingga konstruksi selengkapnya ditulis sebagai berikut.

```
data.pelanggan[ basic_pattern_analysis(data.pelanggan$Kode.Pelanggan)=="AA-9999" , ]
```

Cobalah gunakan konstruksi ini pada tugas praktek berikut.

Tugas Praktek

Tampilkan data.pelanggan dengan isi data pada kolom Kode.Pelanggan yang memiliki pola anomali "AA-9999" .

Jika berjalan dengan lancar maka akan diperoleh hasil berikut.

```
   Kode.Pelanggan  Nama.Lengkap                                Alamat
51        KD-0047 Puspita Citra Perum Bimasakti Raya, Blok A No. 10
   Tanggal.Lahir Aktif Kode.Pos        No.Telepon Nilai.Belanja.Setahun
51 19 Maret 1950     1   764450 +6282793268821143               950200
```

Berikut adalah keterangan detil dari hasil tersebut.

Komponen	Keterangan
51	Posisi data yang ditemukan ada pada baris ke 51.
Kode.Pelanggan KD-0047	Detil data dimana kolom Kode.Pelanggan dari data yang ditemukan memiliki nilai KD-0047
Nama.Lengkap Puspita Citra	Detil data dimana kolom Nama.Lengkap dari data yang ditemukan memiliki nilai Puspita Citra
Alamat Perum Bimasakti Raya, Blok A No. 10	Detil data dimana kolom Alamat dari data yang ditemukan memiliki nilai `Perum Bimasakti Raya, Blok A No. 10`
Tanggal.Lahir 19 Maret 1950	Detil data dimana kolom Tanggal.Lahir dari data yang ditemukan memiliki nilai `19 Maret 1950`
Aktif 1	Detil data dimana kolom Aktif dari data yang ditemukan memiliki nilai `1`
Kode.Pos 764450	Detil data dimana kolom Kode.Pos dari data yang ditemukan memiliki nilai `764450`
No.Telepon +6282793268821143	Detil data dimana kolom No.Telepon dari data yang ditemukan memiliki nilai `+6282793268821143`
Nilai.Belanja.Setahun 950200	Detil data dimana kolom `Nilai.Belanja.Setahun` dari data yang ditemukan memiliki nilai `950200`

Code Editor

```
library(openxlsx)

library(bpa)


#Membaca dataset pelanggan

data.pelanggan <-
read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx",sheet=
"Pelanggan")


#Mengambil dataset yang memiliki pola teks "AA-9999" di kolom Kode.Pelanggan

data.pelanggan[basic_pattern_analysis(data.pelanggan$Kode.Pelanggan)=="AA-9999",
]
```

Console

```
> library(openxlsx)

> library(bpa)

> #Membaca dataset pelanggan
> data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pela
nggan.xlsx",sheet="Pelanggan")

> #Mengambil dataset yang memiliki pola teks "AA-9999" di kolom Kode.Pelanggan
> data.pelanggan[basic_pattern_analysis(data.pelanggan$Kode.Pelanggan)=="AA-9999", ]
   Kode.Pelanggan  Nama.Lengkap                                Alamat
51        KD-0047 Puspita Citra Perum Bimasakti Raya, Blok A No. 10
   Tanggal.Lahir Aktif Kode.Pos        No.Telepon Nilai.Belanja.Setahun
51 19 Maret 1950     1   764450 +6282793268821143               950200
```

Profiling terhadap kolom Nama

Melanjutkan profiling kita, praktek kali ini kita akan memfokuskan diri pada kolom **Nama.Lengkap** dengan masih menggunakan function yang sama.

Nah, pada praktek kali ini kita juga akan memberikan satu tip, disini kita akan mengambil kolom tersebut bukan dengan mencantumkan **Nama.Lengkap**, tapi dengan **Nama** saja.

Ini memungkinkan karena kolom dengan awalan **Nama** hanya ada satu. Sebagai perbandingan, kalau mencamtumkan **Kode** tidak akan bisa karena awalan **Kode** ada di dua kolom, yaitu **Kode.Pelanggan** dan **Kode.Pos**.

Tugas Praktek

Gunakan function basic_pattern_analysis untuk mengidentifikasi pola pada kolom

Nama

dengan mengganti bagian […].

Jika berjalan dengan lancar maka akan diperoleh hasil yang sebagian terlihat sebagai berikut.

```
                                A.wAaaaaaaa                                AA.wAaaaaawAaaaaa
                                          1                                          1
                             AA.wAaaaawAaaaaaa                          AAA.wAaaaawAaaaaaaaaaaa
                                          1                                          1
…
…
                                          3                                          1
AaaaaaaawAaaaaaaaaaaw(999-9999999999)   AaaaaaaawAaaaaaaaaw-w99999999999999)
                                          1                                          1
…
… s
                                     AwAaaaa                                AwAaaawAaaaaaa
                                          1                                          1
                           aa.wAaaawAaaaaaaaaa
                                          2
```

Terlihat ada pola nama yang mengandung 9. Ini artinya pada nama tersebut mengandung angka, sesuatu yang tidak lazim.

Code Editor

```
library(openxlsx)

library(bpa)


#Membaca dataset pelanggan

data.pelanggan <-
read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx",sheet=
"Pelanggan")


#Menggunakan function basic_pattern_analysis pada kolom Nama

basic_pattern_analysis(data.pelanggan$Nama, unique_only=TRUE)
```

Console

```
> library(openxlsx)

> library(bpa)

> #Membaca dataset pelanggan
> data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pela
nggan.xlsx",sheet="Pelanggan")

> #Menggunakan function basic_pattern_analysis pada kolom Nama
> basic_pattern_analysis(data.pelanggan$Nama, unique_only=TRUE)

                          A.wAaaaaaaa                     AA.wAaaaaawAaaaaa
                                    1                                     1
                    AA.wAaaaawAaaaaaa                AAA.wAaaaawAaaaaaaaaaaa
                                    1                                     1
                AAA.wAaaawAaaaaaaaaa                     Aa.wAaawwwwAaaaaaa
                                    1                                     1
                               Aaa                                   Aaaaa
                                    1                                     2
                           Aaaaaaa                        Aaaaaaa,wAaaa
                                    5                                     1
                          Aaaaaaaa                      AaaaaaaaaaawAaaa
                                    1                                     1
                  AaaaaaaaaawAaaaaa                     AaaaaaaaawAaaaa
                                    1                                     2
                  AaaaaaaaawAaaaaaa        AaaaaaaaawAaaaaaaaaaa,wAAAA,wAA
                                    1                                     1
               AaaaaaaaawwwwwAaaaaaa                      AaaaaaaawAaaa
                                    1                                     2
                     AaaaaaaawAaaaa                     AaaaaaaawAaaaaa
```

```
                                3                                 2
                  AaaaaaaawAaaaaaa                  AaaaaaaawAaaaaaaa
                                3                                 1
AaaaaaaawAaaaaaaaaaaw(999-9999999999)  AaaaaaaawAaaaaaaaaw-w99999999999999)
                                1                                 1
           AaaaaaaawAaaaaaaawAaaaaaa                       AaaaaaawAaa
                                1                                 1
                   AaaaaaawAaaa                      AaaaaaawAaaaa
                                2                                 4
                  AaaaaaawAaaaaaa                AaaaaaawAaaaaaaaawAaaaa
                                7                                 1
                  AaaaaaawAaaawA.                       AaaaaawA.w
                                1                                 1
                   AaaaaawAaaa                       AaaaaawAaaaa
                                1                                 2
                  AaaaaawAaaaaaa                    AaaaaawAaaaaaaa
                                5                                 1
               AaaaaawAaaaaaaaaa                 AaaaaawAaaaaawAaaaa
                                1                                 1
           AaaaaawAaaaaawAaaaaaa              AaaaaawAaaaawAaaaaaaa
                                1                                 1
            AaaaaawAaaaawAaaaaaa                 AaaaaawAaaawAaaaaaa
                                1                                 2
                  AaaaaawAaawAaaa                        AaaaawA.
                                1                                 1
                   AaaaawAaaa                        AaaaawAaaa
                                2                                 2
                  AaaaawAaaaa                       AaaaawAaaaa
                                5                                 3
                  AaaaawAaaaaa                   AaaaawAaaaaa,wAaa.
                                9                                 1
                  AaaaawAaaaaaa                     AaaaawAaaaaaaa
                                6                                 4
               AaaaawAaaaaaaa,wAA                AaaaawAaaaaaawAaaaaa
                                1                                 1
           AaaaawAaaaaaawAaaaaaa              AaaaawAaaaaaawAaaaaaaaa
                                1                                 2
            AaaaawAaaaawAaaaaa                   AaaaawAaawAaaaa
                                2                                 1
           AaaaawAaawAaaaaaa                 AaaawAa.wAaaaawAaaaaaa
                                1                                 1
                  AaaawAaaaa                        AaaawAaaaaa
                                5                                 2
                  AaaawAaaaaaa                      AaaawAaaaaaa'a
                                4                                 1
                  AaaawAaaaaaa                      AaaawAaaaaaaa
                                5                                 1
AaaawAaaaaaawAaaaaw(AaaaaaaawAAA)                 AaaawAaaaaawAaaaaaa
                                1                                 1
           AaaawAaaaaawAaaaaa                   AaaawAaaawAaaaaaa
                                1                                 1
            AaaawAaawAaaa                       AaaawAa|aaaa
                                1                                 1
           Aaaw%$wAaaaaaaa                      AaawAaaaa
                                1                                 2
```

```
                AaawAaaaaa                 AaawAaaaaaaa
                        2                            2
          AaawAaaaaaaa,wA.A.          AaawAaaaaaaaaaa
                        1                            1
             AaawAaaaawAaa          AaawAaaaawAaaaaw
                        1                            1
          AaawAaaaawwAaaaaa          AaawAaawAaaaaaa@,wAA
                        1                            1
                 AwAaaaa                 AwAaaawAaaaaaa
                        1                            1
        aa.wAaaawAaaaaaaaa
                        2
```

Perkenalan function grepl

Teknik filtering dengan menggunakan operator == hanya dapat digunakan untuk mengenal teks yang spesifik dan sama persis. Sebagai contoh kita ingin mencari pola "AA-9999".

Jika ingin mencari teks yang mengandung karakter tertentu di dalamnya, seperti pada kasus profiling **Nama** – kita ingin mencari teks yang mengandung karakter tapi bukan huruf. Ini tentunya akan banyak pola yang bisa terjadi, misalkan "Aaaaaaaa", "AaaaaawAA", dan lain-lain.

Jika dilakukan demikian, bentuk filteringnya adalah daftar teks yang akan panjang sekali dan belum tentu benar. Kita tidak menginginkan hal tersebut, tapi kita ingin kepastian akan satu mekanisme filtering yang pasti benar dan tidak merepotkan?

Untuk hal ini perlu penyaringan menggunakan konstruksi bernama **regular expression (regex)** dan diimplementasikan di R dengan function bernama **grepl**.

Function **grepl** digunakan untuk menyaring suatu data berdasarkan pola regex. Regex adalah suatu bahasa yang sangat lengkap untuk mendeteksi pola teks yang beragam.

Regex juga sangat kompleks, dan agar menjaga fokus di course ini maka kita tidak akan membahas regex secara mendalam. Tapi sebagai gantinya, akan diberikan penjelasan apa yang dilakukan oleh pola regex yang diberikan sebagai petunjuk untuk melakukan filter.

Penggunaan function grepl adalah sebagai berikut:

```
grepl(pattern=pola_pattern_regex, x = data)
```

dimana:

- **pola_pattern_regex**: adalah pola regular expression (regex) yang dapat digunakan untuk filter data.
- **data**: adalah data berupa **teks** atau vector dari character.

Hasil output dari grepl adalah nilai TRUE jika ada pola yang terdapat di dalam teks / data, sebaliknya FALSE jika tidak ada pola yang terdapat di dalam teks / data.

Berikut adalah beberapa pola regex dan penjelasannya.

Pola regex	Apa yang dilakukan	Penjelasan Detil
[a]	Mencari karakter a di dalam teks	[] = adalah character class, dimana kumpulan karakter yang akan dicari dikumpulkan di tanda kurung siku ini a = karakter a
aa	Mencari dua karakter a berurutan di dalam teks	aa = dua karakter a berurutan

[ab]	Mencari karakter a atau b di dalam teks	[] = adalah character class, dimana kumpulan karakter yang akan dicari dikumpulkan di tanda kurung siku ini ab = karakter a atau b
[^a]	Mencari karakter bukan a di dalam teks	[] = adalah character class, dimana kumpulan karakter yang akan dicari dikumpulkan di tanda kurung siku ini ^ = tanda ^ sebelum karakter merupakan tanda negasi (bukan) ^a = bukan karakter a
[^ab]	Mencari karakter yang bukan a dan b dalam teks	[] = adalah character class, dimana kumpulan karakter yang akan dicari dikumpulkan di tanda kurung siku ini ^ = tanda ^ sebelum karakter merupakan tanda negasi (bukan) ^a = bukan karakter a

Tugas Praktek

Pada code editor telah terdapat code yang terdiri dari beberapa perintah grepl yang "setengah jadi".

Gantilah bagian-bagian berikut sesuai petunjuk:

- […1…] dengan [a]
- […2…] dengan [^a]
- […3…] dengan [bc]
- […4…] dengan [^bc]
- […5…] dengan [s]
- […6…] dengan [^s]
- […7…] dengan aa

Jika berjalan dengan lancar maka akan diperoleh output sebagai berikut.

```
> grepl(pattern="[a]", "pelanggan")
[1] TRUE


> grepl(pattern="[^a]", "pelanggan")
[1] TRUE
```

```
> grepl(pattern="[bc]", "pelanggan")
[1] FALSE

> grepl(pattern="[^bc]", "pelanggan")
[1] TRUE

> grepl(pattern="[s]", "pelanggan")
[1] FALSE

> grepl(pattern="[^s]", "pelanggan")
[1] TRUE
```

Berikut keterangan hasil-hasilnya

Perintah	Hasil	Penjelasan Detil
`grepl(pattern="[a]", "pelanggan")`	TRUE	Teks "pelanggan" mengandung karakter "a"
`grepl(pattern="[^a]", "pelanggan")`	TRUE	Teks "pelanggan" mengandung karakter bukan "a" seperti "p", "e", "l", "n", "g"
`grepl(pattern="[bc]", "pelanggan")`	FALSE	Teks "pelanggan" tidak mengandung karakter "b" ataupun "c"
`grepl(pattern="[^bc]", "pelanggan")`	TRUE	Teks "pelanggan" mengandung karakter bukan "b" dan "c"
`grepl(pattern="[s]", "pelanggan")`	FALSE	Teks "pelanggan" tidak mengandung karakter "s"
`grepl(pattern="[^s]", "pelanggan")`	TRUE	Teks "pelanggan" mengandung karakter bukan "s"
`grepl(pattern="aa", "pelanggan")`	FALSE	Teks "pelanggan" mengandung dua karakter "a" secara berurutan

Code Editor

```
grepl(pattern="[a]", x="pelanggan")
grepl(pattern="[^a]", x="pelanggan")
grepl(pattern="[bc]", x="pelanggan")
grepl(pattern="[^bc]", x="pelanggan")
grepl(pattern="[s]", x="pelanggan")
grepl(pattern="[^s]", x="pelanggan")
grepl(pattern="aa", x="pelanggan")
```

Console

```
> grepl(pattern="[a]", x="pelanggan")
[1] TRUE

> grepl(pattern="[^a]", x="pelanggan")
[1] TRUE

> grepl(pattern="[bc]", x="pelanggan")
[1] FALSE

> grepl(pattern="[^bc]", x="pelanggan")
[1] TRUE

> grepl(pattern="[s]", x="pelanggan")
[1] FALSE

> grepl(pattern="[^s]", x="pelanggan")
[1] TRUE

> grepl(pattern="aa", x="pelanggan")
[1] FALSE
```

Menemukan nama yang mengandung karakter tidak lazim

Dengan mengenal function grepl pada satu praktek sebelum ini, kita sudah siap untuk melakukan filter terhadap hasil profiling dari kolom Nama – dimana isi yang tidak lazim untuk suatu nama kita perlu temukan.

Langkah pertama, tentunya kita perlu mendefinisikan apa yang disebut tidak lazim?

Secara sederhana kita dapat mengatakan nama tidak lazim bila:

- Mengandung karakter bukan huruf, spasi, titik dan koma.
- Memiliki spasi lebih dari satu secara berurutan.

Dengan menggunakan simbol pola dari library bpa, maka definisi di atas dapat dimodelkan sebagai berikut dalam regex:

- [^Aaw.,]
- ww

Dan jika menggunakan grepl, maka konstruksinya adalah sebagai berikut:

- grepl(pattern="[^Aaw.,]", x=basic_pattern_analysis(data.pelanggan$Nama))
- grepl(pattern="ww", x=basic_pattern_analysis(data.pelanggan$Nama))

Tugas Praktek

Gantilah bagian […1…] dan […2…] pada code editor untuk mengenali penulisan nama yang tidak lazim:

- […1…] dengan grepl(pattern="[^Aaw.,]", x=basic_pattern_analysis(data.pelanggan$Nama))
- […2…] dengan grepl(pattern="ww", x=basic_pattern_analysis(data.pelanggan$Nama))

Jika berjalan dengan lancar maka akan diperoleh output sebagai berikut.

```
> data.pelanggan[grepl(pattern="[^Aaw.,]", x=basic_pattern_analysis(data.pelanggan$Na
ma)),]
    Kode.Pelanggan                          Nama.Lengkap
15        KD-00113                     Edi %$ Alexander
36        KD-00010                 Ibu Sri Wahyuni@, IR
50        KD-00039    Joko Wiryanto Abadi (Pelanggan OKE)
68        KD-00005 Prihatin Setyonugroho (021-5555555544)
```

```
70        KD-00001                          Agus Cahyono's
88        KD-00063   Widianto Nuryajaya - 08222222999111)
107       KD-00120                            Dewi Sr|yani
                                      Alamat    Tanggal.Lahir Aktif Kode.Pos
15     Taman Bunga Langit, Jl. Selatan No. 12 22 Februari 2000     0   712984
36            Perum Venus, Gg. Harimau No. 1A  23 Oktober 1991     1   987453
50            Perum Indah Supernova II, No. 9       05-09-1990     1   764449
68   Jln. Tegal Sari Indah, No. D87 -- Kota H  19 Agustus 1986     1   476511
70  Jl. Pulo Bambu No. 15, Kota Tenggara Lama  8 Februari 1967     1   876511
88                  Jl. Macan Buntung, No. 4F       29-02-1969     1   768091
107            Jalan Ring Road Konstan, No. 5       11/29/1967  TRUE   567120
           No.Telepon Nilai.Belanja.Setahun
15   6281413705348345                311000
36   6284079659289143                389400
50   6289122766908102               1086300
68   6286843623971825               1488900
70     08298911112222               1082900
88   6285463027900499               1100200
107 +6285239934324639                273400
```

Dari syarat pertama ketidaklaziman data nama, ternyata ada 7 data yang ditemukan pada baris 15, 36, 50, 68, 70, 88, dan 107. Perhatikan isi kolom Nama yang telah ditandai dengan warna merah.

```
> data.pelanggan[grepl(pattern="ww", x=basic_pattern_analysis(data.pelanggan$Nama)),]
    Kode.Pelanggan           Nama.Lengkap
9         KD-00046     Ir. Ita    Nugraha
35        KD-00117 Florensia     Novianti
145       KD-00108      Ibu Jujur  Suwito
                                      Alamat Tanggal.Lahir Aktif Kode.Pos
9      Vila Bukit Sagitarius, Gang Kelapa No. 6    14-03-1879     1   877521
35  Perumahan Bina Andromeda, Jl. Salmon No. 22    19/08/1950     0   987452
145        Apartement Clifften, Lantai 12 No. 3    02/28/1969     1   768035
           No.Telepon Nilai.Belanja.Setahun
9   6288267903981205                541300
```

```
35  6283166638654813                854400
145 6284037884325249                851600
```

Dari syarat kedua ketidaklaziman data nama, ternyata ada 3 data yang ditemukan pada baris 9, 35, dan 145. Perhatikan isi kolom Nama yang telah ditandai dengan warna merah.

Code Editor

```
library(bpa)


#Membaca dataset pelanggan

data.pelanggan <-
read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx",
sheet="Pelanggan")

#Menggunakan function grepl untuk mengambil pola nama tidak lazim

data.pelanggan[grepl(pattern="[^Aaw.,]",x=basic_pattern_analysis(data.pelanggan$Na
ma)),]

data.pelanggan[grepl(pattern="WW",x=basic_pattern_analysis(data.pelanggan$Nama)),
]
```

Console

```
> library(openxlsx)

> library(bpa)

> #Membaca dataset pelanggan
> data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pela
nggan.xlsx", sheet="Pelanggan")

> #Menggunakan function grepl untuk mengambil pola nama tidak lazim
> data.pelanggan[grepl(pattern="[^Aaw.,]",x=basic_pattern_analysis(data.pelanggan$Nam
a)),]
    Kode.Pelanggan                            Nama.Lengkap
15         KD-00113                        Edi %$ Alexander
36         KD-00010                     Ibu Sri Wahyuni@, IR
50         KD-00039     Joko Wiryanto Abadi (Pelanggan OKE)
68         KD-00005 Prihatin Setyonugroho (021-5555555544)
70         KD-00001                          Agus Cahyono's
88         KD-00063   Widianto Nuryajaya - 08222222999111)
107        KD-00120                            Dewi Sr|yani
```

```
                                      Alamat    Tanggal.Lahir Aktif Kode.Pos
15     Taman Bunga Langit, Jl. Selatan No. 12 22 Februari 2000     0   712984
36            Perum Venus, Gg. Harimau No. 1A  23 Oktober 1991     1   987453
50            Perum Indah Supernova II, No. 9       05-09-1990     1   764449
68   Jln. Tegal Sari Indah, No. D87 -- Kota H  19 Agustus 1986     1   476511
70  Jl. Pulo Bambu No. 15, Kota Tenggara Lama  8 Februari 1967     1   876511
88                  Jl. Macan Buntung, No. 4F       29-02-1969     1   768091
107           Jalan Ring Road Konstan, No. 5       11/29/1967  TRUE   567120
          No.Telepon Nilai.Belanja.Setahun
15   6281413705348345                311000
36   6284079659289143                389400
50   6289122766908102               1086300
68   6286843623971825               1488900
70     08298911112222               1082900
88   6285463027900499               1100200
107 +6285239934324639                273400

> data.pelanggan[grepl(pattern="WW",x=basic_pattern_analysis(data.pelanggan$Nama)),]
[1] Kode.Pelanggan        Nama.Lengkap          Alamat
[4] Tanggal.Lahir         Aktif                 Kode.Pos
[7] No.Telepon            Nilai.Belanja.Setahun
<0 rows> (or 0-length row.names)
```

Profiling terhadap seluruh kolom

Akan lebih tepat jika kita tetap melakukan profiling per tiap kolom. Tapi function **basic_pattern_analysis** juga dapat melakukan untuk seluruh kolom dari data.frame Pelanggan.

Untuk melakukan hal ini, kita gunakan input data.frame langsung di dalam function **basic_pattern_analysis**.

Tugas Praktek

Lakukan profiling untuk seluruh kolom dari variable data.pelanggan dengan mengganti bagian […] pada code editor.

Code Editor

```
library(openxlsx)

library(bpa)


#Membaca dataset pelanggan

data.pelanggan <-
read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx",
sheet="Pelanggan")


#Profiling pola seluruh kolom

basic_pattern_analysis(data.pelanggan)
```

Console

```
> library(openxlsx)

> library(bpa)

> #Membaca dataset pelanggan
> data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pela
nggan.xlsx", sheet="Pelanggan")

> #Profiling pola seluruh kolom
> basic_pattern_analysis(data.pelanggan)
    Kode.Pelanggan                          Nama.Lengkap
1         AA-99999                   AaawAaaaaaaa,wA.A.
2         AA-99999                        AaawAaaaawAaa
```

```
3         AA-99999                                AaaaawAaaaaaa
4         AA-99999                             AaaaaaaawAaaaaaa
5         AA-99999                                  AaaaawAaaaaa
6         AA-99999                                 AaaaawAaaaaaa
7         AA-99999                                  AaaawAaaaaaa
8         AA-99999                                  AaaaawAaaaaa
9         AA-99999                             Aa.wAaawwwwAaaaaaa
10        AA-99999                            AaaaawAaaaaaa,wAaa.
11        AA-99999                                 AaaaaaawAaaaa
12        AA-99999                                 AaaawAaawAaaa
13        AA-99999                            AaaaaawAaaawAaaaaaa
14        AA-99999                                 AaaaaaaawAaaa
15        AA-99999                               Aaaw%$wAaaaaaaaa
16        AA-99999                      AaaaawAaaaaaawAaaaaaaaaa
17        AA-99999                           AaaaaaawAaaaaaa
18        AA-99999                                       Aaa
19        AA-99999                            AaawAaaaaaaaaa
20        AA-99999                             AaaaaawAaaaaaa
21        AA-99999                                 AaaaawAaaa
22        AA-99999                            AaaaaaawAaaaaaa
23        AA-99999                                 AaaawAaaaa
24        AA-99999                         AaaaaawAaaaaaawAaaaa
25        AA-99999                              AaaaawAaaaaaaa
26        AA-99999                               Aaaaaa,wAaaa
27        AA-99999                                     Aaaaa
28        AA-99999                              AaaaaawAaaaa
29        AA-99999                           AaaaaaaaawAaaaaaa
30        AA-99999                         AaaawAa.wAaaaawAaaaaaa
31        AA-99999                              AaaaawAaaaaaa
32        AA-99999                                AaaawAaaaa
33        AA-99999                               AaaaaawAaaa
34        AA-99999                               AaaaawAaaaa
35        AA-99999                         AaaaaaaaawwwwwAaaaaaa
36        AA-99999                           AaawAaawAaaaaaa@,wAA
37        AA-99999                              AaaawAaaaaaa
38        AA-99999                                AaaawAaaaa
39        AA-99999                           AaaaaaaaaawAaaaaa
40        AA-99999                              AaawAaaaaaa
41        AA-99999                             AaaaaaawAaaaaa
42        AA-99999                            AA.wAaaaaawAaaaa
43        AA-99999                             AaaaaaawAaaaaa
44        AA-99999                                     Aaaaa
45        AA-99999                              AaaaawAaaaaaa
46        AA-99999                              AaaaaaaawAaaaa
47        AA-99999                                AaaawAaaaaa
48        AA-99999                           AaaawAaaaaawAaaaaa
49        AA-99999                              AaaawAaaaaaa
50        AA-99999    AaaawAaaaaaaawAaaaaw(AaaaaaaaawAAA)
51         AA-9999                              AaaaaaawAaaaa
52        AA-99999                             AaaaaaawAaaaaa
53        AA-99999                              AaaaaawAaaaaa
54        AA-99999                              AaaaaaawAaaaa
55        AA-99999                                AaaawAaaaa
56        AA-99999                            AaaaawAaawAaaaaa
57        AA-99999                             AaaaaaaaawAaaaa
```

```
58        AA-99999                          AaaaaawAaaaaaaa
59        AA-99999                            AaawAaaaaaaa
60        AA-99999                       aa.wAaaawAaaaaaaaa
61        AA-99999                                AaaaawAaa
62        AA-99999                     AaaawAaaaaaawAaaaaaa
63        AA-99999                          AaaaaaawAaaaaaa
64        AA-99999                             AaaaaaawAaa
65        AA-99999                          AaaaaaawAaaaaaa
66        AA-99999                       AAA.wAaaawAaaaaaaaa
67        AA-99999                            AaaaawAaaaaaa
68        AA-99999 AaaaaaaawAaaaaaaaaaaw(999-9999999999)
69        AA-99999                           AaaawAaaaaaaaa
70        AA-99999                           AaaawAaaaaaa'a
71        AA-99999                           AaaaaawAaaaaaa
72        AA-99999                AaaaaaaawAaaaaaaawAaaaaaa
73        AA-99999                            AaaawAaaaaaa
74        AA-99999                            AaaaawAaaaaa
75        AA-99999                              AaaaawAaa
76        AA-99999                           AaaaawAaaaaaa
77        AA-99999                              AaaaawAaaa
78        AA-99999                      AaaaawAaaaawAaaaaa
79        AA-99999                   AaaaaaawAaaaaaaaawAaaaa
80        AA-99999                         AaaaaaawAaaawA.
81        AA-99999                            AaaaaaawAaaa
82        AA-99999                           AaaaaaaawAaaa
83        AA-99999                           AaaaaaaawAaaaa
84        AA-99999                           AaaaawAaaaaaaa
85        AA-99999                            AaaaawAaaaaa
86        AA-99999                            AaaaaaawAaaaa
87        AA-99999                                AaaaawA.
88        AA-99999   AaaaaaaawAaaaaaaaaw-w99999999999999)
89        AA-99999                                Aaaaaaaa
90        AA-99999                       AA.wAaaaawAaaaaaa
91        AA-99999                             AaawAaaaaa
92        AA-99999                            AaaawAaaaaa
93        AA-99999                            AaaawAaaaa
94        AA-99999                           AaaaawAaaaaa
95        AA-99999                            AaaaawAaaaa
96        AA-99999                          AaaaaaaawAaaaaa
97        AA-99999                        AaaaaawAaaaaaaaaa
98        AA-99999                             AaaaawAaaaa
99        AA-99999                           AaaaawAaaaaaaa
100       AA-99999                            AaaawAaaaaaaa
101       AA-99999                           AaaaawAaaaaaaa
102       AA-99999                          AaaaaaaawAaaaaaa
103       AA-99999                                AwAaaaa
104       AA-99999                           AwAaaawAaaaaaa
105       AA-99999                             AaaaawAaaaa
106       AA-99999                                Aaaaaaa
107       AA-99999                            AaaawAa|aaaa
108       AA-99999                           AaaaawAaaaaaa
109       AA-99999                            AaaaawAaaaaa
110       AA-99999                           AaaaawAaaaaaa
111       AA-99999                              AaawAaaaaa
112       AA-99999                              A.wAaaaaaa
```

```
113       AA-99999                          AaaaawAaaaaaa                                  
114       AA-99999                         AaaaaawAaaaaaa                                  
115       AA-99999                          AaaaaawAaaaa                                   
116       AA-99999                            AaawAaaaa                                    
117       AA-99999                        AaaaaaaaawAaaaa                                  
118       AA-99999                        AaaaaawAaawAaaa                                  
119       AA-99999                         AaaaawAaaaaaa                                   
120       AA-99999                AaaaaawAaaaaawAaaaaaaa                                   
121       AA-99999                AaaaaawAaaaaaawAaaaaaa                                   
122       AA-99999                          AaawAaaaa                                      
123       AA-99999                            Aaaaaaa                                      
124       AA-99999                         AaaawAaaaaaa                                    
125       AA-99999                        AaaaaaawAaaaaaa                                  
126       AA-99999              AAA.wAaaaawAaaaaaaaaaaa                                    
127       AA-99999                            Aaaaaaa                                      
128       AA-99999                            Aaaaaaa                                      
129       AA-99999                            Aaaaaaa                                      
130       AA-99999                          AaaaawAaaaa                                    
131       AA-99999                 AaaaaawAaaaawAaaaaaa                                    
132       AA-99999                      AaaaawAaawAaaaa                                    
133       AA-99999                     AaaaaaaawAaaaaa                                     
134       AA-99999                AaaaawAaaaaaawAaaaaaa                                    
135       AA-99999                      AaaawAaaaaaaa                                      
136       AA-99999                      AaaawAaaaaaa                                       
137       AA-99999                   AaaaaaaaaaawAaaa                                      
138       AA-99999                    AaaaawAaaaaaaaa                                      
139       AA-99999        AaaaaaaaawAaaaaaaaaa,wAAAA,wAA                                   
140       AA-99999                AaaaawAaaaaaawAaaaaa                                     
141       AA-99999                   AaawAaaaawAaaaaw                                      
142       AA-99999                  AaaawAaaawAaaaaaa                                      
143       AA-99999                  AaaaawAaaaawAaaaa                                      
144       AA-99999                       AaaaaaawAaaa                                      
145       AA-99999                   AaawAaaaawwAaaaa                                      
146       AA-99999                     AaaaawAaaaaaa                                       
147       AA-99999                     AaaaaaaawAaaaa                                      
148       AA-99999                   AaaaaaaawAaaaaa                                       
149       AA-99999                  AaaaawAaaaaaaa,wAA                                     
150       AA-99999                       AaaaaawAaaaaa                                     
151       AA-99999                         AaaaaawA.w                                      
152       AA-99999                      AaaawAaaaaaa                                       
153       AA-99999              AaaaawAaaaaaawAaaaaaaaa                                    
154       AA-99999                   AaaaaawAaaawAaaaaa                                    
155       AA-99999                   aa.wAaaawAaaaaaaaa                                    
                                              Alamat     Tanggal.Lahir Aktif
1                    AaaawAaaaaaa,wAa.w99w-wAaaawA      9wAaaaaw9999 AAAAA
2                    AaaawAaaaaaa,wAa.w99w-wAaaawA        99-99-9999     9
3                     AaaawAaaaaaa,wAa.w9w-wAaaawA        99-99-9999 AAAAA
4           AaaawAaaaaaawAaaaawAaaaaaa,wAaaawA9-9          99/99/99     9
5           AaaawAaaaaaawAaaaawAaaaaaa,wAaaawA9/9        99-99-9999     9
6              AaaawAaaaaawAaaaa,wAaaawA9w-wAa.w9        99-99-9999     9
7              AaaawAaaaaawAaaaa,wAaaawA9w-wAa.w9        99-99-9999     9
8          AaaawAaaaawAaaaaaaaa,wAaaa.wAaaaawAa.w9       99-99-9999     9
9          AaaawAaaaawAaaaaaaaa,wAaaawAaaaawAa.w9        99-99-9999     9
10             AaaawAaaaawAaaaaaaaa,wAaaawA9wAa.w9       99-99-9999     9
11                   AaaaawAaaawAaaaa,wAaaawAAwAa.w9       99/99/99     9
```

12	AaaaawAaaawAaaaa,wAaaawAAwAa.w9	99/99/9999	9
13	AaaaawAaaaawAaaaaa,wAa.wAaaaawAa.w9	99/99/9999	9
14	AaaaawAaaaawAaaaaa,wAa.wAaaaawAa.w9	99wAaaaaaaw9999	9
15	AaaaawAaaaawAaaaaa,wAa.wAaaaaaawAa.w99	99wAaaaaaaaw9999	A
16	AaaaawAaaaawAaaaaa,wAa.wAaaaawAaaawAa.w9	99wAaaaaaaw9999	9
17	AaaaawAaaaaaawAaaaa,wAa.w9wAa.w9	99-99-9999	9
18	AaaaawAaaaaaawAaaaa,wAa.w9wAa.w9	99/99/99	9
19	AaaaawAaaaawAaaaaa,wAaaaaaw9wAa.w99	99/99/99	9
20	AaaaawAaaaawAaaaa,wAaaaaaw9wAa.w9	99/99/9999	9
21	AaaawAaaaaa,wAa.w9wAAw999/999	99-99-9999	9
22	AaaawAaaaaawAaaaa,wAaaawA9/9	99/99/9999	9
23	AaaaawAaaa,wAa.w99A,wAaaawA	9wAaaaaaaaw9999	9
24	AaaaawAaaa,wAa.w99A,wAaaawA	99wAaaaaw9999	9
25	AaaawAaaaawAa.w99,wAaaawAaaaaaaawAaaa	99/99/99	AAAA
26	AaaawAaaaawAa.w99,wAaaawAaaaaaaawAaaa	99/99/9999	9
27	AaaaaawAaaawAaaaa,wAa.wAaaaaawAaaaaaawAa.w99AA	99wAaaaaaaaw9999	9
28	AaaaaaaaawAaaaaawAaaaaaaaaaaaa,wAa.wAaaaaawAa.w9	99-99-9999	9
29	AaaaaaaaawAaaaaawAaaaaaaaaa,wAa.wAaaaawAa.w9A	99wAaaaw9999	A
30	AaaaaaaaawAaaaaawAaaaaaaaaa,wAa.wAaaaawAa.w9A	99-99-9999	9
31	AaaaaaaaawAaaawAaaa,wAaaawAaaaaaaw-wAa.w99	99-99-9999	AAAA
32	AaaaaaaaawAaaawAaaa,wAaaawAaaaaaaawAa.w9	99wAaaaw9999	9
33	AaaaaaaaawAaaaaaaa,wAa.wAaaaaawAaawAa.w99	99/99/9999	9
34	AaaaaaaaawAaaawAaaaaaaa,wAa.wAaaawAa.w9	99/99/9999	AAAAA
35	AaaaaaaaawAaaawAaaaaaaa,wAa.wAaaaaawAa.w99	99/99/9999	9
36	AaaaawAaaaa,wAa.wAaaaaaawAa.w9A	99wAaaaaaaw9999	9
37	AaaaawAaaaa,wAaaa.wAaaaaaawAa.w99	99wAaaaaaaw9999	9
38	AaaaawAaaaa,wAaaa.wAaaaaaawAa.w9A	99/99/99	9
39	AaaaawAaaaa,wAaaaa.wAaaaaawAa.w9	99wAaaaw9999	9
40	AaaaawAaaaa,wAaaaa.wAaaaaaawAa.w99	99/99/9999	9
41	AaaaawAaaaaaw99,wAaaawAaaaaawAa.w9	99/99/9999	AAAA
42	AaaaawAaaaa,wAaaawAwAa.w9	99wAaaaaaaaaw9999	9
43	AaaaawAaaawAaaaaawAaaaaaawAaaaa,wAaaawAaaaaawAa.w9	99/99/9999	9
44	AaaaawAaaawAaaaaawA.A.,wAaaawAaaaaaawAa.w9	99/99/9999	9
45	AaaaawAaaawAaaaaawA.A.,wAaaawAaaaaaawAa.w9	99-99-9999	9
46	AaaaawAaaawAaaaaaa,wAa.wA99	99-99-9999	9
47	AaaaawAaaawAaaaaaa,wAa.w9A	99/99/9999	9
48	AaaaawAaaawAaaaaaa,wAa.w9A	99/99/9999	9
49	AaaaawAaaaawAaaaaaaaa,wAa.w9	99/99/9999	9
50	AaaaawAaaaawAaaaaaaaawAA,wAa.w9	99-99-9999	9
51	AaaaawAaaaaaaawAaaa,wAaaawAwAa.w99	99wAaaaaw9999	9
52	AaaaawAaaaaaaawAaaa,wAaaawAwAa.w99	99-99-9999	9
53	AaaawAaaaaaaaaa,wAa.w99A	99wAaaaaaaaw9999	AAAA
54	AaaawAaaaaaaaaa,wAa.w9A	99/99/9999	9
55	AaaawA,wwAaaaawAaaaawAaaaaaawAa.w99999	9wAaaaaw9999	9
56	AaaaaaaawAaaaaaa-Aaaaaaa,wAa.w999w	99/99/9999	9
57	AaaaaaaawAaaaaaa-Aaaaaaa,wAa.w999	99-99-9999	9
58	AaaaaaaawAaaaaawAaaaaaa,wAaaawAwAa.w99	99/99/9999	9
59	AaaaaaaawAaaaaawAaaaaaa,wAaaawAwAa.w99	99-99-9999	AAAA
60	AaaaaaaawAaaaaawAaaaaaa,wAa.w9A	99wAaaw9999	9
61	AaaaaaaawAaaaaaawAaaaaa,wAaaawAAw-w99/99	99/99/9999	9
62	AaaaaaaawAaaaaaawAaaaaaa,wAaaawAaaaaaawAa.w9	99/99/9999	AAAA
63	AaaaaaawAaaaaa,wAaaawA9w-wAa.w99	99/99/99	9
64	AaaaaaawAaaaaaa,wAa.w99,wAaaawA	99-99-9999	9
65	AaaaaaawAaaaaaa,wAa.w9	99wAaaaaaaaw9999	9
66	AaaaaaaawAaaaaaa,wAa.w99w-wAaaawA	99-99-9999	AAAA

67	AaaawAaaawAaaaaaa,wAa.w99A	99-99-9999	9
68	Aaa.wAaaaawAaaawAaaaa,wAa.wA99w--wAaaawA	99wAaaaaaaw9999	9
69	Aaa.wAaaaawAaaawAaaaa,wAa.wA99w--wAaaawA	99-99-9999	9
70	Aa.wAaaawAaaaawAa.w99,wAaaawAaaaaaaawAaaa	9wAaaaaaaaw9999	9
71	Aaa.wAaaaaaawAA,wAaaawAaaaawAa.w9	99-99-9999	9
72	Aaa.wAaaaaaawAAw-wAaaawAaaaawAa.w9	99wAaaaaaaaw9999	9
73	Aaa.wAaaaaaaawAa.w99,wAAw999w-waaaawA	99/99/9999	9
74	Aaaw.wA.wAaaawAaaaawAaaaawAaaawAwAa.w9	99/99/99	9
75	Aa.wAaaaawAaaaaaaawAaaa,wAa.wA99	99-99-9999	9
76	Aa.wAaaaawAaaaaaaawAaaa,wAa.wA99	99-99-9999	AAAA
77	Aa.wAaaaawAaaaaaawAa.w99999,wAaaawA	99-99-9999	9
78	Aa.wAaaawAaaaawAaaaaaaa,wAaaawAAw999	99/99/9999	9
79	Aa.wAaaawAaaaawAaaaaa,wAa.w999w-wAaaawA	99-99-9999	9
80	Aaaaa.wAaaaawAaaaawAa.w99w-wAaaawAaaaaaaawAaaa	99-99-9999	AAAA
81	Aa.wAaaaawAaaaaaawAa.w999	99-99-9999	9
82	Aaaaa.wAaaaawAaaaaaawAa.w999	99wAaaaaw9999	9
83	Aa.wAaaaawAaaaaaawAa.w9999	99-99-9999	9
84	AaaaaaaaawAaaaawAaaaaaaa,wAa.w9wAa.w9	99wAaaaaaaaw9999	9
85	Aa.wAaaaawAaaaaaawAa.w999	99/99/9999	9
86	AA.wAaaaaaaa,wAa.w99AAA	99/99/9999	9
87	AA.wAaaaaaaa,wAa.w99AAAw	99wAaaw9999	9
88	Aa.wAaaaawAaaaaaa,wAa.w9A	99-99-9999	9
89	Aa.wAaaaawAaaaaaa,wAa.w9Aw-wAaaawA	99wAaaaaaaw9999	9
90	Aa.wAaaaawAaaaaaa,wAa.w9A	99/99/9999	9
91	Aa.wAa.wAaaaaa,wAaaawA9w-wAa.w9Aw	99-99-9999	AAAAA
92	Aa.wAaaaaaaaaawAaaaa,wAaaawA9wAa.w99wAAw99	99/99/99	AAAA
93	Aa.wAaaaaaaaaawAaaaa,wAaaawA9wAa.w99	99wAaaaaaaaw9999	9
94	Aa.wAaaaawAaaa,wAaaawAaaaaAaaaaa	99wAaaaaaaw9999	9
95	Aa.wAaaaaaawAaaaaa,wAaaawA9w-wAa.w9	99/99/9999	9
96	Aa.wAaaaaaawAaaaaaaaa,wAa.w99	99-99-9999	9
97	Aa.wAaaaaaawAaaaaaaaa,wAa.w99	99/99/9999	AAAA
98	Aa.wAaaaaaw999,wAaaawA	99wAaaaaaaaaw9999	AAAAA
99	AaaaaaaaawAaaaawAaaaaaaaa,wAa.w9wAa.w9	99/99/99	9
100	AaaaaaaaawAaaaawAaaaa,wAaaaawAaaaa,wAaw9wAa.wA9	9wAaaaaw9999	AAAAA
101	AaaaaaaaawAaaaawAaaaawAaaaawAaaaa,wAaw9wAa.wA9	9wAaaaaw9999	9
102	Aaaaa.wAaaaawAaaa,wAa.wA99w-wAaaawA	-	AAAA
103	Aaaaa.wAaaaawAaawAa.w9,wAa.w999	99-99-9999	9
104	AaaaawAaaaawAaaaaaaawAaaawAa.wA-99	99wAaaaaaaw9999	9
105	AaaaawAaaaaaaawAaaaawAaaaaaaa,wAa.w9999	99wAaaaw9999	AAAAA
106	AaaaawAaaawAaaawAaaaaaaa,wAa.w9wAAw9	99/99/9999	9
107	AaaaawAaaawAaaawAaaaaaa,wAa.w9	99/99/9999	AAAA
108	AaaaawAaaawAaaaaaawAaaaa,wAa.w99	99-99-9999	9
109	AaaaawAaaawAaaawAaaaa,wAa.w99w-wAaaawA	99/99/9999	9
110	AaaaawAaaawAaaawAaaaa,wAaaawA,wAa.w99	99wAaaaaaaaw9999	9
111	AaaaawAaaawAaaaaaaa,wAaaawAwAa.w9	99/99/99	9
112	AaaaawAaaaaaawAa.w9,wAaaaaaawAaaawAaaaawAaaaaaa	99/99/9999	9
113	AaaaawAaaawAaaa,wAa.w99,wAaaawAA	99-99-9999	AAAA
114	AaaaawAaaawAaaa,wAa.w99,wAaaawAA	99wAaaaw9999	9
115	AaaaawAaaawAaaaa,wAa.w999	99-99-9999	AAAAA
116	AaaaawAaaawAaaaa,wAa.w999	99-99-9999	9
117	AaaaawAaaawAaaaa,wAa.w999	99-99-9999	AAAAA
118	AaaaawAaaawAa.w99,wAaaaaaawAaaaaaawAaaawA	99/99/9999	AAAAA
119	AaaaawAaaawAaaaaa,wAa.w99A	99/99/9999	9
120	AaaaawAaaawAaaaawAaaaa,wAaaawAwAa.w9	99wAaaaaaaaw9999	9
121	AaaawAaaaa,wAa.w9w-wAaaawAA	99-99-9999	9

```
122            AaaawAaaaawAaaawAa.w999,wAaaawAaawA99999        99-99-9999     9
123                    AaaawAaaaaaa,wAa.w9w-wAaaawAA           99-99-9999     9
124                    AaaawAaaaaaa,wAa.w99w-wAaaawAA            99/99/99     9
125                    AaaawAaaaaaa,wAa.w999w-wAaaawA          99/99/9999     9
126                    AaaawAaaaawAaaaaaaawAAA,wAa.w9  99wAaaaaaaaw9999     9
127                  AaaawAaaaawAaaaaaaawAAA,wAa.w999          99/99/9999     9
128                  AaaawAaaaawAaaaaaaawAAA,wAa.w999  99wAaaaaaaaw9999  AAAA
129                       AaaawAaaaaa,wAa.w9w-wAaaawA  99wAaaaaaaaw9999     9
130                        AaaawAaaaaawAa.w99,wAaaawA           99/99/99     9
131               AaaaaawAaaaaaaaaawAaaaaaaaa,wAa.w9999     99wAaaaaw9999     9
132                  AaaaaaaaaaawAaaaaawAaaaa,wAa.w9999          99/99/9999     9
133                 AaaaaaawAaaawAaa,wAaaaawAaaaawAa.w9   99wAaaaaaaw9999 AAAAA
134                 AaaaaaawAaaawAaa,wAaaaawAaaawAa.w99          99/99/99     9
135              AaaaaaawAaaaawAaaawAaaaa,wAaaawAwAa.w9          99/99/9999     9
136                      AaaaawAaaawAaaaa,wAaaawAw9/9            99/99/9999     A
137                      AaaaawAaaawAaaaa,wAaaawAw9/9             99/99/99  AAAA
138         AaaaaaaaawAaaawAaaaaaaaaa,wAaaawAA9wAa.w999  99wAaaaaaaaw9999     -
139         AaaaaaaaawAaaawAaaaaaaaaa,wAaaawAA9wAa.w99w          99/99/99  AAAA
140                      AaaawAw9/9,wAaaaawAaaawAaaaa  99wAaaaaaaaw9999     9
141                  AaaaaawAaaaaaawAA,wAa.w9w-wAaaawA   9wAaaaaaaaw9999     9
142                  AaaaaawAaaaaaawAA,wAa.w9w-wAaaawA     99wAaaaw9999 AAAAA
143         AaaaaawAaaaaaawAa.w99wAw-wAaaaaawAaaawAaaaa  99wAaaaaaaaw9999     9
144         AaaaaawAaaaaaawAa.w99wAw-wAaaaaawAaaawAaaaa          99/99/9999     9
145                AaaaaaaaaawAaaaaaaa,wAaaaaaw99wAa.w9          99/99/9999     9
146              Aa.wAaaawAaaaaawAaaa,wAa.w99w-wAaaawA   99wAaaaaaaw9999     9
147              Aa.wAaaawAaaaaawAaaa,wAa.w99w-wAaaawA     99wAaaaw9999     9
148            AaaaaaaaawAaaaaawAaaaa,wAa.w99wAa.w9999        99-99-9999 AAAAA
149            AaaaaaaaawAaaaaawAaaaa,wAa.w99wAa.w9999          99/99/99     9
150                      Aaa.wAaaaawAaaaw99w-wAaaawA       9wAaaaw9999     9
151                 Aaaaa.wAaaaaawAaaawAa.w99w-wAaaawwA          99/99/99     9
152           AaaaaaaaawAaaaawAaaa,wAaaaaawAaaaa,wAa.wA9        99-99-9999     9
153             AaaaawAaaaawAaaaaa,wAa.wAaaaawAaaawAa.w9   99wAaaaaaaw9999     9
154                  AaaaawAaaaawAaaaaa,wAa.wAaaaawAa.w9          99/99/9999     9
155                      AaaaaaaawAaaaaawAaaaaaa,wAa.w9A       99wAaaw9999     9
    Kode.Pos        No.Telepon Nilai.Belanja.Setahun
1     999999   999999999999999               9999999
2     999999  9999999999999999                999999
3     999999 +9999999999999999               9999999
4     999999  9999999999999999               9999999
5     999999   999999999999999                999999
6     999999  9999999999999999               9999999
7     999999   999999999999999                999999
8     999999 +9999999999999999                999999
9     999999  9999999999999999                999999
10    999999 +9999999999999999                999999
11    999999 +9999999999999999               9999999
12    999999   999999999999999               9999999
13    999999 +9999999999999999                999999
14    999999  9999999999999999                999999
15    999999  9999999999999999                999999
16    999999 +9999999999999999               9999999
17    999999 +9999999999999999                999999
18    999999  9999999999999999                999999
19    999999  9999999999999999                999999
20    999999   999999999999999               9999999
```

```
21    999999 +9999999999999999          9999999
22    999999 +9999999999999999           999999
23    999999    999999999999999          9999999
24    999999   9999999999999999           999999
25    999999 +9999999999999999           999999
26    999999    +99999999999999           999999
27    999999 +9999999999999999           999999
28    999999   9999999999999999           999999
29    999999    999999999999999           999999
30         -    999999999999999          9999999
31    999999   9999999999999999           999999
32    999999    999999999999999           999999
33    999999   9999999999999999           999999
34    999999    999999999999999           999999
35    999999   9999999999999999           999999
36    999999   9999999999999999           999999
37    999999 +9999999999999999           999999
38         - +9999999999999999          9999999
39    999999    999999999999999          9999999
40    999999 +9999999999999999           999999
41    999999 +9999999999999999           999999
42    999999   9999999999999999           999999
43    999999    999999999999999          9999999
44    999999    999999999999999           999999
45    999999   9999999999999999           999999
46    999999 +9999999999999999          9999999
47    999999   9999999999999999          9999999
48    999999 +9999999999999999          9999999
49    999999   9999999999999999           999999
50    999999   9999999999999999          9999999
51    999999 +9999999999999999           999999
52    999999 +9999999999999999           999999
53    999999    999999999999999           999999
54    999999   9999999999999999           999999
55    999999    999999999999999           999999
56    999999 +9999999999999999           999999
57    999999 +9999999999999999             <NA>
58    999999 +9999999999999999          9999999
59    999999   9999999999999999          9999999
60    999999 +9999999999999999           999999
61    999999   9999999999999999          9999999
62    999999    999999999999999           999999
63    999999 +9999999999999999          9999999
64    999999   9999999999999999           999999
65    999999    999999999999999           999999
66    999999    999999999999999           999999
67    999999   9999999999999999           999999
68    999999   9999999999999999          9999999
69    999999 +9999999999999999           999999
70    999999     99999999999999          9999999
71    999999    999999999999999           999999
72    999999 +9999999999999999           999999
73    999999 +9999999999999999           999999
74    999999    999999999999999          9999999
75    999999 +9999999999999999           999999
```

```
76          -   999999999999999              999999
77     999999   999999999999999             9999999
78     999999   999999999999999             9999999
79     999999 +9999999999999999              999999
80     999999   +99999999999999              999999
81     999999                 -              999999
82     999999    99999999999999              999999
83     999999   999999999999999              999999
84     999999   999999999999999              999999
85     999999 +9999999999999999              999999
86     999999  9999999999999999              999999
87     999999   999999999999999              999999
88     999999  9999999999999999             9999999
89     999999 +9999999999999999                <NA>
90          -   999999999999999             9999999
91     999999  9999999999999999             9999999
92     999999   999999999999999             9999999
93     999999 +9999999999999999              999999
94     999999 +9999999999999999              999999
95     999999   999999999999999             9999999
96     999999   999999999999999             9999999
97     999999 +9999999999999999             9999999
98     999999 +9999999999999999             9999999
99     999999   999999999999999              999999
100    999999     9999999999999              999999
101    999999    99999999999999              999999
102    999999   999999999999999              999999
103    999999   999999999999999              999999
104    999999 +9999999999999999              999999
105    999999 +9999999999999999             9999999
106    999999 +9999999999999999              999999
107    999999 +9999999999999999              999999
108    999999 +9999999999999999              999999
109    999999   999999999999999             9999999
110    999999 +9999999999999999             9999999
111    999999   999999999999999             9999999
112    999999  9999999999999999             9999999
113    999999   999999999999999             9999999
114    999999  9999999999999999             9999999
115    999999 +9999999999999999              999999
116    999999   999999999999999              999999
117    999999 +9999999999999999              999999
118    999999 +9999999999999999              999999
119    999999   999999999999999             9999999
120    999999 +9999999999999999              999999
121    999999 +9999999999999999                <NA>
122    999999 +9999999999999999             9999999
123    999999  9999999999999999              999999
124    999999 +9999999999999999              999999
125    999999 +9999999999999999             9999999
126    999999  9999999999999999              999999
127    999999 +9999999999999999             9999999
128    99999A   999999999999999              999999
129    999999   999999999999999              999999
130    999999  9999999999999999             9999999
```

```
131   999999   999999999999999                999999
132   999999 +9999999999999999                999999
133   999999   999999999999999               9999999
134   999999  9999999999999999                999999
135   999999 +9999999999999999               9999999
136   999999  9999999999999999                999999
137   999999   999999999999999               9999999
138   999999   999999999999999               9999999
139        -  9999999999999999                999999
140   999999  9999999999999999                999999
141   999999   999999999999999                999999
142   999999   999999999999999               9999999
143   999999   999999999999999                999999
144   999999 +9999999999999999                  <NA>
145   999999  9999999999999999                999999
146   999999   999999999999999                999999
147   9999A9   999999999999999               9999999
148   999999   999999999999999                999999
149   999999   999999999999999                999999
150   999999   999999999999999               9999999
151   9999A9   999999999999999                999999
152   999999 +9999999999999999               9999999
153   999999 +9999999999999999               9999999
154   999999 +9999999999999999                999999
155   999999 +9999999999999999                999999
```

Menggabungkan hasil profiling ke dalam dataset awal

Data pola yang sudah kita dapatkan di praktek terakhir akan menarik jika digabungkan kembali ke sumber data asal, terutama untuk dua alasan berikut:

- Kita tidak perlu scan berulang-ulang untuk mendapatkan nama dengan pola tertentu, ini akan menghemat resource komputasi terutama jika datanya sangat besar. Cukup memfilter kolom pola terkait.
- Hasil penggabungan menjadi dataset baru yang bisa kita olah dengan aplikasi lain misalkan Excel atau SQL – dimana kita saat ini lebih terbiasa.

Tiga langkah proses penggabungan ini adalah sebagai berikut:

- Melakukan profiling terhadap seluruh kolom dari variable **data.pelanggan** dan disimpan ke variable baru, pada contoh berikut namanya adalah **pola.data.pelanggan**.

```
pola.data.pelanggan <-
basic_pattern_analysis(data.pelanggan)
```

- Mengganti nama-nama kolom pada variable pola.data.pelanggan dengan menambahkan prefix "Pola." dengan perintah berikut.

```
names(pola.data.pelanggan)<-
paste("Pola",names(pola.data.pelanggan),sep=".")
```

- Menggabungkan kedua data.frame data.pelanggan dan pola.data.pelanggan dengan function cbind, dan disimpan kembali ke variable **data.pelanggan**.

```
data.pelanggan <- cbind(data.pelanggan,
pola.data.pelanggan)
```

Kita akan lakukan langsung praktek untuk melihat hasil akhir.

Tugas Praktek

Gantilah bagian […1…], […2…] dan […3…] pada code editor dengan perintah dari langkah pertama sampai ketiga dari Lesson di atas.

Jika berjalan dengan lancar maka akan diperoleh hasil berikut.

```
> str(data.pelanggan)
'data.frame':    155 obs. of  16 variables:
 $ Kode.Pelanggan            : chr  "KD-00032" "KD-00053" "KD-00133" "KD-00056" ...
```

```
 $ Nama.Lengkap              : chr  "Eva Novianti, S.H." "Ibu Heidi Goh" "Unang Handoko" "Jokolono Sukarman" ...

 $ Alamat                    : chr  "Vila Sempilan, No. 67 - Kota B" "Vila Sempilan, No. 11 - Kota B" "Vila Sempilan, No. 1 - Kota B" "Vila Permata Intan Berkilau, Blok C 5-7" ...

 $ Tanggal.Lahir             : chr  "1 April 2028" "19-08-1986" "11-07-1981" "10/13/79" ...

 $ Aktif                     : chr  "FALSE" "1" "FALSE" "0" ...

 $ Kode.Pos                  : chr  "567130" "567130" "567130" "876551" ...

 $ No.Telepon                : chr  "085419651438216" "6282189517223455" "+6282952955586979" "6289278629437370" ...

 $ Nilai.Belanja.Setahun     : num  1275600 317800 1537200 1524700 655400 ...

 $ Pola.Kode.Pelanggan       : Factor w/ 2 levels "AA-9999","AA-99999": 2 2 2 2 2 2 2 2 2 2 ...

 $ Pola.Nama.Lengkap         : Factor w/ 85 levels "A.wAaaaaaaa",..: 77 79 49 22 48 51 63 48 6 50 ...

 $ Pola.Alamat               : Factor w/ 125 levels "AA.wAaaaaaaa,wAa.w99AAA",..: 107 107 108 111 112 115 115 117 119 118 ...

 $ Pola.Tanggal.Lahir        : Factor w/ 13 levels "-","99-99-9999",..: 12 2 2 3 2 2 2 2 2 2 ...

 $ Pola.Aktif                : Factor w/ 5 levels "-","9","A","AAAA",..: 5 2 5 2 2 2 2 2 2 2 ...

 $ Pola.Kode.Pos             : Factor w/ 4 levels "-","999999","99999A",..: 2 2 2 2 2 2 2 2 2 2 ...

 $ Pola.No.Telepon           : Factor w/ 7 levels "+99999999999999",..: 6 7 2 7 6 7 6 2 7 2 ...

 $ Pola.Nilai.Belanja.Setahun: Factor w/ 2 levels "999999","9999999": 2 1 2 2 1 2 1 1 1 1 ...
```

Dengan hasil perintah str di atas, kita dapat mengambil kesimpulan penggabungan telah berhasil dilakukan dimana kolom-kolom beprefix "Pola" telah digabungkan dan isinya konsisten dengan keluaran function **basic_pattern_analysis**.

Code Editor

```
#Melakukan profiling terhadap seluruh kolom data.pelanggan
pola.data.pelanggan <- basic_pattern_analysis(data.pelanggan)

#Merubah nama kolom
names(pola.data.pelanggan) <- paste("pola",names(pola.data.pelanggan),sep=".")

#Menggabungkan dua data.frame
data.pelanggan <- cbind(data.pelanggan, pola.data.pelanggan)
#Menampilkan struktur
str(data.pelanggan)
```

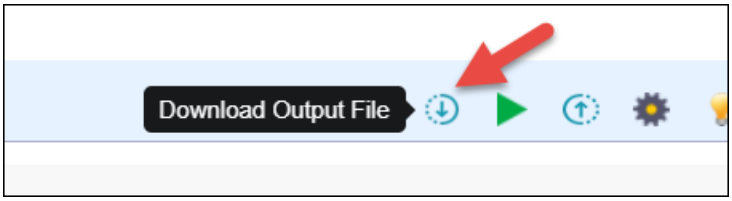
Console

```
> library(openxlsx)

> library(bpa)

> #Membaca dataset pelanggan
> data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pela
nggan.xlsx", sheet="Pelanggan")

> #Melakukan profiling terhadap seluruh kolom data.pelanggan
> pola.data.pelanggan <- basic_pattern_analysis(data.pelanggan)

> #Merubah nama kolom
> names(pola.data.pelanggan) <- paste("pola",names(pola.data.pelanggan),sep=".")

> #Menggabungkan dua data.frame
> data.pelanggan <- cbind(data.pelanggan, pola.data.pelanggan)

> #Menampilkan struktur
> str(data.pelanggan)
'data.frame':    155 obs. of  16 variables:
 $ Kode.Pelanggan            : chr  "KD-00032" "KD-00053" "KD-00133" "KD-00056" ...
 $ Nama.Lengkap              : chr  "Eva Novianti, S.H." "Ibu Heidi Goh" "Unang Hando
ko" "Jokolono Sukarman" ...
 $ Alamat                    : chr  "Vila Sempilan, No. 67 - Kota B" "Vila Sempilan,
No. 11 - Kota B" "Vila Sempilan, No. 1 - Kota B" "Vila Permata Intan Berkilau, Blok C
5-7" ...
```

```
 $ Tanggal.Lahir             : chr  "1 April 2028" "19-08-1986" "11-07-1981" "10/13/7
9" ...
 $ Aktif                     : chr  "FALSE" "1" "FALSE" "0" ...
 $ Kode.Pos                  : chr  "567130" "567130" "567130" "876551" ...
 $ No.Telepon                : chr  "085419651438216" "6282189517223455" "+6282952955
586979" "6289278629437370" ...
 $ Nilai.Belanja.Setahun     : num  1275600 317800 1537200 1524700 655400 ...
 $ pola.Kode.Pelanggan       : Factor w/ 2 levels "AA-9999","AA-99999": 2 2 2 2 2 2 2
2 2 2 ...
 $ pola.Nama.Lengkap         : Factor w/ 85 levels "A.wAaaaaaaa",..: 77 79 49 22 48 5
1 63 48 6 50 ...
 $ pola.Alamat               : Factor w/ 125 levels "AA.wAaaaaaaa,wAa.w99AAA",..: 107
107 108 111 112 115 115 117 119 118 ...
 $ pola.Tanggal.Lahir        : Factor w/ 13 levels "-","99-99-9999",..: 12 2 2 3 2 2
2 2 2 2 ...
 $ pola.Aktif                : Factor w/ 5 levels "-","9","A","AAAA",..: 5 2 5 2 2 2
2 2 2 2 ...
 $ pola.Kode.Pos             : Factor w/ 4 levels "-","999999","99999A",..: 2 2 2 2 2
2 2 2 2 2 ...
 $ pola.No.Telepon           : Factor w/ 7 levels "+99999999999999",..: 6 7 2 7 6 7 6
2 7 2 ...
 $ pola.Nilai.Belanja.Setahun: Factor w/ 2 levels "999999","9999999": 2 1 2 2 1 2 1 1
1 1 ...
```

Menuliskan hasil ke dalam file Excel

Fitur DQLab terbaru memungkinkan member menuliskan hasil operasi di Live Code Editor ke dalam file yang dapat Anda download dengan mengakses icon Download Output File. Icon ini terletak di sebelah kiri tombol Run Code.



Pada praktek kali ini kita akan menggunakan fitur tersebut dengan menuliskan data hasil penggabungan data source dan profile.

Perintah penulisan file yang kita gunakan adalah **write.xlsx** dengan contoh perintah sebagai berikut:

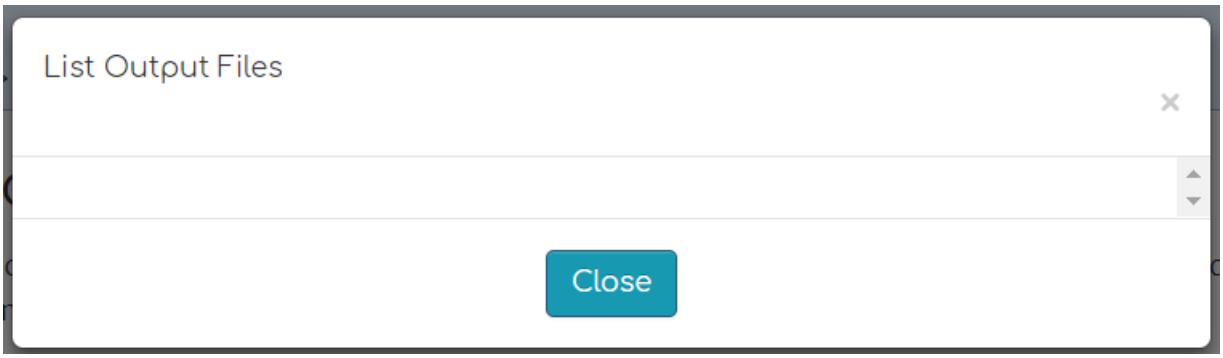
```
write.xlsx(data.pelanggan, file="data.pelanggan.xlsx")
```

yang jika dijalankan akan menulis satu file bernama data.pelanggan.xlsx dengan isi variable **data.pelanggan**.

Kita akan lakukan langsung praktek untuk melakukan hal ini.

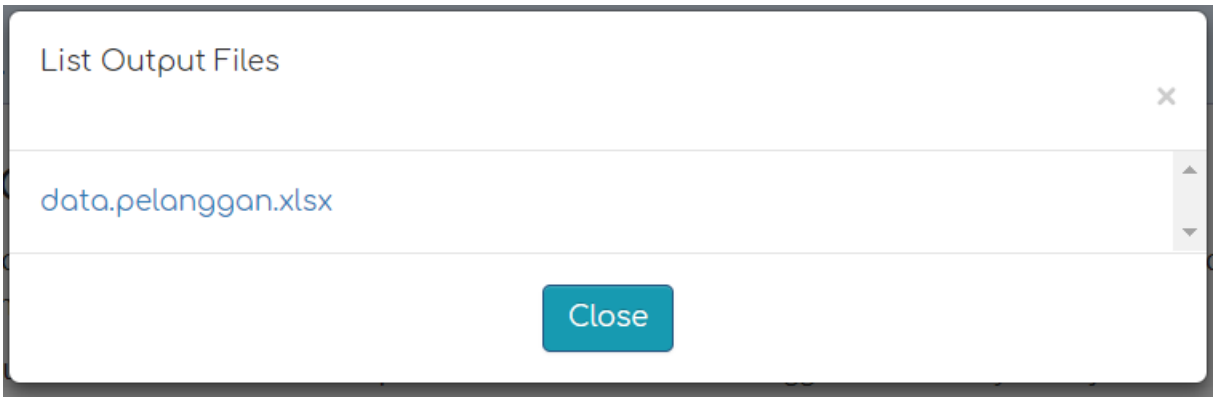
Tugas Praktek

Cobalah klik icon "**Download Output File**" pada code editor, akan muncul pop up window dengan tampilan berikut.



Gantilah bagian […] pada code editor dengan perintah write.xlsx(data.pelanggan, file="data.pelanggan.xlsx").

Klik tombol "Run", dan jika berjalan dengan lancar pada saat kita klik kembali icon "Download Output File" akan muncul file data.pelanggan.xlsx pada daftar output file (List Output Files) seperti pada gambar berikut.



Download file tersebut dengan mengklik nama filenya, dan jika dibuka di Excel maka tampilan sebagian datanya akan terlihat sebagai berikut.

E	F	G	H	I	J	
	Kode.Pos	No.Telepon	Nilai.Belanja.Setahun	Pola.Kode.Pelanggan	Pola.Nama.Lengkap	Pola.Alamat
E	567130	085419651438216	1275600	AA-99999	AaawAaaaaaaa,wA.A.	AaaawAaaaaaaa,v
	567130	6282189517223455	317800	AA-99999	AaawAaaaawAaa	AaaawAaaaaaaa,v
E	567130	+6282952955586979	1537200	AA-99999	AaaaawAaaaaaa	AaaawAaaaaaaa,v
	876551	6289278629437370	1524700	AA-99999	AaaaaaaawAaaaaaaa	AaaawAaaaaaawA
	876551	084384621977881	655400	AA-99999	AaaaawAaaaaa	AaaawAaaaaaawA
	876552	6285842418573681	1444400	AA-99999	AaaaawAaaaaaaa	AaaawAaaaaawAa
	876552	089522699290044	350400	AA-99999	AaaawAaaaaaa	AaaawAaaaaawAa
	877521	+6288389541238485	354600	AA-99999	AaaaawAaaaaa	AaaawAaaaawAaa
	877521	6288267903981205	541300	AA-99999	Aa.wAaawwwwAaaaaaa	AaaawAaaaawAaa
	877521	+6284871003581659	536000	AA-99999	AaaaawAaaaaaa,wAaa.	AaaawAaaaawAaa
	712983	+6287132221371404	1336200	AA-99999	AaaaaaawAaaaa	AaaaawAaaawAaa
	712983	083309536733507	1316500	AA-99999	AaaawAaawAaaa	AaaaawAaaawAaa
	712984	+6286815308308264	725600	AA-99999	AaaaaawAaaawAaaaaaa	AaaaawAaaaawAa
	712984	6286725681847845	398200	AA-99999	AaaaaaaawAaaa	AaaaawAaaaawAa
	712984	6281413705348345	311000	AA-99999	Aaaw%$wAaaaaaaaa	AaaaawAaaaawAa

Code Editor

```
library(openxlsx)

library(bpa)


#Membaca dataset pelanggan

data.pelanggan <-
read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pelanggan.xlsx",
sheet="Pelanggan")
```

#Melakukan profiling terhadap seluruh kolom data.pelanggan

pola.data.pelanggan <- basic_pattern_analysis(data.pelanggan)

#Merubah nama kolom

names(pola.data.pelanggan)<-paste("Pola",names(pola.data.pelanggan),sep=".")

#Menggabungkan dua data.frame

data.pelanggan <- cbind(data.pelanggan, pola.data.pelanggan)

#Menulis File Excel

write.xlsx(data.pelanggan,file="data.pelanggan.xlsx")

Console

```
> library(openxlsx)

> library(bpa)

> #Membaca dataset pelanggan
> data.pelanggan <- read.xlsx("https://academy.dqlab.id/dataset/dqlab_messy_data_pela
nggan.xlsx", sheet="Pelanggan")

> #Melakukan profiling terhadap seluruh kolom data.pelanggan
> pola.data.pelanggan <- basic_pattern_analysis(data.pelanggan)

> #Merubah nama kolom
> names(pola.data.pelanggan)<-paste("Pola",names(pola.data.pelanggan),sep=".")

> #Menggabungkan dua data.frame
> data.pelanggan <- cbind(data.pelanggan, pola.data.pelanggan)

> #Menulis File Excel
> write.xlsx(data.pelanggan,file="data.pelanggan.xlsx")
```

Kesimpulan

Data profiling adalah tahap awal untuk melakukan data cleansing. Di dalam proses ini melakukan aktifitas yang sederhana tapi penting:

- Identifikasi berbagai pola yang terdapat pada satu kolom data.
- Melakukan perbandingan dengan ekspektasi atau ukuran scientific yang wajar, untuk menemukan data yang perlu diperbaiki.

Kedua proses ini telah dipraktekkan dengan sangat detil menggunakan fungsi dan operator berikut:

- Function **summary** dari paket bawaan R.
- Function **basic_pattern_analysis** dari library bpa di R.
- Menggunakan operator == dan function **grepl** untuk menarik data untuk pola hasil temuan.

Dengan penguasaan keterampilan profiling ini, Anda bisa mengenal outlier dan mengambil datanya – telah ditunjukkan untuk kolom Kode.Pelanggan dan Nama. Tanpa kemampuan identifikasi dan pengambilan data ini, tentu proses data cleansing atau perbaikan data tidak dapat dilakukan.

Terakhir, kita lakukan penggabungan pola dan data asal menjadi satu dataset dan dituliskan ke file agar bisa dilihat atau dikelola menggunakan aplikasi lain seperti Excel. Pada bab berikutnya, kita akan mulai menggunakan dataset gabungan ini tetapi dibaca dari sistem database MySQL, bukan file lagi.

Klik tombol Next untuk melanjutkan.

Pendahuluan

Kemampuan data wrangling tidak terlepas dari kemampuan untuk membaca berbagai sumber data, salah satu yang paling populer adalah membaca sistem database relasional.

Walaupun proses pembelajaran pengolahan data cleansing tidak perlu melibatkan sistem database relasional. Namun karena sedemikian populernya, maka DQLab memutuskan untuk tetap memberikan materi SQL setelah di course sebelumnya kita selalu membaca data dari file teks maupun Excel.

Dengan demikian dari bab ini sampai penutup, kita akan tetap membaca dari sistem database. Namun tetap diingat, sumber data tetap bisa dibaca dari teks file maupun Excel.

Walaupun cukup banyak konsep yang perlu dikenalkan, DQLab akan usahakan untuk memberikan penjelasang yang gamblang dengan menghilangkan banyak detil yang tidak diperlukan.

Klik tombol Next untuk melanjutkan.

Apa itu Sistem Database Relasional?

Ada dua kategori sistem database yang sangat populer saat ini, yaitu:

- Sistem database relasional atau SQL based database: adalah sistem database yang mengusung konsep objek database yang saling berelasi dengan skema dari objek-objek tersebut telah didefinisikan dengan jelas.
  Contoh produk: Microsoft Access, MySQL, Oracle, SQL Server, PostgreSQL, dan lain-lain.
- NoSQL: adalah sistem database yang mengusung konsep objek database dengan skema yang fleksibel dan tidak kaku seperti relasional.
  Contoh produk: seperti MongoDB, Apache Cassandra, Apache HBase, dan lain-lain.

Sistem database yang pertama atau relasional adalah yang paling banyak digunakan di hampir seluruh perusahaan di Indonesia yang menggunakan sistem informasi komputer.

Sebuah sistem database dirancang untuk melakukan tiga fungsi berikut:

- Menyimpan Data
- Mengorganisasikan Data
- dan Mengambil Data

Dan untuk relasional database, kemampuan untuk melakukan tiga hal tersebut bisa menggunakan bahasa khusus yang dinamakan SQL (Structured Query Language). Dengan SQL kita memiliki konstruksi bahasa yang lebih mudah untuk berinteraksi dengan objek-objek data seperti database, table, kolom, dan lain-lain.

Sepanjang course ini kita akan fokus menggunakan SQL untuk fungsi terakhir, yaitu mengambil data. Produk yang akan kita gunakan adalah MySQL – yang bisa dikatakan sebagai produk database open source paling populer.

Klik tombol Next untuk melanjutkan.

Mana pernyataan yang benar mengenai sistem database relasional?

Mana pernyataan yang benar mengenai sistem database relasional?

- [x] Sistem yang digunakan untuk menyimpan data.
- [x] Sistem yang digunakan untuk mengorganisasikan data.
- [ ] Sistem yang digunakan untuk menganalisa data.
- [ ] Semua salah.
- [x] Sistem yang digunakan untuk mengambil data.

Mana pernyataan yang benar mengenai bahasa SQL?

Mana pernyataan yang benar mengenai bahasa SQL?

- [ ] Bahasa SQL digunakan untuk programming data science.
- [x] Bahasa SQL digunakan untuk berinteraksi dengan objek-objek database.
- [ ] Bahasa SQL hanya digunakan oleh aplikasi lama seperti Foxpro.
- [ ] Semua benar.
- [ ] Bahasa SQL tidak digunakan lagi karena telah digantikan oleh R ataupun Python.

Server, Database, Table, Row dan Column

SQL tentunya membutuhkan interaksi dengan objek-objek sistem database dimana isi atau datanya sendiri disimpan.

Disini, DQLab akan mengambil konsep spreadsheet Excel sebagai analogi untuk menjelaskan objek-objek database sebagai berikut:

- Database: adalah satu file spreadsheet Excel yang memiliki banyak sheet.
- Table: adalah sheet pada Excel. Dengan demikian database terdiri dari beberapa table.
- Row: Tiap sheet memiliki table data yang memiliki row data.

Sedangkan MySQL server adalah analoginya adalah lokasi folder di komputer dimana kita bisa menyimpan banyak file.

Jika dikaitkan ke R, table, kolom, dan baris dapat disamakan dengan data.frame, kolom data.frame, dan isi data.frame.

Klik tombol Next untuk melanjutkan.

Mana pernyataan yang benar mengenai objek-objek database?

Mana pernyataan yang benar mengenai objek-objek database?

- [ ] Table terdiri dari beberapa column.
- [ ] Table terdiri dari beberapa rows.
- [ ] Database terdiri dari beberapa table.
- [ ] Database dapat disamakan sebagai satu file spreadsheet Excel.
- [x] Semua benar.

Package RMySQL

Untuk dapat berinteraksi dengan sistem database MySQL di dalam R, kita bisa gunakan package RMySQL – yang sudah terinstalasi di server.

Sisa praktek dari bab ini akan diperkenalkan.

- Function-function yang akan digunakan untuk melakukan koneksi dan mengambil data dari database yang disimpan di MySQL server.
- Perintah SELECT yang merupakan bagian dari SQL untuk mengambil kolom tertentu dan dengan filter isi dengan pola yang dimengerti oleh MySQL.

Klik tombol Next untuk melanjutkan.

Koneksi Database

Seperti halnya produk server umumnya, MySQL juga memerlukan koneksi dari aplikasi lain sebelum dapat memberikan data kepada aplikasi.

Dengan RMySQL, kita gunakan function dbConnect untuk melakukan koneksi ke MySQL. Berikut adalah contoh penggunaannya.

```
dbConnect(MySQL(), host="localhost", user="root",
password="",
                  dbname="dqlabdatawrangling")
```

Keterangan parameter yang digunakan.

- **MySQL()**: adalah function yang merupakan keharusan untuk load driver MySQL ke R.
- **host**: ini merupakan lokasi server, bisa dalam bentuk alamat IP atau nama host. Untuk praktek course ini, kita gunakan nama host "mysqlhost".
- **user**: ini merupakan nama user yang diperbolehkan untuk melakukan koneksi ke server. Untuk praktek course ini, kita gunakan user c.
- **password**: password yang digunakan oleh user. Untuk praktek course ini, kita gunakan password "demo".
- **dbname**: nama database yang digunakan. Untuk praktek course ini, kita gunakan database dengan nama "dqlabdatawrangling".

Function tersebut akan mengembalikan objek connection yang perlu kita simpan dalam suatu variable. Berikut adalah contoh dengan variable bernama con.

```
con <- dbConnect(MySQL(), host="localhost", user="root",
password="",
                  dbname="dqlabdatawrangling")
```

Pada akhir script kita perlu memutuskan koneksi dengan function dbDisconnect dengan input variable connection yang telah kita buat. Selengkapnya adalah sebagai berikut.

```
dbDisconnect(con)
```

Namun, kadangkala koneksi yang telah dibuat pada setiap sesi R tidak terputus secara otomatis dengan berbagai alasan, sehingga kita tambahkan perintah berikut untuk memutuskan semua koneksi yang masih dikenali oleh sesi R.

```
all_cons <- dbListConnections(MySQL())

for(con in all_cons) dbDisconnect(con)
```

Tugas Praktek

Di dalam code editor telah dilengkapi perintah untuk membuat dan menutup koneksi database ke MySQL seperti yang telah dijelaskan dengan contoh pada Lesson.

Cobalah dijalankan dengan tombol Run. Anda akan mendapatkan hasil sebagai berikut.

```
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling2")
Failed to connect to database: Error: Access denied for user 'demo'@'%' to database '
dqlabdatawrangling2'
```

Akan terjadi error. Ini karena nama database "dqlabdatawranling2" tidak terdapat pada server MySQL.

Gantilah "dqlabdatawrangling2" dengan "dqlabdatawrangling" dan jalankan kembali. Jika tidak ada lagi error yang muncul, maka koneksi telah berhasil dilakukan.

Klik tombol Submit untuk mengumpulkan soal.

Code Editor

```
library(RMySQL)

#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",

                 dbname="dqlabdatawrangling")

#Menutup Koneksi

all_cons <- dbListConnections(MySQL())

for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Mengambil data dengan SELECT

Untuk mengambil selurh data dari suatu table koneksi database MySQL yang telah kita buat, maka diperlukan beberapa proses berikut.

Pertama tentunya adalah membuat perintah sql untuk mengambil data, yaitu SELECT dengan contoh konstruksi lengkap berikut.

```
SELECT * FROM nama_table
```

Dimana:

- tanda bintang (*) disini memiliki arti mengambil semua kolom dari table.
- nama_table yang akan kita ambil datanya, seperti disebutkan di awal course, nama table kita adalah dqlab_messy_data.

Kita bisa membuat variable untuk menampung perintah SQL ini seperti contoh berikut:

```
sql <- "SELECT * FROM dqlab_messy_data"
```

Proses kedua adalah mengirimkan perintah SQL tadi ke server dengan function dbSendQuery seperti terlihat pada contoh berikut.

```
rs <- dbSendQuery(con, sql)
```

Perintah ini mengirimkan perintah SQL – yang kemudian kita akan sebut query, dan telah disimpan pada variable bernama **sql** – ke server MySQL – yang sudah disimpan koneksinya dengan variable bernama **con**. Hasil pengiriman ini ditampung pada variable **rs**.

Perintah tersebut biasanya dimodifikasi dengan function tryCatch. Yang berfungsi untuk menagani error apabila ada perintah yang salah atau koneksi terputus di tengah jalan, sehingga perintah akhirnya menjadi seperti bentuk berikut.

```
rs <- tryCatch(dbSendQuery(con, sql), finally =
print("query ok"))
```

rs hanyalah sebuah pointer atau masih berupa connector ke data, belum mengambil data sebenarnya. Untuk mengambil data, kita perlu tambahan satu function lagi yaitu fetch.

Function fetch memerlukan dua parametre yaitu res yang meminta hasil eksekusi dari dbSendQuery, dan jumlah data yang diinginkan pada parameter n – jika nilainya -1 maka kita mengambil semua data.

Pada contoh kita, bentuknya adalah sebagai berikut.

```
fetch(res=rs, n=-1)
```

Hasil dari fetch bisa disimpan dalam bentuk variable.

```
data.pelanggan <- fetch(res=rs, n=-1)
```

Tugas Praktek

Berdasarkan keterangan dan contoh pada Lesson, ganti bagian […] pada code editor dengan perintah SQL untuk mengambil seluruh kolom data dari table dqlab_messy_data.

Jika semua berjalan lancar, maka pada hasil akan terdapat tampilan struktur data.frame dari hasil pembacaan table MySQL seperti berikut.

```
> str(data.pelanggan)
'data.frame':    155 obs. of  16 variables:
 $ kode_pelanggan            : chr  "KD-00032" "KD-00053" "KD-00133" "KD-00056" ...
 $ nama                      : chr  "Eva Novianti, S.H." "Ibu Heidi Goh" "Unang Handoko" "Jokolono Sukarman" ...
 $ alamat                    : chr  "Vila Sempilan, No. 67 - Kota B" "Vila Sempilan, No. 11 - Kota B" "Vila Sempilan, No. 1 - Kota B" "Vila Permata Intan Berkilau, Blok C 5-7" ...
 $ tanggal_lahir             : chr  "1 April 2028" "19-08-1986" "11-07-1981" "10/13/79" ...
 $ aktif                     : chr  "FALSE" "1" "FALSE" "0" ...
 $ kode_pos                  : chr  "567130" "567130" "567130" "876551" ...
 $ no_telepon                : chr  "085419651438216" "6282189517223455" "+6282952955586979" "6289278629437370" ...
 $ nilai_belanja_setahun     : chr  " 1275600.0" " 317800.0" " 1537200.0" " 1524700.0" ...
 $ pola_kode_pelanggan       : chr  "AA-99999" "AA-99999" "AA-99999" "AA-99999" ...
 $ pola_nama                 : chr  "AaawAaaaaaaa,wA.A." "AaawAaaaawAaa" "AaaaawAaaaaaa" "AaaaaaawAaaaaaaa" ...
 $ pola_alamat               : chr  "AaaawAaaaaaaa,wAa.w99w-wAaaawA" "AaaawAaaaaaaa,wAa.w99w-wAaaawA" "AaaawAaaaaaaa,wAa.w9w-wAaaawA" "AaaawAaaaaaawAaaaawAaaaaaaa,wAaaawA9-9" ...
 $ pola_tanggal_lahir        : chr  "9wAaaaaw9999" "99-99-9999" "99-99-9999" "99/99/99" ...
 $ pola_aktif                : chr  "AAAAA" "9" "AAAAA" "9" ...
 $ pola_kode_pos             : chr  "999999" "999999" "999999" "999999" ...
 $ pola_no_telepon           : chr  "999999999999999" "9999999999999999" "+9999999999999999" "9999999999999999" ...
 $ pola_nilai_belanja_setahun: chr  "9999999" "999999" "9999999" "9999999" ...
```

Terlihat ada 16 kolom pada table yang isinya sama dengan sumber dataset pelanggan yang telah kita analisa pada bab sebelumnya, dengan tambahan pola teksnya.

Code Editor

```
library(RMySQL)
#Membuka koneksi
con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
                 dbname="dqlabdatawrangling")

#Konstruksi SQL
sql <- "SELECT * FROM dqlab_messy_data"

#Mengirimkan query
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data
data.pelanggan <- fetch(rs, n=-1)
str(data.pelanggan)

#Menutup Koneksi
all_cons <- dbListConnections(MySQL())
for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> #Konstruksi SQL
> sql <- "SELECT * FROM dqlab_messy_data"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> data.pelanggan <- fetch(rs, n=-1)

> str(data.pelanggan)
'data.frame':    155 obs. of  16 variables:
 $ kode_pelanggan            : chr  "KD-00032" "KD-00053" "KD-00133" "KD-00056" ...
 $ nama                      : chr  "Eva Novianti, S.H." "Ibu Heidi Goh" "Unang Handoko" "Jokolono Sukarman" ...
 $ alamat                    : chr  "Vila Sempilan, No. 67 - Kota B" "Vila Sempilan, No. 11 - Kota B" "Vila Sempilan, No. 1 - Kota B" "Vila Permata Intan Berkilau, Blok C 5-7" ...
 $ tanggal_lahir             : chr  "1 April 2028" "19-08-1986" "11-07-1981" "10/13/79" ...
 $ aktif                     : chr  "FALSE" "1" "FALSE" "0" ...
 $ kode_pos                  : chr  "567130" "567130" "567130" "876551" ...
 $ no_telepon                : chr  "085419651438216" "6282189517223455" "+6282952955586979" "6289278629437370" ...
 $ nilai_belanja_setahun     : chr  " 1275600.0" " 317800.0" " 1537200.0" " 1524700.0" ...
 $ pola_kode_pelanggan       : chr  "AA-99999" "AA-99999" "AA-99999" "AA-99999" ...
 $ pola_nama                 : chr  "AaawAaaaaaaa,wA.A." "AaawAaaaawAaa" "AaaaawAaaaaaa" "AaaaaaawAaaaaaaa" ...
 $ pola_alamat               : chr  "AaaawAaaaaaaa,wAa.w99w-wAaaawA" "AaaawAaaaaaaa,wAa.w99w-wAaaawA" "AaaawAaaaaaaa,wAa.w9w-wAaaawA" "AaaawAaaaaaawAaaaawAaaaaaaa,wAaaawA9-9" ...
 $ pola_tanggal_lahir        : chr  "9wAaaaaw9999" "99-99-9999" "99-99-9999" "99/99/99" ...
 $ pola_aktif                : chr  "AAAAA" "9" "AAAAA" "9" ...
 $ pola_kode_pos             : chr  "999999" "999999" "999999" "999999" ...
 $ pola_no_telepon           : chr  "999999999999999" "9999999999999999" "+9999999999999999" "9999999999999999" ...
 $ pola_nilai_belanja_setahun: chr  "9999999" "999999" "9999999" "9999999" ...

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Mengambil kolom Kode Pelanggan dan Nama

Dengan perintah SELECT kita bisa mengambil beberapa kolom tertentu saja. Kolom-kolom yang ingin kita ambil kita daftarkan dengan pemisah koma.

Berkaitan dengan contoh kita, jika kita ingin mengambil kolom nama dan tanggal_lahir maka perintahnya adalah sebagai berikut.

```
SELECT nama, tanggal_lahir FROM dqlab_messy_data
```

Sangat mudah bukan?

Tugas Praktek

Berdasarkan contoh pada Lesson, ganti bagian […] pada code editor dengan perintah SQL untuk mengambil kolom kode_pelanggan dan nama.

Jika semua berjalan lancar, maka pada hasil akan terdapat tampilan data.frame yang terlihat sebagian sebagai berikut.

```
> data.pelanggan
                                    nama     tanggal_lahir
1                       Eva Novianti, S.H.      1 April 2028
2                            Ibu Heidi Goh        19-08-1986
3                            Unang Handoko        11-07-1981
4                         Jokolono Sukarman          10/13/79
5                              Tommy Sinaga        24-03-1976
6                           Irwan Setianto        20-02-1970
7                              Agus Cahyono        14-11-1987
8                              Maria Sirait        12-01-1968
9                       Ir. Ita    Nugraha        14-03-1879
…
…
…
150                          Frenki Pranata       7 Juli 1968
151                             Frenki P.           07/07/68
152                          Tedi Rahmanto        14-12-2003
153                 Bapak Sanjaya Priyantoro   26 Agustus 1983
154                      Safira Hana Sahrani        02/20/1970
155                      dr. Yati Octavianus       21 Mei 1980
```

Code Editor

```
library(RMySQL)

#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo",
host="mysqlhost",dbname="dqlabdatawrangling")


#Konstruksi SQL

sql <- "SELECT kode_pelanggan, nama FROM dqlab_messy_data"

#Mengirimkan query

rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data

data.pelanggan <- fetch(rs, n=-1)

data.pelanggan

#Menutup Koneksi

all_cons <- dbListConnections(MySQL())

for(con in all_cons) dbDisconnect(con
```

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dq
labdatawrangling")

> #Konstruksi SQL
> sql <- "SELECT kode_pelanggan, nama FROM dqlab_messy_data"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> data.pelanggan <- fetch(rs, n=-1)

> data.pelanggan
    kode_pelanggan                                    nama
```

1	KD-00032	Eva Novianti, S.H.
2	KD-00053	Ibu Heidi Goh
3	KD-00133	Unang Handoko
4	KD-00056	Jokolono Sukarman
5	KD-00111	Tommy Sinaga
6	KD-00036	Irwan Setianto
7	KD-00126	Agus Cahyono
8	KD-00137	Maria Sirait
9	KD-00046	Ir. Ita Nugraha
10	KD-00027	Djoko Wardoyo, Drs.
11	KD-00002	Khairul Nissa
12	KD-00075	Kaka Ari Lima
13	KD-00076	Safira Hana Sahrani
14	KD-00035	Sidharta Paul
15	KD-00113	Edi %$ Alexander
16	KD-00099	Bapak Sanjaya Priyantoro
17	KD-00132	Rachmat Chandra
18	KD-00088	Ayu
19	KD-00119	Tri Sulistianti
20	KD-00096	Rahmat Chandra
21	KD-00139	Agnes Rita
22	KD-00090	Andreas Sutanto
23	KD-00074	Taka Teguh
24	KD-00021	Paulus Angkasa Putra
25	KD-00045	Usman Pandajaya
26	KD-00012	Cahyono, Agus
27	KD-00030	Hendi
28	KD-00129	Edward Salim
29	KD-00122	Christine Angkasa
30	KD-00059	Prof Dr. Sadli Masikun
31	KD-00079	Meiti Kuswara
32	KD-00134	Budi Yahya
33	KD-00064	Fauzan Amir
34	KD-00038	Putri Utomo
35	KD-00117	Florensia Novianti
36	KD-00010	Ibu Sri Wahyuni@, IR
37	KD-00028	Aman Pakpahan
38	KD-00125	Tedi Halim
39	KD-00069	Syarifuddin Mahmud
40	KD-00114	Tri Iskandar
41	KD-00062	Zulkifli Kirana
42	KD-00006	DR. Candra Wijaya
43	KD-00024	Solihin Chaerul
44	KD-00084	Surya
45	KD-00104	Iqbal Setiawan
46	KD-00103	Yonathan Bagus
47	KD-00143	Hari Wibowo
48	KD-00034	Rita Meutia Latief
49	KD-00087	Budi Setiawan
50	KD-00039	Joko Wiryanto Abadi (Pelanggan OKE)
51	KD-0047	Puspita Citra
52	KD-00149	Chandra Rachmat
53	KD-00003	Slamet Wiyanto
54	KD-00043	Suharno Jamar
55	KD-00135	Tiah Feris

```
56        KD-00050                       Intan Tri Wahyuni
57        KD-00110                         Sumartono Salim
58        KD-00049                         Dianto Laksmana
59        KD-00141                            Edi Sumantri
60        KD-00044                     dr. Yati Octavianus
61        KD-00124                               Yakob Tan
62        KD-00105                    Urip Chandra Effendi
63        KD-00107                         Rachmat Chandra
64        KD-00086                             Sisilia Lai
65        KD-00123                         Rakhmat Chandra
66        KD-00025                    DRG. Euis Rosidawati
67        KD-00008                           Willy Sanjaya
68        KD-00005 Prihatin Setyonugroho (021-5555555544)
69        KD-00101                          Fera Kurniawan
70        KD-00001                          Agus Cahyono's
71        KD-00020                          Hendri Winarto
72        KD-00080              Cristian Pakpahan Winarno
73        KD-00102                            Leny Sarmini
74        KD-00146                            Roger Sirait
75        KD-00048                               Lilis Ong
76        KD-00019                          Maria Yuniarti
77        KD-00151                              Ferry Thia
78        KD-00130                     Bapak Jujur Suwito
79        KD-00073                  Takashi Yudistira Arief
80        KD-00778                          Cahyono Agus H.
81        KD-00066                             Purnomo Hadi
82        KD-00041                           Poernomo Hadi
83        KD-00140                          Leonardo Tedja
84        KD-00116                         Risma Sihombing
85        KD-00127                           Herdi Rivanto
86        KD-00057                           Sumardi Utomo
87        KD-00016                                Indra K.
88        KD-00063   Widianto Nuryajaya - 08222222999111)
89        KD-00148                                Kuswanto
90        KD-00023                      IR. Yahya Permata
91        KD-00029                              Sri Rahayu
92        KD-00136                            Joko Wibawa
93        KD-00106                             Budi Yahya
94        KD-00026                           Anton Winarta
95        KD-00145                             Lilis Kasim
96        KD-00018                       Sudirman Kartono
97        KD-00058                     Fineli Rahmadianto
98        KD-00051                             Abdul Kadir
99        KD-00144                         Risma Sihombing
100       KD-00128                           Tedi Rahmanto
101       KD-00115                          Teddy Rahmanto
102       KD-00009                        Antonius Winarta
103       KD-00092                                 M Hasbi
104       KD-00070                         I Made Mulyana
105       KD-00118                            Abdul Kadir
106       KD-00052                                Iriawan
107       KD-00120                          Dewi Sr|yani
108       KD-00055                         Maria Wiryawan
109       KD-00089                           Acmad Junaidi
110       KD-00042                           Ahmad Junaidi
```

```
111       KD-00112                              Ari Masbun
112       KD-00098                            B. Sulaiman
113       KD-00033                           Citra Permana
114       KD-00013                          Danang Santosa
115       KD-00138                            Teddja Yanto
116       KD-00094                               Sri Utami
117       KD-00054                          Yudistira Utomo
118       KD-00100                          Rahayu Sri Asih
119       KD-00121                            Diana Sumirah
120       KD-00061                  Tjipto Kesuma Wardhaya
121       KD-00031                  Risman Suparyo Permata
122       KD-00040                               Sri Utami
123       KD-00068                                 Miliana
124       KD-00131                             Dewi Pratiwi
125       KD-00097                           Frenkie Pranata
126       KD-00004                DRS. Maria Simangunsong
127       KD-00071                                 Suparta
128       KD-00093                                 Partono
129       KD-00082                                 Darmadi
130       KD-00150                             Maria Utami
131       KD-00065                   Civara Intan Wahyudi
132       KD-00067                        Niken Sri Utami
133       KD-00011                          Rosalina Kurnia
134       KD-00091                  Indri Nourina Marthia
135       KD-00147                           Budi Setiawan
136       KD-00081                            Andy Gunawan
137       KD-00109                         Purwadianto Hadi
138       KD-00072                          Harry Widijanto
139       KD-00014        Elisabeth Suryadinata, SKOM, ST
140       KD-00078                   Gugun Gunawan Wijaya
141       KD-00095                       Sri Resti Agung
142       KD-00022                     Mbak Dian Sukowati
143       KD-00017                     Irfan Putra Wijaya
144       KD-00037                           Cynthia Agus
145       KD-00108                     Ibu Jujur  Suwito
146       KD-00015                        Mario Setiawan
147       KD-00083                        Setiawan Mario
148       KD-00060                      Sulaiman Baskara
149       KD-00007                    Indra Kurniawan, ST
150       KD-00077                        Frenki Pranata
151       KD-00085                            Frenki P.
152       KD-00142                         Tedi Rahmanto
153       KD-00192                Bapak Sanjaya Priyantoro
154       KD-00298                     Safira Hana Sahrani
155       KD-00492                     dr. Yati Octavianus

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Melakukan filtering data dengan where

Sejauh ini kita sudah bisa mengambil hanya beberapa kolom dari table database, namun masih seluruh mengambil seluruh baris data. Yang tentunya untuk banyak kasus tidak diperlukan.

Proses filtering data ini dapat dilakukan dengan perintah SELECT dengan menambahkan WHERE setelah FROM [nama_table].

Mengikuti where adalah operasi logika yang melibatkan nama kolom dan nilai pengecekan dari table.

Lebih jelasnya, kita perhatikan contoh berikut.

```
     SELECT kode_pelanggan, nama FROM dqlab_messy_data WHERE
nama = 'agus cahyono'
```

Terlihat WHERE ditempatkan setelah nama table dqlab_messy_data, diikuti oleh kolom nama, operator perbandingan =, dan teks 'agus cahyono'.

Catatan: Tanda kutip untuk filter teks harus merupakan kutip tunggal.

Dengan konstruksi demikian, kita akan memperoleh data berikut.

```
kode_pelanggan         nama
1       KD-00126 Agus Cahyono
```

Perhatikan pada konfigurasi MySQL di DQLab, huruf besar dan kecil tidak menjadi masalah untuk filter dengan operator sama dengan (=).

Tugas Praktek

Ganti bagian […] pada code editor dengan perintah SQL untuk mengambil kolom kode_pelanggan dan nama, dan dengan filter nama = 'Safira Hana Sahrani'.

Catatan: Gunakan huruf besar dan kecil sesuai permintaan soal di atas, karena sistem DQLab masih sensitif mengenali code yang menggunakan huruf besar dan kecil, dan hanya menerima satu opsi.

Jika semua berjalan lancar, maka pada hasil akan terdapat tampilan data.frame yang terlihat sebagian sebagai berikut.

```
kode_pelanggan                nama
1       KD-00076 Safira Hana Sahrani
2       KD-00298 Safira Hana Sahrani
```

Code Editor

library(RMySQL)

#Membuka koneksi

```
con <- dbConnect(MySQL(), user="demo", password="demo",
host="mysqlhost",dbname="dqlabdatawrangling")


#Konstruksi SQL

sql <- "SELECT kode_pelanggan, nama FROM dqlab_messy_data WHERE nama =
'Safira Hana Sahrani'"


#Mengirimkan query

rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))


#Mengambil data

data.pelanggan <- fetch(rs, n=-1)

data.pelanggan


#Menutup Koneksi

all_cons <- dbListConnections(MySQL())

for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dq
labdatawrangling")

> #Konstruksi SQL
> sql <- "SELECT kode_pelanggan, nama FROM dqlab_messy_data WHERE nama = 'Safira Hana
Sahrani'"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> data.pelanggan <- fetch(rs, n=-1)

> data.pelanggan
```

```
  kode_pelanggan                nama
1       KD-00076 Safira Hana Sahrani
2       KD-00298 Safira Hana Sahrani

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Melakukan filtering data dengan where … like

Like adalah operator pada bahasa SQL untuk mencari isi data yang memiliki karakter-karakter tertentu. Like sangat berguna untuk mencari isi yang mengandung prefix, infix ataupun suffix.

Untuk melakukan hal tersebut, like berpasangan dengan satu simbol yaitu simbol % yang mewakili karakter apapun – atau disebut juga wild character.

Misalkan kita memiliki data nama berikut:

Nama
Maria Sirait
Ir. Ita Nugraha
Djoko Wardoyo, Drs.
Khairul Nissa
Kaka Ari Lima

Jika kita ingin mencari data yang memiliki tanda koma, maka kita bisa menulis konstruksi like sebagai berikut.

```
Nama like '%,%'
```

Ini artinya Nama tanda koma diapit oleh karakter apapun – tanda persen di sebelah kiri dan kanan tanda koma.

Kemudian jika kita ingin mencari Nama yang berawalan huruf K, maka kita bisa gunakan konstruksi like berikut.

```
Nama like 'K%'
```

Ini artinya Nama diawali huruf K dan diikuti dengan karakter apapun.

Lebih jelasnya, kita perhatikan contoh berikut.

```
SELECT kode_pelanggan, nama FROM dqlab_messy_data WHERE
nama like 'a%'
```

Konstruksi filter ini akan mencari semua data nama yang berawalan a. Huruf besar dan kecil tidak menjadi masalah.

Dengan konstruksi demikian, kita akan memperoleh data berikut.

```
   kode_pelanggan              nama
1        KD-00126     Agus Cahyono
2        KD-00088              Ayu
3        KD-00139       Agnes Rita
4        KD-00090  Andreas Sutanto
5        KD-00028    Aman Pakpahan
6        KD-00001   Agus Cahyono's
7        KD-00026    Anton Winarta
8        KD-00051      Abdul Kadir
9        KD-00009 Antonius Winarta
10       KD-00118      Abdul Kadir
11       KD-00089    Acmad Junaidi
12       KD-00042    Ahmad Junaidi
13       KD-00112       Ari Masbun
14       KD-00081     Andy Gunawan
```

Tugas Praktek

Ganti bagian […] pada code editor dengan perintah SQL untuk mengambil kolom kode_pelanggan dan nama, dan dengan filter nama berawalan huruf B.

Catatan: Gunakan huruf B besar, karena sistem DQLab masih sensitif mengenali code yang menggunakan huruf besar dan kecil, dan hanya menerima satu opsi.

Jika semua berjalan lancar, maka pada hasil akan terdapat tampilan data.frame yang terlihat sebagian sebagai berikut.

```
kode_pelanggan                     nama
1       KD-00099 Bapak Sanjaya Priyantoro
2       KD-00134               Budi Yahya
3       KD-00087            Budi Setiawan
4       KD-00130       Bapak Jujur Suwito
5       KD-00106               Budi Yahya
6       KD-00098              B. Sulaiman
7       KD-00147            Budi Setiawan
```

```
8       KD-00192 Bapak Sanjaya Priyantoro
```

Code Editor

```
library(RMySQL)

#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dqlabdatawrangling")


#Konstruksi SQL

sql <- "SELECT kode_pelanggan, nama FROM dqlab_messy_data WHERE nama like 'b%'"


#Mengirimkan query

rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))


#Mengambil data

data.pelanggan <- fetch(rs, n=-1)

data.pelanggan


#Menutup Koneksi

all_cons <- dbListConnections(MySQL())

for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dq
labdatawrangling")

> #Konstruksi SQL
> sql <- "SELECT kode_pelanggan, nama FROM dqlab_messy_data WHERE nama like 'b%'"
```

```
> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> data.pelanggan <- fetch(rs, n=-1)

> data.pelanggan
  kode_pelanggan                     nama
1       KD-00099 Bapak Sanjaya Priyantoro
2       KD-00134               Budi Yahya
3       KD-00087            Budi Setiawan
4       KD-00130       Bapak Jujur Suwito
5       KD-00106               Budi Yahya
6       KD-00098              B. Sulaiman
7       KD-00147            Budi Setiawan
8       KD-00192 Bapak Sanjaya Priyantoro

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Melakukan filtering data dengan where … REGEXP

REGEXP adalah operator khusus untuk melakukan filter berdasarkan regular expression pada MySQL. Fungsinya seperti function grepl di R.

REGEXP dapat memberikan solusi filtering yang tidak dapat dilakukan oleh operator = maupun LIKE.

Sebagai contoh, kita ingin data yang yang mengandung huruf q atau z maka konstruksi perintahnya terlihat sebagai berikut.

```
SELECT kode_pelanggan, nama FROM dqlab_messy_data WHERE nama REGEXP '[qz]'
```

Dengan konstruksi demikian, kita akan memperoleh data berikut.

```
  kode_pelanggan            nama
1       KD-00064     Fauzan Amir
2       KD-00062 Zulkifli Kirana
3       KD-00104  Iqbal Setiawan
```

Tugas Praktek

Ganti bagian […] pada code editor dengan perintah SQL untuk mengambil kolom kode_pelanggan dan nama, dan dengan filter nama yang mengandung huruf x atau z.

Jika semua berjalan lancar, maka pada hasil akan terdapat tampilan data.frame yang terlihat sebagian sebagai berikut.

```
  kode_pelanggan             nama
1       KD-00113 Edi %$ Alexander
2       KD-00064      Fauzan Amir
3       KD-00062  Zulkifli Kirana
```

Code Editor

```
library(RMySQL)

#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dqlabdatawrangling")


#Konstruksi SQL

sql <- "SELECT kode_pelanggan, nama FROM dqlab_messy_data WHERE nama REGEXP '[xz]'"


#Mengirimkan query

rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))


#Mengambil data

data.pelanggan <- fetch(rs, n=-1)

data.pelanggan


#Menutup Koneksi

all_cons <- dbListConnections(MySQL())

for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dq
labdatawrangling")

> #Konstruksi SQL
> sql <- "SELECT kode_pelanggan, nama FROM dqlab_messy_data WHERE nama REGEXP '[xz]'"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> data.pelanggan <- fetch(rs, n=-1)

> data.pelanggan
  kode_pelanggan             nama
1       KD-00113 Edi %$ Alexander
2       KD-00064      Fauzan Amir
3       KD-00062  Zulkifli Kirana

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Kesimpulan

Kemampuan data wrangling tidak terlepas dari kemampuan untuk membaca berbagai sumber data, termasuk di dalamnya sistem database relasional.

Pada bab ini kita telah mempelajari dasar:

- Perkenalan singkat konsep database relasional, SQL dan NoSQL.
- Perkenalan singkat MySQL sebagai salah produk database relasional.
- Penggunaaan function-function pada library RMySQL untuk melakukan koneksi dan mengambil data dengan eksekusi perintah SQL
- Perkenalan konstruksi SELECT untuk mengambil keseluruhan data, sebagian kolom, dan sebagian data dengan filtering.

Dengan bekal pengetahuan dan keterampilan ini kita akan mengkombinasikan skillset R dan SQL sehingga kontrol pengolahan data menjadi semakin baik di tangan kita.

Klik tombol Next untuk melanjutkan ke bab berikutnya.

Pendahuluan

Bab ini akan memfokuskan diri memperbaiki teks nama yang tidak lazim - sesuai dengan definisi pada bab "Data Profiling".

Kita akan banyak memperkenalkan function-function dan pola regex untuk mengolah isi sepanjang bab dan sisa course.

Pada kesempatan ini juga, DQLab akan akan memperkenalkan cara kerja atau metodologi sistematis kita bekerja dengan data pelanggan ini sebagai berikut:

- Sumber data dibaca dari **database MySQL** – dengan demikian membiasakan diri Anda untuk menggunakan perintah SELECT.
- Data cleansing akan dilakukan **per kolom** – dengan demikian Anda membiasakan diri mengolah data per tahap yang selain memberi fokus juga lebih "memberi nafas" kepada sistem R sehingga meningkatkan kinerja secara total.
- Perkenalan **pola regex** untuk mengidentifikasi dan mengganti isi data.
- Regex dan pola dari bpa akan banyak tumpang tindihnya, keduanya akan digunakan bergantian. Rule of thumbnya: **regex dari sisi performa jauh lebih lambat** karena butuh komputasi yang besar. Jadi gunakan regex ketika pola bpa tidak bisa.
- Hasilnya akan disimpan per file Excel dengan format nama "staging.nama.kolom.xlsx" – dengan demikian kontrol pengolahan data masih ada pada Anda.
- File-file ini pada akhir course akan **disatukan**, dan ditulis ke dalam satu file – menyambung point kedua di atas, akhirnya file-file "serpihan" ini harus dapat diintegrasikan kembali.

Dengan demikian sepanjang praktek, server database MySQL hanya digunakan untuk "read only" – dalam arti tidak ada penulisan apapun ke dalam database – sesuai kondisi riil yang dihadapi para data analis di lapangan.

Klik tombol Next untuk melanjutkan.

Menghilangkan spasi tambahan

Salah satu definisi nama yang tidak lazim atau salah tulis adalah spasi berlebih. Kebutuhan kita adalah menemukan spasi berlebih tersebut dan menggantinya dengan satu spasi saja.

Spasi berlebih sendiri jika dirinci lebih lanjut adalah karakter spasi dengan jumlah lebih dari satu dan posisinya berurutan, atau singkatnya spasi berulang.

Pola regex konkrit dari spasi berulang adalah `" {2,}"`, dimana:

- : karakter spasi.
- {2,} : adalah pola repetisi dimana karakter muncul berulang, minimal dua kali sampai dengan tidak terhingga.

Pola regex ini kita gunakan dengan function **gsub** untuk mengganti nilainya. Contoh penggunaan **gsub** untuk mengakomodir kebutuhan kita adalah sebagai berikut.

```
gsub(" {2,}", " ", data.pelanggan$nama)
```

Keterangan:

- `gsub`: function untuk mencari dan mengganti teks.
- `" {2,}"`: pola regex spasi berulang yang akan dicari.
- `" "`: karakter satu spasi.
- `pelanggan$nama`: sumber data yang kita gunakan.

Catatan: Regex memiliki shorthand class \s yang dapat mewakili spasi dan tab, namun karena perlu tambahan adaptasi code penulisan ke MySQL dan gsub perlu dilakukan, maka shorthand ini tidak akan dibahas pada bab ini.

Terakhir, ada kemungkinan spasi hanya satu tapi letaknya sebelum atau sesudah nama. Ini juga sesuatu yang kita tidak inginkan. Kondisi ini bisa kita perbaiki dengan menggunakan function **trimws**. Contoh penggunaan **trimws** untuk mengakomodir kebutuhan kita sebagai berikut.

```
trimws(data.pelanggan$nama,which="both")
```

Keterangan:

- `trimws`: function untuk menghilangkan spasi di awal dan/atau setelah teks.
- `pelanggan$nama`: sumber data yang kita gunakan.
- `which="both"` : parameter yang menginformasikan kepada triwms untuk menghilangkan spasi sebelum dan sesudah nama.

Tugas Praktek

Gantilah seluruh spasi berulang pada kolom Nama dari data pelanggan yang dibaca dari database MySQL. Pola regex " {2,}" akan digunakan untuk hal ini, mulai dari filtering di SQL sampai dengan pada saat penggunaan function gsub untuk mengganti teks.

Pada code editor telah disediakan potongan code yang mencerminkan keseluruhan proses tersebut. Anda tinggal mengganti "puzzle" dalam bentuk […1…] dengan pola regex spasi berulang untuk kepentingan identifikasi (ada di dua tempat).

Kemudian ganti […2…] dengan satu karakter spasi sebagai teks pengganti. Terakhir, lengkapi penggunaan function trimws untuk menggantikan bagian […3…]

```
> data.pelanggan
  kode_pelanggan                    nama
1       KD-00046     Ir. Ita    Nugraha
2       KD-00117 Florensia     Novianti
3       KD-00108      Ibu Jujur  Suwito


...


> data.pelanggan
  kode_pelanggan                nama
1       KD-00046    Ir. Ita Nugraha
2       KD-00117 Florensia Novianti
3       KD-00108   Ibu Jujur Suwito
```

Terlihat kalau data sebelum dan sesudah terjadi penggantian spasi berlebih menjadi satu spasi saja.

- "Ir. Ita Nugraha" menjadi "Ir. Ita Nugraha"
- "Florensia Novianti" menjadi "Florensia Novianti"
- "Ibu Jujur Suwito" menjadi "Ibu Jujur Suwito"

Code Editor

```
library(RMySQL)
#Membuka koneksi
con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
                 dbname="dqlabdatawrangling")

#Konstruksi SQL
sql <- "SELECT kode_pelanggan, nama from dqlab_messy_data where nama REGEXP
' {2,}'"

#Mengirimkan query
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data
data.pelanggan <- fetch(rs, n=-1)
data.pelanggan
data.pelanggan$nama <- gsub(" {2,}", " ", data.pelanggan$nama)
data.pelanggan$nama <- trimws(data.pelanggan$nama, which="both")
data.pelanggan

#Menutup Koneksi
all_cons <- dbListConnections(MySQL())
for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> #Konstruksi SQL
> sql <- "SELECT kode_pelanggan, nama from dqlab_messy_data where nama REGEXP ' {2,}'
"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> data.pelanggan <- fetch(rs, n=-1)

> data.pelanggan
  kode_pelanggan                    nama
1       KD-00046     Ir. Ita    Nugraha
2       KD-00117 Florensia     Novianti
3       KD-00108      Ibu Jujur  Suwito

> data.pelanggan$nama <- gsub(" {2,}", " ", data.pelanggan$nama)

> data.pelanggan$nama <- trimws(data.pelanggan$nama, which="both")

> data.pelanggan
  kode_pelanggan                nama
1       KD-00046    Ir. Ita Nugraha
2       KD-00117 Florensia Novianti
3       KD-00108   Ibu Jujur Suwito

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Menghilangkan angka dan simbol

Perbaikan selanjutnya untuk kolom nama adalah menghilangkan angka-angka dan simbol selain tanda baca titik dan koma. Kebutuhan kita adalah menemukan pola tersebut dan menggantinya dengan karakter kosong.

Dengan mendefinisikan ulang kebutuhan tersebut, polanya menjadi seluruh karakter yang **bukan** huruf, spasi, tanda titik dan tanda koma.

Pola regex konkrit dari definisi terakhir adalah "[^A-Za-z .,]", dimana:

- [] : pasangan kurung siku pembuka dan penutup merupakan tempat kita memasukkan karakter-karakter yang kita perlukan.
- ^ : tanda ini di awal artinya **negasi** atau pernyataan **bukan** untuk karakter-karakter yang akan mengikuti.
- A-Z: karakter huruf besar dari A sampai Z.
- a-z: karakter huruf kecil dari a sampai z.
- : tanda spasi
- . : tanda titik.
- , : tanda koma.

Pola regex ini kita gunakan dengan function **gsub** untuk menghilangkan karakternya. Contoh penggunaan **gsub** untuk mengakomodir kebutuhan kita adalah sebagai berikut.

```
gsub("[^A-Za-z .,]", "", data.pelanggan$Nama)
```

Keterangan:

- `gsub`: function untuk mencari dan mengganti teks.
- `[^A-Za-z .,]`: pola regex bukan huruf, spasi, tanda titik dan tanda koma.
- `""`: karakter kosong.
- `pelanggan$Nama`: sumber data yang kita gunakan.

Tugas Praktek

Masih sama dengan potongan code pada praktek sebelumnya, gantilah bagian […1…] dengan pola regex bukan huruf, spasi, tanda titik dan tanda koma untuk kepentingan identifikasi (ada di dua tempat).

Kemudian ganti […2…] dengan karakter kosong. Dengan demikian karakter yang ditemukan dengan pola di atas akan dihilangkan.

Jika berjalan dengan lancar, maka sebagian tampilan outputnya akan muncul sebagai berikut.

```
  kode_pelanggan                                  nama
1       KD-00113                       Edi %$ Alexander
2       KD-00010                     Ibu Sri Wahyuni@, IR
```

```
3       KD-00039    Joko Wiryanto Abadi (Pelanggan OKE)
4       KD-00005 Prihatin Setyonugroho (021-5555555544)
5       KD-00001                          Agus Cahyono's
6       KD-00063   Widianto Nuryajaya - 08222222999111)
7       KD-00120                            Dewi Sr|yani


…


  kode_pelanggan                                 nama
1       KD-00113                        Edi  Alexander
2       KD-00010                   Ibu Sri Wahyuni, IR
3       KD-00039 Joko Wiryanto Abadi Pelanggan OKE
4       KD-00005                 Prihatin Setyonugroho
5       KD-00001                         Agus Cahyonos
6       KD-00063                    Widianto Nuryajaya
7       KD-00120                           Dewi Sryani
```

Terlihat kalau data sebelum dan sesudah terjadi penggantian teks, nama menjadi lebih rapi. Tapi perhatikan kalau masih ada tiga data yang masih aneh yaitu :

- "Edi Alexander" masih memiliki spasi tambahan setelah terjadi pergantian menjadi " Edi Alexander"
- "Joko Wiryanto Abadi Pelanggan OKE", harusnya "" karena " Joko Wiryanto Abadi" hanya keterangan.
- "Ibu Sri Wahyuni, IR" memiliki kata panggilan "Ibu" yang kemungkinan besar adalah panggilan dan bukan nama. Dan "IR" sendiri jika gelar maka penulisannya salah. Tapi ini memang di luar pola definisi kita.

Hasil ini menunjukkan dua hal:

- Bahwa setiap pola harus dilakukan berantai. Sebagai contoh untuk kasus ini, setelah menghilangkan angka dan simbol kita perlu melanjutkan dengan menghilangkan spasi berulang. Urutan pola mana yang dieksekusi terlebih dahulu menjadi faktor penting.
- Pola definisi yang sudah kita pikirkan sudah benar ternyata tidak bisa berlaku universal, akan ada kondisi lain yang membuat salah – keterangan-keterangan tambahan pada contoh di atas menunjukkan hal itu dengan jelas.

Dari poin terakhir, jelas sekali kita selalu bisa melewatkan pola kotor baru sehingga definisi dan urutan eksekusi pola cleansing yang ada menjadi tidak berlaku lagi.

Inilah yang menjadi basis argumen bahwa data tidak bisa 100 persen menjadi bersih atau clean. Harus ada intervensi manusia dengan proses yang dinamakan ***data stewarding*** – dimana pengecekan manual perlu dilakukan namun tentunya tidak seluruh data.

Teknik dan metodologi untuk membantu *data stewarding* cukup banyak, namun iterasi bertahap dari proses profiling data dan pembuatan definisi adalah metodologi yang terbukti efektif dan ampuh untuk mempermudah sekaligus mempercepat proses *data stewarding*.

Code Editor

```
library(RMySQL)

#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo",
host="mysqlhost",dbname="dqlabdatawrangling")


#Konstruksi SQL

sql <- "SELECT kode_pelanggan, nama from dqlab_messy_data where nama REGEXP
'[^A-Za-z .,]'"


#Mengirimkan query

rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))


#Mengambil data

data.pelanggan <- fetch(rs, n=-1)

data.pelanggan

data.pelanggan$nama <- gsub("[^A-Za-z .,]", "", data.pelanggan$nama)

data.pelanggan$nama <- trimws(data.pelanggan$nama, which="both")

data.pelanggan


#Menutup Koneksi

all_cons <- dbListConnections(MySQL())
```

for(con in all_cons) dbDisconnect(con)

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dq
labdatawrangling")

> #Konstruksi SQL
> sql <- "SELECT kode_pelanggan, nama from dqlab_messy_data where nama REGEXP '[^A-Za
-z .,]'"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> data.pelanggan <- fetch(rs, n=-1)

> data.pelanggan
  kode_pelanggan                                  nama
1       KD-00113                       Edi %$ Alexander
2       KD-00010                     Ibu Sri Wahyuni@, IR
3       KD-00039    Joko Wiryanto Abadi (Pelanggan OKE)
4       KD-00005 Prihatin Setyonugroho (021-5555555544)
5       KD-00001                          Agus Cahyono's
6       KD-00063   Widianto Nuryajaya - 08222222999111)
7       KD-00120                            Dewi Sr|yani

> data.pelanggan$nama <- gsub("[^A-Za-z .,]", "", data.pelanggan$nama)

> data.pelanggan$nama <- trimws(data.pelanggan$nama, which="both")

> data.pelanggan
  kode_pelanggan                                nama
1       KD-00113                    Edi  Alexander
2       KD-00010                Ibu Sri Wahyuni, IR
3       KD-00039 Joko Wiryanto Abadi Pelanggan OKE
4       KD-00005              Prihatin Setyonugroho
5       KD-00001                       Agus Cahyonos
6       KD-00063                 Widianto Nuryajaya
7       KD-00120                       Dewi Sryani

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Menghilangkan kata panggilan dan perbaikan penulisan gelar

Pada contoh data pelanggan ditemukan lagi masalah baru, yaitu terdapat panggilan seperti "Ibu", "Bapak", dan lain-lain – yang tidak cocok dimasukkan. Kecuali memang nama yang diberikan adalah seperti itu, namun kemungkinan yang lebih besar adalah salah tulis.

Selain itu, pada praktek sebelumnya juga ada gelar yang salah tulis dari segi huruf besar dan kecil.

Contoh: "Ir" yang merupakan singkatan dari "Insinyur" ditulis dengan "IR".

Kedua kondisi di atas tidak dapat ditemukan dengan teknik profiling yang telah kita pelajari. Agar standar kita buat rentetan perintah gsub dengan input berupa nama panggilan dan gelar yang salah tulis beserta teks standar penggantinya. Untuk selanjutnya, proses ini kita sebut **standarisasi data**.

Kita tetap gunakan function **gsub** dengan pola regex **\b**. Dan karena setiap pola regex yang memiliki backslash tunggal \ harus ditulis dua kali pada gsub, maka **\b** diubah menjadi **\\b** pada saat digunakan pada function gsub.

Pola **\\b** menunjukkan bahwa kata yang menjadi input itu harus sama persis. Sebagai contoh pola "**\\b**ibu**\\b**" akan cocok dengan kata "Ibu", "ibu" dan "iBu". Tapi tidak dengan kata "sibuk" dan "ribut".

Kemudian pada function tersebut juga ada tambahan parameter ignore.case = TRUE dengan tujuan pola berlaku untuk huruf besar maupun kecil.

```
data.pelanggan$nama <- gsub("\\bir\\b",
"Ir",data.pelanggan$nama, ignore.case = TRUE)

data.pelanggan$nama <- gsub("\\bibu\\b",
"",data.pelanggan$nama, ignore.case = TRUE)
```

Agar seragam, maka teks yang dicari seperti "ir" dan "ibu" semua ditulis dengan huruf kecil.

Tugas Praktek

Gantilah bagian […1…], […2…], […3…] dengan function gsub yang tepat secara berurut untuk kata "bapak", "ibu", dan "ir".

Jika berjalan dengan lancar, maka sebagian tampilan outputnya akan muncul sebagai berikut.

```
> data.pelanggan
  kode_pelanggan                    nama
1       KD-00053           Ibu Heidi Goh
```

```
2       KD-00099 Bapak Sanjaya Priyantoro
3       KD-00010     Ibu Sri Wahyuni@, IR
4       KD-00130       Bapak Jujur Suwito
5       KD-00108        Ibu Jujur  Suwito
6       KD-00192 Bapak Sanjaya Priyantoro

...

> data.pelanggan
  kode_pelanggan                nama
1       KD-00053           Heidi Goh
2       KD-00099  Sanjaya Priyantoro
3       KD-00010     Sri Wahyuni, Ir
4       KD-00130        Jujur Suwito
5       KD-00108        Jujur Suwito
6       KD-00192  Sanjaya Priyantoro
```

Catatan: Perhatikan juga bahwa pola untuk cleansing spasi berulang dan simbol non nama telah dimasukkan juga dalam code editor.

Code Editor

```
library(RMySQL)

#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo",
host="mysqlhost",dbname="dqlabdatawrangling")


#Konstruksi SQL

sql <- "SELECT kode_pelanggan, nama from dqlab_messy_data where nama like
'%ibu%' or nama like '%bapak%'"
```

#Mengirimkan query

rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data

data.pelanggan <- fetch(rs, n=-1)

data.pelanggan

data.pelanggan$nama <- gsub("[^A-Za-z .,]", "", data.pelanggan$nama)

data.pelanggan$nama <- gsub("\\bbapak\\b", "",data.pelanggan$nama, ignore.case = TRUE)

data.pelanggan$nama <- gsub("\\bibu\\b", "",data.pelanggan$nama, ignore.case = TRUE)

data.pelanggan$nama <- gsub("\\bir\\b", "Ir",data.pelanggan$nama, ignore.case = TRUE)

data.pelanggan$nama <- gsub("[ ]{2,}", " ", data.pelanggan$nama)

data.pelanggan$nama <- trimws(data.pelanggan$nama, which="both")

data.pelanggan

#Menutup Koneksi

all_cons <- dbListConnections(MySQL())

for(con in all_cons) dbDisconnect(con)

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dq
labdatawrangling")

> #Konstruksi SQL
> sql <- "SELECT kode_pelanggan, nama from dqlab_messy_data where nama like '%ibu%' o
r nama like '%bapak%'"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"
```

```
> #Mengambil data
> data.pelanggan <- fetch(rs, n=-1)

> data.pelanggan
  kode_pelanggan                      nama
1       KD-00053            Ibu Heidi Goh
2       KD-00099 Bapak Sanjaya Priyantoro
3       KD-00010     Ibu Sri Wahyuni@, IR
4       KD-00130       Bapak Jujur Suwito
5       KD-00108        Ibu Jujur  Suwito
6       KD-00192 Bapak Sanjaya Priyantoro

> data.pelanggan$nama <- gsub("[^A-Za-z .,]", "", data.pelanggan$nama)

> data.pelanggan$nama <- gsub("\\bbapak\\b", "",data.pelanggan$nama, ignore.case = TR
UE)

> data.pelanggan$nama <- gsub("\\bibu\\b", "",data.pelanggan$nama, ignore.case = TRUE
)

> data.pelanggan$nama <- gsub("\\bir\\b", "Ir",data.pelanggan$nama, ignore.case = TRU
E)

> data.pelanggan$nama <- gsub("[ ]{2,}", " ", data.pelanggan$nama)

> data.pelanggan$nama <- trimws(data.pelanggan$nama, which="both")

> data.pelanggan
  kode_pelanggan                nama
1       KD-00053          Heidi Goh
2       KD-00099 Sanjaya Priyantoro
3       KD-00010    Sri Wahyuni, Ir
4       KD-00130       Jujur Suwito
5       KD-00108       Jujur Suwito
6       KD-00192 Sanjaya Priyantoro

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Konsolidasi Proses Cleansing

Pola dan proses cleansing berupa standarisasi kolom **nama** yang kita lakukan sejauh ini hanya kita terapkan pada sebagian porsi baris data untuk menjaga fokus.

Sekarang saatnya kita konsolidasi prosesnya dengan mengambil keseluruhan data untuk kolom **nama**, dan kemudian diolah dengan seluruh pola yang telah kita buat.

Hasil ini akan ditulis ke dalam file Excel yang akan kita gabungkan dengan standarisasi kolom lain di akhir course ini.

Agar dapat diintegrasikan, kita juga akan mengambil kolom **kode_pelanggan**. Dan juga supaya bisa dibandingkan, kolom nama diduplikasi dengan nama lain agar kondisi sebelum standarisasi juga akan diikutkan. Perintahnya adalah sebagai berikut:

```
data.pelanggan$nama.before <- data.pelanggan$nama
```

Tugas Praktek

Code editor telah dilengkapi dengan pembacaan seluruh dataset dari MySQL dan penerapan seluruh pola standarisasi.

Gantilah bagian […1…] dengan perintah duplikasi kolom berikut:

```
data.pelanggan$nama.before <- data.pelanggan$nama
```

Kemudian ganti bagian […2…] dengan teks `"staging.nama.xlsx"` yang merupakan nama file output kita.

Jika berjalan dengan lancar, maka pada saat kita klik menu "**Download Output File**" maka file "**staging.nama.xls**" sudah ada dalam list.

Dan berikut adalah tampilan file Excel tersebut setelah di-download dan dibuka dengan aplikasi Excel.

	A	B	C
1	kode_pelanggan	nama	nama.before
2	KD-00032	Eva Novianti, S.H.	Eva Novianti, S.H.
3	KD-00053	Heidi Goh	Ibu Heidi Goh
4	KD-00133	Unang Handoko	Unang Handoko
5	KD-00056	Jokolono Sukarman	Jokolono Sukarman
6	KD-00111	Tommy Sinaga	Tommy Sinaga
7	KD-00036	Irwan Setianto	Irwan Setianto
8	KD-00126	Agus Cahyono	Agus Cahyono
9	KD-00137	Maria Sirait	Maria Sirait
10	KD-00046	Ir. Ita Nugraha	Ir. Ita Nugraha
11	KD-00027	Djoko Wardoyo, Drs.	Djoko Wardoyo, Drs.
12	KD-00002	Khairul Nissa	Khairul Nissa
13	KD-00075	Kaka Ari Lima	Kaka Ari Lima
14	KD-00076	Safira Hana Sahrani	Safira Hana Sahrani
15	KD-00035	Sidharta Paul	Sidharta Paul
16	KD-00113	Edi Alexander	Edi %$ Alexander
17	KD-00099	Saniava Privantoro	Ranak Saniava Privantoro

Code Editor

```
library(openxlsx)

library(RMySQL)

#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo",
host="mysqlhost",dbname="dqlabdatawrangling")


#Konstruksi SQL

sql <- "SELECT kode_pelanggan, nama from dqlab_messy_data"


#Mengirimkan query

rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))


#Mengambil data

data.pelanggan <- fetch(rs, n=-1)
```

```
data.pelanggan$nama.before <- data.pelanggan$nama

data.pelanggan$nama <- gsub("[^A-Za-z .,]", "", data.pelanggan$nama)

data.pelanggan$nama <- gsub("\\bbapak\\b", "",data.pelanggan$nama, ignore.case = TRUE)

data.pelanggan$nama <- gsub("\\bibu\\b", "",data.pelanggan$nama, ignore.case = TRUE)

data.pelanggan$nama <- gsub("\\bir\\b", "Ir",data.pelanggan$nama, ignore.case = TRUE)

data.pelanggan$nama <- gsub("[ ]{2,}", " ", data.pelanggan$nama)

data.pelanggan$nama <- trimws(data.pelanggan$nama, which="both")

write.xlsx(data.pelanggan, file="staging.nama.xlsx")


#Menutup Koneksi

all_cons <- dbListConnections(MySQL())

for(con in all_cons) dbDisconnect(con)
```

Console

```
> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dq
labdatawrangling")

> #Konstruksi SQL
> sql <- "SELECT kode_pelanggan, nama from dqlab_messy_data"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> data.pelanggan <- fetch(rs, n=-1)

> data.pelanggan$nama.before <- data.pelanggan$nama

> data.pelanggan$nama <- gsub("[^A-Za-z .,]", "", data.pelanggan$nama)

> data.pelanggan$nama <- gsub("\\bbapak\\b", "",data.pelanggan$nama, ignore.case = TR
UE)

> data.pelanggan$nama <- gsub("\\bibu\\b", "",data.pelanggan$nama, ignore.case = TRUE
)
```

```
> data.pelanggan$nama <- gsub("\\bir\\b", "Ir",data.pelanggan$nama, ignore.case = TRU
E)

> data.pelanggan$nama <- gsub("[ ]{2,}", " ", data.pelanggan$nama)

> data.pelanggan$nama <- trimws(data.pelanggan$nama, which="both")

> write.xlsx(data.pelanggan, file="staging.nama.xlsx")

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Kesimpulan

Sepanjang bab ini, Anda telah menyelesaikan teknik *cleansing* atau standarisasi untuk kolom nama dengan tahapan berikut:

- Membaca sumber data dari database MySQL dengan filter regex dan operator like.
- Mengenal berbagai pola regex untuk menangani spasi berulang, non huruf, dan daftar kata yang fix.
- Melakukan konsolidasi atau penyusunan pola regex dalam urutan yang tepat dan diterapkan ke seluruh data.
- Hasil konsolidasi disimpan dalam file Excel yang akan kita integrasikan di akhir course ini.

Klik tombol Next untuk melanjutkan ke bab berikutnya.

Pendahuluan

Setelah fokus ke kolom nama dan membiasakan diri dengan proses cleansing, pada bab ini kita akan fokus mengenali pola outlier dan standarisasi untuk sisa kolom character lain dengan urutan berikut:

- No Telepon
- Kode Pos
- Alamat
- Aktif

Klik tombol Nex untuk melanjutkan.

Profiling kolom Nomor Telepon (1)

Pola yang kita dapatkan dari bab "Data Profiling" sebelumnya telah disimpan di dalam table **dqlab_messy_data** di dalam MySQL. Hal ini membuat kita tidak perlu setiap kali menggunakan function dari bpa, dan juga akan memberi banyak keuntungan performa ketika data kita sangat besar.

Kembali ke dataset kita, kolom **no_telepon** adalah data yang sangat penting. Komunikasi dengan aplikasi populer seperti Whatsapp sekarang ini semuanya berbasiskan nomor telepon. Dengan demikian, jika ada anomali dari data ini perlu cepat diketahui sehingga dapat segera ditindaklanjuti.

Kita akan lakukan analisa kolom **no_telepon** melalui kolom lain,
yaitu **pola_no_telepon** –kolom hasil output dari function **basic_pattern_analysis**.

Gunakan perintah SQL berikut untuk melakukan hal tersebut.

```
SELECT pola_no_telepon, length(pola_no_telepon) as
panjang_text, count(*) as jumlah_data
from dqlab_messy_data
group by pola_no_telepon
```

Keterangan dari perintah SQL di atas.

Elemen Perintah	Keterangan
SELECT	Konstruksi awal perintah SQL untuk mengambil kolom data
pola_no_telepon	Mengambil kolom "pola_no_telepon"
count(*) as jumlah_data	• Menggunakan fungsi **count** di SQL untuk menghitung jumlah baris data. • ***** mewakili seluruh kolom yang di-count. • **as jumlah_data**: merupakan penamaan kolom dari hasil count(*) menjadi **jumlah_data**.
length(pola_no_telepon) as panjang_text	• Menggunakan fungsi **length** di SQL untuk menghitung jumlah karakter. • **pola_no_telepon** : kolom yang di-count. • **as panjang_text**: merupakan penamaan kolom dari hasil length(pola_no_telepon) menjadi **panjang_text**.
group by pola_no_telepon	Pengelompokan nilai count dan length berdasarkan kolom pola_no_telepon

Hasil dari eksekusi perintah tersebut tampak sebagai berikut.

```
    pola_no_telepon panjang_pola jumlah_data
1   +99999999999999           15           2
2 +9999999999999999           17          57
3                   -            1           1
4     9999999999999           13           1
5    99999999999999           14           3
6   999999999999999           15          53
7  9999999999999999           16          38
```

Berikut adalah analisa dari hasil di atas.

- Ada tujuh pola yang teridentifikasi.
- Mayoritas jumlah ada pada pola nomor 2, 6 dan 7.
- Panjang data pada 2, 6 dan 7 adalah 17, 15 dan 16. Namun karena pola no 2 ada tanda +, jadi kemungkinan panjangnya sebenarnya adalah 16. Sama dengan pola no 7.
- Tidak ada karakter lain selain angka dan tanda +. Kecuali satu data di pola no. 3 hanya tanda minus (-). Ini bisa dianggap missing value.
- Pola lain yang tidak biasa adalah pola no 1 (+99999999999999, jumlahnya hanya 2 data), 4 (9999999999999, jumlahnya hanya 1 data), dan 5 (99999999999999, hanya 3 data).
- Pola nomor 1, 3, 4 dan 5 akan kita flag sebagai outlier yang perlu ditindaklanjuti pada praktek selanjutnya.
- Ada perbedaan jumlah teks antara pola no 5, 6 dan 2 (jumlah teks masing-masing adalah 15, 16, dan 17), kita asumsikan ini disebabkan angka awal mobile 0, 62 dan +62 di depannya. Ini akan kita cek dengan query tambahan pada profiling selanjutnya.

Mari kita jalankan langsung tugas praktek untuk mendapatkan output profiling seperti di atas.

Tugas Praktek

Gantilah bagian […1…] dengan memasukkan perintah SQL yang dicontohkan pada Lesson.

Code Editor

```
library(RMySQL)

#Membuka koneksi
```

```
con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
                 dbname="dqlabdatawrangling")

#Konstruksi SQL
sql <- "SELECT pola_no_telepon, length(pola_no_telepon) as panjang_text, count(*) as jumlah_data
from dqlab_messy_data
group by pola_no_telepon"

#Mengirimkan query
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data
profil_no_telepon <- fetch(rs, n=-1)
print(profil_no_telepon)

#Clear resultset
dbClearResult(rs)

#Menutup Koneksi
all_cons <- dbListConnections(MySQL())
for(con in all_cons) dbDisconnect(con)
```

Console

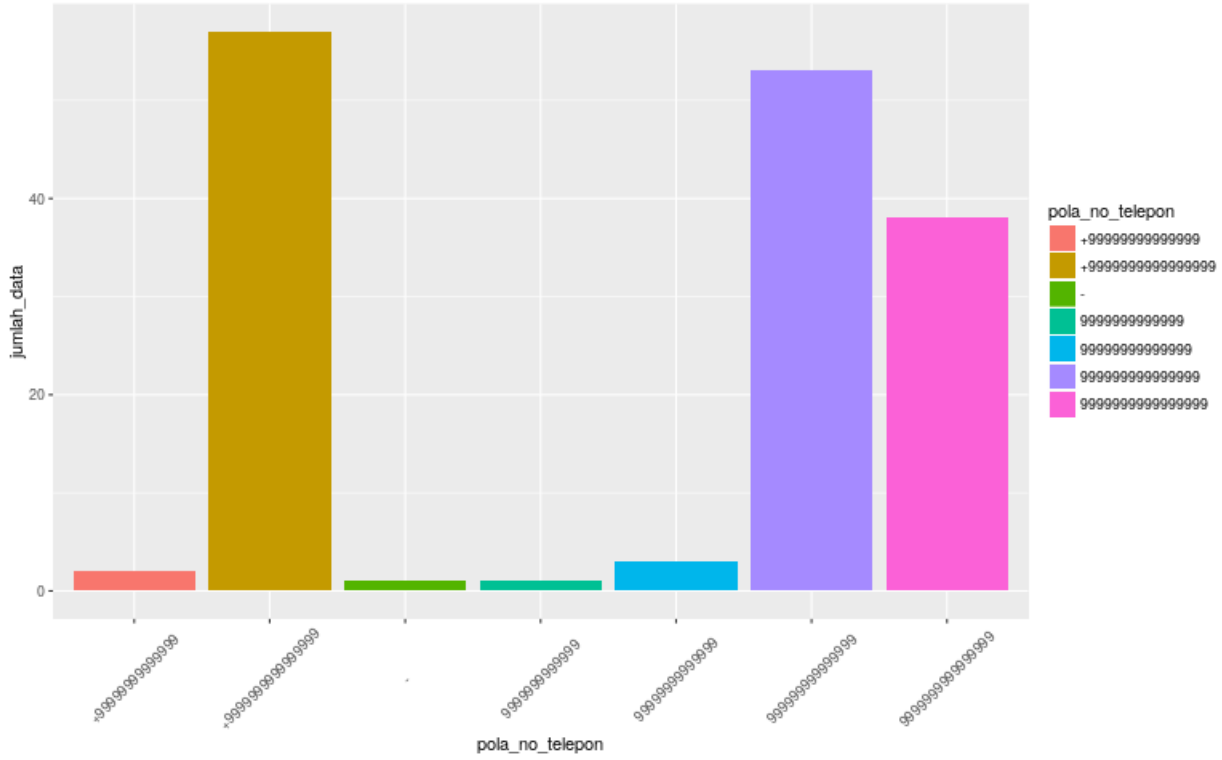
```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> #Konstruksi SQL
```

```
> sql <- "SELECT pola_no_telepon, length(pola_no_telepon) as panjang_text, count(*) a
s jumlah_data
+ from dqlab_messy_data
+ group by pola_no_telepon"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> profil_no_telepon <- fetch(rs, n=-1)

> print(profil_no_telepon)
    pola_no_telepon panjang_text jumlah_data
1   +99999999999999           15           2
2 +9999999999999999           17          57
3                 -            1           1
4     9999999999999           13           1
5    99999999999999           14           3
6   999999999999999           15          53
7  9999999999999999           16          38

> #Clear resultset
> dbClearResult(rs)
[1] TRUE

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Profiling kolom Nomor Telepon (2)

Menganalisa jumlah data dari pola pada praktek sebelumnya dengan memperhatikan angka akan lebih menarik jika langsung divisualisasikan.

Untuk analisa distribusi dari pola no_telepon ini kita bisa gunakan visualisasi bar chart seperti gambar berikut.



Disini langsung terlihat ada tiga pola mayoritas dan yang jauh lebih kecil sebagai outlier.

Untuk menghasilkan chart tersebut, kita tambahkan code ggplot berikut dari praktek sebelumnya.

```
plot.profile <- ggplot(data=profil_no_telepon, aes(x =
pola_no_telepon, y = jumlah_data, fill = pola_no_telepon))

plot.profile <- plot.profile + theme(axis.text.x =
element_text(angle=45,vjust = 0.5))

plot.profile + geom_bar(stat="identity")
```

Tugas Praktek

Tambahkan code yang dicontohkan di atas untuk menggantikan bagian […] pada code editor. Jika dijalankan maka plot distribusi pola akan terlihat.

Code Editor

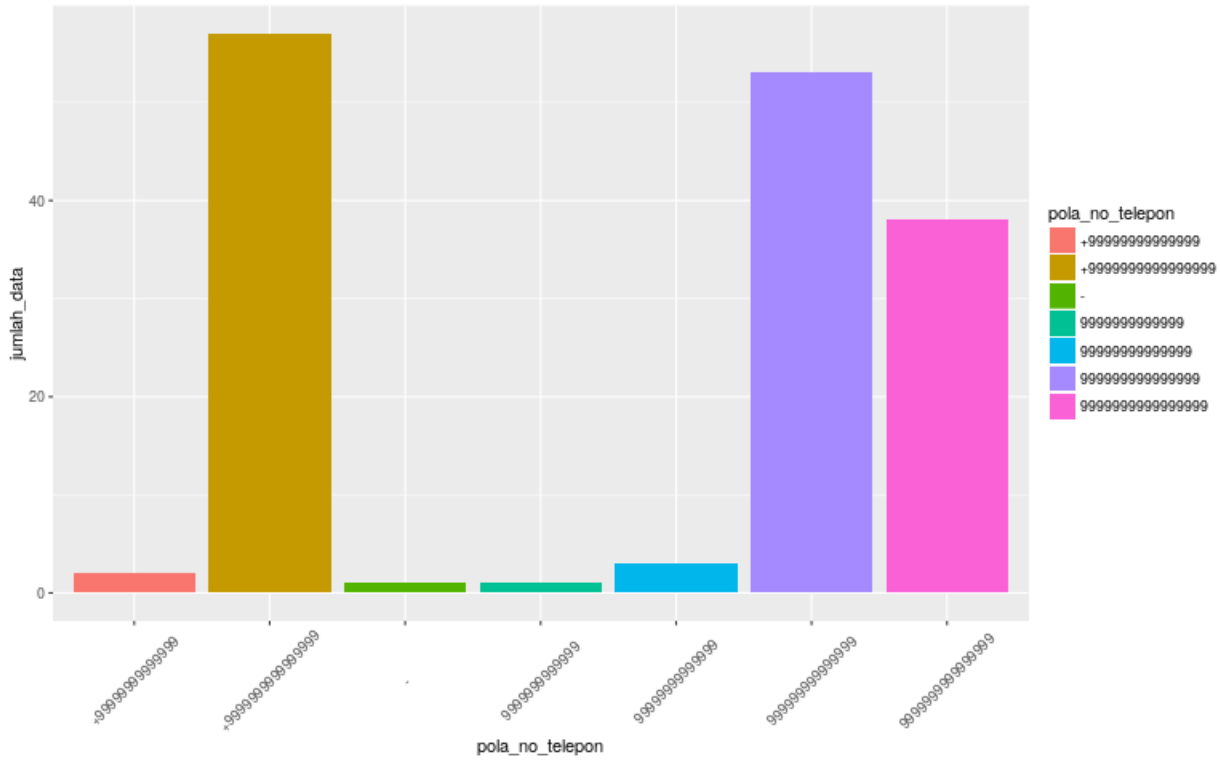
```
library(ggplot2)

#Membuka koneksi
con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
                 dbname="dqlabdatawrangling")

#Konstruksi SQL
sql <- "SELECT pola_no_telepon, length(pola_no_telepon) as panjang_text, count(*) as jumlah_data from dqlab_messy_data group by pola_no_telepon"

#Mengirimkan query
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data
profil_no_telepon <- fetch(rs, n=-1)
print(profil_no_telepon)

#Plotting data
plot.profile <- ggplot(data=profil_no_telepon, aes(x = pola_no_telepon, y = jumlah_data, fill = pola_no_telepon))

plot.profile <- plot.profile + theme(axis.text.x = element_text(angle=45,vjust = 0.5))

plot.profile + geom_bar(stat="identity")

#Clear resultset
dbClearResult(rs)
```

#Menutup Koneksi

all_cons <- dbListConnections(MySQL())

for(con in all_cons) dbDisconnect(con)

Console



Profiling kolom Nomor Telepon (3)

Kembali ke hasil profiling pada praktek sebelumnya, sebagai berikut.

	pola_no_telepon	panjang_pola	jumlah_data
1	+99999999999999	15	2
2	+9999999999999999	17	57
3	-	1	1
4	9999999999999	13	1
5	99999999999999	14	3
6	999999999999999	15	53
7	9999999999999999	16	38

Terlihat ada perbedaan antara pola 6, 7 dan 1 dari sisi jumlah teks (15, 16 dan 17). Kita asumsikan ini dikarenakan variasi penulisan awal nomor mobile berupa angka 0, 62 dan +62.

Kita akan uji asumsi ini dengan mengambil:

- satu karakter pertama dari isi kolom **no_telepon** untuk pola nomor 6 (999999999999999).
- dua karakter pertama dari isi kolom **no_telepon** untuk pola nomor 7 (9999999999999999).
- Tiga karakter pertama dari isi kolom **no_telepon** untuk pola nomor 2 (+9999999999999999).

Mari kita langsung praktekkan saja pada tugas berikut.

Tugas Praktek

Gantilah bagian […1…] pada code editor dengan perintah SQL berikut sebagai pengecekan asumsi pertama.

```
SELECT left(no_telepon,1) as prefix_no_telepon,
pola_no_telepon
from dqlab_messy_data where pola_no_telepon =
'999999999999999'
group by left(no_telepon,1), pola_no_telepon
```

Disini diperkenalkan function **left** untuk mengambil sejumlah karakter pertama dari kolom no_telepon. Perhatikan kalau function **left** ini turut serta dimasukkan ke dalam grouping. Ini dikarenakna function **left** ini mengambil karakter yang bisa angka maupun bukan, dan bukan operasi matematika seperti function **length** maupun **count** di praktek sebelumnya.

Gantilah bagian […2…] pada code editor dengan perintah SQL berikut sebagai pengecekan asumsi kedua.

```
SELECT left(no_telepon,2) as prefix_no_telepon,
pola_no_telepon
from dqlab_messy_data where pola_no_telepon =
'9999999999999999'
group by left(no_telepon,2), pola_no_telepon
```

Dan terakhir, gantilah bagian […3…] pada code editor dengan perintah SQL berikut sebagai pengecekan asumsi kedua.

```
SELECT left(no_telepon,3) as prefix_no_telepon,
pola_no_telepon
from dqlab_messy_data where pola_no_telepon =
'+99999999999999999'
group by left(no_telepon,3), pola_no_telepon
```

Jika berjalan dengan baik maka akan muncul potongan hasil berikut.

```
…
> print(profil_no_telepon)
  prefix_no_telepon     pola_no_telepon
1                 0 999999999999999


…


> print(profil_no_telepon)
  prefix_no_telepon      pola_no_telepon
1                62 9999999999999999
…


  prefix_no_telepon   pola_no_telepon
1               +62 +99999999999999999
```

Dengan masing-masing hanya keluarin hasil satu baris data, artinya tidak ada variasi lain. Kita bisa anggap kedua pola tersebut sangat konsisten dari isi data, dengan demikian tinggal diputuskan untuk angka 0, 62 dan +62 mana yang akan jadi standar.

Code Editor

library(RMySQL)

```
#Membuka koneksi
con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
                 dbname="dqlabdatawrangling")

#Konstruksi SQL untuk Profil 1
sql <- "SELECT left(no_telepon,1) as prefix_no_telepon, pola_no_telepon
from dqlab_messy_data where pola_no_telepon = '999999999999999'
group by left(no_telepon,1), pola_no_telepon"

#Mengirimkan query untuk Profil 1
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data untuk Profil 1
profil_no_telepon <- fetch(rs, n=-1)
print(profil_no_telepon)

#Clear resultset untuk Profil 1
dbClearResult(rs)

#Konstruksi SQL untuk Profil 2
sql <- "SELECT left(no_telepon,2) as prefix_no_telepon, pola_no_telepon
from dqlab_messy_data where pola_no_telepon = '9999999999999999'
group by left(no_telepon,2), pola_no_telepon"

#Mengirimkan query untuk Profil 2
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data untuk Profil 2
```

```
profil_no_telepon <- fetch(rs, n=-1)
print(profil_no_telepon)

#Clear resultset untuk Profil 2
dbClearResult(rs)

#Konstruksi SQL untuk Profil 3
sql <- "SELECT left(no_telepon,3) as prefix_no_telepon, pola_no_telepon
from dqlab_messy_data where pola_no_telepon = '+9999999999999999'
group by left(no_telepon,3), pola_no_telepon"

#Mengirimkan query untuk Profil 3
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data untuk Profil 3
profil_no_telepon <- fetch(rs, n=-1)
print(profil_no_telepon)

#Clear resultset untuk Profil 3
dbClearResult(rs)

#Menutup Koneksi
all_cons <- dbListConnections(MySQL())
for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> #Konstruksi SQL untuk Profil 1
> sql <- "SELECT left(no_telepon,1) as prefix_no_telepon, pola_no_telepon
+ from dqlab_messy_data where pola_no_telepon = '999999999999999'
+ group by left(no_telepon,1), pola_no_telepon"

> #Mengirimkan query untuk Profil 1
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data untuk Profil 1
> profil_no_telepon <- fetch(rs, n=-1)

> print(profil_no_telepon)
  prefix_no_telepon pola_no_telepon
1                 0 999999999999999

> #Clear resultset untuk Profil 1
> dbClearResult(rs)
[1] TRUE

> #Konstruksi SQL untuk Profil 2
> sql <- "SELECT left(no_telepon,2) as prefix_no_telepon, pola_no_telepon
+ from dqlab_messy_data where pola_no_telepon = '9999999999999999'
+ group by left(no_telepon,2), pola_no_telepon"

> #Mengirimkan query untuk Profil 2
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data untuk Profil 2
> profil_no_telepon <- fetch(rs, n=-1)

> print(profil_no_telepon)
  prefix_no_telepon  pola_no_telepon
1                62 9999999999999999

> #Clear resultset untuk Profil 2
> dbClearResult(rs)
[1] TRUE

> #Konstruksi SQL untuk Profil 3
> sql <- "SELECT left(no_telepon,3) as prefix_no_telepon, pola_no_telepon
+ from dqlab_messy_data where pola_no_telepon = '+9999999999999999'
+ group by left(no_telepon,3), pola_no_telepon"

> #Mengirimkan query untuk Profil 3
```

```
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data untuk Profil 3
> profil_no_telepon <- fetch(rs, n=-1)

> print(profil_no_telepon)
  prefix_no_telepon   pola_no_telepon
1               +62 +9999999999999999

> #Clear resultset untuk Profil 3
> dbClearResult(rs)
[1] TRUE

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Standarisasi kolom Nomor Telepon

Pada umumnya nomor telepon diberi kode negara dan pada saat dial kita menggunakan tanda +. Berdasarkan kedua hal tersebut, untuk dataset kita diputuskan untuk menggunakan standarisasi dimana nomor telepon dimulai dari tanda + diikuti 62 (kode negara untuk Indonesia).

Untuk melakukan hal ini maka ada dua hal yang perlu kita lakukan berdasarkan profiling kita pada praktek sebelumnya:

- Menambahkan awalan "+" untuk pola "9999999999999999" (16 digit angka).
- Mengganti awalan "0" menjadi "+62" untuk pola "999999999999999" (15 digit angka).

Pertama yang kita perlu ketahui adalah filter dataset setelah dibaca di R, karena dari MySQL akan kita tarik seluruh data.

Jika data yang ditarik dinamakan data.telepon, maka berikut adalah cara melakukan filter:

```
    data.telepon[data.telepon$pola_no_telepon=="9999999999999999",]
```

Teks berwarna merah adalah konstruksi filter yang kemudian dijadikan index untuk mengambil data.telepon sesuai filter.

Untuk menambahkan "+" untuk pola "9999999999999999" kita gunakan function **paste** – yang membutuhkan input berupa beberapa teks yang akan digabungkan, dan pemisah teks dengan parameter **sep=""**.

Berikut adalah perintah lengkapnya untuk kasus kita.

```
    paste("+",
data.telepon[data.telepon$pola_no_telepon=="9999999999999999",]$
no_telepon, sep="")
```

Terakhir, untuk mengganti awalah "0" menjadi "+62" untuk data berpola "999999999999999" kita tetap gunakan function **gsub** dengan pola regex "^0" dimana tanda topi (^) menunjukkan bahwa pola 0 harus di bagian awal dari teks, bukan di tengah atau di akhir.

Berikut adalah perintah lengkapnya untuk kasus kita.

```
    gsub("^0","+62",  data.telepon[data.telepon$pola_no_telepon
=="999999999999999",]$no_telepon)
```

Tugas Praktek

Lakukan standarisasi isi **no_telepon** pada dataset kita dengan melakukan hal berikut di code editor:

- Mengganti bagian […1…] dengan function **paste** untuk menambahkan tanda "+" pada isi no_telepon untuk pola "9999999999999999".
- Mengganti bagian […2…] dengan function **gsub** untuk mengganti awalan "0" dengan tanda "+62" untuk pola "999999999999999".
- Mengganti bagian […3…] dengan dengan nama file output bernama "**staging.no_telepon.xlsx**".

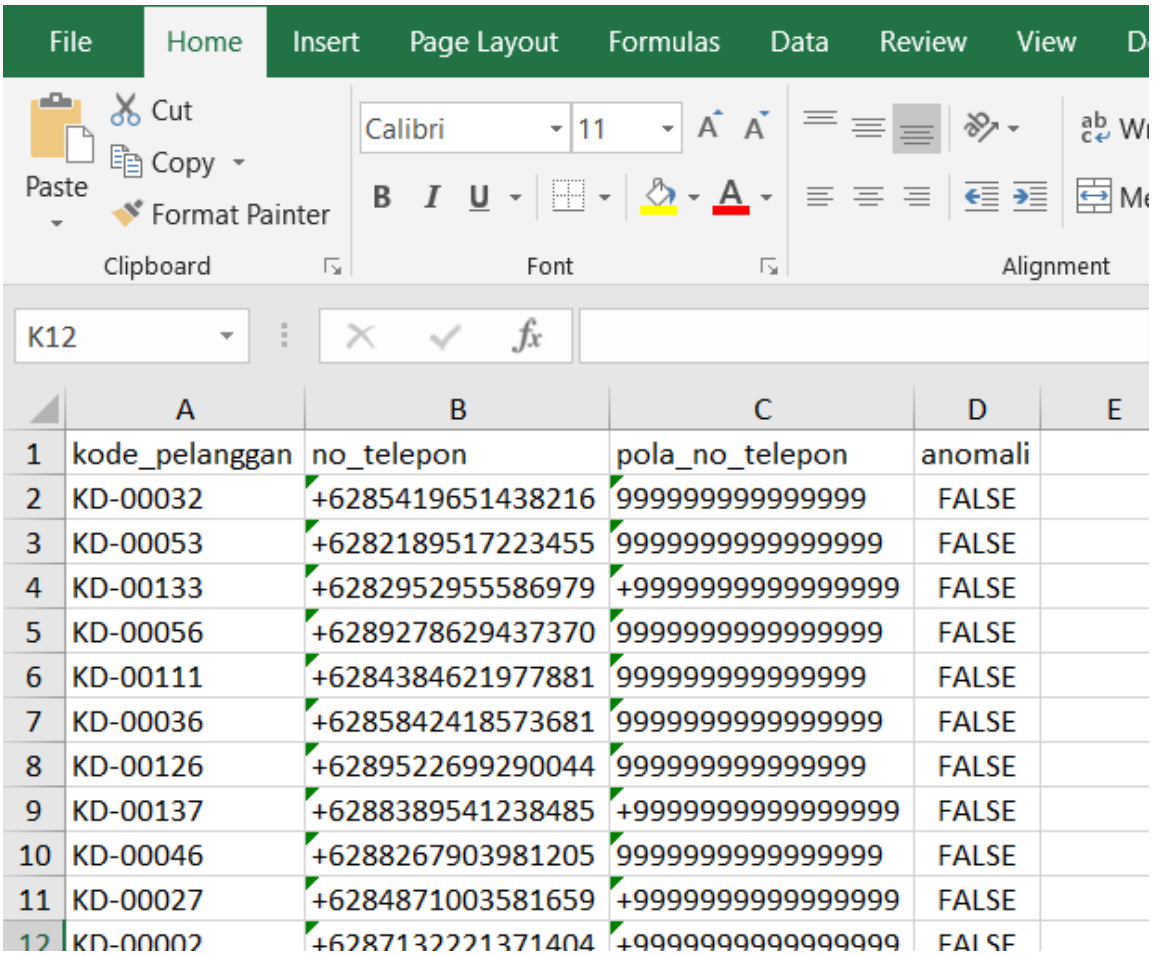
Catatan: Pada code juga sudah dilengkapi code untuk menambahkan kolom **anomali_no_telepon** yang bernilai TRUE/FALSE untuk menandai data mana dengan pola nomor telepon anomali.

Jika berhasil dijalankan dengan lancar, maka ada dua output yang akan kita peroleh:

- Tampilan console yang sebagian terlihat sebagai berikut.

```
   kode_pelanggan        no_telepon   pola_no_telepon anomali_no_telepon
1         KD-00032 +6285419651438216   999999999999999   FALSE
2         KD-00053 +6282189517223455  9999999999999999   FALSE
3         KD-00133 +6282952955586979 +9999999999999999   FALSE
4         KD-00056 +6289278629437370  9999999999999999   FALSE
5         KD-00111 +6284384621977881   999999999999999   FALSE
…
```

- File Excel "**staging.no_telepon.xlsx**" dengan sebagian tampilan ketika dibuka di aplikasi Excel terlihat seperti di bawah ini.



	A	B	C	D	E
1	kode_pelanggan	no_telepon	pola_no_telepon	anomali	
2	KD-00032	+6285419651438216	999999999999999	FALSE	
3	KD-00053	+6282189517223455	9999999999999999	FALSE	
4	KD-00133	+6282952955586979	+9999999999999999	FALSE	
5	KD-00056	+6289278629437370	9999999999999999	FALSE	
6	KD-00111	+6284384621977881	999999999999999	FALSE	
7	KD-00036	+6285842418573681	9999999999999999	FALSE	
8	KD-00126	+6289522699290044	999999999999999	FALSE	
9	KD-00137	+6288389541238485	+9999999999999999	FALSE	
10	KD-00046	+6288267903981205	9999999999999999	FALSE	
11	KD-00027	+6284871003581659	+9999999999999999	FALSE	
12	KD-00002	+6287132221371404	+9999999999999999	FALSE	

Code Editor

```
library(RMySQL)

library(openxlsx)


#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo",
host="mysqlhost",dbname="dqlabdatawrangling")


#Konstruksi SQL untuk Profil 1

sql <- "select kode_pelanggan, no_telepon, pola_no_telepon from dqlab_messy_data"
```

```
#Mengirimkan query untuk standarisasi no_telepon
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data untuk standarisasi no_telepon
data.telepon <- fetch(rs, n=-1)
data.telepon$anomali_no_telepon <- TRUE

data.telepon[data.telepon$pola_no_telepon=="9999999999999999",]$no_telepon <-
paste("+",
data.telepon[data.telepon$pola_no_telepon=="9999999999999999",]$no_telepon,
sep="")
data.telepon[data.telepon$pola_no_telepon=="999999999999999",]$no_telepon <-
gsub("^0","+62",
data.telepon[data.telepon$pola_no_telepon=="999999999999999",]$no_telepon)
data.telepon[data.telepon$pola_no_telepon=="+9999999999999999",]$anomali_no_tel
epon <- FALSE
data.telepon[data.telepon$pola_no_telepon=="9999999999999999",]$anomali_no_tele
pon <- FALSE
data.telepon[data.telepon$pola_no_telepon=="999999999999999",]$anomali_no_telep
on <- FALSE
print(data.telepon)

write.xlsx(file="staging.no_telepon.xlsx", x=data.telepon)

#Clear resultset untuk standarisasi
dbClearResult(rs)

#Menutup Koneksi
all_cons <- dbListConnections(MySQL())
for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> library(openxlsx)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dq
labdatawrangling")

> #Konstruksi SQL untuk Profil 1
> sql <- "select kode_pelanggan, no_telepon, pola_no_telepon from dqlab_messy_data"

> #Mengirimkan query untuk standarisasi no_telepon
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data untuk standarisasi no_telepon
> data.telepon <- fetch(rs, n=-1)

> data.telepon$anomali_no_telepon <- TRUE

> data.telepon[data.telepon$pola_no_telepon=="9999999999999999",]$no_telepon <- paste
("+", data.telepon[data.telepon$pola_no_telepon=="9999999999999999",]$no_telepon, sep
="")

> data.telepon[data.telepon$pola_no_telepon=="999999999999999",]$no_telepon <- gsub("
^0","+62",  data.telepon[data.telepon$pola_no_telepon=="999999999999999",]$no_telepon
)

> data.telepon[data.telepon$pola_no_telepon=="+9999999999999999",]$anomali_no_telepon
<- FALSE

> data.telepon[data.telepon$pola_no_telepon=="9999999999999999",]$anomali_no_telepon
<- FALSE

> data.telepon[data.telepon$pola_no_telepon=="999999999999999",]$anomali_no_telepon <
- FALSE

> print(data.telepon)
    kode_pelanggan        no_telepon   pola_no_telepon anomali_no_telepon
1         KD-00032 +6285419651438216   999999999999999              FALSE
2         KD-00053 +6282189517223455  9999999999999999              FALSE
3         KD-00133 +6282952955586979 +9999999999999999              FALSE
4         KD-00056 +6289278629437370  9999999999999999              FALSE
5         KD-00111 +6284384621977881   999999999999999              FALSE
6         KD-00036 +6285842418573681  9999999999999999              FALSE
7         KD-00126 +6289522699290044   999999999999999              FALSE
8         KD-00137 +6288389541238485 +9999999999999999              FALSE
9         KD-00046 +6288267903981205  9999999999999999              FALSE
10        KD-00027 +6284871003581659 +9999999999999999              FALSE
11        KD-00002 +6287132221371404 +9999999999999999              FALSE
12        KD-00075 +6283309536733507   999999999999999              FALSE
13        KD-00076 +6286815308308264 +9999999999999999              FALSE
```

14	KD-00035	+6286725681847845	9999999999999999	FALSE
15	KD-00113	+6281413705348345	9999999999999999	FALSE
16	KD-00099	+6281729600654645	+9999999999999999	FALSE
17	KD-00132	+6282352225142570	+9999999999999999	FALSE
18	KD-00088	+6283203183708137	9999999999999999	FALSE
19	KD-00119	+6289176501199576	9999999999999999	FALSE
20	KD-00096	+6286210781145764	999999999999999	FALSE
21	KD-00139	+6285986817540683	+9999999999999999	FALSE
22	KD-00090	+6287066745737382	+9999999999999999	FALSE
23	KD-00074	+6281902807450191	999999999999999	FALSE
24	KD-00021	+6285991672131933	9999999999999999	FALSE
25	KD-00045	+6282607473168157	+9999999999999999	FALSE
26	KD-00012	+62829891111222	+99999999999999	TRUE
27	KD-00030	+6282261101749552	+9999999999999999	FALSE
28	KD-00129	+6289323214692782	9999999999999999	FALSE
29	KD-00122	+6286663398617904	999999999999999	FALSE
30	KD-00059	+6283468728620812	999999999999999	FALSE
31	KD-00079	+6284927709580269	9999999999999999	FALSE
32	KD-00134	+6284094392278758	999999999999999	FALSE
33	KD-00064	+6285526151431004	9999999999999999	FALSE
34	KD-00038	+6286621940809359	999999999999999	FALSE
35	KD-00117	+6283166638654813	9999999999999999	FALSE
36	KD-00010	+6284079659289143	9999999999999999	FALSE
37	KD-00028	+6289311313046417	+9999999999999999	FALSE
38	KD-00125	+6286353637542265	+9999999999999999	FALSE
39	KD-00069	+6281298730359784	999999999999999	FALSE
40	KD-00114	+6284122970381517	+9999999999999999	FALSE
41	KD-00062	+6286916223612856	+9999999999999999	FALSE
42	KD-00006	+6284063423953696	9999999999999999	FALSE
43	KD-00024	+6281718632538241	999999999999999	FALSE
44	KD-00084	+6286837329291803	999999999999999	FALSE
45	KD-00104	+6286401899308998	9999999999999999	FALSE
46	KD-00103	+6283481690089399	+9999999999999999	FALSE
47	KD-00143	+6281672571203724	9999999999999999	FALSE
48	KD-00034	+6284588563149814	+9999999999999999	FALSE
49	KD-00087	+6285318844151067	9999999999999999	FALSE
50	KD-00039	+6289122766908102	9999999999999999	FALSE
51	KD-0047	+6282793268821143	+9999999999999999	FALSE
52	KD-00149	+6289337617505007	+9999999999999999	FALSE
53	KD-00003	+6285725955303368	999999999999999	FALSE
54	KD-00043	+6285158186394886	9999999999999999	FALSE
55	KD-00135	+6283674655321990	999999999999999	FALSE
56	KD-00050	+6283594524411404	+9999999999999999	FALSE
57	KD-00110	+6288942588082822	+9999999999999999	FALSE
58	KD-00049	+6284311691840121	+9999999999999999	FALSE
59	KD-00141	+6286730629494828	9999999999999999	FALSE
60	KD-00044	+6285879131063825	+9999999999999999	FALSE
61	KD-00124	+6284366427534780	9999999999999999	FALSE
62	KD-00105	+6288507258756263	999999999999999	FALSE
63	KD-00107	+6282792175097533	+9999999999999999	FALSE
64	KD-00086	+6281334304509664	9999999999999999	FALSE
65	KD-00123	+6286051245623557	999999999999999	FALSE
66	KD-00025	+6286035230854391	999999999999999	FALSE
67	KD-00008	+6285312577710538	9999999999999999	FALSE
68	KD-00005	+6286843623971825	9999999999999999	FALSE

69	KD-00101	+6285375019511143	+9999999999999999	FALSE
70	KD-00001	08298911112222	99999999999999	TRUE
71	KD-00020	+6287384329533477	999999999999999	FALSE
72	KD-00080	+6284032125604618	+9999999999999999	FALSE
73	KD-00102	+6281941958971086	+9999999999999999	FALSE
74	KD-00146	+6288888862370254	999999999999999	FALSE
75	KD-00048	+6285317681095918	+9999999999999999	FALSE
76	KD-00019	+6289317147992822	999999999999999	FALSE
77	KD-00151	+6287896807815060	999999999999999	FALSE
78	KD-00130	+6282833816760984	999999999999999	FALSE
79	KD-00073	+6281859313870200	+9999999999999999	FALSE
80	KD-00778	+62829891112222	+99999999999999	TRUE
81	KD-00066	-	-	TRUE
82	KD-00041	08763322558899	99999999999999	TRUE
83	KD-00140	+6289699357035892	999999999999999	FALSE
84	KD-00116	+6287642929298977	999999999999999	FALSE
85	KD-00127	+6284991627085550	+9999999999999999	FALSE
86	KD-00057	+6286996345317721	9999999999999999	FALSE
87	KD-00016	+6289222405928430	999999999999999	FALSE
88	KD-00063	+6285463027900499	9999999999999999	FALSE
89	KD-00148	+6289756523291187	+9999999999999999	FALSE
90	KD-00023	+6287660464098623	999999999999999	FALSE
91	KD-00029	+6283177123456315	9999999999999999	FALSE
92	KD-00136	+6288841308560422	999999999999999	FALSE
93	KD-00106	+6283460823430150	+9999999999999999	FALSE
94	KD-00026	+6284204043307629	+9999999999999999	FALSE
95	KD-00145	+6281980423349356	999999999999999	FALSE
96	KD-00018	+6282283957103749	999999999999999	FALSE
97	KD-00058	+6289503422652894	+9999999999999999	FALSE
98	KD-00051	+6283835679381969	+9999999999999999	FALSE
99	KD-00144	+6287642929298977	999999999999999	FALSE
100	KD-00128	0898198765432	9999999999999	TRUE
101	KD-00115	08765439876543	99999999999999	TRUE
102	KD-00009	+6282722234294686	999999999999999	FALSE
103	KD-00092	+6284298240961859	999999999999999	FALSE
104	KD-00070	+6281950071656111	+9999999999999999	FALSE
105	KD-00118	+6281693345459608	+9999999999999999	FALSE
106	KD-00052	+6282695676827512	+9999999999999999	FALSE
107	KD-00120	+6285239934324639	+9999999999999999	FALSE
108	KD-00055	+6288385590443770	+9999999999999999	FALSE
109	KD-00089	+6281550391417945	999999999999999	FALSE
110	KD-00042	+6284399241602502	+9999999999999999	FALSE
111	KD-00112	+6285734298900666	999999999999999	FALSE
112	KD-00098	+6283382626807712	9999999999999999	FALSE
113	KD-00033	+6286734992308497	999999999999999	FALSE
114	KD-00013	+6282672925000608	9999999999999999	FALSE
115	KD-00138	+6286357357965169	+9999999999999999	FALSE
116	KD-00094	+6284941125391866	999999999999999	FALSE
117	KD-00054	+6288743246116630	+9999999999999999	FALSE
118	KD-00100	+6282208807303229	+9999999999999999	FALSE
119	KD-00121	+6288508083942658	999999999999999	FALSE
120	KD-00061	+6283534357190274	+9999999999999999	FALSE
121	KD-00031	+6287382247200814	+9999999999999999	FALSE
122	KD-00040	+6287263432705516	+9999999999999999	FALSE
123	KD-00068	+6284941004806026	9999999999999999	FALSE

```
124       KD-00131 +6284939933374036 +9999999999999999              FALSE
125       KD-00097 +6282055715061873 +9999999999999999              FALSE
126       KD-00004 +6283376770990635  9999999999999999              FALSE
127       KD-00071 +6285361733615048 +9999999999999999              FALSE
128       KD-00093 +6287029784792141   999999999999999              FALSE
129       KD-00082 +6284338493742386   999999999999999              FALSE
130       KD-00150 +6287188198226353  9999999999999999              FALSE
131       KD-00065 +6287500842511771   999999999999999              FALSE
132       KD-00067 +6286546368604671 +9999999999999999              FALSE
133       KD-00011 +6288339032314103   999999999999999              FALSE
134       KD-00091 +6288718681168878  9999999999999999              FALSE
135       KD-00147 +6282891052016637 +9999999999999999              FALSE
136       KD-00081 +6288590906353243  9999999999999999              FALSE
137       KD-00109 +6286240577462157   999999999999999              FALSE
138       KD-00072 +6288942438259785   999999999999999              FALSE
139       KD-00014 +6285455084014504  9999999999999999              FALSE
140       KD-00078 +6283670227924527  9999999999999999              FALSE
141       KD-00095 +6285736296760607   999999999999999              FALSE
142       KD-00022 +6285796817992325   999999999999999              FALSE
143       KD-00017 +6289984358708389   999999999999999              FALSE
144       KD-00037 +6283155468652762 +9999999999999999              FALSE
145       KD-00108 +6284037884325249  9999999999999999              FALSE
146       KD-00015 +6282989111122220   999999999999999              FALSE
147       KD-00083 +6282989111122220   999999999999999              FALSE
148       KD-00060 +6286106166597558   999999999999999              FALSE
149       KD-00007 +6283840529196797   999999999999999              FALSE
150       KD-00077 +6283957775331152   999999999999999              FALSE
151       KD-00085 +6289781665737911   999999999999999              FALSE
152       KD-00142 +6289859935888974 +9999999999999999              FALSE
153       KD-00192 +6281729600654645 +9999999999999999              FALSE
154       KD-00298 +6286815308308264 +9999999999999999              FALSE
155       KD-00492 +6285879131063825 +9999999999999999              FALSE

> write.xlsx(file="staging.no_telepon.xlsx", x=data.telepon)

> #Clear resultset untuk standarisasi
> dbClearResult(rs)
[1] TRUE

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Profiling kolom Kode Pos (1)

Kolom selanjutnya yang akan kita profiling adalah kode_pos. Teknik profilenya sama dengan proses profiling **no_telepon** pada subbab **"**Menganalisa Profil kolom Nomor Telepon (1)".

Kita langsung masuk ke praktek saja untuk melakukan teknik ini.

Tugas Praktek

Gunakan SQL grouping dengan contoh untuk melakukan profiling terhadap kolom kode_pos dengan memanfaatkan kolom pola_kode_pos.

Isi SQL tersebut untuk menggantikan bagian […] pada code editor.

Jika berhasil dijalankan, maka diantara output yang dihasilkan terdapat teks berikut.

```
pola_kode_pos panjang_text jumlah_data
1             -            1           5
2        999999            6         147
3        99999A            6           1
4        9999A9            6           2
```

Terlihat selain ada 5 data yang diisi dengan tanda minus (-). Data lainnya sama panjang semua, namun pola 3 dan 4 berisi anomali.

Kode pos pada contoh kita harusnya berisi 6 digit angka semua seperti terlihat pada pola no 2. Namun pada pola nomor 3 dan 4, di antara angka terdapat huruf. Ini yang akan kita lihat isi datanya pada praktek selanjutnya.

Code Editor

```
library(RMySQL)

#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
                 dbname="dqlabdatawrangling")


#Konstruksi SQL

sql <- "SELECT pola_kode_pos, length(pola_kode_pos) as panjang_text, count(*) as
jumlah_data

from dqlab_messy_data
```

```
group by pola_kode_pos"

#Mengirimkan query
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data
profil_kode_pos <- fetch(rs, n=-1)
print(profil_kode_pos)

dbClearResult(rs)

#Menutup Koneksi
all_cons <- dbListConnections(MySQL())
for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> #Konstruksi SQL
> sql <- "SELECT pola_kode_pos, length(pola_kode_pos) as panjang_text, count(*) as ju
mlah_data
+ from dqlab_messy_data
+ group by pola_kode_pos"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> profil_kode_pos <- fetch(rs, n=-1)

> print(profil_kode_pos)
  pola_kode_pos panjang_text jumlah_data
1             -            1           5
2        999999            6         147
```

```
3        99999A            6           1
4        9999A9            6           2

> dbClearResult(rs)
[1] TRUE

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Profiling kolom Kode Pos (2)

Teks '99999A' dan '9999A9' adalah dua pola anomali yang kita temukan pada praktek di subbab sebelumnya. Dari pola ini terlihat ada huruf diantara angka. Harusnya kode pos terdiri dari angka semua.

Ok, kita sudah tau pola. Apa isinya? Dengan mengetahui isi kita bisa mengetahui apa yang perlu diganti.

Untuk dua teks ini kita bisa menggunakan perintah SELECT… WHERE… IN, seperti berikut.

```
SELECT kode_pos, pola_kode_pos from dqlab_messy_data where
pola_kode_pos in ('99999A', '9999A9')"
```

Dimana dengan menggunakan operator **IN** kita bisa filter daftar nilai yang terdapat pada kolom pola_kode_pos.

Tugas Praktek

Gunakan code yang sesuai untuk menggantikan […1...] dan […2...] pada code editor untuk mendaptakan hasil berikut.

```
  kode_pelanggan kode_pos pola_kode_pos
1       KD-00093   967220        99999A
2       KD-00083   8765I1        9999A9
3       KD-00085   98760I        9999A9
```

Terlihat terdapat huruf O, yang harusnya 0 (nol) untuk kode_pelanggan = 'KD-00093' dan untuk kode_pelanggan = 'KD-00085'. Setelah itu terdapat huruf I yang mungkin maksudnya adalah angka 1.

Pada tahap selanjutnya kita coba gantikan dulu dengan asumsi kita, O menjadi 0. Dan I menjadi 1.

Catatan: jangan copy paste SQL dari contoh pada Lesson, perhatikan hasil eksekusinya dari output di atas.

Code Editor

```
library(RMySQL)

#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
                 dbname="dqlabdatawrangling")
```

#Konstruksi SQL

sql <- "SELECT kode_pelanggan, kode_pos, pola_kode_pos from dqlab_messy_data where pola_kode_pos in ('99999A', '9999A9')"

#Mengirimkan query

rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data

data_kode_pos <- fetch(rs, n=-1)

print(data_kode_pos)

#Clear resultset

dbClearResult(rs)

#Menutup Koneksi

all_cons <- dbListConnections(MySQL())

for(con in all_cons) dbDisconnect(con)

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> #Konstruksi SQL
> sql <- "SELECT kode_pelanggan, kode_pos, pola_kode_pos from dqlab_messy_data where
pola_kode_pos in ('99999A', '9999A9')"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> data_kode_pos <- fetch(rs, n=-1)
```

```
> print(data_kode_pos)
  kode_pelanggan kode_pos pola_kode_pos
1       KD-00093   96722O        99999A
2       KD-00083   8765I1        9999A9
3       KD-00085   9876O1        9999A9

> #Clear resultset
> dbClearResult(rs)
[1] TRUE

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Standarisasi kolom Kode Pos (1)

Untuk standarisasi kolom kode pos pada dataset kita tidak ada hal yang khusus, hanya mengganti apa yang salah – dan kebetulan minor.

```
  kode_pelanggan kode_pos pola_kode_pos
1       KD-00093   967220O        99999A
2       KD-00083   8765I1         9999A9
3       KD-00085   9876O1         9999A9
```

Kembali ke hasil profiling di atas, maka *action item* kita adalah:

- Mengganti huruf O menjadi 0.
- Mengganti huruf I menjadi 1.

Dan untuk ini cukup menggunakan function **gsub** seperti yang sudah kita praktekkan beberapa kali. Sebagai contoh, untuk mengganti huruf O menjadi 0 pada data frame data_kode_pos$kode_pos adalah sebagai berikut.

```
    data_kode_pos$kode_pos <-
gsub("O","0",  data_kode_pos$kode_pos)
```

Mari kita langsung jalankan tugas berikut.

Tugas Praktek

Masukkan dua perintah gsub – masing-masing untuk menggati O menjadi 0, dan I menjadi 1 – untuk untuk menggantikan bagian […1…] dan […2…] pada code editor.

Jika berhasil, maka sebagian hasil output akan tampak seperti di bawah ini.

```
kode_pelanggan kode_pos pola_kode_pos
1       KD-00093   967220         99999A
2       KD-00083   876511         9999A9
3       KD-00085   987601         9999A9
```

Terlihat huruf O dan I telah digantikan dengan huruf semua. Dan pola_kode_pos di sampingnya hanya menunjukkan kondisi pola sebelum pergantian.

Code Editor

```
library(RMySQL)

#Membuka koneksi
```

```
con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
                 dbname="dqlabdatawrangling")
#Konstruksi SQL
sql <- "SELECT kode_pelanggan, kode_pos, pola_kode_pos from dqlab_messy_data
where pola_kode_pos in ('99999A', '9999A9')"

#Mengirimkan query
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data
data_kode_pos <- fetch(rs, n=-1)

#Merubah nilai O dan I
data_kode_pos$kode_pos <- gsub("O","0",  data_kode_pos$kode_pos)
data_kode_pos$kode_pos <- gsub("I","1",  data_kode_pos$kode_pos)
print(data_kode_pos)

#Clear resultset
dbClearResult(rs)

#Menutup Koneksi
all_cons <- dbListConnections(MySQL())
for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> #Konstruksi SQL
> sql <- "SELECT kode_pelanggan, kode_pos, pola_kode_pos from dqlab_messy_data where
pola_kode_pos in ('99999A', '9999A9')"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> data_kode_pos <- fetch(rs, n=-1)

> #Merubah nilai O dan I
> data_kode_pos$kode_pos <- gsub("O","0",  data_kode_pos$kode_pos)

> data_kode_pos$kode_pos <- gsub("I","1",  data_kode_pos$kode_pos)

> print(data_kode_pos)
  kode_pelanggan kode_pos pola_kode_pos
1       KD-00093   967220        99999A
2       KD-00083   876511        9999A9
3       KD-00085   987601        9999A9

> #Clear resultset
> dbClearResult(rs)
[1] TRUE

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Standarisasi kolom Kode Pos (2)

Seperti halnya pada kolom nama dan nomor telepon, pada praktek kali ini kita akan membaca seluruh data, melakukan pergantian (untuk seluruh data – karena O dan I tidak akan ditemukan di data lain – dan data masih cukup kecil) dan menulis ke staging file.

Klik menu icon "**Download Output File**" dan pelajari hasilnya.

Tugas Praktek

Ganti bagian […1…], […2…] dan […3…] dengan perintah yang sesuai. Hasil akhir adalah output dengan nama file "**staging.kode_pos.xlsx**".

Code Editor

```
library(RMySQL)

library(openxlsx)


#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",

                 dbname="dqlabdatawrangling")

#Konstruksi SQL

sql <- "SELECT kode_pelanggan, kode_pos, pola_kode_pos from dqlab_messy_data"


#Mengirimkan query

rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data

data_kode_pos <- fetch(rs, n=-1)


#Merubah nilai O menjadi 0 pada kolom kode_pos

data_kode_pos$kode_pos <- gsub("O","0",  data_kode_pos$kode_pos)

#Merubah nilai I menjadi 1 pada kolom kode_pos

data_kode_pos$kode_pos <- gsub("I","1",  data_kode_pos$kode_pos)
```

```
print(data_kode_pos)

#Menulis data ke file

write.xlsx(file="staging.kode_pos.xlsx", x=data_kode_pos)


#Clear resultset

dbClearResult(rs)


#Menutup Koneksi

all_cons <- dbListConnections(MySQL())

for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> library(openxlsx)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> #Konstruksi SQL
> sql <- "SELECT kode_pelanggan, kode_pos, pola_kode_pos from dqlab_messy_data"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> data_kode_pos <- fetch(rs, n=-1)

> #Merubah nilai O menjadi 0 pada kolom kode_pos
> data_kode_pos$kode_pos <- gsub("O","0",  data_kode_pos$kode_pos)

> #Merubah nilai I menjadi 1 pada kolom kode_pos
> data_kode_pos$kode_pos <- gsub("I","1",  data_kode_pos$kode_pos)

> print(data_kode_pos)
    kode_pelanggan kode_pos pola_kode_pos
1         KD-00032   567130        999999
2         KD-00053   567130        999999
3         KD-00133   567130        999999
4         KD-00056   876551        999999
5         KD-00111   876551        999999
```

6	KD-00036	876552	999999
7	KD-00126	876552	999999
8	KD-00137	877521	999999
9	KD-00046	877521	999999
10	KD-00027	877521	999999
11	KD-00002	712983	999999
12	KD-00075	712983	999999
13	KD-00076	712984	999999
14	KD-00035	712984	999999
15	KD-00113	712984	999999
16	KD-00099	712984	999999
17	KD-00132	633429	999999
18	KD-00088	633429	999999
19	KD-00119	633430	999999
20	KD-00096	633431	999999
21	KD-00139	511431	999999
22	KD-00090	511431	999999
23	KD-00074	511432	999999
24	KD-00021	511432	999999
25	KD-00045	876511	999999
26	KD-00012	876511	999999
27	KD-00030	349922	999999
28	KD-00129	986454	999999
29	KD-00122	986455	999999
30	KD-00059	-	-
31	KD-00079	986456	999999
32	KD-00134	986456	999999
33	KD-00064	987451	999999
34	KD-00038	987452	999999
35	KD-00117	987452	999999
36	KD-00010	987453	999999
37	KD-00028	987453	999999
38	KD-00125	-	-
39	KD-00069	349981	999999
40	KD-00114	349981	999999
41	KD-00062	487451	999999
42	KD-00006	487851	999999
43	KD-00024	811613	999999
44	KD-00084	811613	999999
45	KD-00104	811613	999999
46	KD-00103	877613	999999
47	KD-00143	877614	999999
48	KD-00034	877615	999999
49	KD-00087	764449	999999
50	KD-00039	764449	999999
51	KD-0047	764450	999999
52	KD-00149	764450	999999
53	KD-00003	764550	999999
54	KD-00043	764550	999999
55	KD-00135	876612	999999
56	KD-00050	321321	999999
57	KD-00110	321321	999999
58	KD-00049	321321	999999
59	KD-00141	321321	999999
60	KD-00044	321321	999999

61	KD-00124	321321	999999
62	KD-00105	321321	999999
63	KD-00107	893422	999999
64	KD-00086	813442	999999
65	KD-00123	813442	999999
66	KD-00025	813442	999999
67	KD-00008	813444	999999
68	KD-00005	476511	999999
69	KD-00101	476511	999999
70	KD-00001	876511	999999
71	KD-00020	476533	999999
72	KD-00080	476533	999999
73	KD-00102	666122	999999
74	KD-00146	666123	999999
75	KD-00048	866162	999999
76	KD-00019	-	-
77	KD-00151	876612	999999
78	KD-00130	876614	999999
79	KD-00073	876512	999999
80	KD-00778	876511	999999
81	KD-00066	896549	999999
82	KD-00041	896549	999999
83	KD-00140	896549	999999
84	KD-00116	986455	999999
85	KD-00127	896549	999999
86	KD-00057	896550	999999
87	KD-00016	896550	999999
88	KD-00063	768091	999999
89	KD-00148	768091	999999
90	KD-00023	-	-
91	KD-00029	896566	999999
92	KD-00136	896555	999999
93	KD-00106	896555	999999
94	KD-00026	896555	999999
95	KD-00145	896555	999999
96	KD-00018	896555	999999
97	KD-00058	896555	999999
98	KD-00051	696193	999999
99	KD-00144	986455	999999
100	KD-00128	986455	999999
101	KD-00115	986455	999999
102	KD-00009	896555	999999
103	KD-00092	696193	999999
104	KD-00070	696193	999999
105	KD-00118	696193	999999
106	KD-00052	567120	999999
107	KD-00120	567120	999999
108	KD-00055	696193	999999
109	KD-00089	696193	999999
110	KD-00042	696193	999999
111	KD-00112	696193	999999
112	KD-00098	696193	999999
113	KD-00033	666122	999999
114	KD-00013	666122	999999
115	KD-00138	896549	999999

```
116       KD-00094   896549        999999
117       KD-00054   896549        999999
118       KD-00100   896549        999999
119       KD-00121   896112        999999
120       KD-00061   896113        999999
121       KD-00031   896114        999999
122       KD-00040   896115        999999
123       KD-00068   567151        999999
124       KD-00131   567151        999999
125       KD-00097   567120        999999
126       KD-00004   967220        999999
127       KD-00071   967220        999999
128       KD-00093   967220        99999A
129       KD-00082   967221        999999
130       KD-00150   967221        999999
131       KD-00065   967222        999999
132       KD-00067   967223        999999
133       KD-00011   967223        999999
134       KD-00091   967223        999999
135       KD-00147   967224        999999
136       KD-00081   967229        999999
137       KD-00109   967229        999999
138       KD-00072   817321        999999
139       KD-00014        -             -
140       KD-00078   817324        999999
141       KD-00095   768031        999999
142       KD-00022   768031        999999
143       KD-00017   768034        999999
144       KD-00037   768034        999999
145       KD-00108   768035        999999
146       KD-00015   876511        999999
147       KD-00083   876511        9999A9
148       KD-00060   986455        999999
149       KD-00007   986455        999999
150       KD-00077   987601        999999
151       KD-00085   987601        9999A9
152       KD-00142   986455        999999
153       KD-00192   712984        999999
154       KD-00298   712984        999999
155       KD-00492   321321        999999

> #Menulis data ke file
> write.xlsx(file="staging.kode_pos.xlsx", x=data_kode_pos)

> #Clear resultset
> dbClearResult(rs)
[1] TRUE

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Profiling kolom Alamat

Kolom alamat sebenarnya cukup sulit diprofile. Tapi pastinya harus memiliki karakteristik dimana kolom ini merupakan campuran huruf dan angka dengan mayoritas berupa huruf.

Berbasiskan karakteristik tersebut kita bisa cari anomali selain isi yang kosong, alamat juga tidak boleh terdiri dari huruf (plus spasi) semua dan angka (plus spasi) semua.

Karena kita ada dua kolom yang berkaitan dengan alamat, yaitu kolom alamat dan pola_alamat. Kedua-duanya bisa digunakan untuk filtering dengan REGEX untuk karakteristik di atas, tapi lebih mudah dengan menggunakan pola_regex.

Jika kita gunakan regex untuk kolom **alamat** maka kita gunakan:

- ^[A-Za-z]+$
- ^[0-9]+$

Jika kita gunakan regex untuk kolom **pola_alamat** maka kita gunakan:

- ^[aAw]+$
- ^[9w]+$

Catatan: Tanda topi (^) di awal pola regex dan $ (dollar) di akhir pola regex adalah penanda bahwa pola berlaku dari awal sampai akhir teks.

Tugas Praktek

Masukkan dua perintah gsub – masing-masing untuk untuk menggantikan bagian […1…] dan […2…] pada code editor.

Catatan: perhatikan kolom yang dipakai pada potongan code editor.

Jika berhasil, maka sebagian hasil output akan tampak sebagai berikut.

```
[1] kode_pelanggan alamat         pola_alamat
<0 rows> (or 0-length row.names)
```

Ini artinya tidak ada pola yang terdiri dari karakter semua atau angka semua. Tahap berikutnya hanya melakukan standarisasi yang kita perlukan, misalkan singkatan "jln." menjadi "jalan ".

Code Editor

```
library(RMySQL)

#Membuka koneksi
```

```
con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
                 dbname="dqlabdatawrangling")
#Konstruksi SQL
sql <- "SELECT kode_pelanggan, alamat, pola_alamat from dqlab_messy_data where
pola_alamat REGEXP '^[aAw]+$' or pola_alamat REGEXP '^[9w]+$'"

#Mengirimkan query
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data
data_alamat <- fetch(rs, n=-1)
print(data_alamat)

#Clear resultset
dbClearResult(rs)

#Menutup Koneksi
all_cons <- dbListConnections(MySQL())
for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> #Konstruksi SQL
> sql <- "SELECT kode_pelanggan, alamat, pola_alamat from dqlab_messy_data where pola
_alamat REGEXP '^[aAw]+$' or pola_alamat REGEXP '^[9w]+$'"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"
```

```
> #Mengambil data
> data_alamat <- fetch(rs, n=-1)

> print(data_alamat)
[1] kode_pelanggan alamat         pola_alamat
<0 rows> (or 0-length row.names)

> #Clear resultset
> dbClearResult(rs)
[1] TRUE

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Standarisasi kolom Alamat

Penulisan alamat dengan sistem yang paling kaku sekalipun biasanya harus memiliki input free text dimana user bisa bebas melakukan input.

Akitabnya banyak penulisan yang tidak standar, misalkan untuk "jalan" menjadi "jl" dan "jln" seperti terlihat pada sebagian dataset kita sebagai berikut.

```
Jl. Pulo Bambu No. 15, Kota Tenggara Lama
Jln. Tegal Sari Indah, No. D87 -- Kota H
Jalan Hang Tuah, No. 11, Kota DM
Jl. Puri Arteri Raya, No. 88 - Kota T
Jl. Pahlawan, No. 69CCD
Jl. Bintang Supernova, No. 78
Jl. Wisma Tenteram Saja, No. A22
Jln. Manggis II, Gang Buntu No. 1
Jalan. Kebon Jahe, No. F16 - Kota E
```

Untuk kondisi ini, tidak ada jalan lain selain mengumpulkan daftar "kesalahan umum" (*common mistakes*) ini dan perbaiki berdasarkan daftar tersebut.

Dan untuk contoh di atas, kita akan ganti semua variasi singkatan di atas dengan "Jalan".

Pola regexnya adalah sebagai berikut:

```
jln[ ]*\\.
\\bjln\\b
jl[ ]*\\.
\\bjl\\b
jalan\\.
```

dimana

- \\b adalah penanda batas kata.
- \\. adalah penulisan titik.
- [ ]*\\. menunjukkan perulangan spasi yang mungkin mengikuti sebelum tanda titik.

Catatan: Pola ini hanya contoh untuk kasus kita, pada prakteknya Anda perlu mengumpulkan pola-pola ini untuk melakukan standarisasi.

Tugas Praktek

Dengan perintah **gsub** yang telah Anda pelajari sebelumnya, gunakan opsi **ignore.case = TRUE** dan lengkapi ganti bagian […1...], […2…], […3…], […4…] dan […5…] pada code editor dengan pola yang telah diberikan contohnya di atas.

Jika berjalan lancar, maka hasil output lengkap dari variable **data_alamat** akan terlihat sebagai berikut. Perhatikan jika tidak ada lagi singkatan "jalan".

	kode_pelanggan	alamat
1	KD-00032	Vila Sempilan, No. 67 - Kota B
2	KD-00053	Vila Sempilan, No. 11 - Kota B
3	KD-00133	Vila Sempilan, No. 1 - Kota B
4	KD-00056	Vila Permata Intan Berkilau, Blok C5-7
5	KD-00111	Vila Permata Intan Berkilau, Blok A1/2
6	KD-00036	Vila Gunung Seribu, Blok O1 - No. 1
7	KD-00126	Vila Gunung Seribu, Blok F4 - No. 8
8	KD-00137	Vila Bukit Sagitarius, Gang. Sawit No. 3
9	KD-00046	Vila Bukit Sagitarius, Gang Kelapa No. 6
10	KD-00027	Vila Bukit Sagitarius, Blok A1 No. 1
11	KD-00002	Taman Vivo Indah, Blok AA No. 7
12	KD-00075	Taman Vivo Indah, Blok AA No. 7
13	KD-00076	Taman Bunga Langit, Jalan Utara No. 3
14	KD-00035	Taman Bunga Langit, Jalan Timur No. 1
15	KD-00113	Taman Bunga Langit, Jalan Selatan No. 12
16	KD-00099	Taman Bunga Langit, Jalan Barat Laut No. 6
17	KD-00132	Rusun Kerinci Indah, Lt. 6 No. 1
18	KD-00088	Rusun Kerinci Indah, Lt. 5 No. 6
19	KD-00119	Rumah Susun Gelora, Lantai 1 No. 12
20	KD-00096	Rumah Susun Eunos, Lantai 2 No. 2
21	KD-00139	Ruko Azalea, No. 3 RT 001/002
22	KD-00090	Ruko Almond Manis, Blok C7/8
23	KD-00074	Puspa Loka, No. 98F, Kota Y
24	KD-00021	Puspa Loka, No. 98B, Kota Y
25	KD-00045	Pulo Bambu No. 57, Kota Tenggara Lama
26	KD-00012	Pulo Bambu No. 15, Kota Tenggara Lama
27	KD-00030	Pondok Bima Sakti, Jalan Asrama Pelajar No. 11FF

28	KD-00129	Perumahan Sektor Telekomunikasi, Jalan Afrika No. 3
29	KD-00122	Perumahan Sektor Bougenville, Jalan Sawit No. 8A
30	KD-00059	Perumahan Sektor Bougenville, Jalan Karet No. 7P
31	KD-00079	Perumahan Duku Satu, Gang Merpati - No. 41
32	KD-00134	Perumahan Duku Lima, Gang Perkutut No. 1
33	KD-00064	Perumahan Catalina, Jalan Kereta Api No. 77
34	KD-00038	Perumahan Bina Andromeda, Jalan Teri No. 4
35	KD-00117	Perumahan Bina Andromeda, Jalan Salmon No. 22
36	KD-00010	Perum Venus, Gg. Harimau No. 1A
37	KD-00028	Perum Venus, Gang. Kelinci No. 12
38	KD-00125	Perum Venus, Gang. Harimau No. 4A
39	KD-00069	Perum Titan, Jalan Trobos No. 8
40	KD-00114	Perum Titan, Jalan Kelinci No. 12
41	KD-00062	Perum Sektor 50, Gang Permai No. 5
42	KD-00006	Perum Pluto, Blok C No. 1
43	KD-00024	Perum Maju Permai Persada Indah, Gang Kenari No. 3
44	KD-00084	Perum Maju Permai P.I., Gang Kesturi No. 5
45	KD-00104	Perum Maju Permai P.I., Gang Kesturi No. 5
46	KD-00103	Perum Kali Meksiko, No. D22
47	KD-00143	Perum Kali Meksiko, No. 8F
48	KD-00034	Perum Kali Meksiko, No. 8C
49	KD-00087	Perum Indah Supernova, No. 1
50	KD-00039	Perum Indah Supernova II, No. 9
51	KD-0047	Perum Bimasakti Raya, Blok A No. 10
52	KD-00149	Perum Bimasakti Raya, Blok A No. 10
53	KD-00003	Meta Residences, No. 32C
54	KD-00043	Meta Residences, No. 1A
55	KD-00135	Kota T, Jalan Taman Kencana No. 11112
56	KD-00050	Kompleks Selatan-Selatan, No. 121
57	KD-00110	Kompleks Selatan-Selatan, No. 111
58	KD-00049	Kompleks Permai Angkasa, Blok M No. 10
59	KD-00141	Kompleks Permai Angkasa, Blok J No. 09
60	KD-00044	Kompleks Pelaut Tangguh, No. 5A
61	KD-00124	Kompleks Nelayan Permai, Blok DD - 98/99

62	KD-00105	Kompleks Akademi Perawat, Gang Farmasi No. 3
63	KD-00107	Kampung Kijang, Blok D3 - No. 12
64	KD-00086	Kampung Harimau, No. 88, Kota K
65	KD-00123	Kampung Harimau, No. 3
66	KD-00025	Kampoeng Harimau, No. 81 - Kota K
67	KD-00008	Kali Mars Cluster, No. 24C
68	KD-00005	Jalan Tegal Sari Indah, No. D87 -- Kota H
69	KD-00101	Jalan Tegal Sari Indah, No. D77 -- Kota H
70	KD-00001	Jalan Pulo Bambu No. 15, Kota Tenggara Lama
71	KD-00020	Jalan Manggis II, Gang Buntu No. 1
72	KD-00080	Jalan Manggis II - Gang Buntu No. 4
73	KD-00102	Jalan Kangguru No. 92, RT 005 - kota R
74	KD-00146	Jalan G. Asri Mawar Harum Blok G No. 9
75	KD-00048	Jalan Wisma Tenteram Saja, No. A31
76	KD-00019	Jalan Wisma Tenteram Saja, No. A22
77	KD-00151	Jalan Taman Kencana No. 11112, Kota T
78	KD-00130	Jalan Raya Griya Barbarosa, Blok AF 789
79	KD-00073	Jalan Puri Indah Menawan, No. 818 - Kota T
80	KD-00778	Jalan Pulau Bambu No. 15 - Kota Tenggara Lama
81	KD-00066	Jalan Pulau Sentosa No. 133
82	KD-00041	Jalan Pulau Sentosa No. 133
83	KD-00140	Jalan Pulau Sentosa No. 1335
84	KD-00116	Apartemen Lucky Beruntung, Lt. 5 No. 4
85	KD-00127	Jalan Pulau Sentosa No. 133
86	KD-00057	Jalan Pahlawan, No. 69FFF
87	KD-00016	Jalan Pahlawan, No. 69CCD
88	KD-00063	Jalan Macan Buntung, No. 4F
89	KD-00148	Jalan Macan Buntung, No. 1F - Kota D
90	KD-00023	Jalan Macan Buntung, No. 1F
91	KD-00029	Jalan Kp. Kijang, Blok A1 - No. 2F
92	KD-00136	Jalan Kemenangan Besar, Blok C8 No. 22 RT 02
93	KD-00106	Jalan Kemenangan Besar, Blok C8 No. 22
94	KD-00026	Jalan Kebon Jahe, Kota EntahDimana
95	KD-00145	Jalan Kampung Kijang, Blok C5 - No. 9

96	KD-00018	Jalan Bintang Supernova, No. 78
97	KD-00058	Jalan Bintang Supernova, No. 78
98	KD-00051	Jalan Binjai 200, Kota L
99	KD-00144	Apartemen Lucky Beruntung, Lt. 3 No. 4
100	KD-00128	Apartemen Bukit Merah, Annex Plaza, Lt 3 No. A1
101	KD-00115	Apartemen Bukit Merah Annex Plaza, Lt 3 No. A1
102	KD-00009	Jalan Kebon Jahe, No. F16 - Kota E
103	KD-00092	Jalan Bukit Tol Km. 3, No. 971
104	KD-00070	Jalan Wisma Tenteram Saja No. B-01
105	KD-00118	Jalan Semantik Semut Berjalan, No. 3333
106	KD-00052	Jalan Ring Road Neolitik, No. 1 RT 5
107	KD-00120	Jalan Ring Road Konstan, No. 5
108	KD-00055	Jalan Raya Jupiter Titan, No. 55
109	KD-00089	Jalan Raya Hang Lekir, No. 62 - Kota Z
110	KD-00042	Jalan Raya Hang Lekir, Kota Z, No. 62
111	KD-00112	Jalan Raya Andromeda, Blok D No. 3
112	KD-00098	Jalan Pesisir No. 5, Kampoeng Maju Surya Gemilang
113	KD-00033	Jalan Hang Tuah, No. 31, Kota DM
114	KD-00013	Jalan Hang Tuah, No. 11, Kota DM
115	KD-00138	Jalan Gula Pahit, No. 081
116	KD-00094	Jalan Gula Pahit, No. 015
117	KD-00054	Jalan Gula Pahit, No. 001
118	KD-00100	Jalan Asia No. 55, Kompleks Pelajar Kota C
119	KD-00121	Indah Mars Cluster, No. 22F
120	KD-00061	Griya Asri Mawar Harum, Blok G No. 1
121	KD-00031	Gang Tupai, No. 7 - Desa CL
122	KD-00040	Gang Samun Saja No. 132, Kode Pos A99222
123	KD-00068	Gang Piranha, No. 3 - Desa BT
124	KD-00131	Gang Piranha, No. 13 - Desa BT
125	KD-00097	Gang Kelinci, No. 666 - Kota B
126	KD-00004	Gang Bulan Desember III, No. 9
127	KD-00071	Gang Bulan Desember III, No. 155
128	KD-00093	Gang Bulan Desember III, No. 145
129	KD-00082	Gang Arwana, No. 6 - Kota S

130	KD-00150	Gang Arwana No. 12, Kota S
131	KD-00065	Corina Residences Apartment, No. 0612
132	KD-00067	Condominium Pesona Indah, No. 0708
133	KD-00011	Cluster Ikan Mas, Taman Intan No. 2
134	KD-00091	Cluster Ikan Mas, Taman Baru No. 96
135	KD-00147	Cluster Griya Bima Sakti, Blok A No. 1
136	KD-00081	Bukit Vivo Indah, Blok C 2/4
137	KD-00109	Bukit Vivo Indah, Blok C 2/4
138	KD-00072	Boulevard Raya Residences, Blok AB2 No. 102
139	KD-00014	Boulevard Raya Residences, Blok AA2 No. 88
140	KD-00078	Blok C 2/4, Bukit Vivo Indah
141	KD-00095	Asrama Perawat IV, No. 2 - Kota D
142	KD-00022	Asrama Perawat IV, No. 1 - Kota D
143	KD-00017	Asrama Pelajar No. 22 A - Pondok Bima Sakti
144	KD-00037	Asrama Pelajar No. 11 B - Pondok Bima Sakti
145	KD-00108	Apartement Clifften, Lantai 12 No. 3
146	KD-00015	Jalan Puri Arteri Raya, No. 88 - Kota T
147	KD-00083	Jalan Puri Arteri Raya, No. 88 - Kota T
148	KD-00060	Apartemen Kecapi Indah, Lt. 18 No. 1801
149	KD-00007	Apartemen Kecapi Indah, Lt. 16 No. 1610
150	KD-00077	Jalan Sutomo Baru 21 - Kota M
151	KD-00085	Jalan Sutomo Baru No. 21 - Kota M
152	KD-00142	Apartemen Bukit Baru, Dahlia Tower, No. A3
153	KD-00192	Taman Bunga Langit, Jalan Barat Laut No. 6
154	KD-00298	Taman Bunga Langit, Jalan Utara No. 3
155	KD-00492	Kompleks Pelaut Tangguh, No. 5A

Code Editor

```
library(RMySQL)
library(openxlsx)
#Membuka koneksi
con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
```

```
                dbname="dqlabdatawrangling")
#Konstruksi SQL
sql <- "SELECT kode_pelanggan, alamat from dqlab_messy_data"

#Mengirimkan query
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
#Mengambil data
data_alamat <- fetch(rs, n=-1)

#Merubah singkatan jl, jln, jl. dan jln. menjadi Jalan
data_alamat$alamat <- gsub("jln[ ]*\\.", "Jalan", data_alamat$alamat,
ignore.case=TRUE)
data_alamat$alamat <- gsub("\\bjln\\b", "Jalan", data_alamat$alamat,
ignore.case=TRUE)
data_alamat$alamat <- gsub("jl[ ]*\\.", "Jalan", data_alamat$alamat, ignore.case=TRUE)
data_alamat$alamat <- gsub("\\bjl\\b", "Jalan", data_alamat$alamat,
ignore.case=TRUE)
data_alamat$alamat <- gsub("jalan\\.", "Jalan", data_alamat$alamat,
ignore.case=TRUE)
print(data_alamat)
#Menulis data ke file
write.xlsx(file="staging.alamat.xlsx", x= data_alamat)

#Clear resultset
dbClearResult(rs)

#Menutup Koneksi
all_cons <- dbListConnections(MySQL())
for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> library(openxlsx)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> #Konstruksi SQL
> sql <- "SELECT kode_pelanggan, alamat from dqlab_messy_data"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> data_alamat <- fetch(rs, n=-1)

> #Merubah singkatan jl, jln, jl. dan jln. menjadi Jalan
> data_alamat$alamat <- gsub("jln[ ]*\\.", "Jalan", data_alamat$alamat, ignore.case=T
RUE)

> data_alamat$alamat <- gsub("\\bjln\\b", "Jalan", data_alamat$alamat, ignore.case=TR
UE)

> data_alamat$alamat <- gsub("jl[ ]*\\.", "Jalan", data_alamat$alamat, ignore.case=TR
UE)

> data_alamat$alamat <- gsub("\\bjl\\b", "Jalan", data_alamat$alamat, ignore.case=TRU
E)

> data_alamat$alamat <- gsub("jalan\\.", "Jalan", data_alamat$alamat, ignore.case=TRU
E)

> print(data_alamat)
    kode_pelanggan                                        alamat
1         KD-00032                    Vila Sempilan, No. 67 - Kota B
2         KD-00053                    Vila Sempilan, No. 11 - Kota B
3         KD-00133                     Vila Sempilan, No. 1 - Kota B
4         KD-00056             Vila Permata Intan Berkilau, Blok C5-7
5         KD-00111             Vila Permata Intan Berkilau, Blok A1/2
6         KD-00036                Vila Gunung Seribu, Blok O1 - No. 1
7         KD-00126                Vila Gunung Seribu, Blok F4 - No. 8
8         KD-00137           Vila Bukit Sagitarius, Gang. Sawit No. 3
9         KD-00046           Vila Bukit Sagitarius, Gang Kelapa No. 6
10        KD-00027               Vila Bukit Sagitarius, Blok A1 No. 1
11        KD-00002                    Taman Vivo Indah, Blok AA No. 7
12        KD-00075                    Taman Vivo Indah, Blok AA No. 7
13        KD-00076               Taman Bunga Langit, Jalan Utara No. 3
14        KD-00035               Taman Bunga Langit, Jalan Timur No. 1
15        KD-00113           Taman Bunga Langit, Jalan Selatan No. 12
16        KD-00099          Taman Bunga Langit, Jalan Barat Laut No. 6
```

```
17        KD-00132                    Rusun Kerinci Indah, Lt. 6 No. 1
18        KD-00088                    Rusun Kerinci Indah, Lt. 5 No. 6
19        KD-00119                 Rumah Susun Gelora, Lantai 1 No. 12
20        KD-00096                   Rumah Susun Eunos, Lantai 2 No. 2
21        KD-00139                       Ruko Azalea, No. 3 RT 001/002
22        KD-00090                     Ruko Almond Manis, Blok C7/8
23        KD-00074                      Puspa Loka, No. 98F, Kota Y
24        KD-00021                      Puspa Loka, No. 98B, Kota Y
25        KD-00045              Pulo Bambu No. 57, Kota Tenggara Lama
26        KD-00012              Pulo Bambu No. 15, Kota Tenggara Lama
27        KD-00030    Pondok Bima Sakti, Jalan Asrama Pelajar No. 11FF
28        KD-00129 Perumahan Sektor Telekomunikasi, Jalan Afrika No. 3
29        KD-00122    Perumahan Sektor Bougenville, Jalan Sawit No. 8A
30        KD-00059    Perumahan Sektor Bougenville, Jalan Karet No. 7P
31        KD-00079          Perumahan Duku Satu, Gang Merpati - No. 41
32        KD-00134            Perumahan Duku Lima, Gang Perkutut No. 1
33        KD-00064         Perumahan Catalina, Jalan Kereta Api No. 77
34        KD-00038          Perumahan Bina Andromeda, Jalan Teri No. 4
35        KD-00117       Perumahan Bina Andromeda, Jalan Salmon No. 22
36        KD-00010                     Perum Venus, Gg. Harimau No. 1A
37        KD-00028                   Perum Venus, Gang. Kelinci No. 12
38        KD-00125                   Perum Venus, Gang. Harimau No. 4A
39        KD-00069                     Perum Titan, Jalan Trobos No. 8
40        KD-00114                   Perum Titan, Jalan Kelinci No. 12
41        KD-00062                  Perum Sektor 50, Gang Permai No. 5
42        KD-00006                            Perum Pluto, Blok C No. 1
43        KD-00024  Perum Maju Permai Persada Indah, Gang Kenari No. 3
44        KD-00084          Perum Maju Permai P.I., Gang Kesturi No. 5
45        KD-00104          Perum Maju Permai P.I., Gang Kesturi No. 5
46        KD-00103                        Perum Kali Meksiko, No. D22
47        KD-00143                         Perum Kali Meksiko, No. 8F
48        KD-00034                         Perum Kali Meksiko, No. 8C
49        KD-00087                       Perum Indah Supernova, No. 1
50        KD-00039                    Perum Indah Supernova II, No. 9
51         KD-0047                 Perum Bimasakti Raya, Blok A No. 10
52        KD-00149                 Perum Bimasakti Raya, Blok A No. 10
53        KD-00003                           Meta Residences, No. 32C
54        KD-00043                            Meta Residences, No. 1A
55        KD-00135              Kota T,  Jalan Taman Kencana No. 11112
56        KD-00050                  Kompleks Selatan-Selatan, No. 121
57        KD-00110                   Kompleks Selatan-Selatan, No. 111
58        KD-00049              Kompleks Permai Angkasa, Blok M No. 10
59        KD-00141              Kompleks Permai Angkasa, Blok J No. 09
60        KD-00044                    Kompleks Pelaut Tangguh, No. 5A
61        KD-00124           Kompleks Nelayan Permai, Blok DD - 98/99
62        KD-00105        Kompleks Akademi Perawat, Gang Farmasi No. 3
63        KD-00107                    Kampung Kijang, Blok D3 - No. 12
64        KD-00086                    Kampung Harimau, No. 88, Kota K
65        KD-00123                           Kampung Harimau, No. 3
66        KD-00025                  Kampoeng Harimau, No. 81 - Kota K
67        KD-00008                         Kali Mars Cluster, No. 24C
68        KD-00005           Jalan Tegal Sari Indah, No. D87 -- Kota H
69        KD-00101           Jalan Tegal Sari Indah, No. D77 -- Kota H
70        KD-00001         Jalan Pulo Bambu No. 15, Kota Tenggara Lama
71        KD-00020                  Jalan Manggis II, Gang Buntu No. 1
```

72	KD-00080	Jalan Manggis II - Gang Buntu No. 4
73	KD-00102	Jalan Kangguru No. 92, RT 005 - kota R
74	KD-00146	Jalan G. Asri Mawar Harum Blok G No. 9
75	KD-00048	Jalan Wisma Tenteram Saja, No. A31
76	KD-00019	Jalan Wisma Tenteram Saja, No. A22
77	KD-00151	Jalan Taman Kencana No. 11112, Kota T
78	KD-00130	Jalan Raya Griya Barbarosa, Blok AF 789
79	KD-00073	Jalan Puri Indah Menawan, No. 818 - Kota T
80	KD-00778	Jalan Pulau Bambu No. 15 - Kota Tenggara Lama
81	KD-00066	Jalan Pulau Sentosa No. 133
82	KD-00041	Jalan Pulau Sentosa No. 133
83	KD-00140	Jalan Pulau Sentosa No. 1335
84	KD-00116	Apartemen Lucky Beruntung, Lt. 5 No. 4
85	KD-00127	Jalan Pulau Sentosa No. 133
86	KD-00057	Jalan Pahlawan, No. 69FFF
87	KD-00016	Jalan Pahlawan, No. 69CCD
88	KD-00063	Jalan Macan Buntung, No. 4F
89	KD-00148	Jalan Macan Buntung, No. 1F - Kota D
90	KD-00023	Jalan Macan Buntung, No. 1F
91	KD-00029	Jalan Kp. Kijang, Blok A1 - No. 2F
92	KD-00136	Jalan Kemenangan Besar, Blok C8 No. 22 RT 02
93	KD-00106	Jalan Kemenangan Besar, Blok C8 No. 22
94	KD-00026	Jalan Kebon Jahe, Kota EntahDimana
95	KD-00145	Jalan Kampung Kijang, Blok C5 - No. 9
96	KD-00018	Jalan Bintang Supernova, No. 78
97	KD-00058	Jalan Bintang Supernova, No. 78
98	KD-00051	Jalan Binjai 200, Kota L
99	KD-00144	Apartemen Lucky Beruntung, Lt. 3 No. 4
100	KD-00128	Apartemen Bukit Merah, Annex Plaza, Lt 3 No. A1
101	KD-00115	Apartemen Bukit Merah Annex Plaza, Lt 3 No. A1
102	KD-00009	Jalan Kebon Jahe, No. F16 - Kota E
103	KD-00092	Jalan Bukit Tol Km. 3, No. 971
104	KD-00070	Jalan Wisma Tenteram Saja No. B-01
105	KD-00118	Jalan Semantik Semut Berjalan, No. 3333
106	KD-00052	Jalan Ring Road Neolitik, No. 1 RT 5
107	KD-00120	Jalan Ring Road Konstan, No. 5
108	KD-00055	Jalan Raya Jupiter Titan, No. 55
109	KD-00089	Jalan Raya Hang Lekir, No. 62 - Kota Z
110	KD-00042	Jalan Raya Hang Lekir, Kota Z, No. 62
111	KD-00112	Jalan Raya Andromeda, Blok D No. 3
112	KD-00098	Jalan Pesisir No. 5, Kampoeng Maju Surya Gemilang
113	KD-00033	Jalan Hang Tuah, No. 31, Kota DM
114	KD-00013	Jalan Hang Tuah, No. 11, Kota DM
115	KD-00138	Jalan Gula Pahit, No. 081
116	KD-00094	Jalan Gula Pahit, No. 015
117	KD-00054	Jalan Gula Pahit, No. 001
118	KD-00100	Jalan Asia No. 55, Kompleks Pelajar Kota C
119	KD-00121	Indah Mars Cluster, No. 22F
120	KD-00061	Griya Asri Mawar Harum, Blok G No. 1
121	KD-00031	Gang Tupai, No. 7 - Desa CL
122	KD-00040	Gang Samun Saja No. 132, Kode Pos A99222
123	KD-00068	Gang Piranha, No. 3 - Desa BT
124	KD-00131	Gang Piranha, No. 13 - Desa BT
125	KD-00097	Gang Kelinci, No. 666 - Kota B
126	KD-00004	Gang Bulan Desember III, No. 9

```
127       KD-00071                    Gang Bulan Desember III, No. 155
128       KD-00093                    Gang Bulan Desember III, No. 145
129       KD-00082                         Gang Arwana, No. 6 - Kota S
130       KD-00150                          Gang Arwana No. 12, Kota S
131       KD-00065               Corina Residences Apartment, No. 0612
132       KD-00067                  Condominium Pesona Indah, No. 0708
133       KD-00011                 Cluster Ikan Mas, Taman Intan No. 2
134       KD-00091                 Cluster Ikan Mas, Taman Baru No. 96
135       KD-00147              Cluster Griya Bima Sakti, Blok A No. 1
136       KD-00081                        Bukit Vivo Indah, Blok C 2/4
137       KD-00109                        Bukit Vivo Indah, Blok C 2/4
138       KD-00072         Boulevard Raya Residences, Blok AB2 No. 102
139       KD-00014         Boulevard Raya Residences, Blok AA2 No. 88
140       KD-00078                        Blok C 2/4, Bukit Vivo Indah
141       KD-00095                  Asrama Perawat IV, No. 2 - Kota D
142       KD-00022                  Asrama Perawat IV, No. 1 - Kota D
143       KD-00017         Asrama Pelajar No. 22 A - Pondok Bima Sakti
144       KD-00037         Asrama Pelajar No. 11 B - Pondok Bima Sakti
145       KD-00108                Apartement Clifften, Lantai 12 No. 3
146       KD-00015             Jalan Puri Arteri Raya, No. 88 - Kota T
147       KD-00083             Jalan Puri Arteri Raya, No. 88 - Kota T
148       KD-00060             Apartemen Kecapi Indah, Lt. 18 No. 1801
149       KD-00007             Apartemen Kecapi Indah, Lt. 16 No. 1610
150       KD-00077                       Jalan Sutomo Baru 21 - Kota M
151       KD-00085                  Jalan Sutomo Baru No. 21 - Kota  M
152       KD-00142          Apartemen Bukit Baru, Dahlia Tower, No. A3
153       KD-00192          Taman Bunga Langit, Jalan Barat Laut No. 6
154       KD-00298               Taman Bunga Langit, Jalan Utara No. 3
155       KD-00492                     Kompleks Pelaut Tangguh, No. 5A

> #Menulis data ke file
> write.xlsx(file="staging.alamat.xlsx", x= data_alamat)

> #Clear resultset
> dbClearResult(rs)
[1] TRUE

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Profiling kolom Aktif (1)

Kolom terakhir yang perlu Anda tangani adalah kolom Aktif. Dengan pengalaman Anda sejauh ini, cobalah langsung kerjakan tugas berikut.

Tugas Praktek

Isilah konstruksi SQL yang sesuai untuk mengganti isi […] pada code editor berikut untuk melakukan profiling kolom **pola_aktif**.

Jika berjalan dengan baik maka hasil keluarannya terlihat seperti di bawah ini.

```
  pola_aktif jumlah_data
1          -           1
2          9         121
3          A           3
4       AAAA          17
5      AAAAA          13
```

Terlihat angka (pola 9) merupakan mayoritas, ada tiga data yang merupakan satu huruf (A). Dan seperti pernah diprofile di bab "Data Profiling" ada dua teks yang juga ada di kolom Aktif ini, yaitu TRUE dan FALSE. Ini tercerminkan di dua pola: AAAA untuk TRUE dan AAAAA untuk FALSE.

Jika kita analisa, harusnya seluruh data kita konversi ke dalam bentuk angka. Angka 1 untuk mewakili pelanggan aktif dan 0 untuk mewakili pelanggan tidak aktif.

Code Editor

```
library(RMySQL)

#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
                 dbname="dqlabdatawrangling")


#Konstruksi SQL

sql <- "select pola_aktif, count(*) as jumlah_data from dqlab_messy_data group by
pola_aktif"


#Mengirimkan query
```

```
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data
profil_aktif <- fetch(rs, n=-1)
print(profil_aktif)

#Menutup Koneksi
all_cons <- dbListConnections(MySQL())
for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> #Konstruksi SQL
> sql <- "select pola_aktif, count(*) as jumlah_data from dqlab_messy_data group by p
ola_aktif"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> profil_aktif <- fetch(rs, n=-1)

> print(profil_aktif)
  pola_aktif jumlah_data
1          -           1
2          9         121
3          A           3
4       AAAA          17
5      AAAAA          13

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Profiling kolom Aktif (2)

Dengan lima pola yang telah kita identifikasi berikut ini.

```
  pola_aktif jumlah_data
1          -           1
2          9         121
3          A           3
4       AAAA          17
5      AAAAA          13
```

Jika kita keluarin isi riilnya dan digrouping, ekspektasi kita hanya akan ada tujuh nilai dengan detil berikut.

- Satu nilai, yaitu – untuk pola pertama.
- Dua nilai, yaitu 0 dan 1 untuk pola kedua.
- Dua nilai, yaitu "0" dan "1" untuk pola ketiga.
- Satu nilai, yaitu "TRUE" untuk pola keempat.
- Satu nilai, yaitu "FALSE" untuk pola kelima.

Mari kita periksa dengan tugas berikut.

Tugas Praktek

Isilah konstruksi SQL yang sesuai untuk mengganti isi […] pada code editor berikut untuk melakukan profiling kolom **aktif** dan **pola_aktif**.

Jika berjalan lancar maka akan muncul hasil berikut.

```
  aktif pola_aktif jumlah_data
1     -          -           1
2     0          9          23
3     1          9          98
4 FALSE      AAAAA          13
5     I          A           1
6     O          A           2
7  TRUE       AAAA          17
```

Melihat hasil tersebut, dapat kita simpulkan bahwa asumsi kita sebelumnya benar untuk jumlah data, namun salah untuk isi data pada pola "A". Pola "A" ternyata isinya adalah "I" dan "O", salah tulis untuk angka "1" dan "0".

Kita akan perbaiki dan lakukan standarisasi pada praktek selanjutnya.

Code Editor

```
library(RMySQL)
#Membuka koneksi
con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
                 dbname="dqlabdatawrangling")

#Konstruksi SQL
sql <- "select aktif, pola_aktif, count(*) as jumlah_data from dqlab_messy_data group by aktif, pola_aktif"

#Mengirimkan query
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

#Mengambil data
profil_aktif <- fetch(rs, n=-1)
print(profil_aktif)

#Menutup Koneksi
all_cons <- dbListConnections(MySQL())
for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> #Konstruksi SQL
> sql <- "select aktif, pola_aktif, count(*) as jumlah_data from dqlab_messy_data gro
up by aktif, pola_aktif"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> profil_aktif <- fetch(rs, n=-1)

> print(profil_aktif)
  aktif pola_aktif jumlah_data
1     -          -           1
2     0          9          23
3     1          9          98
4 FALSE      AAAAA          13
5     I          A           1
6     O          A           2
7  TRUE       AAAA          17

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Standarisasi Profil kolom Aktif

Dari hasil profil, kita akan lakukan tugas berikut untuk melakukan standarisasi nilai.

Tugas Praktek

Gantilah […1…] sampai dengan […4…] dengan perintah gsub untuk merubah text pada kolom **aktif** dengan urutan berikut berikut.

1. "I" akan diubah menjadi 1.
2. "O" akan diubah menjadi 0.
3. "TRUE" akan diubah menjadi 1.
4. "FALSE" akan diubah menjadi 0.

Dan kemudian lengkapi juga […5…] dengan nama file "**staging.aktif.xlsx**".

Code Editor

```
library(RMySQL)

library(openxlsx)


#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",

                 dbname="dqlabdatawrangling")


#Konstruksi SQL

sql <- "select kode_pelanggan, aktif, pola_aktif from dqlab_messy_data"


#Mengirimkan query

rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))


#Mengambil data

data_aktif <- fetch(rs, n=-1)

data_aktif$aktif <- gsub("I", "1", data_aktif$aktif)

data_aktif$aktif <- gsub("O", "0", data_aktif$aktif)
```

```
data_aktif$aktif <- gsub("TRUE", "1", data_aktif$aktif)
data_aktif$aktif <- gsub("FALSE", "0", data_aktif$aktif)
print(data_aktif)

#Menulis output ke file Excel
write.xlsx(file="staging.aktif.xlsx", x=data_aktif)

#Menutup Koneksi
all_cons <- dbListConnections(MySQL())
for(con in all_cons) dbDisconnect(con)
```

Console

```
> library(RMySQL)

> library(openxlsx)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> #Konstruksi SQL
> sql <- "select kode_pelanggan, aktif, pola_aktif from dqlab_messy_data"

> #Mengirimkan query
> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> #Mengambil data
> data_aktif <- fetch(rs, n=-1)

> data_aktif$aktif <- gsub("I", "1", data_aktif$aktif)

> data_aktif$aktif <- gsub("O", "0", data_aktif$aktif)

> data_aktif$aktif <- gsub("TRUE", "1", data_aktif$aktif)

> data_aktif$aktif <- gsub("FALSE", "0", data_aktif$aktif)

> print(data_aktif)
    kode_pelanggan aktif pola_aktif
1         KD-00032     0      AAAAA
2         KD-00053     1          9
3         KD-00133     0      AAAAA
```

4	KD-00056	0	9
5	KD-00111	1	9
6	KD-00036	1	9
7	KD-00126	1	9
8	KD-00137	1	9
9	KD-00046	1	9
10	KD-00027	0	9
11	KD-00002	1	9
12	KD-00075	1	9
13	KD-00076	1	9
14	KD-00035	0	9
15	KD-00113	0	A
16	KD-00099	1	9
17	KD-00132	1	9
18	KD-00088	1	9
19	KD-00119	1	9
20	KD-00096	1	9
21	KD-00139	1	9
22	KD-00090	0	9
23	KD-00074	1	9
24	KD-00021	1	9
25	KD-00045	1	AAAA
26	KD-00012	0	9
27	KD-00030	0	9
28	KD-00129	1	9
29	KD-00122	1	A
30	KD-00059	1	9
31	KD-00079	1	AAAA
32	KD-00134	1	9
33	KD-00064	1	9
34	KD-00038	0	AAAAA
35	KD-00117	0	9
36	KD-00010	1	9
37	KD-00028	1	9
38	KD-00125	1	9
39	KD-00069	1	9
40	KD-00114	1	9
41	KD-00062	1	AAAA
42	KD-00006	1	9
43	KD-00024	1	9
44	KD-00084	1	9
45	KD-00104	1	9
46	KD-00103	0	9
47	KD-00143	0	9
48	KD-00034	1	9
49	KD-00087	1	9
50	KD-00039	1	9
51	KD-0047	1	9
52	KD-00149	1	9
53	KD-00003	1	AAAA
54	KD-00043	1	9
55	KD-00135	1	9
56	KD-00050	1	9
57	KD-00110	1	9
58	KD-00049	0	9

```
59        KD-00141     1       AAAA
60        KD-00044     1          9
61        KD-00124     0          9
62        KD-00105     1       AAAA
63        KD-00107     1          9
64        KD-00086     0          9
65        KD-00123     1          9
66        KD-00025     1       AAAA
67        KD-00008     1          9
68        KD-00005     1          9
69        KD-00101     1          9
70        KD-00001     1          9
71        KD-00020     1          9
72        KD-00080     1          9
73        KD-00102     0          9
74        KD-00146     1          9
75        KD-00048     0          9
76        KD-00019     1       AAAA
77        KD-00151     1          9
78        KD-00130     0          9
79        KD-00073     1          9
80        KD-00778     1       AAAA
81        KD-00066     1          9
82        KD-00041     1          9
83        KD-00140     1          9
84        KD-00116     1          9
85        KD-00127     1          9
86        KD-00057     1          9
87        KD-00016     0          9
88        KD-00063     1          9
89        KD-00148     1          9
90        KD-00023     1          9
91        KD-00029     0      AAAAA
92        KD-00136     1       AAAA
93        KD-00106     1          9
94        KD-00026     1          9
95        KD-00145     1          9
96        KD-00018     1          9
97        KD-00058     1       AAAA
98        KD-00051     0      AAAAA
99        KD-00144     1          9
100       KD-00128     0      AAAAA
101       KD-00115     0          9
102       KD-00009     1       AAAA
103       KD-00092     1          9
104       KD-00070     1          9
105       KD-00118     0      AAAAA
106       KD-00052     1          9
107       KD-00120     1       AAAA
108       KD-00055     1          9
109       KD-00089     1          9
110       KD-00042     1          9
111       KD-00112     1          9
112       KD-00098     1          9
113       KD-00033     1       AAAA
```

```
114       KD-00013     1          9
115       KD-00138     0      AAAAA
116       KD-00094     1          9
117       KD-00054     0      AAAAA
118       KD-00100     0      AAAAA
119       KD-00121     1          9
120       KD-00061     1          9
121       KD-00031     1          9
122       KD-00040     1          9
123       KD-00068     0          9
124       KD-00131     1          9
125       KD-00097     1          9
126       KD-00004     0          9
127       KD-00071     0          9
128       KD-00093     1       AAAA
129       KD-00082     1          9
130       KD-00150     1          9
131       KD-00065     1          9
132       KD-00067     1          9
133       KD-00011     0      AAAAA
134       KD-00091     1          9
135       KD-00147     1          9
136       KD-00081     0          A
137       KD-00109     1       AAAA
138       KD-00072     -          -
139       KD-00014     1       AAAA
140       KD-00078     1          9
141       KD-00095     0          9
142       KD-00022     0      AAAAA
143       KD-00017     1          9
144       KD-00037     1          9
145       KD-00108     1          9
146       KD-00015     1          9
147       KD-00083     1          9
148       KD-00060     0      AAAAA
149       KD-00007     0          9
150       KD-00077     0          9
151       KD-00085     1          9
152       KD-00142     1          9
153       KD-00192     1          9
154       KD-00298     1          9
155       KD-00492     1          9

> #Menulis output ke file Excel
> write.xlsx(file="staging.aktif.xlsx", x=data_aktif)

> #Menutup Koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons) dbDisconnect(con)
```

Konsolidasi Data

	A	B	C	D	E	F
1	kode_pelanggan	nama	alamat	no_telepon	anomali_no_telepon	kode_pos
2	KD-00001	Agus Cahyonos	Jalan Pulo Bambu No. 15, Kota Tenggara Lama	08298911112222	TRUE	876511
3	KD-00002	Khairul Nissa	Taman Vivo Indah, Blok AA No. 7	+6287132221371404	FALSE	712983
4	KD-00003	Slamet Wiyanto	Meta Residences, No. 32C	+6285725955303368	FALSE	764550
5	KD-00004	DRS. Maria Simangunsong	Gang Bulan Desember III, No. 9	+6283376770990635	FALSE	967220
6	KD-00005	Prihatin Setyonugroho	Jalan Tegal Sari Indah, No. D87 -- Kota H	+6286843623971825	FALSE	476511
7	KD-00006	DR. Candra Wijaya	Perum Pluto, Blok C No. 1	+6284063423953696	FALSE	487851
8	KD-00007	Indra Kurniawan, ST	Apartemen Kecapi Indah, Lt. 16 No. 1610	+6283840529196797	FALSE	986455
9	KD-00008	Willy Sanjaya	Kali Mars Cluster, No. 24C	+6285312577710538	FALSE	813444
10	KD-00009	Antonius Winarta	Jalan Kebon Jahe, No. F16 - Kota E	+6282722234294686	FALSE	896555
11	KD-00010	Sri Wahyuni, Ir	Perum Venus, Gg. Harimau No. 1A	+6284079659289143	FALSE	987453
12	KD-00011	Rosalina Kurnia	Cluster Ikan Mas, Taman Intan No. 2	+6288339032314103	FALSE	967223
13	KD-00012	Cahyono, Agus	Pulo Bambu No. 15, Kota Tenggara Lama	+62829891111222	TRUE	876511
14	KD-00013	Danang Santosa	Jalan Hang Tuah, No. 11, Kota DM	+6282672925000608	FALSE	666122
15	KD-00014	Elisabeth Suryadinata, SKOM, ST	Boulevard Raya Residences, Blok AA2 No. 88	+6285455084014504	FALSE	-
16	KD-00015	Mario Setiawan	Jalan Puri Arteri Raya, No. 88 - Kota T	+6282989111122220	FALSE	876511
17	KD-00016	Indra K.	Jalan Pahlawan, No. 69CCD	+6289222405928430	FALSE	896550

Sampai pada bab ini, Anda telah menyelesaikan profiling dan standarisasi untuk lima kolom berikut:

- Nama
- No Telepon
- Kode Pos
- Alamat
- Aktif

Dan menyimpan hasil standarisasi ke dalam file-file berikut:

- nama.xlsx
- no_telepon.xlsx
- kode_pos.xlsx
- alamat.xlsx
- aktif.xlsx

Kita akan menyatukan seluruh file ini ke dalam satu file: **staging.teks.xlsx** dengan proses berikut:

- Membaca tiap file Excel dan menyimpannya dalam berbagai variable.
- Menggabungkan variable-variable dengan function **merge**.
- Hasil gabungan ini kita ambil field "kode_pelanggan", "nama", "alamat", "no_telepon", "kode_pos", dan "aktif".
- Hasil gabungan kita tulis ke dalm file **teks.xlsx**.

Khusus untuk function merge, berikut adalah contohnya:

```
    staging.teks <- merge(x=staging.nama, y=staging.no_telepon,
by.x = "kode_pelanggan", by.y = "kode_pelanggan", all = TRUE)
```

Berikut adalah penjelasannya.

Elemen	Keterangan
staging.text	Nama variable untuk menyimpan hasil penggabungan. Variable ini akan digunakan secara berulang dan bertahap untuk digabungkan kembali dengan variable lainnya.
x = staging.nama	• x = Merupakan parameter untuk variable pertama • staging.nama = Variable pertama yang akan digabung
y = staging.no_telepon	• y = Merupakan parameter untuk variable kedua • staging.no_telepon = Variable kedua yang akan digabung
`by.x = "kode_pelanggan"`	Penggabungan memerlukan referensi. Dan referensi untuk data frame adalah mana kolom yang nilainya sama dari kedua sisi. Parameter by.x ini adalah menyatakan referensi kolom dari variable pertama, yaitu kolom "kode_pelanggan"
`by.y = "kode_pelanggan"`	Parameter by.y ini adalah menyatakan referensi kolom dari variable kedua, yaitu kolom "kode_pelanggan"

Tugas Praktek

Seluruh code pada proses di atas hampir lengkap dimasukkan ke dalam code editor. Gantilah bagian […1…] s/d […4…] untuk melengkapi apa yang diperlukan sehingga menghasilkan file gabungan bernama "**staging.teks.xlsx**" seperti tampilan Excel berikut.

	A	B	C	D	E	F
1	kode_pelanggan	nama	alamat	no_telepon	anomali_no_telepon	kode_pos
2	KD-00001	Agus Cahyonos	Jalan Pulo Bambu No. 15, Kota Tenggara Lama	08298911112222	TRUE	876511
3	KD-00002	Khairul Nissa	Taman Vivo Indah, Blok AA No. 7	+6287132221371404	FALSE	712983
4	KD-00003	Slamet Wiyanto	Meta Residences, No. 32C	+6285725955303368	FALSE	764550
5	KD-00004	DRS. Maria Simangunsong	Gang Bulan Desember III, No. 9	+6283376770990635	FALSE	967220
6	KD-00005	Prihatin Setyonugroho	Jalan Tegal Sari Indah, No. D87 -- Kota H	+6286843623971825	FALSE	476511
7	KD-00006	DR. Candra Wijaya	Perum Pluto, Blok C No. 1	+6284063423953696	FALSE	487851
8	KD-00007	Indra Kurniawan, ST	Apartemen Kecapi Indah, Lt. 16 No. 1610	+6283840529196797	FALSE	986455
9	KD-00008	Willy Sanjaya	Kali Mars Cluster, No. 24C	+6285312577710538	FALSE	813444
10	KD-00009	Antonius Winarta	Jalan Kebon Jahe, No. F16 - Kota E	+6282722234294686	FALSE	896555
11	KD-00010	Sri Wahyuni, Ir	Perum Venus, Gg. Harimau No. 1A	+6284079659289143	FALSE	987453
12	KD-00011	Rosalina Kurnia	Cluster Ikan Mas, Taman Intan No. 2	+6288339032314103	FALSE	967223
13	KD-00012	Cahyono, Agus	Pulo Bambu No. 15, Kota Tenggara Lama	+62829891111222	TRUE	876511
14	KD-00013	Danang Santosa	Jalan Hang Tuah, No. 11, Kota DM	+6282672925000608	FALSE	666122
15	KD-00014	Elisabeth Suryadinata, SKOM, ST	Boulevard Raya Residences, Blok AA2 No. 88	+6285455084014504	FALSE	-
16	KD-00015	Mario Setiawan	Jalan Puri Arteri Raya, No. 88 - Kota T	+6282989111122220	FALSE	876511
17	KD-00016	Indra K.	Jalan Pahlawan, No. 69CCD	+6289222405928430	FALSE	896550

Code Editor

```
library(openxlsx)


#Membaca tiap file staging Excel dan menyimpannya dalam variable bernama awalan staging

staging.nama <- read.xlsx("staging.nama.xlsx")

staging.no_telepon <- read.xlsx("staging.no_telepon.xlsx")

staging.kode_pos <- read.xlsx("staging.kode_pos.xlsx")

staging.alamat <- read.xlsx("staging.alamat.xlsx")

staging.aktif <- read.xlsx("staging.aktif.xlsx")


#Menggabungkan variable staging dengan function merge

staging.teks <- merge(x=staging.nama, y=staging.no_telepon, by.x = "kode_pelanggan", by.y = "kode_pelanggan", all = TRUE)

staging.teks <- merge(x=staging.teks, y=staging.kode_pos, by.x = "kode_pelanggan", by.y = "kode_pelanggan", all = TRUE)

staging.teks <- merge(x=staging.teks, y=staging.alamat, by.x = "kode_pelanggan", by.y = "kode_pelanggan", all = TRUE)

staging.teks <- merge(x=staging.teks, y=staging.aktif, by.x = "kode_pelanggan", by.y = "kode_pelanggan", all = TRUE)
```

staging.teks <- staging.teks[c("kode_pelanggan","nama", "alamat", "no_telepon", "anomali_no_telepon", "kode_pos")]

write.xlsx(file="staging.teks.xlsx", staging.teks)

Console

```
> library(openxlsx)

> #Membaca tiap file staging Excel dan menyimpannya dalam variable bernama awalan sta
ging
> staging.nama <- read.xlsx("staging.nama.xlsx")

> staging.no_telepon <- read.xlsx("staging.no_telepon.xlsx")

> staging.kode_pos <- read.xlsx("staging.kode_pos.xlsx")

> staging.alamat <- read.xlsx("staging.alamat.xlsx")

> staging.aktif <- read.xlsx("staging.aktif.xlsx")

> #Menggabungkan variable staging dengan function merge
> staging.teks <- merge(x=staging.nama, y=staging.no_telepon, by.x = "kode_pelanggan"
, by.y = "kode_pelanggan", all = TRUE)

> staging.teks <- merge(x=staging.teks, y=staging.kode_pos, by.x = "kode_pelanggan",
by.y = "kode_pelanggan", all = TRUE)

> staging.teks <- merge(x=staging.teks, y=staging.alamat, by.x = "kode_pelanggan", by
.y = "kode_pelanggan", all = TRUE)

> staging.teks <- merge(x=staging.teks, y=staging.aktif, by.x = "kode_pelanggan", by.
y = "kode_pelanggan", all = TRUE)

> staging.teks <- staging.teks[c("kode_pelanggan","nama", "alamat", "no_telepon",  "a
nomali_no_telepon", "kode_pos")]

> write.xlsx(file="staging.teks.xlsx", staging.teks)
```

Kesimpulan

Selamat!!

Anda telah menyelesaikan dua bab yang cukup intensif ini dengan mengidentifikasi pola dan berbagai cara mengganti teks sehingga menjadi standar yang diterima.

Untuk rangkuman, berbeda dengan bab sebelumnya. Kali ini kita berikan rangkuman table untuk daftar fungsi, pola regex dan SQL dari apa yang telah Anda telah pelajari dan lakukan untuk melakukan *profiling* dan standarisasi.

Kolom	Function	Pola Regex	SQL	Deskripsi
Nama	• gsub • trimws • write.xlsx	• " {2,}" • "[^A-Za-z.,]" • \bir\b • \bibu\b • \bbapak\b	• SELECT kode_pelanggan, nama from dqlab_messy_data where nama REGEXP … • SELECT kode_pelanggan, nama from dqlab_messy_data where nama like … or nama like …	Standarisasi disini menghilangkan spasi berulang, spasi di awal dan akhir teks nama, dan menghilangkan kata panggilan.
No Telepon	• gsub • ggplot • theme • geom_bar • write.xlsx	• "^0"	• SELECT pola_no_telepon, length(pola_no_telepon) as panjang_text, count(*) as jumlah_data from dqlab_messy_data group by pola_no_telepon • SELECT left(no_telepon,1) as prefix_no_telepon, pola_no_telepon from dqlab_messy_data where pola_no_telepon = '999999999999999' group by left(no_telepon,1), pola_no_telepon	Standarisasi disini menghilangkan spasi berulang, spasi di awal dan akhir teks nama, dan menghilangkan kata panggilan.
Kode Pos	• gsub • write.xlsx		• SELECT pola_kode_pos, length(pola_kode_pos) as panjang_text, count(*) as jumlah_data	Standarisasi disini mengganti karakter yang salah tulis sehingga

Kolom	Function	Pola Regex	SQL	Deskripsi
			from dqlab_messy_data group by pola_kode_pos • SELECT kode_pos, pola_kode_pos from dqlab_messy_data where pola_kode_pos in ('99999A', '9999A9')"	seluruhnya menjadi enam digit angka kode pos.
Alamat	• gsub • write.xlsx	• ^[A-Za-z]+$ • ^[0-9]+$ • ^[aAw]+$ • ^[9w]+$ • jln[ ]*\. • \bjln\b • jl[ ]*\. • \bjl\b • jalan\.	• SELECT kode_pelanggan, alamat, pola_alamat from dqlab_messy_data where pola_alamat REGEXP '^[aAw]+$' or pola_alamat REGEXP '^[9w]+$'	Standarisasi disini merubah salah satu teks yang paling sering digunakan sebagai singkatan di alamat, yaitu kata "Jalan".
Aktif	• Gsub • write.xlsx		• SELECT pola_aktif, length(pola_aktif) as panjang_text, count(*) as jumlah_data from dqlab_messy_datagroup by pola_aktif	Standarisasi disini mengganti teks "I","O", "TRUE" dan "FALSE" menjadi angka 1 dan 0.
Seluruh Kolom	• merge • read.xlsx • write.xlsx			Menggabungkan seluruh dataset

Pastinya Anda semakin semangat kan? Ternyata pengolahan data walaupun rumit tapi sangat bisa dikelola 😊

Klik tombol **Next** untuk melanjutkan ke bab berikutnya – yaitu mengolah kolom tanggal lahir.

Pendahuluan

Kolom **tanggal lahir** pada data pelanggan adalah kolom lain yang berisi informasi sangat penting. Dengan data tanggal yang benar, kita bisa menghitung umur dan bisa mengenal demografik pelanggan dengan lebih baik.

Ini berimplikasi pada cara kita melakukan pemaketan produk, pemasaran, pendekatan relasi, dan lain-lain.

Pada bab ini kita akan melakukan profiling dan standarisasi yang diperlukan untuk data tanggal lahir yang pada bab awal telah ditampilkan sebagian sebagai berikut.

Tanggal Lahir
1 April 2028
19-08-1986
11-07-1981
10/13/79
24-03-1976
20-02-1970
14-11-1987
12-01-1968
14-03-1879
23-11-1962
10/23/91
02/28/1969
02/20/1970
24 Januari 1952
22 Februari 2000
26 Agustus 1983

Klik tombol Next untuk melanjutkan.

Identifikasi awal kolom Tanggal Lahir

Profiling tanggal lahir jika menggunakan observasi seperti pada pengantar, kita akan mendapatkan tiga pola berikut:

- Terdapat format yang terdiri dari angka hari, nama bulan dan angka tahun dengan pemisah spasi – dengan panjang nama bulan bervariasi. Format ini di beberapa aplikasi biasanya ditulis dengan dd MMM yyyy.
- Terdapat format yang terdiri dari angka hari, angka bulan dan angka tahun dengan pemisah tanda minus. Format ini di beberapa aplikasi biasanya ditulis dengan dd-MM-yyyy. Di R kita bisa menggunakan
- Terdapat format yang terdiri dari angka bulan, angka hari dan angka tahun dengan pemisah garis miring. Format ini di beberapa aplikasi biasanya ditulis dengan MM/dd/yyyy.

Untuk melakukan profiling awal apakah format ini benar semua sebelum kita lakukan standarisasi, maka bisa diambil strategi berikut (nomor urut sesuai format no urut di atas):

- Melakukan pengelompokan terhadap komponen non huruf untuk memastikan nama bulan konsisten semua, misalkan bulan pertama ditulis Januari, dan tidak ada variasi seperti January atau Jan.
- Menggunakan statistik max dan min untuk tiap angka hari, bulan dan tahun. Ini tentunya tidak memberi jaminan akan memberikan pola yang benar. Sebagai contoh untuk tanggal 31-02-1998, ini adalah tanggal yang tidak valid. Tapi jika pisahkan komponennya: 31 untuk hari, 2 untuk bulan dan 1998 untuk tahun – semuanya angka valid untuk min dan max.
- Menggunakan statistik max dan min untuk tiap angka hari, bulan dan tahun. Ini tentunya tidak memberi jaminan tanggal yang benar dengan alasan yang sama dengan poin no 2.

Profiling nomor 2 dan 3 tidak akan kita lakukan karena alasan yang disebutkan. Kita akan fokus ke profiling pertama dengan menggunakan gabungan perintah SQL dan function-function di R.

Berikut adalah tahapannya:

- Gunakan perintah SQL untuk mengambil kolom tanggal lahir dengan filter pola tanggal yang memiliki huruf.

  ```
  SELECT tanggal_lahir from dqlab_messy_data where
  pola_tanggal_lahir like '%A%'
  ```

  Atau menggunakan pola regex [A-Za-z] – artinya mengandung huruf kecil atau besar dari a sampai z – maka perintah SQL nya adalah sebagai berikut.

  ```
  SELECT tanggal_lahir from dqlab_messy_data where tanggal_lahir
  REGEXP '[A-Za-z]'
  ```

-

- Untuk jawaban tugas praktek, kita akan gunakan versi regex.
- Menghapus karakter digit dan spasi dengan function **gsub** dan pola regex [0-9].

```
gsub('[0-9 ]','', data.pelanggan$tanggal_lahir)
```

- Menggunakan fungsi **unique** yang akan melakukan grouping nilai teks nama bulan.

```
unique(data.pelanggan$tanggal_lahir)
```

Tugas Praktek

Gantilah `[…1…]` sampai dengan `[…3…]` dengan perintah yang bersesuaian dengan contoh pada soal sehingga kita mendapatkan hasil akhir nama unik bulan seperti terlihat sebagai berikut.

```
> unique(data.pelanggan$tanggal_lahir)
 [1] "April"     "Januari"   "Februari"  "Agustus"   "Desember"  "Maret"
 [7] "Juni"      "Juli"      "Oktober"   "September" "November"  "Mei"
```

Ini menunjukkan bahwa nama bulan ada 12 dan tidak ada variasi. Dengan demikian data tinggal dirubah sesuai list ini pada praktek selanjutnya.

Code Editor

```
library(RMySQL)


#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo",
host="mysqlhost",dbname="dqlabdatawrangling")


#Melakukan query data untuk format tanggal yang memiliki huruf dengan regex [A-Za-z]

sql <- "SELECT tanggal_lahir from dqlab_messy_data where tanggal_lahir REGEXP
'[A-Za-z]'"

rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

data.pelanggan <- fetch(rs, n=-1)

dbClearResult(rs)
```

Menghapus karakter digit dan spasi dengan function gsub dan pola regex [0-9].

data.pelanggan$tanggal_lahir <- gsub('[0-9]','', data.pelanggan$tanggal_lahir)

#Melakukan grouping nama bulan dengan function unique

unique(data.pelanggan$tanggal_lahir)

#Menutup seluruh koneksi

all_cons <- dbListConnections(MySQL())

for(con in all_cons)

+ dbDisconnect(con)

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dq
labdatawrangling")

> #Melakukan query data untuk format tanggal yang memiliki huruf dengan regex [A-Za-z
]
> sql <- "SELECT tanggal_lahir from dqlab_messy_data where tanggal_lahir REGEXP '[A-Z
a-z]'"

> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> data.pelanggan <- fetch(rs, n=-1)

> dbClearResult(rs)
[1] TRUE

> # Menghapus karakter digit dan spasi dengan function gsub dan pola regex [0-9 ].
> data.pelanggan$tanggal_lahir <- gsub('[0-9 ]','', data.pelanggan$tanggal_lahir)

> #Melakukan grouping nama bulan dengan function unique
> unique(data.pelanggan$tanggal_lahir)
 [1] "April"     "Januari"   "Februari"  "Agustus"   "Desember"  "Maret"
 [7] "Juni"      "Juli"      "Oktober"   "September" "November"  "Mei"

> #Menutup seluruh koneksi
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons)
+     + dbDisconnect(con)
```

Mengganti Januari s/d Desember menjadi angka

Hasil profiling praktek sebelumnya mendapatkan daftar nama bulan yang akan menjadi dasar kita untuk merubah daftar nama ini menjadi angka pada praktek berikut. Untuk melakukannya kita menggunakan aturan sederhana (*simple rule*) atau penggantian sederhana, dari teks satu menjadi teks lainnya.

Kita tetap menggunakan function **gsub** namun tanpa pola regex – mengganti nama bulan dengan angka bulan terkait. Jadi "Januari" diganti menjadi "1", "Februari" diganti menjadi "2", dan seterusnya.

Namun selain angka, kita akan sekalian merubah format ini menjadi dd-MM-yyyy dengan pemisah tanda minus (-). Dengan demikian, karena nama bulan merupakan teks yang diapit oleh hari dan bulan. Maka "Januari" diubah menjadi "-1-", "Februari" diganti menjadi "-2-", dan seterusnya.

Berikut adalah contoh gsub untuk mengganti "**Januari**" menjadi "**1**" untuk variable **data.pelanggan$tanggal_lahir**.

```
gsub("Januari","-01-", data.pelanggan$tanggal_lahir)
```

Mari kita lakukan tugas praktek berikut untuk melakukan standarisasi data ini.

Tugas Praktek

Ganti bagian […1…] s/d […12…] untuk mengganti nama-nama bulan dari "Januari" s/d "Desember" dengan function gsub.

Jika berjalan dengan lancar maka output dari variable data.pelanggan yang digunakan terlihat sebagai berikut.

Code Editor

```
library(RMySQL)

#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo",
host="mysqlhost",dbname="dqlabdatawrangling")


#Melakukan query untuk data yang mengandung huruf alfabet

sql <- "select tanggal_lahir from dqlab_messy_data where tanggal_lahir regexp '[a-z]'"
```

```
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
data.pelanggan <- fetch(rs, n=-1)
dbClearResult(rs)
#Menghilangkan spasi
data.pelanggan$tanggal_lahir <- gsub(" ","", data.pelanggan$tanggal_lahir)

#Melakukan standarisasi nama bulan ke angka bulan
data.pelanggan$tanggal_lahir <- gsub("Januari","-01-", data.pelanggan$tanggal_lahir)
data.pelanggan$tanggal_lahir <- gsub("Februari","-02-", data.pelanggan$tanggal_lahir)
data.pelanggan$tanggal_lahir <- gsub("Maret","-03-", data.pelanggan$tanggal_lahir)
data.pelanggan$tanggal_lahir <- gsub("April","-04-", data.pelanggan$tanggal_lahir)
data.pelanggan$tanggal_lahir <- gsub("Mei","-05-", data.pelanggan$tanggal_lahir)
data.pelanggan$tanggal_lahir <- gsub("Juni","-06-", data.pelanggan$tanggal_lahir)
data.pelanggan$tanggal_lahir <- gsub("Juli","-07-", data.pelanggan$tanggal_lahir)
data.pelanggan$tanggal_lahir <- gsub("Agustus","-08-", data.pelanggan$tanggal_lahir)
data.pelanggan$tanggal_lahir <- gsub("September","-09-",
data.pelanggan$tanggal_lahir)
data.pelanggan$tanggal_lahir <- gsub("Oktober","-10-", data.pelanggan$tanggal_lahir)
data.pelanggan$tanggal_lahir <- gsub("November","-11-",
data.pelanggan$tanggal_lahir)
data.pelanggan$tanggal_lahir <- gsub("Desember","-12-",
data.pelanggan$tanggal_lahir)
data.pelanggan
```

#Menutup seluruh koneksi MySQL

all_cons <- dbListConnections(MySQL())

for(con in all_cons)

+ dbDisconnect(con)

Console

```
> library(RMySQL)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dq
labdatawrangling")

> #Melakukan query untuk data yang mengandung huruf alfabet
> sql <- "select tanggal_lahir from dqlab_messy_data where tanggal_lahir regexp '[a-z
]'"

> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> data.pelanggan <- fetch(rs, n=-1)

> dbClearResult(rs)
[1] TRUE

> #Menghilangkan spasi
> data.pelanggan$tanggal_lahir <- gsub(" ","", data.pelanggan$tanggal_lahir)

> #Melakukan standarisasi nama bulan ke angka bulan
> data.pelanggan$tanggal_lahir <- gsub("Januari","-01-", data.pelanggan$tanggal_lahir
)

> data.pelanggan$tanggal_lahir <- gsub("Februari","-02-", data.pelanggan$tanggal_lahi
r)

> data.pelanggan$tanggal_lahir <- gsub("Maret","-03-", data.pelanggan$tanggal_lahir)

> data.pelanggan$tanggal_lahir <- gsub("April","-04-", data.pelanggan$tanggal_lahir)

> data.pelanggan$tanggal_lahir <- gsub("Mei","-05-", data.pelanggan$tanggal_lahir)

> data.pelanggan$tanggal_lahir <- gsub("Juni","-06-", data.pelanggan$tanggal_lahir)

> data.pelanggan$tanggal_lahir <- gsub("Juli","-07-", data.pelanggan$tanggal_lahir)

> data.pelanggan$tanggal_lahir <- gsub("Agustus","-08-", data.pelanggan$tanggal_lahir
)
```

```
> data.pelanggan$tanggal_lahir <- gsub("September","-09-", data.pelanggan$tanggal_lah
ir)

> data.pelanggan$tanggal_lahir <- gsub("Oktober","-10-", data.pelanggan$tanggal_lahir
)

> data.pelanggan$tanggal_lahir <- gsub("November","-11-", data.pelanggan$tanggal_lahi
r)

> data.pelanggan$tanggal_lahir <- gsub("Desember","-12-", data.pelanggan$tanggal_lahi
r)

> data.pelanggan
   tanggal_lahir
1      1-04-2028
2     24-01-1952
3     22-02-2000
4     26-08-1983
5      1-12-1964
6     14-03-1979
7     28-02-1969
8     20-06-2001
9     14-07-1977
10    23-10-1991
11    23-10-1991
12    24-06-1992
13    05-09-1990
14    19-03-1950
15    23-11-1962
16     8-03-1955
17    21-05-1980
18    13-11-1963
19    19-08-1986
20     8-02-1967
21    13-11-1962
22    19-03-1950
23    20-12-1977
24    28-05-1969
25    17-08-1986
26    30-11-1954
27    12-01-1969
28    17-09-1982
29     8-03-1955
30     8-03-1955
31    10-10-1982
32    04-07-1987
33    17-09-1982
34    22-04-1933
35    30-11-1954
36    17-02-2097
37    30-11-1954
38    26-11-1983
39    14-03-1879
40    12-01-1969
41    20-11-1987
```

```
42    26-11-1983
43     1-12-1964
44    25-07-1974
45    23-11-1962
46    09-08-1972
47    19-03-1950
48     7-07-1968
49    26-08-1983
50    21-05-1980

> #Menutup seluruh koneksi MySQL
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons)
+     + dbDisconnect(con)
```

Melakukan Standarisasi Format dd-MM-yyyy

Pada praktek sebelumnya kita telah mengganti format tanggal yang memiliki nama bulan ke format dd-MM-yyyy. Untuk sisa format satunya lagi yang masih dalam bentuk MM/dd/yyyy, kita akan lakukan standarisasi juga menjadi dd-MM-yyyy.

Banyak cara untuk melakukan ini, tapi untuk praktek ini kita lakukan dengan tahapan berikut.

Memisahkan Kolom

Tahap pertama, kita akan memisahkan teks tanggal menjadi tiga kolom, yaitu hari, bulan dan tahun- dengan pemisah adalah tanda garis miring (/).

Function colsplit dari package reshape2 sangat cocok untuk mencapai tujuan ini. Syntax dari colsplit adalah sebagai berikut.

```
    colsplit(string, pattern, names)
```

dimana:

- **string**: adalah teks yang akan dipisahkan.
- **pattern**: pola regex yang digunakan untuk memisahkan teks.
- **names**: vector yang berisi nama-nama kolom yang dipisahkan.

Dan menyesuaikan kebutuhan kita, penggunaannya adalah sebagai berikut.

```
    colsplit(string, pattern, names)
```

Menggabungkan kembali tiga kolom: hari, bulan dan tahun dengan function paste dan tanda minus (-) sebagai karakter penggabungan.

```
    tanggal.split
<- colsplit(data.pelanggan$tanggal_lahir,"/",c("bulan","hari","t
ahun"))
```

dimana:

- **split**: variable untuk menyimpan hasil split
- **pelanggan$tanggal_lahir**: data dari kolom tanggal_lahir dari variable data.pelanggan.
- **"/"**: tanda garis miring, pola yang digunakan untuk memisahkan teks.
- **c("bulan","hari","tahun")**: vector yang berisi nama-nama kolom yang dipisahkan secara terurut, yaitu dimulai dari **bulan** yang kemudian diikuti **hari** dan **tahun**.

Menggabungkan Kolom

Tahap kedua adalah menggabungkan kembali tiga kolom tersebut dengan urutan hari, bulan dan tahun dengan function paste dan tanda minus (-) sebagai karakter penggabungan. Syntaxnya adalah sebagai berikut.

```
paste(..., sep = " ")
```

dimana:

- … : adalah daftar variable atau teks dengan pemisah koma.
- **`sep = " "`** : adalah karakter antara pada saat penggabungan teks. Pada contoh ini adalah karakter spasi.

Dan menyesuaikan kebutuhan kita, maka function paste yang digunakan adalah sebagai berikut.

```
paste(tanggal.split$hari, tanggal.split$bulan,
tanggal.split$tahun, sep="-")
```

dimana:

- **`split$hari`**: kolom **hari** dari variable **tanggal.split**. Ini kita dapatkan dari hasil split sebelumnya.
- **`split$bulan`**: kolom **bulan** dari variable **tanggal.split**. Ini kita dapatkan dari hasil split sebelumnya.
- **`split$tahun`**: kolom **tahun** dari variable **tanggal.split**. Ini kita dapatkan dari hasil split sebelumnya.
- **`sep = "-"`** : adalah karakter antara pada saat penggabungan teks. Pada contoh ini adalah karakter minus (-).

Tugas Praktek

Gantilah bagian […1…] dan […2…], masing-masing untuk memisahkan kolom tanggal lahir dan menggabungkannya kembali dengan urutan hari, bulan dan tahun.

Gunakan contoh function colsplit dan paste pada Lesson untuk menyelesaikan tugas ini.

Jika berjalan dengan lancar maka Anda akan mendapatkan hasil berikut.

```
> data.pelanggan$tanggal_lahir
 [1] "13-10-79"   "23-10-91"   "28-2-1969"  "20-2-1970"  "1-1-1"
 [6] "1-1-1"      "26-8-1983"  "17-7-1987"  "7-7-77"     "8-2-1967"
[11] "14-11-1987" "12-7-1977"  "8-19-1950"  "31-1-1"     "28-2-1969"
[16] "28-2-1969"  "12-7-1977"  "25-2-1987"  "17-7-1987"  "12-1-1972"
[21] "25-6-1987"  "25-7-1974"  "5-9-1990"   "28-2-1969"  "14-1-1988"
[26] "18-8-1988"  "1-1-1"      "1-12-1964"  "31-1-1"     "24-9-1990"
[31] "24-2-1978"  "29-2-1969"  "12-1-1971"  "31-1-1"     "8-8-2008"
[36] "23-12-1968" "20-12-77"   "15-2-1997"  "29-11-1967" "30-11-1967"
[41] "7-7-68"     "7-7-1968"   "20-10-1987" "14-11-1987" "21-1-1"
[46] "23-6-1968"  "29-12-1963" "20-6-1"     "15-2-1997"  "1-1-1"
```

```
[51] "29-12-1967" "20-10-1987" "17-8-86"    "23-10-95"   "3-10-1988"
[56] "28-2-1969"  "23-10-79"   "7-7-68"     "20-2-1970"
```

Terlihat ada beberapa hasil yang tidak sesuai ekspektasi kita – yang ditandai dengan text warna merah sebagai berikut.

```
> data.pelanggan$tanggal_lahir
 [1] "13-10-79"   "23-10-91"   "28-2-1969"  "20-2-1970"  "1-1-1"
 [6] "1-1-1"      "26-8-1983"  "17-7-1987"  "7-7-77"     "8-2-1967"
[11] "14-11-1987" "12-7-1977"  "8-19-1950"  "31-1-1"     "28-2-1969"
[16] "28-2-1969"  "12-7-1977"  "25-2-1987"  "17-7-1987"  "12-1-1972"
[21] "25-6-1987"  "25-7-1974"  "5-9-1990"   "28-2-1969"  "14-1-1988"
[26] "18-8-1988"  "1-1-1"      "1-12-1964"  "31-1-1"     "24-9-1990"
[31] "24-2-1978"  "29-2-1969"  "12-1-1971"  "31-1-1"     "8-8-2008"
[36] "23-12-1968" "20-12-77"   "15-2-1997"  "29-11-1967" "30-11-1967"
[41] "7-7-68"     "7-7-1968"   "20-10-1987" "14-11-1987" "21-1-1"
[46] "23-6-1968"  "29-12-1963" "20-6-1"     "15-2-1997"  "1-1-1"
[51] "29-12-1967" "20-10-1987" "17-8-86"    "23-10-95"   "3-10-1988"
[56] "28-2-1969"  "23-10-79"   "7-7-68"     "20-2-1970"
```

Nah, terlihat ada ketidakseragaman format tahun. Ada yang 1, 79, 81, dan seterusnya. Ini harus diperbaiki dengan rule sederhana pada praktek selanjutnya.

Code Editor

```
library(RMySQL)

library(reshape2)


#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo",
host="mysqlhost",dbname="dqlabdatawrangling")

#Mengambil data yang memiliki tanda garis miring /

sql <- "select tanggal_lahir from dqlab_messy_data where tanggal_lahir like '%/%'"


rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
```

```
data.pelanggan <- fetch(rs, n=-1)

dbClearResult(rs)

#Melakukan split dan menyimpannya ke variable tanggal.split dengan urutan bulan, hari dan tahun

tanggal.split <- colsplit(data.pelanggan$tanggal_lahir,"/",c("bulan","hari","tahun"))


#Menggabungkan kembali dalam urutan hari, bulan dan tahun dan menyimpannya kembali ke data.pelanggan$tanggal_lahir

data.pelanggan$tanggal_lahir <- paste(tanggal.split$hari, tanggal.split$bulan, tanggal.split$tahun, sep="-")

data.pelanggan$tanggal_lahir

#Menutup seluruh koneksi MySQL

all_cons <- dbListConnections(MySQL())

for(con in all_cons)

    + dbDisconnect(con)
```

Console

```
> library(RMySQL)

> library(reshape2)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dq
labdatawrangling")

> #Mengambil data yang memiliki tanda garis miring /
> sql <- "select tanggal_lahir from dqlab_messy_data where tanggal_lahir like '%/%'"

> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> data.pelanggan <- fetch(rs, n=-1)

> dbClearResult(rs)
[1] TRUE

> #Melakukan split dan menyimpannya ke variable tanggal.split dengan urutan bulan, ha
ri dan tahun
> tanggal.split <- colsplit(data.pelanggan$tanggal_lahir,"/",c("bulan","hari","tahun"
))
```

```
> #Menggabungkan kembali dalam urutan hari, bulan dan tahun dan menyimpannya kembali
ke data.pelanggan$tanggal_lahir
> data.pelanggan$tanggal_lahir <- paste(tanggal.split$hari, tanggal.split$bulan, tang
gal.split$tahun, sep="-")

> data.pelanggan$tanggal_lahir
 [1] "13-10-79"   "23-10-91"   "28-2-1969"  "20-2-1970"  "1-1-1"
 [6] "1-1-1"      "26-8-1983"  "17-7-1987"  "7-7-77"     "8-2-1967"
[11] "14-11-1987" "12-7-1977"  "8-19-1950"  "31-1-1"     "28-2-1969"
[16] "28-2-1969"  "12-7-1977"  "25-2-1987"  "17-7-1987"  "12-1-1972"
[21] "25-6-1987"  "25-7-1974"  "5-9-1990"   "28-2-1969"  "14-1-1988"
[26] "18-8-1988"  "1-1-1"      "1-12-1964"  "31-1-1"     "24-9-1990"
[31] "24-2-1978"  "29-2-1969"  "12-1-1971"  "31-1-1"     "8-8-2008"
[36] "23-12-1968" "20-12-77"   "15-2-1997"  "29-11-1967" "30-11-1967"
[41] "7-7-68"     "7-7-1968"   "20-10-1987" "14-11-1987" "21-1-1"
[46] "23-6-1968"  "29-12-1963" "20-6-1"     "15-2-1997"  "1-1-1"
[51] "29-12-1967" "20-10-1987" "17-8-86"    "23-10-95"   "3-10-1988"
[56] "28-2-1969"  "23-10-79"   "7-7-68"     "20-2-1970"

> #Menutup seluruh koneksi MySQL
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons)
+     + dbDisconnect(con)
```

Memperbaiki format tahun satu dan dua digit

Dan berdasarkan praktek sebelumnya juga, kita perlu definisikan rule standarisasi yang harus dilakukan untuk tahun.

Untuk yang dua digit – angka 10 sampai dengan 99 – secara logika maka harusnya adalah di tahun berawalan 19. Contoh: 79 adalah 1979. Sedangkan jika satu digit – angka 0 sampai dengan 9 – adalah tahun berawalan angka 200. Contoh: 1 adalah tahun 2001.

Ini hal yang tentu perlu diperdebatkan. Tapi dengan menggunakan tingkat kemungkinan dan kesepakatan, maka *rule* inilah yang akan jadi patokan nanti ke praktek selanjutnya.

Dengan kesepakatan ini, jika menggunakan gsub akan sangat panjang. Tapi kita menggunakan function **sapply** untuk iterasi seluruh data dan melakukan pergantian. Perintahnya adalah sebagai berikut.

```
sapply(tanggal.split$tahun, function(x) if(x>=0 & x<10) 2000+x
else if(x>=10 & x<100) 1900+x else x)
```

Berikut adalah keterangan tahap demi tahap dari perintah di atas.

Elemen Perintah	Keterangan
sapply	Adalah fungsi untuk mengakses satu per satu item data frame ataupun list, dan mengolahnya dengan function yang kita definisikan di dalam argumen kedua.
tanggal.split$tahun	Kolom tahun dari variable tanggal.split.
function(x)	Definisi function dengan x mewakili tiap item yang diakses oleh sapply, dalam hal ini tiap item dari tanggal.split$tahun.

Kemudian ada konstruksi **if...else** sebagai berikut.

Elemen Perintah	Keterangan
if(x>=0 & x<10) 2000+x	Penggunaan if untuk cek kondisi. Ini artinya jika item x lebih besar dari angka 0 dan lebih kecil dari angka 10 maka x ditambahkan dengan 2000.
else if(x>=10 & x<100) 1900+x	Jika kondisi di atas tidak dipenuhi, kita cek kondisi berikutnya yaitu jika item x lebih besar sama dengan dari angka 10 dan lebih kecil dari angka 100 maka x ditambahkan dengan 1900.
else x	Jika seluruh kondisi di atas tidak terpenuhi, maka kita kembalikan nilai asli yaitu x.

Tugas Praktek

Gantilah bagian `[…1…]` dengan perintah sapply untuk perbaikan format tahun seperti yang dijelaskan pada soal. Kemudian isi juga bagian `[…2…]` dan `[…3…]` dengan function **colsplit** dan **paste** dari praktek sebelumnya.

Jika berjalan lancar maka output yang dihasilkan adalah sebagai berikut.

```
> data.pelanggan$tanggal_lahir
 [1] "13-10-1979" "23-10-1991" "28-2-1969"  "20-2-1970"  "1-1-2001"
 [6] "1-1-2001"   "26-8-1983"  "17-7-1987"  "7-7-1977"   "8-2-1967"
[11] "14-11-1987" "12-7-1977"  "8-19-1950"  "31-1-2001"  "28-2-1969"
[16] "28-2-1969"  "12-7-1977"  "25-2-1987"  "17-7-1987"  "12-1-1972"
[21] "25-6-1987"  "25-7-1974"  "5-9-1990"   "28-2-1969"  "14-1-1988"
[26] "18-8-1988"  "1-1-2001"   "1-12-1964"  "31-1-2001"  "24-9-1990"
[31] "24-2-1978"  "29-2-1969"  "12-1-1971"  "31-1-2001"  "8-8-2008"
[36] "23-12-1968" "20-12-1977" "15-2-1997"  "29-11-1967" "30-11-1967"
[41] "7-7-1968"   "7-7-1968"   "20-10-1987" "14-11-1987" "21-1-2001"
[46] "23-6-1968"  "29-12-1963" "20-6-2001"  "15-2-1997"  "1-1-2001"
[51] "29-12-1967" "20-10-1987" "17-8-1986"  "23-10-1995" "3-10-1988"
[56] "28-2-1969"  "23-10-1979" "7-7-1968"   "20-2-1970"
```

Terlihat data tanggal lahir untuk porsi data yang memiliki tanda garis miring sekarang sudah terstandarisai sesuai keinginan kita.

Code Editor

```
library(RMySQL)

library(reshape2)


#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo",
host="mysqlhost",dbname="dqlabdatawrangling")


sql <- "select tanggal_lahir from dqlab_messy_data where tanggal_lahir like '%/%'"


rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

data.pelanggan <- fetch(rs, n=-1)

dbClearResult(rs)


#Melakukan split dan menyimpannya ke variable tanggal.split dengan urutan bulan, hari
dan tahun

tanggal.split <- colsplit(data.pelanggan$tanggal_lahir,"/",c("bulan","hari","tahun"))


#Memperbaiki data tahun dengan format satu dan dua digit angka dengan sapply

tanggal.split$tahun <- sapply(tanggal.split$tahun, function(x) if(x>=0 & x<10) 2000+x
else if(x>=10 & x<100) 1900+x else x)


#Menggabungkan kembali dalam urutan hari, bulan dan tahun dengan tanda separator
"-" dan menyimpannya kembali ke data.pelanggan$tanggal_lahir

data.pelanggan$tanggal_lahir <- paste(tanggal.split$hari, tanggal.split$bulan,
tanggal.split$tahun, sep="-")

data.pelanggan$tanggal_lahir
```

```
#Menutup seluruh koneksi MySQL
all_cons <- dbListConnections(MySQL())
for(con in all_cons)
    + dbDisconnect(con)
```

Console

```
> library(RMySQL)

> library(reshape2)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dq
labdatawrangling")

> sql <- "select tanggal_lahir from dqlab_messy_data where tanggal_lahir like '%/%'"

> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> data.pelanggan <- fetch(rs, n=-1)

> dbClearResult(rs)
[1] TRUE

> #Melakukan split dan menyimpannya ke variable tanggal.split dengan urutan bulan, ha
ri dan tahun
> tanggal.split <- colsplit(data.pelanggan$tanggal_lahir,"/",c("bulan","hari","tahun"
))

> #Memperbaiki data tahun dengan format satu dan dua digit angka dengan sapply
> tanggal.split$tahun <- sapply(tanggal.split$tahun, function(x) if(x>=0 & x<10) 2000
+x else if(x>=10 & x<100) 1900+x else x)

> #Menggabungkan kembali dalam urutan hari, bulan dan tahun dengan tanda separator "-
" dan menyimpannya kembali ke data.pelanggan$tanggal_lahir
> data.pelanggan$tanggal_lahir <- paste(tanggal.split$hari, tanggal.split$bulan, tang
gal.split$tahun, sep="-")

> data.pelanggan$tanggal_lahir
 [1] "13-10-1979" "23-10-1991" "28-2-1969"  "20-2-1970"  "1-1-2001"
 [6] "1-1-2001"   "26-8-1983"  "17-7-1987"  "7-7-1977"   "8-2-1967"
[11] "14-11-1987" "12-7-1977"  "8-19-1950"  "31-1-2001"  "28-2-1969"
[16] "28-2-1969"  "12-7-1977"  "25-2-1987"  "17-7-1987"  "12-1-1972"
[21] "25-6-1987"  "25-7-1974"  "5-9-1990"   "28-2-1969"  "14-1-1988"
[26] "18-8-1988"  "1-1-2001"   "1-12-1964"  "31-1-2001"  "24-9-1990"
[31] "24-2-1978"  "29-2-1969"  "12-1-1971"  "31-1-2001"  "8-8-2008"
[36] "23-12-1968" "20-12-1977" "15-2-1997"  "29-11-1967" "30-11-1967"
[41] "7-7-1968"   "7-7-1968"   "20-10-1987" "14-11-1987" "21-1-2001"
```

```
[46] "23-6-1968"  "29-12-1963" "20-6-2001"  "15-2-1997"  "1-1-2001"
[51] "29-12-1967" "20-10-1987" "17-8-1986"  "23-10-1995" "3-10-1988"
[56] "28-2-1969"  "23-10-1979" "7-7-1968"   "20-2-1970"

> #Menutup seluruh koneksi MySQL
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons)
+     + dbDisconnect(con)
```

Menggabungkan data standarisasi dengan rbind

Perhatikan query untuk mengambil data yang digunakan pada dua praktek sebelumnya mewakili dua segmen dataset yang berbeda.

Dua Porsi Data

Standarisasi perlu dilakukan per segmen, dan pada akhirnya perlu digabungkan. Kita akan gunakan dua query berikut, satunya adalah untuk query yang mengembalikan hasil dimana tanggal mengandung garis miring sebagai berikut:

```
    select tanggal_lahir from dqlab_messy_data where
tanggal_lahir like '%/%'
```

Ini akan kita lakukan standarisasi dengan melakukan pemisahan dan penggabungan kolom.

Dan untuk query yang tidak mengandung garis miring sebagai berikut

```
    select tanggal_lahir from dqlab_messy_data where not
tanggal_lahir like '%/%'
```

Porsi data ini akan mengembalikan dua format:

- data yang memiliki nama bulan
- dan data tanggal dengan format dd-MM-yyyy.

Dengan demikian, standarisasi yang kita lakukan akan sama dengan praktek standarisasi simple rule untuk nama bulan dimana porsi format dd-MM-yyyy tidak akan terkena efek apapun.

RBind

Function rbind – yang merupakan singkatan dari row bind – digunakan untuk menggabungkan data secara vertikal dari baris-baris data yang memiliki struktur kolom dan tipe data yang sama perlu.

Syntaxnya sangat simple, isi argumennya adalah kumpulan vector, data frame, ataupun matrix.

```
    rbind(data.frame1, data.frame2, …)
```

Hasilnya adalah vector, data.frame ataupun matrix yang sudah tergabung.

Tugas Praktek

Pada praktek kali ini kita akan melakukan proses-proses berikut:

- Melakukan dua query ke database untuk mendapatkan dua porsi data: satu yang mengandung garis miring dan satu yang tidak.

- Query pertama adalah berikut, hasilnya akan dimasukkan ke variable **pelanggan1**.

```
select kode_pelanggan, tanggal_lahir from dqlab_messy_data where
tanggal_lahir like '%/%'
```

- Query kedua adalah berikut, hasilnya akan dimasukkan ke variable **pelanggan2**.

```
select kode_pelanggan, tanggal_lahir from dqlab_messy_data where
not tanggal_lahir like '%/%'
```

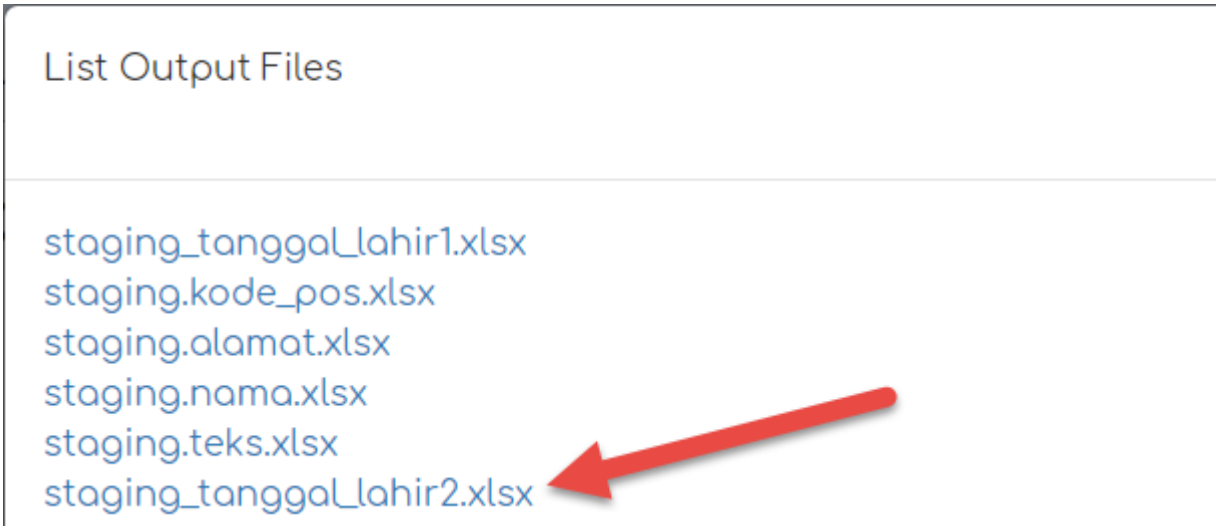
- Setelah itu tiap porsi data akan dilakukan standarisasi berikut
    - Untuk data.pelanggan1 akan dilakukan standarisasi menggunakan **split**, **paste** dan **sapply**.
    - Untuk data.pelanggan2 akan dilakukan standarisasi nama bulan menjadi angka menggunakan *simple rule* dengan **gsub**.
- Setelah standarisasi kedua dataset ini akan digabungkan kembali menggunakan **rbind**.
- Hasil penggabungan ini akan kita tuliskan di file Excel bernama **xlsx** dengan function **write.xlsx**.

Hampir seluruh code tersebut sudah ada pada code editor dengan keterangan pada comment. Lengkapi bagian **rbind** dengan variable data yang sesuai dengan mengganti […1…] dan […2…].

Jika berjalan dengan lancar maka output di console adalah sebagai berikut.

```
> data.gabungan
    kode_pelanggan tanggal_lahir
1         KD-00056    13-10-1979
2         KD-00002    23-10-1991
3         KD-00075     28-2-1969
4         KD-00076     20-2-1970
…
152       KD-00077     7-07-1968
153       KD-00142    14-12-2003
154       KD-00192    26-08-1983
155       KD-00492    21-05-1980
```

Dan output file pada window "List Output Files" akan muncul file **"staging_tanggal_lahir1.xlsx"**.



Code Editor

```
library(RMySQL)

library(reshape2)

library(openxlsx)


#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo",
host="mysqlhost",dbname="dqlabdatawrangling")


#Mengambil data yang memiliki tanda garis miring /

sql <- "select kode_pelanggan, tanggal_lahir from dqlab_messy_data where
tanggal_lahir like '%/%'"


rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

data.pelanggan1 <- fetch(rs, n=-1)

dbClearResult(rs)

#Melakukan split dan menyimpannya ke variable tanggal.split dengan urutan bulan, hari
dan tahun
```

```
tanggal.split <- colsplit(data.pelanggan1$tanggal_lahir,"/",c("bulan","hari","tahun"))


#Memperbaiki data tahun dengan format satu dan dua digit angka

tanggal.split$tahun <- sapply(tanggal.split$tahun, function(x) if(x>=0 & x<10) 2000+x
else if(x>=10 & x<100) 1900+x else x)


#Menggabungkan kembali dalam urutan hari, bulan dan tahun dan menyimpannya
kembali ke data.pelanggan$tanggal_lahir

data.pelanggan1$tanggal_lahir <- paste(tanggal.split$hari, tanggal.split$bulan,
tanggal.split$tahun, sep="-")


#Mengambil data yang tidak memiliki tanda garis miring /

sql <- "select kode_pelanggan, tanggal_lahir from dqlab_messy_data where not
tanggal_lahir like '%/%'"


rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

data.pelanggan2 <- fetch(rs, n=-1)

dbClearResult(rs)


#Mengganti Januari s/d Desember menjadi angka

data.pelanggan2$tanggal_lahir <- gsub(" ","", data.pelanggan2$tanggal_lahir)


data.pelanggan2$tanggal_lahir <- gsub("Januari","-01-",
data.pelanggan2$tanggal_lahir)

data.pelanggan2$tanggal_lahir <- gsub("Februari","-02-",
data.pelanggan2$tanggal_lahir)

data.pelanggan2$tanggal_lahir <- gsub("Maret","-03-", data.pelanggan2$tanggal_lahir)

data.pelanggan2$tanggal_lahir <- gsub("April","-04-", data.pelanggan2$tanggal_lahir)

data.pelanggan2$tanggal_lahir <- gsub("Mei","-05-", data.pelanggan2$tanggal_lahir)

data.pelanggan2$tanggal_lahir <- gsub("Juni","-06-", data.pelanggan2$tanggal_lahir)
```

```
data.pelanggan2$tanggal_lahir <- gsub("Juli","-07-", data.pelanggan2$tanggal_lahir)

data.pelanggan2$tanggal_lahir <- gsub("Agustus","-08-",
data.pelanggan2$tanggal_lahir)

data.pelanggan2$tanggal_lahir <- gsub("September","-09-",
data.pelanggan2$tanggal_lahir)

data.pelanggan2$tanggal_lahir <- gsub("Oktober","-10-",
data.pelanggan2$tanggal_lahir)

data.pelanggan2$tanggal_lahir <- gsub("November","-11-",
data.pelanggan2$tanggal_lahir)

data.pelanggan2$tanggal_lahir <- gsub("Desember","-12-",
data.pelanggan2$tanggal_lahir)

#Menggabungkan dua porsi data pelanggan secara vertikal dengan rbind sesuai urutan
porsi data yang diquery

data.gabungan <- rbind(data.pelanggan1, data.pelanggan2)

data.gabungan

write.xlsx(data.gabungan, file="staging_tanggal_lahir1.xlsx")


#Menutup seluruh koneksi MySQL

all_cons <- dbListConnections(MySQL())

for(con in all_cons)

   + dbDisconnect(con)
```

Console

```
> library(RMySQL)

> library(reshape2)

> library(openxlsx)

> #Membuka koneksi
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dq
labdatawrangling")

> #Mengambil data yang memiliki tanda garis miring /
> sql <- "select kode_pelanggan, tanggal_lahir from dqlab_messy_data where tanggal_la
hir like '%/%'"

> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> data.pelanggan1 <- fetch(rs, n=-1)

> dbClearResult(rs)
[1] TRUE

> #Melakukan split dan menyimpannya ke variable tanggal.split dengan urutan bulan, ha
ri dan tahun
> tanggal.split <- colsplit(data.pelanggan1$tanggal_lahir,"/",c("bulan","hari","tahun
"))

> #Memperbaiki data tahun dengan format satu dan dua digit angka
> tanggal.split$tahun <- sapply(tanggal.split$tahun, function(x) if(x>=0 & x<10) 2000
+x else if(x>=10 & x<100) 1900+x else x)

> #Menggabungkan kembali dalam urutan hari, bulan dan tahun dan menyimpannya kembali
ke data.pelanggan$tanggal_lahir
> data.pelanggan1$tanggal_lahir <- paste(tanggal.split$hari, tanggal.split$bulan, tan
ggal.split$tahun, sep="-")

> #Mengambil data yang tidak memiliki tanda garis miring /
> sql <- "select kode_pelanggan, tanggal_lahir from dqlab_messy_data where not tangga
l_lahir like '%/%'"

> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> data.pelanggan2 <- fetch(rs, n=-1)

> dbClearResult(rs)
[1] TRUE

> #Mengganti Januari s/d Desember menjadi angka
> data.pelanggan2$tanggal_lahir <- gsub(" ","", data.pelanggan2$tanggal_lahir)
```

```
> data.pelanggan2$tanggal_lahir <- gsub("Januari","-01-", data.pelanggan2$tanggal_lah
ir)

> data.pelanggan2$tanggal_lahir <- gsub("Februari","-02-", data.pelanggan2$tanggal_la
hir)

> data.pelanggan2$tanggal_lahir <- gsub("Maret","-03-", data.pelanggan2$tanggal_lahir
)

> data.pelanggan2$tanggal_lahir <- gsub("April","-04-", data.pelanggan2$tanggal_lahir
)

> data.pelanggan2$tanggal_lahir <- gsub("Mei","-05-", data.pelanggan2$tanggal_lahir)

> data.pelanggan2$tanggal_lahir <- gsub("Juni","-06-", data.pelanggan2$tanggal_lahir)

> data.pelanggan2$tanggal_lahir <- gsub("Juli","-07-", data.pelanggan2$tanggal_lahir)

> data.pelanggan2$tanggal_lahir <- gsub("Agustus","-08-", data.pelanggan2$tanggal_lah
ir)

> data.pelanggan2$tanggal_lahir <- gsub("September","-09-", data.pelanggan2$tanggal_l
ahir)

> data.pelanggan2$tanggal_lahir <- gsub("Oktober","-10-", data.pelanggan2$tanggal_lah
ir)

> data.pelanggan2$tanggal_lahir <- gsub("November","-11-", data.pelanggan2$tanggal_la
hir)

> data.pelanggan2$tanggal_lahir <- gsub("Desember","-12-", data.pelanggan2$tanggal_la
hir)

> #Menggabungkan dua porsi data pelanggan secara vertikal dengan rbind sesuai urutan
porsi data yang diquery
> data.gabungan <- rbind(data.pelanggan1, data.pelanggan2)

> data.gabungan
    kode_pelanggan tanggal_lahir
1         KD-00056    13-10-1979
2         KD-00002    23-10-1991
3         KD-00075     28-2-1969
4         KD-00076     20-2-1970
5         KD-00088      1-1-2001
6         KD-00119      1-1-2001
7         KD-00096     26-8-1983
8         KD-00090     17-7-1987
9         KD-00045      7-7-1977
10        KD-00012      8-2-1967
11        KD-00064    14-11-1987
12        KD-00038     12-7-1977
13        KD-00117     8-19-1950
14        KD-00125     31-1-2001
15        KD-00114     28-2-1969
16        KD-00062     28-2-1969
```

17	KD-00024	12-7-1977
18	KD-00084	25-2-1987
19	KD-00143	17-7-1987
20	KD-00034	12-1-1972
21	KD-00087	25-6-1987
22	KD-00043	25-7-1974
23	KD-00050	5-9-1990
24	KD-00049	28-2-1969
25	KD-00124	14-1-1988
26	KD-00105	18-8-1988
27	KD-00107	1-1-2001
28	KD-00102	1-12-1964
29	KD-00146	31-1-2001
30	KD-00130	24-9-1990
31	KD-00127	24-2-1978
32	KD-00057	29-2-1969
33	KD-00023	12-1-1971
34	KD-00136	31-1-2001
35	KD-00145	8-8-2008
36	KD-00058	23-12-1968
37	KD-00144	20-12-1977
38	KD-00052	15-2-1997
39	KD-00120	29-11-1967
40	KD-00089	30-11-1967
41	KD-00112	7-7-1968
42	KD-00098	7-7-1968
43	KD-00100	20-10-1987
44	KD-00121	14-11-1987
45	KD-00131	21-1-2001
46	KD-00097	23-6-1968
47	KD-00071	29-12-1963
48	KD-00150	20-6-2001
49	KD-00067	15-2-1997
50	KD-00091	1-1-2001
51	KD-00147	29-12-1967
52	KD-00081	20-10-1987
53	KD-00109	17-8-1986
54	KD-00014	23-10-1995
55	KD-00037	3-10-1988
56	KD-00108	28-2-1969
57	KD-00007	23-10-1979
58	KD-00085	7-7-1968
59	KD-00298	20-2-1970
60	KD-00032	1-04-2028
61	KD-00053	19-08-1986
62	KD-00133	11-07-1981
63	KD-00111	24-03-1976
64	KD-00036	20-02-1970
65	KD-00126	14-11-1987
66	KD-00137	12-01-1968
67	KD-00046	14-03-1879
68	KD-00027	23-11-1962
69	KD-00035	24-01-1952
70	KD-00113	22-02-2000
71	KD-00099	26-08-1983

```
72        KD-00132    24-01-1987
73        KD-00139    21-05-1980
74        KD-00074     1-12-1964
75        KD-00021    14-03-1979
76        KD-00030    28-02-1969
77        KD-00129    23-04-1978
78        KD-00122    20-06-2001
79        KD-00059    05-07-1987
80        KD-00079    05-12-1979
81        KD-00134    14-07-1977
82        KD-00010    23-10-1991
83        KD-00028    23-10-1991
84        KD-00069    24-06-1992
85        KD-00006    05-09-1990
86        KD-00104    17-08-1986
87        KD-00103    30-11-1954
88        KD-00039    05-09-1990
89         KD-0047    19-03-1950
90        KD-00149    12-01-1968
91        KD-00003    23-11-1962
92        KD-00135     8-03-1955
93        KD-00110    12-12-1950
94        KD-00141    30-11-1954
95        KD-00044    21-05-1980
96        KD-00086    13-11-1962
97        KD-00123    13-11-1963
98        KD-00025    19-03-1950
99        KD-00008    22-07-1973
100       KD-00005    19-08-1986
101       KD-00101    19-08-1950
102       KD-00001     8-02-1967
103       KD-00020    13-11-1962
104       KD-00080    13-11-1962
105       KD-00048    29-02-1969
106       KD-00019    23-11-1962
107       KD-00151    29-03-1967
108       KD-00073    26-01-1979
109       KD-00778    08-02-1967
110       KD-00066    19-03-1905
111       KD-00041    19-03-1950
112       KD-00140    22-12-1993
113       KD-00116    20-12-1977
114       KD-00016    28-05-1969
115       KD-00063    29-02-1969
116       KD-00148    17-08-1986
117       KD-00029    13-11-1962
118       KD-00106    30-11-1954
119       KD-00026    12-01-1969
120       KD-00018    19-03-1950
121       KD-00051    17-09-1982
122       KD-00128     8-03-1955
123       KD-00115     8-03-1955
124       KD-00009             -
125       KD-00092    22-11-1979
126       KD-00070    10-10-1982
```

```
127       KD-00118    04-07-1987
128       KD-00055    29-02-1976
129       KD-00042    17-09-1982
130       KD-00033    21-05-1981
131       KD-00013    22-04-1933
132       KD-00138    12-12-1987
133       KD-00094    16-06-1975
134       KD-00054    01-01-1982
135       KD-00061    30-11-1954
136       KD-00031    27-02-1976
137       KD-00040    12-01-1971
138       KD-00068    05-06-1979
139       KD-00004    17-02-2097
140       KD-00093    30-11-1954
141       KD-00082    26-11-1983
142       KD-00065    14-03-1879
143       KD-00011    12-01-1969
144       KD-00072    20-11-1987
145       KD-00078    26-11-1983
146       KD-00095     1-12-1964
147       KD-00022    25-07-1974
148       KD-00017    23-11-1962
149       KD-00015    09-08-1972
150       KD-00083    19-03-1950
151       KD-00060    24-09-1990
152       KD-00077     7-07-1968
153       KD-00142    14-12-2003
154       KD-00192    26-08-1983
155       KD-00492    21-05-1980

> write.xlsx(data.gabungan, file="staging_tanggal_lahir1.xlsx")

> #Menutup seluruh koneksi MySQL
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons)
+     + dbDisconnect(con)
```

Menggunakan as.Date untuk pengecekan konversi keseluruhan

Dari hasil standarisasi praktek terakhir terlihat ada masalah di data tanggal lahir ini. Untuk melihat data mana yang bermasalah, kita coba gunakan kembalian error dari function **as.Date** – yang mencoba melakukan konversi terhadap kolom tanggal lahir.

Syntax dari function **as.Date** adalah sebagai berikut.

```
as.Date(data, format)
```

dimana:

- **data**: teks yang akan dikonversi menjadi tipe data Date.
- **format**: pola format data tanggal dengan elemen berikut.

Simbol	Representasi	Contoh
%d	hari (0-31)	01-31
%m	bulan (00-12)	00-12
%b %B	Nama bulan yang dipersingkat (Inggris) Nama bulan penuh (Inggris)	Feb February
%y %Y	2-digit tahun 4-digit tahun	92 1992

Dan menyesuaikan kebutuhan kita – dengan anggapan format yang kita butuhkan untuk standarisasi dd-MM-yyyy – maka function as.Date adalah sebagai berikut.

```
as.Date(data.pelanggan$tanggal_lahir, "%d-%m-%Y")
```

Function ini akan mengembalikan tanggal jika berhasil atau *missing value* NA jika tidak berhasil.

Seluruh hasil as.Date perlu dikonversi lagi ke dalam standar format kita dengan tambahan function format berikut.

```
format(as.Date(data.pelanggan$tanggal_lahir, "%d-%m-%Y"),"%d-%m-%Y")
```

Tugas Praktek

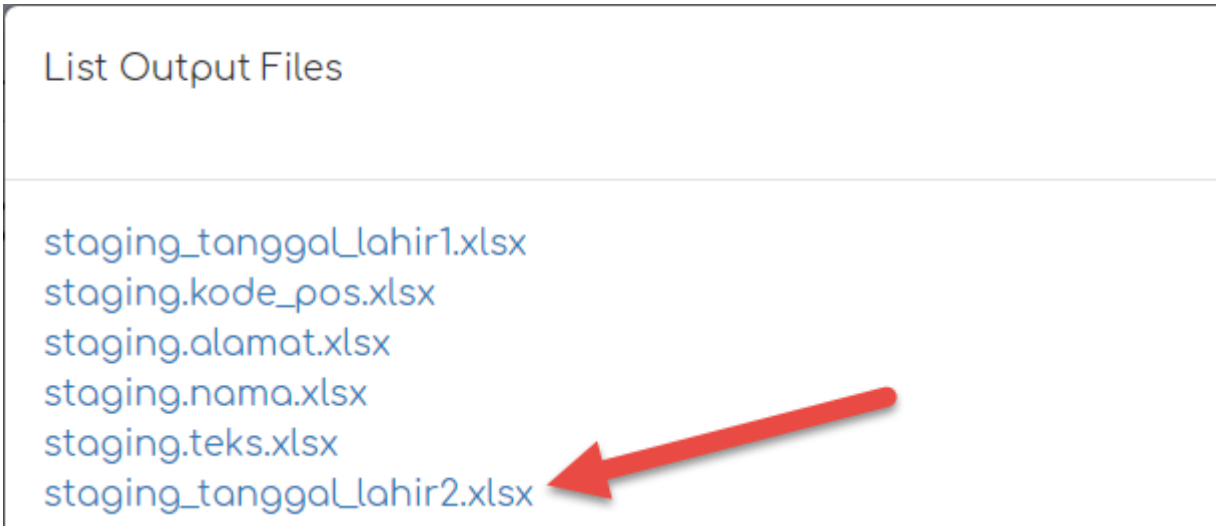
Kita akan membaca file staging hasil standarisasi dari praktek terakhir, kemudian cek valid tidaknya kolom **tanggal_lahir** dengan function **as.Date**, dan kemudian menuliskan ke dalam file **staging_tanggal_lahir2.xlsx**.

Ganti bagian […1…] dengan melakukan konversi as.Date dan lakukan format ulang dengan function **format**.

Jika berjalan dengan baik maka outputnya adalah sebagai berikut.

```
> data.pelanggan
    kode_pelanggan tanggal_lahir
1         KD-00056    13-10-1979
2         KD-00002    23-10-1991
3         KD-00075    28-02-1969
…
11        KD-00064    14-11-1987
12        KD-00038    12-07-1977
13        KD-00117
14        KD-00125    31-01-2001
…
32        KD-00057
…
115       KD-00063
…
152       KD-00077    07-07-1968
153       KD-00142    14-12-2003
154       KD-00192    26-08-1983
155       KD-00492    21-05-1980
```

Dan hasil dari "List Output Files" adalah sebagai berikut.



Catatan: perhatikan kalau kita tidak melakukan query ke MySQL lagi.

Code Editor

```
library(openxlsx)

#Membaca file staging Excel hasil standarisasi tanggal lahir

data.pelanggan <- read.xlsx("staging_tanggal_lahir1.xlsx")


#Menggunakan as.Date untuk melakukan konversi kolom tanggal_lahir dan menyimpannya kembali ke kolom tersebut

data.pelanggan$tanggal_lahir <- format(as.Date(data.pelanggan$tanggal_lahir, "%d-%m-%Y"),"%d-%m-%Y")

data.pelanggan


#Menulis hasil ke file staging_tanggal_lahir2.xlsx

write.xlsx(data.pelanggan, file="staging_tanggal_lahir2.xlsx")
```

Console

```
> library(openxlsx)

> #Membaca file staging Excel hasil standarisasi tanggal lahir
> data.pelanggan <- read.xlsx("staging_tanggal_lahir1.xlsx")

> #Menggunakan as.Date untuk melakukan konversi kolom tanggal_lahir dan menyimpannya
kembali ke kolom tersebut
> data.pelanggan$tanggal_lahir <- format(as.Date(data.pelanggan$tanggal_lahir, "%d-%m
-%Y"),"%d-%m-%Y")

> data.pelanggan
    kode_pelanggan tanggal_lahir
1         KD-00056    13-10-1979
2         KD-00002    23-10-1991
3         KD-00075    28-02-1969
4         KD-00076    20-02-1970
5         KD-00088    01-01-2001
6         KD-00119    01-01-2001
7         KD-00096    26-08-1983
8         KD-00090    17-07-1987
9         KD-00045    07-07-1977
10        KD-00012    08-02-1967
11        KD-00064    14-11-1987
12        KD-00038    12-07-1977
13        KD-00117          <NA>
14        KD-00125    31-01-2001
15        KD-00114    28-02-1969
16        KD-00062    28-02-1969
17        KD-00024    12-07-1977
18        KD-00084    25-02-1987
19        KD-00143    17-07-1987
20        KD-00034    12-01-1972
21        KD-00087    25-06-1987
22        KD-00043    25-07-1974
23        KD-00050    05-09-1990
24        KD-00049    28-02-1969
25        KD-00124    14-01-1988
26        KD-00105    18-08-1988
27        KD-00107    01-01-2001
28        KD-00102    01-12-1964
29        KD-00146    31-01-2001
30        KD-00130    24-09-1990
31        KD-00127    24-02-1978
32        KD-00057          <NA>
33        KD-00023    12-01-1971
34        KD-00136    31-01-2001
35        KD-00145    08-08-2008
36        KD-00058    23-12-1968
37        KD-00144    20-12-1977
38        KD-00052    15-02-1997
39        KD-00120    29-11-1967
40        KD-00089    30-11-1967
```

41	KD-00112	07-07-1968
42	KD-00098	07-07-1968
43	KD-00100	20-10-1987
44	KD-00121	14-11-1987
45	KD-00131	21-01-2001
46	KD-00097	23-06-1968
47	KD-00071	29-12-1963
48	KD-00150	20-06-2001
49	KD-00067	15-02-1997
50	KD-00091	01-01-2001
51	KD-00147	29-12-1967
52	KD-00081	20-10-1987
53	KD-00109	17-08-1986
54	KD-00014	23-10-1995
55	KD-00037	03-10-1988
56	KD-00108	28-02-1969
57	KD-00007	23-10-1979
58	KD-00085	07-07-1968
59	KD-00298	20-02-1970
60	KD-00032	01-04-2028
61	KD-00053	19-08-1986
62	KD-00133	11-07-1981
63	KD-00111	24-03-1976
64	KD-00036	20-02-1970
65	KD-00126	14-11-1987
66	KD-00137	12-01-1968
67	KD-00046	14-03-1879
68	KD-00027	23-11-1962
69	KD-00035	24-01-1952
70	KD-00113	22-02-2000
71	KD-00099	26-08-1983
72	KD-00132	24-01-1987
73	KD-00139	21-05-1980
74	KD-00074	01-12-1964
75	KD-00021	14-03-1979
76	KD-00030	28-02-1969
77	KD-00129	23-04-1978
78	KD-00122	20-06-2001
79	KD-00059	05-07-1987
80	KD-00079	05-12-1979
81	KD-00134	14-07-1977
82	KD-00010	23-10-1991
83	KD-00028	23-10-1991
84	KD-00069	24-06-1992
85	KD-00006	05-09-1990
86	KD-00104	17-08-1986
87	KD-00103	30-11-1954
88	KD-00039	05-09-1990
89	KD-0047	19-03-1950
90	KD-00149	12-01-1968
91	KD-00003	23-11-1962
92	KD-00135	08-03-1955
93	KD-00110	12-12-1950
94	KD-00141	30-11-1954
95	KD-00044	21-05-1980

```
96        KD-00086    13-11-1962
97        KD-00123    13-11-1963
98        KD-00025    19-03-1950
99        KD-00008    22-07-1973
100       KD-00005    19-08-1986
101       KD-00101    19-08-1950
102       KD-00001    08-02-1967
103       KD-00020    13-11-1962
104       KD-00080    13-11-1962
105       KD-00048          <NA>
106       KD-00019    23-11-1962
107       KD-00151    29-03-1967
108       KD-00073    26-01-1979
109       KD-00778    08-02-1967
110       KD-00066    19-03-1905
111       KD-00041    19-03-1950
112       KD-00140    22-12-1993
113       KD-00116    20-12-1977
114       KD-00016    28-05-1969
115       KD-00063          <NA>
116       KD-00148    17-08-1986
117       KD-00029    13-11-1962
118       KD-00106    30-11-1954
119       KD-00026    12-01-1969
120       KD-00018    19-03-1950
121       KD-00051    17-09-1982
122       KD-00128    08-03-1955
123       KD-00115    08-03-1955
124       KD-00009          <NA>
125       KD-00092    22-11-1979
126       KD-00070    10-10-1982
127       KD-00118    04-07-1987
128       KD-00055    29-02-1976
129       KD-00042    17-09-1982
130       KD-00033    21-05-1981
131       KD-00013    22-04-1933
132       KD-00138    12-12-1987
133       KD-00094    16-06-1975
134       KD-00054    01-01-1982
135       KD-00061    30-11-1954
136       KD-00031    27-02-1976
137       KD-00040    12-01-1971
138       KD-00068    05-06-1979
139       KD-00004    17-02-2097
140       KD-00093    30-11-1954
141       KD-00082    26-11-1983
142       KD-00065    14-03-1879
143       KD-00011    12-01-1969
144       KD-00072    20-11-1987
145       KD-00078    26-11-1983
146       KD-00095    01-12-1964
147       KD-00022    25-07-1974
148       KD-00017    23-11-1962
149       KD-00015    09-08-1972
150       KD-00083    19-03-1950
```

```
151       KD-00060    24-09-1990
152       KD-00077    07-07-1968
153       KD-00142    14-12-2003
154       KD-00192    26-08-1983
155       KD-00492    21-05-1980

> #Menulis hasil ke file staging_tanggal_lahir2.xlsx
> write.xlsx(data.pelanggan, file="staging_tanggal_lahir2.xlsx")
```

Mengidentifikasi Tanggal Lahir Tidak Logis

Sampai tahap ini, kita bisa berkesimpulan kalau seluruh tanggal sudah distandarisasi dengan baik, dimana tanggal yang tidak valid dikonversi menjadi missing value (NA).

Tahap berikutnya berkaitan dengan proses bisnis. Katakanlah bisnis kita baru berdiri 5 tahun dan hanya menerima pelanggan maksimum 75 tahun. Dengan demikian maksimum umur pelanggan yang tercatat haruslah 80 tahun (75 + 5). Jika lebih dari itu maka diasumsikan salah catat.

Untuk menghitung umur maka kita sebenarnya menghitung perbedaan diantara tanggal lahir dengan tanggal sekarang atau tanggal referensi yang kita gunakan. Tujuan ini bisa tercapati dengan menggunakan function difftime.

Berikut adalah contoh penggunaannya.

```
    difftime(tanggal_referensi , data.pelanggan$tanggal_lahir ,
units = "days")
```

dimana:

- **difftime**: adalah function untuk menghitung perbedaan dua tanggal.
- **tanggal_referensi**: variable dengan tipe data Date yang akan menjadi referensi untuk menghitung umur.
- **pelanggan$tanggal_lahir**: variable dengan tipe data Date yang akan menjadi pembanding.
- **unit = "days"**: perbedaan dalam unit hari (days). Selain hari kita bisa memasukkan detik (secs), menit (mins), jam (hours), dan minggu (weeks).

Perhatikan, karena tidak ada tahun – maka hitungan difftime ini perlu kita konversi ke tahun dengan menggunakna function **as.Numeric** dan kemudian dibagi dengan angka 365. Lengkapnya sebagai berikut.

```
    as.numeric(difftime(tanggal_referensi ,
data.pelanggan$tanggal_lahir , units = "days"))/365
```

dimana:

- **numeric**: adalah function untuk melakukan konversi data ke numerik.
- **/365** : membagi angka dengan 365.

Tugas Praktek

Pada code editor telah dimasukkan potongan code untuk membaca file **staging_tanggal_lahir2.xlsx** dan dimasukkan ke dalam variable **data.pelanggan**. Karena semua dibaca sebagai char oleh function **read.xlsx**, kolom **tanggal_lahir** pada variable ini kemudian perlu dikonversi lagi menjadi Date dengan menggunakan function **as.Date**.

Selanjutnya, kita perlu memasukkan 27 April 2018 sebagai tanggal referensi menggantikan bagian `[…1…]` pada code editor. Dan kemudian ganti bagian `[…2…]` dengan kombinasi function **difftime**, **as.numeric**, dan pembagian dengan 365 untuk menghitung umur berdasarkan perbedaan **tanggal_lahir** dengan **tanggal_referensi**.

Pada code editor telah dimasukkan pengecekan apakah umur di atas 80 tahun dengan perintah berikut.

```
data.pelanggan$umur_valid <- data.pelanggan$umur <= 80
```

Dan terakhir, hasil akan disimpan pada satu file bernama **staging_tanggal_lahir3.xlsx**.

Jika semua berjalan dengan lancar maka hasilnya akan terlihat sebagai sebagai berikut. Perhatikan ada tambahan kolom umur dan umur_valid. Untuk umur_valid yang false, kita perhatikan umur dan tanggal lahirnya – pada tampilan telah ditandai dengan warna font merah.

```
> data.pelanggan
    kode_pelanggan tanggal_lahir       umur umur_valid
1         KD-00056    13-10-1979  38.564384       TRUE
2         KD-00002    23-10-1991  26.528767       TRUE
…
…
67        KD-00046    14-03-1879 139.213699      FALSE
…
…
110       KD-00066    19-03-1905 113.183562      FALSE
…
…
131       KD-00013    22-04-1933  85.071233      FALSE
…
…
142       KD-00065    14-03-1879 139.213699      FALSE
…
…
```

Code Editor

```
library(openxlsx)
data.pelanggan <- read.xlsx("staging_tanggal_lahir2.xlsx")

#Membaca data tanggal_lahir sebagai tipe data Date
data.pelanggan$tanggal_lahir <- as.Date(data.pelanggan$tanggal_lahir, "%d-%m-%Y")

#Set tanggal referensi ke 27 April 2018
tanggal_referensi <- as.Date("27-4-2018", "%d-%m-%Y")

#Menghitung perbedaan tanggal dalam tahun
data.pelanggan$umur <- as.numeric(difftime(tanggal_referensi,
data.pelanggan$tanggal_lahir, units="days"))/365

#Pengecekan umur maksimal 80 tahun
data.pelanggan$umur_valid <- data.pelanggan$umur <= 80

#Format ulang tanggal lahir
data.pelanggan$tanggal_lahir <- format(data.pelanggan$tanggal_lahir, "%d-%m-%Y")

#Menampilkan data.pelanggan
data.pelanggan

#Menulis hasil ke file staging_tanggal_lahir3.xlsx
write.xlsx(data.pelanggan, file="staging_tanggal_lahir3.xlsx")
```

Console

```
> library(openxlsx)

> data.pelanggan <- read.xlsx("staging_tanggal_lahir2.xlsx")

> #Membaca data tanggal_lahir sebagai tipe data Date
> data.pelanggan$tanggal_lahir <- as.Date(data.pelanggan$tanggal_lahir, "%d-%m-%Y")

> #Set tanggal referensi ke 27 April 2018
> tanggal_referensi <- as.Date("27-4-2018", "%d-%m-%Y")

> #Menghitung perbedaan tanggal dalam tahun
> data.pelanggan$umur <- as.numeric(difftime(tanggal_referensi, data.pelanggan$tangga
l_lahir, units="days"))/365

> #Pengecekan umur maksimal 80 tahun
> data.pelanggan$umur_valid <- data.pelanggan$umur <= 80

> #Format ulang tanggal lahir
> data.pelanggan$tanggal_lahir <- format(data.pelanggan$tanggal_lahir, "%d-%m-%Y")

> #Menampilkan data.pelanggan
> data.pelanggan
    kode_pelanggan tanggal_lahir       umur umur_valid
1         KD-00056    13-10-1979  38.564384       TRUE
2         KD-00002    23-10-1991  26.528767       TRUE
3         KD-00075    28-02-1969  49.191781       TRUE
4         KD-00076    20-02-1970  48.213699       TRUE
5         KD-00088    01-01-2001  17.328767       TRUE
6         KD-00119    01-01-2001  17.328767       TRUE
7         KD-00096    26-08-1983  34.693151       TRUE
8         KD-00090    17-07-1987  30.800000       TRUE
9         KD-00045    07-07-1977  40.832877       TRUE
10        KD-00012    08-02-1967  51.249315       TRUE
11        KD-00064    14-11-1987  30.471233       TRUE
12        KD-00038    12-07-1977  40.819178       TRUE
13        KD-00117          <NA>         NA         NA
14        KD-00125    31-01-2001  17.246575       TRUE
15        KD-00114    28-02-1969  49.191781       TRUE
16        KD-00062    28-02-1969  49.191781       TRUE
17        KD-00024    12-07-1977  40.819178       TRUE
18        KD-00084    25-02-1987  31.189041       TRUE
19        KD-00143    17-07-1987  30.800000       TRUE
20        KD-00034    12-01-1972  46.320548       TRUE
21        KD-00087    25-06-1987  30.860274       TRUE
22        KD-00043    25-07-1974  43.786301       TRUE
23        KD-00050    05-09-1990  27.660274       TRUE
24        KD-00049    28-02-1969  49.191781       TRUE
25        KD-00124    14-01-1988  30.304110       TRUE
26        KD-00105    18-08-1988  29.709589       TRUE
27        KD-00107    01-01-2001  17.328767       TRUE
28        KD-00102    01-12-1964  53.438356       TRUE
29        KD-00146    31-01-2001  17.246575       TRUE
```

30	KD-00130	24-09-1990	27.608219	TRUE
31	KD-00127	24-02-1978	40.197260	TRUE
32	KD-00057	<NA>	NA	NA
33	KD-00023	12-01-1971	47.320548	TRUE
34	KD-00136	31-01-2001	17.246575	TRUE
35	KD-00145	08-08-2008	9.723288	TRUE
36	KD-00058	23-12-1968	49.375342	TRUE
37	KD-00144	20-12-1977	40.378082	TRUE
38	KD-00052	15-02-1997	21.208219	TRUE
39	KD-00120	29-11-1967	50.443836	TRUE
40	KD-00089	30-11-1967	50.441096	TRUE
41	KD-00112	07-07-1968	49.838356	TRUE
42	KD-00098	07-07-1968	49.838356	TRUE
43	KD-00100	20-10-1987	30.539726	TRUE
44	KD-00121	14-11-1987	30.471233	TRUE
45	KD-00131	21-01-2001	17.273973	TRUE
46	KD-00097	23-06-1968	49.876712	TRUE
47	KD-00071	29-12-1963	54.364384	TRUE
48	KD-00150	20-06-2001	16.863014	TRUE
49	KD-00067	15-02-1997	21.208219	TRUE
50	KD-00091	01-01-2001	17.328767	TRUE
51	KD-00147	29-12-1967	50.361644	TRUE
52	KD-00081	20-10-1987	30.539726	TRUE
53	KD-00109	17-08-1986	31.715068	TRUE
54	KD-00014	23-10-1995	22.526027	TRUE
55	KD-00037	03-10-1988	29.583562	TRUE
56	KD-00108	28-02-1969	49.191781	TRUE
57	KD-00007	23-10-1979	38.536986	TRUE
58	KD-00085	07-07-1968	49.838356	TRUE
59	KD-00298	20-02-1970	48.213699	TRUE
60	KD-00032	01-04-2028	-9.936986	TRUE
61	KD-00053	19-08-1986	31.709589	TRUE
62	KD-00133	11-07-1981	36.819178	TRUE
63	KD-00111	24-03-1976	42.120548	TRUE
64	KD-00036	20-02-1970	48.213699	TRUE
65	KD-00126	14-11-1987	30.471233	TRUE
66	KD-00137	12-01-1968	50.323288	TRUE
67	KD-00046	14-03-1879	139.213699	FALSE
68	KD-00027	23-11-1962	55.463014	TRUE
69	KD-00035	24-01-1952	66.301370	TRUE
70	KD-00113	22-02-2000	18.189041	TRUE
71	KD-00099	26-08-1983	34.693151	TRUE
72	KD-00132	24-01-1987	31.276712	TRUE
73	KD-00139	21-05-1980	37.958904	TRUE
74	KD-00074	01-12-1964	53.438356	TRUE
75	KD-00021	14-03-1979	39.147945	TRUE
76	KD-00030	28-02-1969	49.191781	TRUE
77	KD-00129	23-04-1978	40.038356	TRUE
78	KD-00122	20-06-2001	16.863014	TRUE
79	KD-00059	05-07-1987	30.832877	TRUE
80	KD-00079	05-12-1979	38.419178	TRUE
81	KD-00134	14-07-1977	40.813699	TRUE
82	KD-00010	23-10-1991	26.528767	TRUE
83	KD-00028	23-10-1991	26.528767	TRUE
84	KD-00069	24-06-1992	25.857534	TRUE

85	KD-00006	05-09-1990	27.660274	TRUE
86	KD-00104	17-08-1986	31.715068	TRUE
87	KD-00103	30-11-1954	63.449315	TRUE
88	KD-00039	05-09-1990	27.660274	TRUE
89	KD-0047	19-03-1950	68.153425	TRUE
90	KD-00149	12-01-1968	50.323288	TRUE
91	KD-00003	23-11-1962	55.463014	TRUE
92	KD-00135	08-03-1955	63.180822	TRUE
93	KD-00110	12-12-1950	67.419178	TRUE
94	KD-00141	30-11-1954	63.449315	TRUE
95	KD-00044	21-05-1980	37.958904	TRUE
96	KD-00086	13-11-1962	55.490411	TRUE
97	KD-00123	13-11-1963	54.490411	TRUE
98	KD-00025	19-03-1950	68.153425	TRUE
99	KD-00008	22-07-1973	44.794521	TRUE
100	KD-00005	19-08-1986	31.709589	TRUE
101	KD-00101	19-08-1950	67.734247	TRUE
102	KD-00001	08-02-1967	51.249315	TRUE
103	KD-00020	13-11-1962	55.490411	TRUE
104	KD-00080	13-11-1962	55.490411	TRUE
105	KD-00048	<NA>	NA	NA
106	KD-00019	23-11-1962	55.463014	TRUE
107	KD-00151	29-03-1967	51.115068	TRUE
108	KD-00073	26-01-1979	39.276712	TRUE
109	KD-00778	08-02-1967	51.249315	TRUE
110	KD-00066	19-03-1905	113.183562	FALSE
111	KD-00041	19-03-1950	68.153425	TRUE
112	KD-00140	22-12-1993	24.361644	TRUE
113	KD-00116	20-12-1977	40.378082	TRUE
114	KD-00016	28-05-1969	48.947945	TRUE
115	KD-00063	<NA>	NA	NA
116	KD-00148	17-08-1986	31.715068	TRUE
117	KD-00029	13-11-1962	55.490411	TRUE
118	KD-00106	30-11-1954	63.449315	TRUE
119	KD-00026	12-01-1969	49.320548	TRUE
120	KD-00018	19-03-1950	68.153425	TRUE
121	KD-00051	17-09-1982	35.632877	TRUE
122	KD-00128	08-03-1955	63.180822	TRUE
123	KD-00115	08-03-1955	63.180822	TRUE
124	KD-00009	<NA>	NA	NA
125	KD-00092	22-11-1979	38.454795	TRUE
126	KD-00070	10-10-1982	35.569863	TRUE
127	KD-00118	04-07-1987	30.835616	TRUE
128	KD-00055	29-02-1976	42.186301	TRUE
129	KD-00042	17-09-1982	35.632877	TRUE
130	KD-00033	21-05-1981	36.958904	TRUE
131	KD-00013	22-04-1933	85.071233	FALSE
132	KD-00138	12-12-1987	30.394521	TRUE
133	KD-00094	16-06-1975	42.893151	TRUE
134	KD-00054	01-01-1982	36.342466	TRUE
135	KD-00061	30-11-1954	63.449315	TRUE
136	KD-00031	27-02-1976	42.191781	TRUE
137	KD-00040	12-01-1971	47.320548	TRUE
138	KD-00068	05-06-1979	38.920548	TRUE
139	KD-00004	17-02-2097	-78.865753	TRUE

```
140       KD-00093    30-11-1954  63.449315       TRUE
141       KD-00082    26-11-1983  34.441096       TRUE
142       KD-00065    14-03-1879 139.213699      FALSE
143       KD-00011    12-01-1969  49.320548       TRUE
144       KD-00072    20-11-1987  30.454795       TRUE
145       KD-00078    26-11-1983  34.441096       TRUE
146       KD-00095    01-12-1964  53.438356       TRUE
147       KD-00022    25-07-1974  43.786301       TRUE
148       KD-00017    23-11-1962  55.463014       TRUE
149       KD-00015    09-08-1972  45.745205       TRUE
150       KD-00083    19-03-1950  68.153425       TRUE
151       KD-00060    24-09-1990  27.608219       TRUE
152       KD-00077    07-07-1968  49.838356       TRUE
153       KD-00142    14-12-2003  14.378082       TRUE
154       KD-00192    26-08-1983  34.693151       TRUE
155       KD-00492    21-05-1980  37.958904       TRUE

> #Menulis hasil ke file staging_tanggal_lahir3.xlsx
> write.xlsx(data.pelanggan, file="staging_tanggal_lahir3.xlsx")
```

Konsolidasi Data

Tiba saatnya kita konsolidasi data antara hasil standarisasi bab sebelumnya (kolom bertipe teks) dengan bab ini (kolom bertipe tanggal).

Kita akan menggabungkan file "**staging.teks.xlsx**" dan file "**staging_tanggal_lahir3.xlsx**" dengan function merge.

Tugas Praktek

Ganti isi […1…] sampai dengan […3…] pada code editor dengan function yang telah Anda pelajari mendapatkan satu file konsolidasi bernama "staging.final.xlsx" yang dapat Anda download.

Ketika dibuka di aplikasi Microsoft Excel, maka sebagian hasilnya terlihat sebagai berikut.

	A	B	C	D	E	F	G	H	I
1	kode_pelanggan	nama	alamat	no_telepon	anomali_no_telepon	kode_pos	tanggal_lahir	umur	umur_valid
2	KD-00001	Agus Cahyonos	Jalan Pulo Bambu No. 15, Kota Tenggara Lama	08298911112222	TRUE	876511	08-02-1967	51.24931507	TRUE
3	KD-00002	Khairul Nissa	Taman Vivo Indah, Blok AA No. 7	+6287132221371404	FALSE	712983	23-10-1991	26.52876712	TRUE
4	KD-00003	Slamet Wiyanto	Meta Residences, No. 32C	+6285725955303368	FALSE	764550	23-11-1962	55.4630137	TRUE
5	KD-00004	DRS. Maria Simangunsong	Gang Bulan Desember III, No. 9	+6283376770990635	FALSE	967220	17-02-2097	-78.86575342	TRUE
6	KD-00005	Prihatin Setyonugroho	Jalan Tegal Sari Indah, No. D87 -- Kota H	+6286843623971825	FALSE	476511	19-08-1986	31.70958904	TRUE
7	KD-00006	DR. Candra Wijaya	Perum Pluto, Blok C No. 1	+6284063423953696	FALSE	487851	05-09-1990	27.66027397	TRUE
8	KD-00007	Indra Kurniawan, ST	Apartemen Kecapi Indah, Lt. 16 No. 1610	+6283840529196797	FALSE	986455	23-10-1979	38.5369863	TRUE
9	KD-00008	Willy Sanjaya	Kali Mars Cluster, No. 24C	+6285312577710538	FALSE	813444	22-07-1973	44.79452055	TRUE
10	KD-00009	Antonius Winarta	Jalan Kebon Jahe, No. F16 - Kota E	+6282722234294686	FALSE	896555			
11	KD-00010	Sri Wahyuni, Ir	Perum Venus, Gg. Harimau No. 1A	+6284079659289143	FALSE	987453	23-10-1991	26.52876712	TRUE
12	KD-00011	Rosalina Kurnia	Cluster Ikan Mas, Taman Intan No. 2	+6288339032314103	FALSE	967223	12-01-1969	49.32054795	TRUE
13	KD-00012	Cahyono, Agus	Pulo Bambu No. 15, Kota Tenggara Lama	+62829891111222	TRUE	876511	08-02-1967	51.24931507	TRUE
14	KD-00013	Danang Santosa	Jalan Hang Tuah, No. 11, Kota DM	+6282672925000608	FALSE	666122	22-04-1933	85.07123288	FALSE
15	KD-00014	Elisabeth Suryadinata, SKOM, ST	Boulevard Raya Residences, Blok AA2 No. 88	+6285455084014504	FALSE	-	23-10-1995	22.5260274	TRUE
16	KD-00015	Mario Setiawan	Jalan Puri Arteri Raya, No. 88 - Kota T	+62829891111222220	FALSE	876511	09-08-1972	45.74520548	TRUE
17	KD-00016	Indra K.	Jalan Pahlawan, No. 69CCD	+6289222405928430	FALSE	896550	28-05-1969	48.94794521	TRUE
18	KD-00017	Irfan Putra Wijaya	Asrama Pelajar No. 22 A - Pondok Bima Sakti	+6289984358708389	FALSE	768034	23-11-1962	55.4630137	TRUE
19	KD-00018	Sudirman Kartono	Jalan Bintang Supernova, No. 78	+6282283957103749	FALSE	896555	19-03-1950	68.15342466	TRUE

Code Editor

```
library(openxlsx)

#Membaca tiap file staging Excel dan menyimpannya dalam variable bernama awalan staging

staging.teks <- read.xlsx("staging.teks.xlsx")

staging.tanggal <- read.xlsx("staging_tanggal_lahir3.xlsx")


#Menggabungkan variable staging dengan function merge

staging.final <- merge(x=staging.teks, y=staging.tanggal, by.x = "kode_pelanggan", by.y = "kode_pelanggan", all = TRUE)


#Menulis hasil ke file staging.final.xlsx

write.xlsx(staging.final, file="staging.final.xlsx")
```

Console

```
> library(openxlsx)

> #Membaca tiap file staging Excel dan menyimpannya dalam variable bernama awalan sta
ging
> staging.teks <- read.xlsx("staging.teks.xlsx")

> staging.tanggal <- read.xlsx("staging_tanggal_lahir3.xlsx")

> #Menggabungkan variable staging dengan function merge
> staging.final <- merge(x=staging.teks, y=staging.tanggal, by.x = "kode_pelanggan",
by.y = "kode_pelanggan", all = TRUE)

> #Menulis hasil ke file staging.final.xlsx
> write.xlsx(staging.final, file="staging.final.xlsx")
```

Kesimpulan

Pada bab ini Anda telah menyelesaikan standarisasi untuk tanggal lahir, dengan demikian berarti standarisasi dari seluruh kolom data. Berikut adalah rangkuman sebagai pelengkap dari bab sebelumnya. Terlihat walaupun sederhana standarisasi dari tanggal ini, tapi function yang digunakan cukup banyak.

Kolom	Function	Pola Regex	SQL	Deskripsi
Tanggal Lahir	• gsub • colsplit • paste • sapply • if…else… • rbind • merge • as.Date • difftime		• SELECT kode_pelanggan, tanggal_lahir from dqlab_messy_data where tanggal_lahir REGEXP … • SELECT kode_pelanggan, nama from dqlab_messy_data where	Standarisasi disini mengganti nama bulan menjadi angka, melakukan pemisahan dan penggabungan kolom, melakukan konversi dengan tanggal valid dan pengecekan umur yang logis.

Kolom	Function	Pola Regex	SQL	Deskripsi
			tanggal_lahir like … • SELECT kode_pelanggan, tanggal_lahir from dqlab_messy_data where **not** tanggal_lahir like …	

Pada bab ini juga konsolidasi data secara keseluruhan ke dalam satu file yang dapat kita gunakan untuk menjalankan praktek menemukan duplikasi data (data deduplication) dan mealakukan pengisian kolom yang kosong (data enrichment).

Klik tombol Next untuk melanjutkan.

Pendahuluan

Duplikasi data adalah kondisi dimana dalam suatu dataset terdapat lebih dari satu data yang sebenarnya mewakili satu entity tapi tidak berhasil dikelompokkan menjadi satu.

Contohnya adalah sebagai berikut, terdapat tiga data pelanggan dengan variasi nama dan alamat yang berbeda, tapi sebenarnya merujuk ke satu orang. Terlihat kodenya juga berbeda-beda.

kode_pelangga	nama	alamat	no_telepon
KD-00001	Agus Cahyonos	Jalan Pulo Bambu No. 15, Kota Tenggara Lama	082989111122
KD-00012	Cahyono, Agus	Pulo Bambu No. 15, Kota Tenggara Lama	+6282989111
KD-00778	Cahyono Agus H.	Jalan Pulau Bambu No. 15 - Kota Tenggara Lama	+6282989111

Dengan demikian jika kita menganalisa transaksi juga akan terbagi menjadi tiga. Jika duplikasi ini tidak diperbaiki dengan mengelompokkannya jadi satu – maka kita tidak akan pernah mendapatkan *single customer view* atau gambaran menyeluruh untuk pelanggan ini.

Dari sisi bisnis, ini bisa mengakibatkan kesempatan yang hilang (*opportunity lost*). Sebagai contoh, di suatu group bisnis andaikan kita bisa mengintegrasikan seluruh pelanggan, maka kita bisa tahu perilaku belanja tiap orang dan bisa melakukan penawaran yang lebih baik.

Dengan demikian, menekan biaya pemasaran dan malah meningkatkan penjualan. Tetapi, kesempatan untuk mendapatkan privilege ini hilang karena data yang tidak berhasil ditemukan duplikat dan dengan demikian tidak bisa disatukan.

Masalah ini adalah salah satu hal yang paling memusingkan buat banyak pihak, terutama para untuk para analis termasuk di dalamnya analis bisnis (business analyst) dan data scientist. Dan saking umumnya, dapat diestimasi bahwa 90 persen perusahaan di seluruh Indonesia mengalami hal seperti ini.

PHI-Integration – perusahaan data management – bahkan mengklaim kalau data duplikat dan kotor seperti ini merupakan indikasi growth bisnis yang bagus.

"Bisnis selalu lebih cepat daripada pengembangan sistem dan sop yang stabil. Dengan demikian pengawalan terhadap data sering dikompromi, ini diperparah dengan kemungkinan human error yang tinggi"

Disclaimer: PHI-Integration adalah perusahaan yang berkontribusi terhadap content di DQLab.id.

Bab ini akan memfokuskan diri membahas teori dan praktek untuk memecahkan masalah ini dengan R.

Klik tombol Next untuk melanjutkan.

Jarak Teks

Dasar dari penemuan duplikat dari kebanyakan sistem database bisnis – yang kebanyakan masih menyimpan teks – adalah mencari persamaan atau mencari "jarak"-nya.

Sebagai contoh:

Teks "**Agus Cahyono**" dan "**Agus Cahyono**" memiliki persamaan 100% atau tidak memiliki jarak, atau jaraknya 0.

Lalu bagaimana dengan teks "**Agus Cahyono**" dan "**Cahyono Agus**"? Berapa tingkat persamaannya? Berapa jaraknya? Pertanyaan yang sama juga untuk teks "**Agus Cahyono**" dan "**Cahyono, Agus**".

Dengan menggunakan ukuran jarak teks, maka dua data dapat dikatakan duplikat jika jaraknya semakin dekat.

Jika dilihat dengan mata manusia seluruh teks tersebut "harusnya" sama dan jaraknya 0. Tapi bagaimana dengan sistem software yang mengenali teks karakter per karakter. Kedua contoh terakhir pasti dikenali tidak sama.

Namun beruntung, di area text mining telah dikembangkan banyak variasi metode dan algoritma untuk menghitung jarak teks dengan berbagai kasus, seperti nama kata yang terbalik di atas.

Pada R, terdapat package yang bernama "stringdist" yang akan kita gunakan untuk menghitung jarak teks dengan function yang kita gunakan adalah **stringdist**. Berikut adalah contoh langsung penggunaannya.

```
     stringdist("Agus Cahyono", "Cahyono, Agus",
method="cosine")
```

dimana:

- **stringdist**: function untuk menghitung jarak antar teks.
- **"Agus Cahyono"**: teks pertama yang akan dibandingkan.
- **"Cahyono, Agus"**: teks kedua yang akan dibandingkan.
- **method="cosine"**: metode perhitungan jarak teks, dalam hal ini "cosine". Metode ini digunakan karena menggunakan teks dipecah menjadi vector dari sejumlah pasangan karakter (2 karakter, 3 karakter, dan seterusnya) dan tidak melihat posisi karakter.

Catatan: Function **stringdist** adalah case sensitive, artinya huruf besar dan kecil dari alfabet yang sama dianggap berbeda.

Metode **cosine** adalah yang akan kita gunakan secara intensif di bab ini. Namun, selain cosine ada metode-metode lain seperti:

- **lv**: Levenstein distance. Perhitungan jarak berdasarkan berapa banyak karakter yang dihapus, ditambahkan, dan dirubah sehingga kedua teks menjadi sama. Nilai jaraknya adalah bilangan integer dari 0 sampai dengan nilai integer tertentu.

- **dl**: Damerau-Levenstein distance. Pengembangan dari Levenstein distance dimana memperbolehkan transposisi karakter (berpindah tempat). Nilai jaraknya adalah bilangan integer dari 0 sampai dengan nilai integer tertentu.
- **hamming**: jarak Hamming – jumlah karakter yang berbeda antara kedua teks – dan panjang kedua teks harus sama. Jika tidak, akan mengembalikan **Inf**. Nilai jaraknya adalah bilangan integer dari 0 sampai dengan nilai integer tertentu.
- **osa**: Optimal string alignment - mirip dengan dl tapi setiap teks hanya boleh diedit sekali. Nilai jaraknya adalah bilangan integer dari 0 sampai dengan nilai integer tertentu. Ini adalah metode *default* untuk stringdist.
- **lcs**: longest common substring – berapa banyak karakter yang harus dibuang dari kedua teks sehingga menjadi teks yang sama. Nilai jaraknya adalah bilangan integer dari 0 sampai dengan nilai integer tertentu.
- **qgram**: Berapa banyak pasangan n-gram (yaitu potongan n karakter dari teks) yang berbeda. Nilai jaraknya adalah bilangan integer dari 0 sampai dengan nilai integer tertentu.
- **jaccard**: Adalah jarak yang dihitung berdasarkan berapa banyaknya pasangan n-gram yang berbeda dibagi jumlah pasangan n-gram total. Nilai jaraknya adalah nilai desimal antara 0 sampai dengan 1.
- **jw**: metode Jaro Wrinkler menghitung perpindahan karakter minimum yang diperlukan sehingga satu teks bertransformasi menjadi teks lain. Nilai jaraknya adalah nilai desimal antara 0 sampai dengan 1.
- **soundex**: metode jarak antara teks berdasarkan perbedaan pengucapan dalam bahasa Inggris.

Tugas Praktek

Pada code editor telah diisi dengan potongan code R untuk menghitung jarak berbagai teks. Ganti seluruh bagian […] pada code editor untuk menghitung jarak teks-teks berikut dengan metode cosine.

- "Agus Cahyono" dengan "Agus Cahyono".
- "Agus Cahyono" dengan "agus cahyono".
- "Agus Cahyono" dengan "Agus Tjahyono".
- "Agus Cahyono" dengan "Cahyono Agus".
- "Agus Cahyono" dengan "Cahyono, Agus".
- "Agus Cahyono" dengan "Justin Bieber".

Jika berjalan dengan lancar maka hasilnya akan terlihat sebagai berikut.

```
> stringdist("Agus Cahyono", "Agus Cahyono", method="cosine")
[1] 0


> stringdist("Agus Cahyono", "agus cahyono", method="cosine")
[1] 0.131401


> stringdist("Agus Cahyono", "Agus Tjahyono", method="cosine")
[1] 0.1029148
```

```
> stringdist("Agus Cahyono", "Cahyono Agus", method="cosine")
[1] 0

> stringdist("Agus Cahyono", "Cahyono, Agus", method="cosine")
[1] 0.03390822

> stringdist("Agus Cahyono", "Justin Bieber", method="cosine")
[1] 0.7407185
```

Dari hasil tersebut, dapat dirangkum hal berikut:

- Hasil pertama jaraknya adalah 0 atau tidak ada jarak sama sekali, kedua teks adalah sama.
- Hasil kedua jaraknya adalah 0.131401 karena kedua teks walaupun mengandung susunan alfabet yang persis sama, tapi memiliki huruf besar dan kecil.
- Hasil ketiga jaraknya adalah 0.1029148 karena kedua teks "Agus Cahyono" dan "Agus Tjahyono" ternyata berbeda di bagian "Cahyono" dan "Tjahyono". Angka hasil perhitungan menunjukkan jaraknya masih sangat dekat.
- Hasil keempat jaraknya adalah 0. Artinya sama persis jaraknya walaupun teks katanya terbalik: "Agus Cahyono" dan "Cahyono Agus".
- Hasil kelima jaraknya adalah 0.03390822. Artinya sangat dekat juga jaraknya walaupun teks katanya terbalik: "Agus Cahyono" dan "Cahyono, Agus". Perbedaan kecil ini dikarenakan adanya tanda koma.
- Hasil keenam jaraknya adalah 0.7407185. Artinya memiliki jarak jauh antara "Agus Cahyono" dan "Justin Bieber" dan sangat berbeda dibanding hasil-hasil di atas.

Code Editor

```
#Load library stringdist
library("stringdist")

#Melakukan perhitungan jarak teks
stringdist("Agus Cahyono", "Agus Cahyono", method="cosine")
stringdist("Agus Cahyono", "agus cahyono", method="cosine")
stringdist("Agus Cahyono", "Agus Tjahyono", method="cosine")
stringdist("Agus Cahyono", "Cahyono Agus", method="cosine")
stringdist("Agus Cahyono", "Cahyono, Agus", method="cosine")
```

stringdist("Agus Cahyono", "Justin Bieber", method="cosine")

Console

```
> #Load library stringdist
> library("stringdist")

> #Melakukan perhitungan jarak teks
> stringdist("Agus Cahyono", "Agus Cahyono", method="cosine")
[1] 0

> stringdist("Agus Cahyono", "agus cahyono", method="cosine")
[1] 0.131401

> stringdist("Agus Cahyono", "Agus Tjahyono", method="cosine")
[1] 0.1029148

> stringdist("Agus Cahyono", "Cahyono Agus", method="cosine")
[1] 0

> stringdist("Agus Cahyono", "Cahyono, Agus", method="cosine")
[1] 0.03390822

> stringdist("Agus Cahyono", "Justin Bieber", method="cosine")
[1] 0.7407185
```

Mencari Duplikat pada Vector

Dengan mengerti konsep jarak teks (*string distance*) maka kita bisa kaitkan kembali permasalahan awal kita, yaitu mencari duplikat di antara sekian banyak data.

Duplikat dapat didefinisikan sebagai suatu angka hasil *stringdist* antara dua teks yang di bawah batas maksimal yang diperbolehkan atau *threshold*.

Misalkan, kita bisa putuskan 0.15 dengan metode cosine sebagai *threshold*. Jarak di bawah 0.15 akan dianggap sebagai duplikat.

Bagaimana kita melakukan hal tersebut?

Pada R, dataset biasanya masuk ke data.frame, dan pengecekan duplikat dimulai dari tiap kolom, dan tiap kolom biasanya disimpan dalam bentuk vector.

Dan dengan demikian, pengecekan kita lakukan di vector. Agar mudah dipahami, mari kita lihat contoh berikut.

Kita ada satu teks nama "Agus Cahyono" yang perlu dibandingkan. Kita biasakan simpan teks ini dalam suatu nama variable – misalkan dengan nama **referensi**, sehingga ketika pengembangan code R nya nanti akan lebih mudah.

```
referensi <- "Agus Cahyono"
```

Kemudian kita buat vector yang berisi empat teks nama, kita simpan dengan variable **nama.pelanggan**.

```
nama.pelanggan <- c("Agus Cahyono", "Justin Bieber", "Agus Tjahyono", "Cahyono Agus")
```

Tahap berikutnya, kita hitung jarak teks antara referensi dengan vector dengan **stringdist** menggunakan metode **cosine**. Hasil perhitungan ini kita simpan ke satu variable, misalkan **jarak.teks**.

```
jarak.teks <- stringdist(referensi, nama.pelanggan, method="cosine")
```

Jika ditampilkan, hasilnya akan terlihat sebagai berikut.

```
[1] 0.0000000 0.7407185 0.1029148 0.0000000
```

Terlihat isi vector indeks pertama ("Agus Cahyono") dan keempat ("Cahyono Agus") persis sama. Dan kalau menggunakan *threshold* 0.15, maka isi indeks ketiga ("Agus Tjahyono") masih tergolong sama.

Nah, terakhir kita bisa filter isi nama.pelanggan dengan jarak.teks yang lebih kecil sama dengan angka 0.15.

```
nama.pelanggan[jarak.teks <= 0.15]
```

Hasilnya akan terlihat sebagai berikut.

```
[1] "Agus Cahyono"  "Agus Tjahyono" "Cahyono Agus"
```

Dengan demikian tiga item yang dikenali sebagai duplikat telah didapatkan semua.

Catatan: Threshold 0.15 ini adalah contoh. Pada praktek sebenarnya, threshold ini tidak menjadi jaminan akan mendapatkan hasil yang bagus. Sebagai contoh, "Budi Sanjaya" dan "Rudi Sanjaya" adalah entitas yang bisa berbeda tapi masih masuk threshold.

Contoh ini menunjukkan bahwa otomatisasi pencarian duplikat tidak bisa 100%.

Code Editor

```
#Load library stringdist

library("stringdist")


#Membuat variable referensi dan vector nama

referensi <- "Agus Cahyono"

nama.pelanggan <- c("Agus Cahyono", "Justin Bieber", "Agus Tjahyono", "Cahyono Agus")


#Menghitung jarak referensi dengan vector nama

jarak.teks <- stringdist(referensi, nama.pelanggan, method="cosine")


#Menampilkan variable jarak.teks

jarak.teks


#Data nama pelanggan yang telah difilter dengan threshold 0.15

nama.pelanggan[jarak.teks <= 0.15]
```

Console

```
> #Load library stringdist
> library("stringdist")

> #Membuat variable referensi dan vector nama
> referensi <- "Agus Cahyono"
```

```
> nama.pelanggan <- c("Agus Cahyono", "Justin Bieber", "Agus Tjahyono", "Cahyono Agus
")

> #Menghitung jarak referensi dengan vector nama
> jarak.teks <- stringdist(referensi, nama.pelanggan, method="cosine")

> #Menampilkan variable jarak.teks
> jarak.teks
[1] 0.0000000 0.7407185 0.1029148 0.0000000

> #Data nama pelanggan yang telah difilter dengan threshold 0.15
> nama.pelanggan[jarak.teks <= 0.15]
[1] "Agus Cahyono"  "Agus Tjahyono" "Cahyono Agus"
```

Menambahkan Informasi Grouping Duplikat

Dari mekanisme praktek sebelumnya, kita telah belajar bagaimana mengambil referensi dan membandingkan jarak teks ke seluruh item vector.

Pada praktek ini, kita akan melangkah ke tahap selanjutnya yaitu menambahkan informasi *grouping* seperti berikut.

```
  grouping          nama
1        1  Agus Cahyono
2        1 Agus Tjahyono
3        1  Cahyono Agus
4        2 Justin Bieber
```

Nomor grouping yang sama menyatakan bahwa data-datanya dianggap sama (duplikat). Pada contoh di atas grouping 1 memiliki tiga data, sedangkan grouping 2 hanya 1 data (tidak ada duplikat).

Tugas Praktek

Untuk melakukan hal ini, banyak cara. Pada code editor telah diberikan algoritma yang telah dibuat oleh tim DQLab dengan logika sebagai berikut:

1. Variable **pelanggan** diisi dengan vector data awal.
2. Inisialisasi variable nomor grouping (**grouping_no**) ke nilai 1.
3. Proses penemuan duplikat akan dimulai dengan mengambil **referensi** dari item pertama vector.
4. Variable **pelanggan** akan dihilangkan item per penemuan duplikat sehingga akhirnya akan hilang semua atau panjang vector menjadi nol.
5. Hitung **jarak teks** antara referensi dengan seluruh item nama.pelanggan.
6. Filter nama.pelanggan yang memiliki jarak teks sesuai threshold, dan disimpan ke variable **hasil**.
7. Membuat variable **temp** berupa data frame yang berisi nomor grouping saat ini dan hasil duplikat.
8. Menggabungkan var.temp dengan hasil sebelumnya ke variable **akhir**.
9. Menghilangkan item yang sudah didapatkan duplikatnya dari nama.pelanggan, dengan cara filter item dengan jarak teks di atas threshold.
10. Menaikkan nilai grouping_no sebesar 1.
11. Jika item masih ada, maka proses diulangi dari tahap no 2.

Code telah dilengkapi dan tidak ada yang perlu Anda lakukan, tinggal dijalankan dan jika lancar maka hasilnya akan terlihat sebagai berikut.

```
  grouping          nama
1        1  Agus Cahyono
2        1 Agus Tjahyono
```

```
3         1  Cahyono Agus
4         2 Justin Bieber
```

Pelajari detil code ini dengan baik, karena subbab berikutnya adalah pengembangan dari framework code ini.

Code Editor

```
#Load library stringdist
library("stringdist")


#Membuat variable vector nama
nama.pelanggan <- c("Agus Cahyono", "Justin Bieber", "Agus Tjahyono", "Cahyono Agus")


#Inisialisai variable untuk hasil.akhir
hasil.akhir <- NULL


#Inisialiasi variable grouping_no dengan nilai 1
grouping_no <- 1


#Melakukan perulangan proses pencarian dengan perintah while, sampai akhirnya isi vector menjadi kosong (panjang = 0)
while(length(nama.pelanggan)>0)
{
  #Variable referensi diisi dengan item pertama variable nama.pelanggan
  referensi <- nama.pelanggan[1]


  #Menghitung jarak antara referensi dengan item-item nama.pelanggan
  jarak.teks <- stringdist(referensi, nama.pelanggan, method="cosine")
```

```
  #Hasil filter jarak dengan threshold 0.15 disimpan ke variable nama.hasil
  nama.hasil <- nama.pelanggan[jarak.teks <= 0.15]

  #Hasil filter jarak dengan threshold 0.15 disimpan ke variable nama.hasil
  var.temp = data.frame(grouping=grouping_no, nama=nama.hasil)

  #Menggabungkan hasil sebelumnya
  hasil.akhir <- rbind(hasil.akhir, var.temp)

  #Mengambil porsi data yang bukan di dalam threshold dengan menggunakan simbol ! yang mewakili operator not (bukan)
  nama.pelanggan <- nama.pelanggan[!(jarak.teks <= 0.15)]

  #Menambahkan nilai grouping untuk diambil pada iterasi selanjutnya
  grouping_no <- grouping_no + 1
}
#Menampilkan hasil akhir
hasil.akhir
```

```
> #Load library stringdist
> library("stringdist")

> #Membuat variable vector nama
> nama.pelanggan <- c("Agus Cahyono", "Justin Bieber", "Agus Tjahyono", "Cahyono Agus
")

> #Inisialisai variable untuk hasil.akhir
> hasil.akhir <- NULL

> #Inisialiasi variable grouping_no dengan nilai 1
> grouping_no <- 1

> #Melakukan perulangan proses pencarian dengan perintah while, sampai akhirnya isi v
ector menjadi kosong (panjang = 0)
> while(length(nama.pelanggan)>0)
+ {
+   #Variable referensi diisi dengan item pertama variable nama.pelanggan
+   referensi <- nama.pelanggan[1]
+
+   #Menghitung jarak antara referensi dengan item-item nama.pelanggan
+   jarak.teks <- stringdist(referensi, nama.pelanggan, method="cosine")
+
+   #Hasil filter jarak dengan threshold 0.15 disimpan ke variable nama.hasil
+   nama.hasil <- nama.pelanggan[jarak.teks <= 0.15]
+
+   #Hasil filter jarak dengan threshold 0.15 disimpan ke variable nama.hasil
+   var.temp = data.frame(grouping=grouping_no, nama=nama.hasil)
+
+   #Menggabungkan hasil sebelumnya
+   hasil.akhir <- rbind(hasil.akhir, var.temp)
+
+   #Mengambil porsi data yang bukan di dalam threshold dengan menggunakan simbol ! y
ang mewakili operator not (bukan)
+   nama.pelanggan <- nama.pelanggan[!(jarak.teks <= 0.15)]
+
+   #Menambahkan nilai grouping .... [TRUNCATED]

> #Menampilkan hasil akhir
> hasil.akhir
  grouping          nama
1        1  Agus Cahyono
2        1 Agus Tjahyono
3        1  Cahyono Agus
4        2 Justin Bieber
```

Melakukan Grouping Duplikat dari Dataset Awal

Dari framework praktek "Menambahkan Informasi Grouping Duplikat" kita akan melangkah ke penemuan *grouping* duplikat dari dataset kita. Subbab ini akan fokus mengambil dataset awal kita sebelum standarisasi, yaitu data dari table **dqlab_messy_data**.

	A	B	C	D	E
1	grouping	kode_pelanggan	nama	alamat	jumlah_record
14	13	KD-00076	Safira Hana Sahrani	Taman Bunga Langit, Jl. Utara No. 3	2
15	13	KD-00298	Safira Hana Sahrani	Taman Bunga Langit, Jl. Utara No. 3	2
18	16	KD-00099	Bapak Sanjaya Priyantoro	Taman Bunga Langit, Jl. Barat Laut No. 6	2
19	16	KD-00192	Bapak Sanjaya Priyantoro	Taman Bunga Langit, Jl. Barat Laut No. 6	2
29	26	KD-00012	Cahyono, Agus	Pulo Bambu No. 15, Kota Tenggara Lama	3
30	26	KD-00001	Agus Cahyono's	Jl. Pulo Bambu No. 15, Kota Tenggara Lama	3
31	26	KD-00778	Cahyono Agus H.	Jalan. Pulau Bambu No. 15 - Kota Tenggara Lama	3
65	60	KD-00044	dr. Yati Octavianus	Kompleks Pelaut Tangguh, No. 5A	2
66	60	KD-00492	dr. Yati Octavianus	Kompleks Pelaut Tangguh, No. 5A	2
85	79	KD-00066	Purnomo Hadi	Jl. Pulau Sentosa No. 133	2
86	79	KD-00041	Poernomo Hadi	Jalan. Pulau Sentosa No. 133	2
88	81	KD-00116	Risma Sihombing	Apartemen Lucky Beruntung, Lt. 5 No. 4	2
89	81	KD-00144	Risma Sihombing	Apartemen Lucky Beruntung, Lt. 3 No. 4	2
104	96	KD-00128	Tedi Rahmanto	Apartemen Bukit Merah, Annex Plaza, Lt 3 No. A1	2
105	96	KD-00115	Teddy Rahmanto	Apartemen Bukit Merah Annex Plaza, Lt 3 No. A1	2
113	104	KD-00089	Acmad Junaidi	Jalan Raya Hang Lekir, No. 62 - Kota Z	2
114	104	KD-00042	Ahmad Junaidi	Jalan Raya Hang Lekir, Kota Z, No. 62	2
150	140	KD-00015	Mario Setiawan	Jl. Puri Arteri Raya, No. 88 - Kota T	2
151	140	KD-00083	Setiawan Mario	Jl. Puri Arteri Raya, No. 88 - Kota T	2

Dari dataset ini kita akan mencari grouping duplikat berdasarkan dua kolom, yaitu nama dan alamat. Karena pada kasus nyata, satu kolom nama tentunya tidak cukup – sebagai contoh untuk kasus perbankan pelaporan ke OJK biasanya melibatkan sampai enam kolom.

Tugas Praktek

Pada code editor telah diisi semua code yang diperlukan untuk melakukan grouping duplikat. Cobalah ganti bagian […1…] dan […2…] dengan metode yang sesuai dengan komentar pada code.Jika berjalan dengan lancar, maka ada output file Excel bernama "**staging.duplikat.awal.xlsx**". Anda bisa download dan buka file tersebut dengan aplikasi Excel.

Berikut adalah tampilan dari file tersebut ketika difilter berdasarkan kolom **jumlah_record** di atas satu, dan DQLab juga telah highlight grouping antara dengan warna hijau muda.

	A	B	C	D	E
1	grouping	kode_pelanggan	nama	alamat	jumlah_record
14	13	KD-00076	Safira Hana Sahrani	Taman Bunga Langit, Jl. Utara No. 3	2
15	13	KD-00298	Safira Hana Sahrani	Taman Bunga Langit, Jl. Utara No. 3	2
18	16	KD-00099	Bapak Sanjaya Priyantoro	Taman Bunga Langit, Jl. Barat Laut No. 6	2
19	16	KD-00192	Bapak Sanjaya Priyantoro	Taman Bunga Langit, Jl. Barat Laut No. 6	2
29	26	KD-00012	Cahyono, Agus	Pulo Bambu No. 15, Kota Tenggara Lama	3
30	26	KD-00001	Agus Cahyono's	Jl. Pulo Bambu No. 15, Kota Tenggara Lama	3
31	26	KD-00778	Cahyono Agus H.	Jalan. Pulau Bambu No. 15 - Kota Tenggara Lama	3
65	60	KD-00044	dr. Yati Octavianus	Kompleks Pelaut Tangguh, No. 5A	2
66	60	KD-00492	dr. Yati Octavianus	Kompleks Pelaut Tangguh, No. 5A	2
85	79	KD-00066	Purnomo Hadi	Jl. Pulau Sentosa No. 133	2
86	79	KD-00041	Poernomo Hadi	Jalan. Pulau Sentosa No. 133	2
88	81	KD-00116	Risma Sihombing	Apartemen Lucky Beruntung, Lt. 5 No. 4	2
89	81	KD-00144	Risma Sihombing	Apartemen Lucky Beruntung, Lt. 3 No. 4	2
104	96	KD-00128	Tedi Rahmanto	Apartemen Bukit Merah, Annex Plaza, Lt 3 No. A1	2
105	96	KD-00115	Teddy Rahmanto	Apartemen Bukit Merah Annex Plaza, Lt 3 No. A1	2
113	104	KD-00089	Acmad Junaidi	Jalan Raya Hang Lekir, No. 62 - Kota Z	2
114	104	KD-00042	Ahmad Junaidi	Jalan Raya Hang Lekir, Kota Z, No. 62	2
150	140	KD-00015	Mario Setiawan	Jl. Puri Arteri Raya, No. 88 - Kota T	2
151	140	KD-00083	Setiawan Mario	Jl. Puri Arteri Raya, No. 88 - Kota T	2

Code Editor

```
library(RMySQL)

library(stringdist)

library(openxlsx)


#Membuka koneksi

con <- dbConnect(MySQL(), user="demo", password="demo",
host="mysqlhost",dbname="dqlabdatawrangling")


#Mengambil kolom kode_pelanggan, nama dan alamat dari dqlab_messy_data

sql <- "select kode_pelanggan, nama, alamat from dqlab_messy_data"


rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))

data.pelanggan <- fetch(rs, n=-1)

dbClearResult(rs)
```

```
#Inisialisai variable untuk hasil.akhir
hasil.akhir <- NULL

#Inisialiasi variable grouping_no dengan nilai 1
grouping_no <- 1

while(length(data.pelanggan$nama)>0)
{
#Variable referensi nama dan alamat diambil dari item pertama
referensi.nama <- data.pelanggan$nama[1]
referensi.alamat <- data.pelanggan$alamat[1]

  #Menghitung jarak antara referensi dengan item-item nama dan alamat
  #gunakan method "cosine" untuk nama, dan method "lv" untuk alamat
  jarak.teks.nama <- stringdist(referensi.nama, data.pelanggan$nama,
method="cosine")
  jarak.teks.alamat <- stringdist(referensi.alamat, data.pelanggan$alamat, method="lv")

  #Hasil filter jarak dengan threshold
  # - lebih kecil sama dengan angka 0.15 untuk nama
  # - lebih kecil dari angka 15 untuk alamat
  #disimpan ke variable filter.jarak
  filter.jarak <- (jarak.teks.nama <= 0.15 & jarak.teks.alamat < 15)

  #Melakukan filtering pada variable data.pelanggan, dan mengambil tiga kolom
  #untuk disimpan ke tiga variable
  kode_pelanggan.temp <- data.pelanggan[filter.jarak,]$kode_pelanggan
  nama.temp <- data.pelanggan[filter.jarak,]$nama
  alamat.temp <- data.pelanggan[filter.jarak,]$alamat
```

```
  #Konstruksi temporary variable
  var.temp <- data.frame(grouping=grouping_no,
kode_pelanggan=kode_pelanggan.temp, nama=nama.temp, alamat=alamat.temp,
jumlah_record=length(kode_pelanggan.temp))

  #Menggabungkan temporary variable dengan hasil sebelumnya
  hasil.akhir <- rbind(hasil.akhir, var.temp)

  #Menggabungkan hasil sebelumnya
  data.pelanggan <- data.pelanggan[!filter.jarak,]

  #Menambahkan nilai grouping untuk diambil pada iterasi selanjutnya
  grouping_no <- grouping_no + 1
}

#Menulis hasil ke file staging.duplikat.awal.xlsx
write.xlsx(hasil.akhir, file="staging.duplikat.awal.xlsx")

#Menutup seluruh koneksi MySQL
all_cons <- dbListConnections(MySQL())
for(con in all_cons)
    + dbDisconnect(con)
```

Console

```
> library(RMySQL)
> library(stringdist)
> library(openxlsx)
> #Membuka koneksi
```

```
> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",dbname="dq
labdatawrangling")

> #Mengambil kolom kode_pelanggan, nama dan alamat dari dqlab_messy_data
> sql <- "select kode_pelanggan, nama, alamat from dqlab_messy_data"

> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> data.pelanggan <- fetch(rs, n=-1)

> dbClearResult(rs)
[1] TRUE

> #Inisialisai variable untuk hasil.akhir
> hasil.akhir <- NULL

> #Inisialiasi variable grouping_no dengan nilai 1
> grouping_no <- 1

> while(length(data.pelanggan$nama)>0)
+ {
+ #Variable referensi nama dan alamat diambil dari item pertama
+ referensi.nama <- data.pelanggan$nama[1]
+ referensi.alamat <- data.pelanggan$alamat[1]
+
+   #Menghitung jarak antara referensi dengan item-item nama dan alamat
+   #gunakan method "cosine" untuk nama, dan method "lv" untuk alamat
+   jarak.teks.nama <- stringdist(referensi.nama, data.pelanggan$nama, method="cosine
")
+   jarak.teks.alamat <- stringdist(referensi.alamat, data.pelanggan$alamat, method="
lv")
+
+   #Hasil filter jarak dengan threshold
+   # - lebih kecil sama dengan angka 0.15 untuk nama
+   # - lebih kecil dari angka 15 untuk alamat
+   #disimpan ke variable filter.jarak
+   filter.jarak <- (jarak.teks.nama <= 0.15 & jarak.teks.alamat < 15)
+
+   #Melakukan filtering pada variable data.pelanggan, dan mengambil tiga kolom
+   #untuk disimpan ke tiga variable
+   kode_pelanggan.temp <- data.pelanggan[filter.jarak,]$kode_pelanggan
+   nama.temp <- data.pelang .... [TRUNCATED]

> #Menulis hasil ke file staging.duplikat.awal.xlsx
> write.xlsx(hasil.akhir, file="staging.duplikat.awal.xlsx")

> #Menutup seluruh koneksi MySQL
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons)
+     + dbDisconnect(con)
```

Melakukan Grouping Duplikat dari Dataset Hasil Standarisasi

Subbbab berikut ini hampir sama dengan sebelumnya, kecuali kita menggunakan sumber data yang sudah terstandarisasi yang tersimpan di file Excel **staging.final.xlsx**.

Tugas Praktek

Gantilah bagian `[...1...]`, `[...2...]` dan `[...3...]` pada code editor sehingga kita akan mendapatkan file "**staging.duplikat.standarisasi.xlsx**".

Jika dibuka, difilter berdasarkan data duplikat, dan highlight maka hasilnya terlihat sebagai berikut. Perhatikan jika jumlah grouping duplikat kali ini lebih banyak dari sebelumnya sebanyak dua grouping (ditandai dengan anak panah). Namun ada satu grouping yang sebenarnya salah.

Ini juga contoh yang menunjukkan bahwa otomatisasi tidak bisa 100% akurat.

	A	B	C	D	E
1	grouping	kode_pelanggan	nama	alamat	jumlah_record
2	1	KD-00001	Agus Cahyonos	Jalan Pulo Bambu No. 15, Kota Tenggara Lama	3
3	1	KD-00012	Cahyono, Agus	Pulo Bambu No. 15, Kota Tenggara Lama	3
4	1	KD-00778	Cahyono Agus H.	Jalan Pulau Bambu No. 15 - Kota Tenggara Lama	3
12	9	KD-00009	Antonius Winarta	Jalan Kebon Jahe, No. F16 - Kota E	2
13	9	KD-00026	Anton Winarta	Jalan Kebon Jahe, Kota EntahDimana	2
18	14	KD-00015	Mario Setiawan	Jalan Puri Arteri Raya, No. 88 - Kota T	2
19	14	KD-00083	Setiawan Mario	Jalan Puri Arteri Raya, No. 88 - Kota T	2
44	39	KD-00041	Poernomo Hadi	Jalan Pulau Sentosa No. 133	2
45	39	KD-00066	Purnomo Hadi	Jalan Pulau Sentosa No. 133	2
46	40	KD-00042	Ahmad Junaidi	Jalan Raya Hang Lekir, Kota Z, No. 62	2
47	40	KD-00089	Acmad Junaidi	Jalan Raya Hang Lekir, No. 62 - Kota Z	2
49	42	KD-00044	dr. Yati Octavianus	Kompleks Pelaut Tangguh, No. 5A	2
50	42	KD-00492	dr. Yati Octavianus	Kompleks Pelaut Tangguh, No. 5A	2
80	72	KD-00076	Safira Hana Sahrani	Taman Bunga Langit, Jalan Utara No. 3	2
81	72	KD-00298	Safira Hana Sahrani	Taman Bunga Langit, Jalan Utara No. 3	2
102	93	KD-00099	Sanjaya Priyantoro	Taman Bunga Langit, Jalan Barat Laut No. 6	2
103	93	KD-00192	Sanjaya Priyantoro	Taman Bunga Langit, Jalan Barat Laut No. 6	2
119	109	KD-00115	Teddy Rahmanto	Apartemen Bukit Merah Annex Plaza, Lt 3 No. A1	2
120	109	KD-00128	Tedi Rahmanto	Apartemen Bukit Merah, Annex Plaza, Lt 3 No. A1	2
121	110	KD-00116	Risma Sihombing	Apartemen Lucky Beruntung, Lt. 5 No. 4	2
122	110	KD-00144	Risma Sihombing	Apartemen Lucky Beruntung, Lt. 3 No. 4	2
143	131	KD-00138	Teddja Yanto	Jalan Gula Pahit, No. 081	2
144	131	KD-00140	Leonardo Tedja	Jalan Pulau Sentosa No. 1335	2

Code Editor

```
library(stringdist)

library(openxlsx)


#Membaca file staging.final.xlsx

data.pelanggan <- read.xlsx("staging.final.xlsx")
```

```
#Inisialisai variable untuk hasil.akhir
hasil.akhir <- NULL

#Inisialiasi variable grouping_no dengan nilai 1
grouping_no <- 1

while(length(data.pelanggan$nama)>0)
{
  #Variable referensi nama dan alamat diambil dari item pertama
  referensi.nama <- data.pelanggan$nama[1]
  referensi.alamat <- data.pelanggan$alamat[1]

  #Menghitung jarak antara referensi dengan item-item nama dan alamat
  #gunakan method "cosine" untuk nama, dan method "lv" untuk alamat
  jarak.teks.nama <- stringdist(referensi.nama, data.pelanggan$nama,
method="cosine")
  jarak.teks.alamat <- stringdist(referensi.alamat, data.pelanggan$alamat, method="lv")

  #Hasil filter jarak dengan threshold
  # - lebih kecil sama dengan angka 0.15 untuk nama
  # - lebih kecil dari angka 15 untuk alamat
  #disimpan ke variable filter.jarak
  filter.jarak <- (jarak.teks.nama <= 0.15 & jarak.teks.alamat < 15)

  #Melakukan filtering pada variable data.pelanggan, dan mengambil tiga kolom
  #untuk disimpan ke tiga variable
  kode_pelanggan.temp <- data.pelanggan[filter.jarak,]$kode_pelanggan
  nama.temp <- data.pelanggan[filter.jarak,]$nama
  alamat.temp <- data.pelanggan[filter.jarak,]$alamat
```

```
  #Konstruksi temporary variable
  var.temp <- data.frame(grouping=grouping_no,
kode_pelanggan=kode_pelanggan.temp, nama=nama.temp, alamat=alamat.temp,
jumlah_record=length(kode_pelanggan.temp))

  #Menggabungkan temporary variable dengan hasil sebelumnya
  hasil.akhir <- rbind(hasil.akhir, var.temp)

  #Menggabungkan hasil sebelumnya
  data.pelanggan <- data.pelanggan[!filter.jarak,]

  #Menambahkan nilai grouping untuk diambil pada iterasi selanjutnya
  grouping_no <- grouping_no + 1
}

#Menulis hasil ke file staging.duplikat.standarisasi.xlsx
write.xlsx(hasil.akhir, file="staging.duplikat.standarisasi.xlsx")
```

Console

```
> library(stringdist)

> library(openxlsx)

> #Membaca file staging.final.xlsx
> data.pelanggan <- read.xlsx("staging.final.xlsx")

> #Inisialisai variable untuk hasil.akhir
> hasil.akhir <- NULL

> #Inisialiasi variable grouping_no dengan nilai 1
> grouping_no <- 1

> while(length(data.pelanggan$nama)>0)
+ {
+   #Variable referensi nama dan alamat diambil dari item pertama
```

```
+   referensi.nama <- data.pelanggan$nama[1]
+   referensi.alamat <- data.pelanggan$alamat[1]
+
+   #Menghitung jarak antara referensi dengan item-item nama dan alamat
+   #gunakan method "cosine" untuk nama, dan method "lv" untuk alamat
+   jarak.teks.nama <- stringdist(referensi.nama, data.pelanggan$nama, method="cosine
")
+   jarak.teks.alamat <- stringdist(referensi.alamat, data.pelanggan$alamat, method="
lv")
+
+   #Hasil filter jarak dengan threshold
+   # - lebih kecil sama dengan angka 0.15 untuk nama
+   # - lebih kecil dari angka 15 untuk alamat
+   #disimpan ke variable filter.jarak
+   filter.jarak <- (jarak.teks.nama <= 0.15 & jarak.teks.alamat < 15)
+
+   #Melakukan filtering pada variable data.pelanggan, dan mengambil tiga kolom
+   #untuk disimpan ke tiga variable
+   kode_pelanggan.temp <- data.pelanggan[filter.jarak,]$kode_pelanggan
+   nama.temp <- data. .... [TRUNCATED]

> #Menulis hasil ke file staging.duplikat.standarisasi.xlsx
> write.xlsx(hasil.akhir, file="staging.duplikat.standarisasi.xlsx")
```

Kesimpulan

Duplikasi data adalah kondisi dimana dalam suatu dataset terdapat lebih dari satu data yang sebenarnya mewakili satu entity tapi tidak berhasil dikelompokkan menjadi satu.

Keadaan ini dapat berubah menjadi masalah karena bisnis bisa telat menganalisa data karena data tidak dapat dikonsolidasikan dengan baik. Ini berakibat kepada kesempatan bisnis yang hilang (*opportunity lost*).

Dan sepanjang bab ini kita telah membahas cara mengatasi duplikat ini sebagai berikut:

- Jarak teks (*string distance*) sebagai dasar untuk mencari data duplikat di dataset dengan menggunakan function stringdist
- Metode-metode jarak teks yang tersedia seperti Cosine, Levenstein, Jaro Winkler, dan lain-lain. Namun untuk praktek kita gunakan dua metode pertama untuk nama dan alamat.
- Algoritma sederhana untuk mencari duplikat di vector.
- Penerapan algoritma pencarian duplikat di dataset kita sebelum dan sesudah standarisasi.
- Terlihat bahwa penerapan standarisasi selain membuat data rapi, juga menaikkan kemungkinan mendapatkan data duplikat.

Penanganan duplikasi ini sebenarnya banyak kaitannya dengan pembuatan master data – yaitu data mewakili objek dan stakeholder bisnis misalkan customer, produk, lokasi, dan lain-lain. Master data ini sedemikian pentingnya sehingga banyak analisa tidak akan berhasil jika buruk kualitas master datanya.

Banyak sekali kasus yang membuat deduplikasi data ini sangat sulit diolah, lebih dari sekedar penerapan algoritma. Tapi bab ini memberi dasar yang baik sehingga Anda mendapatkan pengetahuan **how-to** dan jalan untuk memulai.

Klik tombol Next untuk melanjutkan ke bab praktek terakhir sebelum penutup – yaitu Data Enrichment.

Apa itu Data Enrichment?

Data enrichment adalah proses pengisian data yang hilang atau menambah data baik dari sumber internal maupun eksternal dengan cara mengkorelasikan berdasarkan beberapa kolom tertentu sehingga analisa data lebih tajam.

Sebagai contoh, data Nilai Belanja Setahun yang kosong dapat diisi dengan nilai rata-rata (mean) dari keseluruhan data.

Contoh lain, data kodepos yang kosong dapat diisi jika kita memiliki master kode pos. Atau dengan cara mencari dari alamat lain yang mirip dan terisi kode posnya.

Klik tombol Next untuk melanjutkan ke praktek untuk melakukan data enrichment.

Mengganti missing value dengan nilai mean

Kolom **nilai_belanja_setahun** dari dataset kita adalah kolom bertipe numerik yang beberapa diantaranya berisi missing value (NA).

Jika kita lihat hasil summary dari kolom **nilai_belanja_setahun**, tampilannya akan terlihat sebagai berikut.

```
nilai_belanja_setahun
 Min.   : 237400
 1st Qu.: 504800
 Median : 851600
 Mean   : 857226
 3rd Qu.:1179800
 Max.   :1537200
 NA's   :4
```

Pada baris terakhir yaitu NA's : 4 menyatakan bahwa kolom ini memiliki empat NA atau *missing value*. Dan untuk mengisinya, biasa menggunakan nilai rata-rata (*mean*) atau nilai tengah (*median*).

Untuk subbab ini, kita akan mengisi dengan nilai rata-rata (*mean*). Untuk melakukan perhitungan rata-rata kita akan gunakan function **mean**.

Berikut adalah contoh menghitung nilai mean dengan contoh variable **data.pelanggan** sebagai perwakilan dataset kita, dan hasilnya disimpan sebagai variable **nilai_rata_rata**.

```
     nilai_rata_rata <-
mean(data.pelanggan$nilai_belanja_setahun, na.rm=TRUE)
```

dimana:

- **nilai_rata_rata** adalah variable untuk menyimpan hasil function mean.
- **mean** adalah function yang digunakan untuk menghitung nilai rata-rata.
- **pelanggan$nilai_belanja_setahun** adalah kolom nilai_belanja_setahun.
- **rm = TRUE** adalah opsi untuk tidak mengikutsertakan *missing value*. Ini wajib disertakan untuk kasus kita.

Kemudian kita isi ke bagian data frame yang telah filter dulu isi yang memiliki missing value NA function `is.na` sebagai berikut.

```
data.pelanggan$nilai_belanja_setahun[is.na(data.pelanggan$nilai_belanja_setahun)] <- nilai_rata_rata
```

Tugas Praktek

Dari penjelasan dan contoh pada Lesson, ganti bagian `[…1…]`, `[…2…]` dan `[…3…]` pada code editor dengan code yang sesuai sehingga tampilan summary setelah diisi missing value adalah sebagai berikut.

```
> summary(data.pelanggan)
 kode_pelanggan     nilai_belanja_setahun
 Length:155         Min.   : 237400
 Class :character   1st Qu.: 520000
 Mode  :character   Median : 857226
                    Mean   : 857226
                    3rd Qu.:1168250
                    Max.   :1537200
```

Ada summary dari dua kolom yang ditampilkan, terlihat untuk nilai_belanja_setahun sudah tidak ada missing value dan nilai mean masih sama dengan sebelumnya (lihat Lesson) tetapi nilai median menjadi berbeda.

Kita juga akan mendapatkan output berupa file bernama **staging.enrichment.mean.xlsx**.

Code Editor

```
library(RMySQL)

library(openxlsx)


con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
                 dbname="dqlabdatawrangling")


sql <- "SELECT kode_pelanggan, nilai_belanja_setahun from dqlab_messy_data"
```

```
rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
data.pelanggan <- fetch(rs, n=-1)
dbClearResult(rs)

#Melakukan konversi nilai_belanja_setahun menjadi numerik
#karena sebelumnya disimpan dalam bentuk character
data.pelanggan$nilai_belanja_setahun <-
as.numeric(data.pelanggan$nilai_belanja_setahun)
summary(data.pelanggan)

#Menghitung rata-rata dengan function mean dan disimpan dalam variable
nilai_rata_rata
nilai_rata_rata <- mean(data.pelanggan$nilai_belanja_setahun, na.rm=TRUE)

#Mengisi missing value dengan nilai rata-rata
data.pelanggan$nilai_belanja_setahun[is.na(data.pelanggan$nilai_belanja_setahun)] <-
nilai_rata_rata

#Melihat summary setelah missing value
summary(data.pelanggan)

#Menulis ke dalam file staging.enrichment.mean.xlsx
write.xlsx(data.pelanggan, file="staging.enrichment.mean.xlsx")
all_cons <- dbListConnections(MySQL())
for(con in all_cons)
  + dbDisconnect(con)
```

Console

```
> library(RMySQL)

> library(openxlsx)

> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> sql <- "SELECT kode_pelanggan, nilai_belanja_setahun from dqlab_messy_data"

> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> data.pelanggan <- fetch(rs, n=-1)

> dbClearResult(rs)
[1] TRUE

> #Melakukan konversi nilai_belanja_setahun menjadi numerik
> #karena sebelumnya disimpan dalam bentuk character
> data.pelanggan$nilai_belanja_setahun <- as.numeric(data.pelanggan$nilai_belanja_set
ahun)

> summary(data.pelanggan)
 kode_pelanggan     nilai_belanja_setahun
 Length:155         Min.   : 237400
 Class :character   1st Qu.: 504800
 Mode  :character   Median : 851600
                    Mean   : 857226
                    3rd Qu.:1179800
                    Max.   :1537200
                    NA's   :4

> #Menghitung rata-rata dengan function mean dan disimpan dalam variable nilai_rata_r
ata
> nilai_rata_rata <- mean(data.pelanggan$nilai_belanja_setahun, na.rm=TRUE)

> #Mengisi missing value dengan nilai rata-rata
> data.pelanggan$nilai_belanja_setahun[is.na(data.pelanggan$nilai_belanja_setahun)] <
- nilai_rata_rata

> #Melihat summary setelah missing value
> summary(data.pelanggan)
 kode_pelanggan     nilai_belanja_setahun
 Length:155         Min.   : 237400
 Class :character   1st Qu.: 520000
 Mode  :character   Median : 857226
                    Mean   : 857226
                    3rd Qu.:1168250
                    Max.   :1537200

> #Menulis ke dalam file staging.enrichment.mean.xlsx
> write.xlsx(data.pelanggan, file="staging.enrichment.mean.xlsx")
```

```
> all_cons <- dbListConnections(MySQL())

> for(con in all_cons)
+     + dbDisconnect(con)
```

Mengganti missing value dengan nilai median

Subbab ini hampir sama penjelasannya dengan subbab sebelumnya. Kita akan mengambil kembali dataset nilai_belanja_setahun dengan empat nilai *missing value* awal seperti terlihat dari *summary* kolom tersebut sebagai berikut.

```
nilai_belanja_setahun
 Min.   : 237400
 1st Qu.: 504800
 Median : 851600
 Mean   : 857226
 3rd Qu.:1179800
 Max.   :1537200
 NA's   :4
```

Kali ini kita akan mengisi dengan nilai tengah (*median*). Untuk melakukan perhitungan rata-rata kita akan gunakan function **median**.

Berikut adalah contoh menghitung nilai mean dengan contoh variable **data.pelanggan** sebagai perwakilan dataset kita, dan hasilnya disimpan sebagai variable **nilai_tengah**.

```
    nilai_tengah <-
median(data.pelanggan$nilai_belanja_setahun, na.rm=TRUE)
```

dimana:

- **nilai_tengah** adalah variable untuk menyimpan hasil function median.
- **median** adalah function yang digunakan untuk menghitung nilai rata-rata.
- **pelanggan$nilai_belanja_setahun** adalah kolom nilai_belanja_setahun.
- **rm = TRUE** adalah opsi untuk tidak mengikutsertakan *missing value*. Ini wajib disertakan untuk kasus kita.

Kemudian kita isi ke bagian data frame yang telah filter dulu isi yang memiliki missing value NA function `is.na` sebagai berikut.

```
    data.pelanggan$nilai_belanja_setahun[is.na(data.pelanggan$n
ilai_belanja_setahun)] <- nilai_tengah
```

Tugas Praktek

Dari penjelasan dan contoh pada Lesson, ganti bagian `[…1…]`, `[…2…]` dan `[…3…]` pada code editor dengan code yang sesuai sehingga tampilan summary setelah diisi missing value adalah sebagai berikut.

```
> summary(data.pelanggan)
 kode_pelanggan     nilai_belanja_setahun
 Length:155         Min.   : 237400
 Class :character   1st Qu.: 520000
 Mode  :character   Median : 851600
                    Mean   : 857081
                    3rd Qu.:1168250
                    Max.   :1537200
```

Ada summary dari dua kolom yang ditampilkan, terlihat untuk nilai_belanja_setahun sudah tidak ada missing value dan nilai median masih sama dengan sebelumnya (lihat Lesson) tetapi nilai mean berbeda.

Kita juga akan mendapatkan output berupa file bernama **staging.enrichment.median.xlsx**.

Code Editor

```
library(RMySQL)
library(openxlsx)

con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
                 dbname="dqlabdatawrangling")

sql <- "SELECT kode_pelanggan, nilai_belanja_setahun from dqlab_messy_data"

rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
data.pelanggan <- fetch(rs, n=-1)
dbClearResult(rs)

#Melakukan konversi nilai_belanja_setahun menjadi numerik
#karena sebelumnya disimpan dalam bentuk character
```

```
data.pelanggan$nilai_belanja_setahun <-
as.numeric(data.pelanggan$nilai_belanja_setahun)

summary(data.pelanggan)


#Menghitung nilai tengah dengan function median dan disimpan dalam variable
nilai_tengah

nilai_tengah <- median(data.pelanggan$nilai_belanja_setahun, na.rm=TRUE)


#Mengisi missing value dengan nilai tengah

data.pelanggan$nilai_belanja_setahun[is.na(data.pelanggan$nilai_belanja_setahun)] <-
nilai_tengah


#Melihat summary setelah missing value

summary(data.pelanggan)


#Menulis ke dalam file staging.enrichment.mean.xlsx

write.xlsx(data.pelanggan, file="staging.enrichment.median.xlsx")

all_cons <- dbListConnections(MySQL())

for(con in all_cons)

    + dbDisconnect(con)
```

Console

```
> library(RMySQL)

> library(openxlsx)

> con <- dbConnect(MySQL(), user="demo", password="demo", host="mysqlhost",
+                  dbname="dqlabdatawrangling")

> sql <- "SELECT kode_pelanggan, nilai_belanja_setahun from dqlab_messy_data"

> rs <- tryCatch(dbSendQuery(con, sql), finally = print("query ok"))
[1] "query ok"

> data.pelanggan <- fetch(rs, n=-1)
```

```
> dbClearResult(rs)
[1] TRUE

> #Melakukan konversi nilai_belanja_setahun menjadi numerik
> #karena sebelumnya disimpan dalam bentuk character
> data.pelanggan$nilai_belanja_setahun <- as.numeric(data.pelanggan$nilai_belanja_set
ahun)

> summary(data.pelanggan)
 kode_pelanggan     nilai_belanja_setahun
 Length:155         Min.   : 237400
 Class :character   1st Qu.: 504800
 Mode  :character   Median : 851600
                    Mean   : 857226
                    3rd Qu.:1179800
                    Max.   :1537200
                    NA's   :4

> #Menghitung nilai tengah dengan function median dan disimpan dalam variable nilai_t
engah
> nilai_tengah <- median(data.pelanggan$nilai_belanja_setahun, na.rm=TRUE)

> #Mengisi missing value dengan nilai tengah
> data.pelanggan$nilai_belanja_setahun[is.na(data.pelanggan$nilai_belanja_setahun)] <
- nilai_tengah

> #Melihat summary setelah missing value
> summary(data.pelanggan)
 kode_pelanggan     nilai_belanja_setahun
 Length:155         Min.   : 237400
 Class :character   1st Qu.: 520000
 Mode  :character   Median : 851600
                    Mean   : 857081
                    3rd Qu.:1168250
                    Max.   :1537200

> #Menulis ke dalam file staging.enrichment.mean.xlsx
> write.xlsx(data.pelanggan, file="staging.enrichment.median.xlsx")

> all_cons <- dbListConnections(MySQL())

> for(con in all_cons)
+     + dbDisconnect(con)
```

Melakukan enrichment Kode Pos

Enrichment selanjutnya sekaligus yang terakhir yang akan kita akan lakukan adalah kode_pos.

	A	B	C	D	E	F
1	groupi	kode_pelangg	kode_p	alamat	jumlah_reco	kode_pos_enri
17	7	KD-00060	986455	Apartemen Kecapi Indah, Lt. 18 No. 1801	2	986455
18	8	KD-00008	813444	Kali Mars Cluster, No. 24C	2	813444
19	8	KD-00121	896112	Indah Mars Cluster, No. 22F	2	896112
20	9	KD-00009	896555	Jalan Kebon Jahe, No. F16 - Kota E	1	896555
21	10	KD-00010	987453	Perum Venus, Gg. Harimau No. 1A	3	987453
22	10	KD-00028	987453	Perum Venus, Gang. Kelinci No. 12	3	987453
23	10	KD-00125	-	Perum Venus, Gang. Harimau No. 4A	3	987453
24	11	KD-00011	967223	Cluster Ikan Mas, Taman Intan No. 2	2	967223
25	11	KD-00091	967223	Cluster Ikan Mas, Taman Baru No. 96	2	967223
26	12	KD-00013	666122	Jalan Hang Tuah, No. 11, Kota DM	2	666122
27	12	KD-00033	666122	Jalan Hang Tuah, No. 31, Kota DM	2	666122
28	13	KD-00014	-	Boulevard Raya Residences, Blok AA2 No. 88	2	817321
29	13	KD-00072	817321	Boulevard Raya Residences, Blok AB2 No. 102	2	817321
30	14	KD-00015	876511	Jalan Puri Arteri Raya, No. 88 - Kota T	2	876511
31	14	KD-00083	876511	Jalan Puri Arteri Raya, No. 88 - Kota T	2	876511
32	15	KD-00016	896550	Jalan Pahlawan, No. 69CCD	2	896550
33	15	KD-00057	896550	Jalan Pahlawan, No. 69FFF	2	896550
34	16	KD-00017	768034	Asrama Pelajar No. 22 A - Pondok Bima Sakti	2	768034
35	16	KD-00037	768034	Asrama Pelajar No. 11 B - Pondok Bima Sakti	2	768034
36	17	KD-00018	896555	Jalan Bintang Supernova, No. 78	2	896555
37	17	KD-00058	896555	Jalan Bintang Supernova, No. 78	2	896555
38	18	KD-00019	-	Jalan Wisma Tenteram Saja, No. A22	3	866162
39	18	KD-00048	866162	Jalan Wisma Tenteram Saja, No. A31	3	866162
40	18	KD-00070	696193	Jalan Wisma Tenteram Saja No. B-01	3	866162
41	19	KD-00020	476533	Jalan Manggis II, Gang Buntu No. 1	2	476533
42	19	KD-00080	476533	Jalan Manggis II - Gang Buntu No. 4	2	476533

Hal pertama yang kita lakukan adalah mencari duplikat. Untuk contoh pada praktek ini kita gunakan satu kolom saja, yaitu alamat dan dengan threshold yang lebih ketat yaitu angka 10 .

Catatan: pada praktek sebelumnya kita menggunakan angka 15.

Hal kedua yang perlu kita lakukan adalah mencari nomor grouping dari kode pos yang kosong. Kemudian berdasarkan grouping, kita ambil kode pos yang tidak kosong.

Ini adalah pendekatan sederhana setelah kita mengenal proses duplikasi namun cukup panjang. Mari kita langsung lakukan tugas praktek untuk memahaminya.

Tugas Praktek

Ada tiga bagian pada code editor, yaitu […1…], […2…] dan […3…] yang harus diisi sehingga proses pencarian grouping duplikat dan pengisian kode pos yang kosong bisa berjalan lancar.

Jika berjalan lancar, akan banyak output yang dihasilkan. Diantaranya untuk pencarian duplikasi hasilnya akan terlihat sebagai berikut.

```
> hasil.akhir
    grouping kode_pelanggan kode_pos
1          1       KD-00001   876511
2          1       KD-00012   876511
3          1       KD-00045   876511
4          1       KD-00778   876511
5          2       KD-00002   712983
6          2       KD-00075   712983
…
15         7       KD-00007   986455
16         7       KD-00060   986455
17         8       KD-00008   813444
18         8       KD-00121   896112
19         9       KD-00009   896555
20        10       KD-00010   987453
21        10       KD-00028   987453
22        10       KD-00125         -
23        11       KD-00011   967223
24        11       KD-00091   967223
25        12       KD-00013   666122
26        12       KD-00033   666122
27        13       KD-00014         -
28        13       KD-00072   817321
29        14       KD-00015   876511
30        14       KD-00083   876511
31        15       KD-00016   896550
32        15       KD-00057   896550
33        16       KD-00017   768034
34        16       KD-00037   768034
35        17       KD-00018   896555
36        17       KD-00058   896555
37        18       KD-00019         -
38        18       KD-00048   866162
```

39	18	KD-00070	696193
40	19	KD-00020	476533
41	19	KD-00080	476533
42	20	KD-00021	511432
43	20	KD-00074	511432
44	21	KD-00022	768031
45	21	KD-00095	768031
46	22	KD-00023	-
47	22	KD-00063	768091
48	22	KD-00148	768091
49	23	KD-00024	811613
50	24	KD-00025	813442

Grouping yang memiliki kode pos kosong ditandai dengan warna biru dan merah. Terlihat yang berwarna merah adalah grouping yang memiliki dua kode pos. Ini hal yang lumrah di kasus nyata, dan kembali menunjukkan proses pencarian duplikasi akurasinya tidak bisa 100 persen. Tapi ada kemungkinan lain, mungkin kode pos yang salah tulis? Ini dapat kita periksa setelah ada output lengkap dengan alamat.

Tapi selalu bisa ditingkatkan dengan membuat sistem yang lebih komplek – seperti yang telah kita bahas di penutupan bab sebelumnya.

Dan output lainnya adalah file yang dihasilkan, yaitu file **staging.enrichment.kode_pos.xlsx**. Tampilannya ketika dibuka dengan aplikasi Excel dan di-highlight bagian grouping kode pos yang hilang adalah sebagai berikut.

	A	B	C	D	E	F
1	groupi	kode_pelangga	kode_p	alamat	jumlah_reco	kode_pos_enri
17	7	KD-00060	986455	Apartemen Kecapi Indah, Lt. 18 No. 1801	2	986455
18	8	KD-00008	813444	Kali Mars Cluster, No. 24C	2	813444
19	8	KD-00121	896112	Indah Mars Cluster, No. 22F	2	896112
20	9	KD-00009	896555	Jalan Kebon Jahe, No. F16 - Kota E	1	896555
21	10	KD-00010	987453	Perum Venus, Gg. Harimau No. 1A	3	987453
22	10	KD-00028	987453	Perum Venus, Gang. Kelinci No. 12	3	987453
23	10	KD-00125	-	Perum Venus, Gang. Harimau No. 4A	3	987453
24	11	KD-00011	967223	Cluster Ikan Mas, Taman Intan No. 2	2	967223
25	11	KD-00091	967223	Cluster Ikan Mas, Taman Baru No. 96	2	967223
26	12	KD-00013	666122	Jalan Hang Tuah, No. 11, Kota DM	2	666122
27	12	KD-00033	666122	Jalan Hang Tuah, No. 31, Kota DM	2	666122
28	13	KD-00014	-	Boulevard Raya Residences, Blok AA2 No. 88	2	817321
29	13	KD-00072	817321	Boulevard Raya Residences, Blok AB2 No. 102	2	817321
30	14	KD-00015	876511	Jalan Puri Arteri Raya, No. 88 - Kota T	2	876511
31	14	KD-00083	876511	Jalan Puri Arteri Raya, No. 88 - Kota T	2	876511
32	15	KD-00016	896550	Jalan Pahlawan, No. 69CCD	2	896550
33	15	KD-00057	896550	Jalan Pahlawan, No. 69FFF	2	896550
34	16	KD-00017	768034	Asrama Pelajar No. 22 A - Pondok Bima Sakti	2	768034
35	16	KD-00037	768034	Asrama Pelajar No. 11 B - Pondok Bima Sakti	2	768034
36	17	KD-00018	896555	Jalan Bintang Supernova, No. 78	2	896555
37	17	KD-00058	896555	Jalan Bintang Supernova, No. 78	2	896555
38	18	KD-00019	-	Jalan Wisma Tenteram Saja, No. A22	3	866162
39	18	KD-00048	866162	Jalan Wisma Tenteram Saja, No. A31	3	866162
40	18	KD-00070	696193	Jalan Wisma Tenteram Saja No. B-01	3	866162
41	19	KD-00020	476533	Jalan Manggis II, Gang Buntu No. 1	2	476533
42	19	KD-00080	476533	Jalan Manggis II - Gang Buntu No. 4	2	476533

Nah, disini diberikan dua kolom kode pos, sebelum dan sesudah di-enrich sehingga Anda dapat membandingkannya dengan manual. Cobalah perhatikan untuk grouping 18, dimana Jalan Wisma ada blok A dan B, dengan kode pos yang sangat ekstrem berbeda. Ini dapat menjadi insight Anda untuk melakukan pengecekan data ke sumber data lebih lanjut.

Tapi tentunya dengan otomatisasi ini sudah meningkatkan produktivitas pembersihan data Anda dengan sangat tinggi.

Code Editor

```
library(stringdist)

library(openxlsx)


#Membaca file staging.final.xlsx

data.pelanggan <- read.xlsx("staging.final.xlsx")
```

```
str(data.pelanggan)

#------------------- 1. PERSIAPAN: MENEMUKAN DUPLIKASI ALAMAT ----------------------
#Konversi Factor
data.pelanggan$kode_pos <- as.factor(data.pelanggan$kode_pos)

#Inisialiasi variable grouping_no dengan nilai 1
grouping_no <- 1

#Inisialisasi hasil.akhir
hasil.akhir <- NULL

while(length(data.pelanggan$alamat)>0)
{
  #Variable referensi alamat diambil dari item pertama
  referensi.alamat <- data.pelanggan$alamat[1]

  #Menghitung jarak antara referensi dengan item alamat dengan method "lv"
  jarak.teks.alamat <- stringdist(referensi.alamat, data.pelanggan$alamat, method="lv")

  #Hasil filter jarak dengan threshold
  # - lebih kecil dari angka 10 untuk alamat
  #disimpan ke variable filter.jarak
  filter.jarak <- (jarak.teks.alamat < 10)

  #Melakukan filtering pada variable data.pelanggan, dan mengambil dua kolom
  #untuk disimpan ke dua variable
  kode_pelanggan.temp <- data.pelanggan[filter.jarak,]$kode_pelanggan
```

```
  alamat.temp <- data.pelanggan[filter.jarak,]$alamat
  kode_pos.temp <- data.pelanggan[filter.jarak,]$kode_pos

  #Konstruksi temporary variable
  var.temp <- data.frame(grouping=grouping_no,
kode_pelanggan=kode_pelanggan.temp, kode_pos=kode_pos.temp,
alamat=alamat.temp, jumlah_record=length(kode_pelanggan.temp))

  #Menggabungkan temporary variable dengan hasil sebelumnya
  hasil.akhir <- rbind(hasil.akhir, var.temp)

  #Membuang porsi yang sudah ditemukan
  data.pelanggan <- data.pelanggan[!filter.jarak,]

  #Menambahkan nilai grouping untuk diambil pada iterasi selanjutnya
  grouping_no <- grouping_no + 1
}
hasil.akhir
summary(hasil.akhir$kode_pos)
```

```
#---------------- 2. MENGISI KODE POS YANG KOSONG --------------------
#Inisialisasi kolom baru kode pos
hasil.akhir$kode_pos_enrich <-hasil.akhir$kode_pos

#Mengambil nomor grouping kode pos yang kosong
kode_pos_kosong <- hasil.akhir[hasil.akhir$kode_pos == "-",]

while(length(kode_pos_kosong$kode_pos)>0)
{
        grouping_no <- kode_pos_kosong$grouping[1]
        #Membuat variable filter
        filter.data <- hasil.akhir$grouping == grouping_no & hasil.akhir$kode_pos != "-"
        #Mengambil data pertama dari hasil filter dengan function head
        temp.data <- head(hasil.akhir[filter.data,],1)
        #Mengisi kolom kode_pos_enrich dengan kolom kode_pos yang ditemukan
hasil.akhir[hasil.akhir$grouping == grouping_no,]$kode_pos_enrich <-
temp.data$kode_pos

   #Menghapus row pertama dari variable kode_pos_kosong
   kode_pos_kosong <- kode_pos_kosong[-1,]
}
hasil.akhir
summary(hasil.akhir$kode_pos_enrich)

#Menulis hasil ke file staging.enrichment.kode_pos.xlsx
write.xlsx(hasil.akhir, file="staging.enrichment.kode_pos.xlsx")
```

Console

```
> library(stringdist)

> library(openxlsx)

> #Membaca file staging.final.xlsx
> data.pelanggan <- read.xlsx("staging.final.xlsx")

> str(data.pelanggan)
'data.frame':    155 obs. of  9 variables:
 $ kode_pelanggan    : chr  "KD-00001" "KD-00002" "KD-00003" "KD-00004" ...
 $ nama              : chr  "Agus Cahyonos" "Khairul Nissa" "Slamet Wiyanto" "DRS. Ma
ria Simangunsong" ...
 $ alamat            : chr  "Jalan Pulo Bambu No. 15, Kota Tenggara Lama" "Taman Vivo
Indah, Blok AA No. 7" "Meta Residences, No. 32C" "Gang Bulan Desember III, No. 9" ...
 $ no_telepon        : chr  "08298911112222" "+6287132221371404" "+6285725955303368"
"+6283376770990635" ...
 $ anomali_no_telepon: logi  TRUE FALSE FALSE FALSE FALSE FALSE ...
 $ kode_pos          : chr  "876511" "712983" "764550" "967220" ...
 $ tanggal_lahir     : chr  "08-02-1967" "23-10-1991" "23-11-1962" "17-02-2097" ...
 $ umur              : num  51.2 26.5 55.5 -78.9 31.7 ...
 $ umur_valid        : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...

> #------------------- 1. PERSIAPAN: MENEMUKAN DUPLIKASI ALAMAT ----------------------
--
> #Konversi Factor
> data.pelanggan$kode_pos <- as.factor(data.pelanggan$kode_pos)

> #Inisialiasi variable grouping_no dengan nilai 1
> grouping_no <- 1

> #Inisialisasi hasil.akhir
> hasil.akhir <- NULL

> while(length(data.pelanggan$alamat)>0)
+ {
+   #Variable referensi alamat diambil dari item pertama
+   referensi.alamat <- data.pelanggan$alamat[1]
+
+   #Menghitung jarak antara referensi dengan item alamat dengan method "lv"
+   jarak.teks.alamat <- stringdist(referensi.alamat, data.pelanggan$alamat, method="
lv")
+
+   #Hasil filter jarak dengan threshold
+   # - lebih kecil dari angka 10 untuk alamat
+   #disimpan ke variable filter.jarak
+   filter.jarak <- (jarak.teks.alamat < 10)
+
+   #Melakukan filtering pada variable data.pelanggan, dan mengambil dua kolom
+   #untuk disimpan ke dua variable
+   kode_pelanggan.temp <- data.pelanggan[filter.jarak,]$kode_pelanggan
+   alamat.temp <- data.pelanggan[filter.jarak,]$alamat
+   kode_pos.temp <- data.pelanggan[filter.jarak,]$kode_pos
```

```
+
+   #Konstruksi temporary variable
+   var.temp <- data.frame(grouping=grouping_no, kode_pelanggan=kode_pelanggan.temp,
kode_pos=kode_pos.temp, alamat=alamat.temp, jumlah_record=length(ko .... [TRUNCATED]

> hasil.akhir
    grouping kode_pelanggan kode_pos
1          1       KD-00001   876511
2          1       KD-00012   876511
3          1       KD-00045   876511
4          1       KD-00778   876511
5          2       KD-00002   712983
6          2       KD-00075   712983
7          3       KD-00003   764550
8          3       KD-00043   764550
9          4       KD-00004   967220
10         4       KD-00071   967220
11         4       KD-00093   967220
12         5       KD-00005   476511
13         5       KD-00101   476511
14         6       KD-00006   487851
15         7       KD-00007   986455
16         7       KD-00060   986455
17         8       KD-00008   813444
18         8       KD-00121   896112
19         9       KD-00009   896555
20        10       KD-00010   987453
21        10       KD-00028   987453
22        10       KD-00125        -
23        11       KD-00011   967223
24        11       KD-00091   967223
25        12       KD-00013   666122
26        12       KD-00033   666122
27        13       KD-00014        -
28        13       KD-00072   817321
29        14       KD-00015   876511
30        14       KD-00083   876511
31        15       KD-00016   896550
32        15       KD-00057   896550
33        16       KD-00017   768034
34        16       KD-00037   768034
35        17       KD-00018   896555
36        17       KD-00058   896555
37        18       KD-00019        -
38        18       KD-00048   866162
39        18       KD-00070   696193
40        19       KD-00020   476533
41        19       KD-00080   476533
42        20       KD-00021   511432
43        20       KD-00074   511432
44        21       KD-00022   768031
45        21       KD-00095   768031
46        22       KD-00023        -
47        22       KD-00063   768091
48        22       KD-00148   768091
```

```
49          23          KD-00024    811613
50          24          KD-00025    813442
51          24          KD-00086    813442
52          25          KD-00026    896555
53          26          KD-00027    877521
54          27          KD-00029    896566
55          28          KD-00030    349922
56          29          KD-00031    896114
57          29          KD-00068    567151
58          30          KD-00032    567130
59          30          KD-00053    567130
60          30          KD-00133    567130
61          31          KD-00034    877615
62          31          KD-00103    877613
63          31          KD-00143    877614
64          32          KD-00035    712984
65          32          KD-00076    712984
66          32          KD-00113    712984
67          32          KD-00298    712984
68          33          KD-00036    876552
69          33          KD-00126    876552
70          34          KD-00038    987452
71          34          KD-00117    987452
72          35          KD-00039    764449
73          35          KD-00087    764449
74          36          KD-00040    896115
75          37          KD-00041    896549
76          37          KD-00066    896549
77          37          KD-00127    896549
78          37          KD-00140    896549
79          38          KD-00042    696193
80          39          KD-00044    321321
81          39          KD-00492    321321
82          40          KD-00046    877521
83          40          KD-00137    877521
84          41          KD-00049    321321
85          41          KD-00141    321321
86          42          KD-00050    321321
87          42          KD-00110    321321
88          43          KD-00051    696193
89          44          KD-00052    567120
90          45          KD-00054    896549
91          45          KD-00094    896549
92          45          KD-00138    896549
93          46          KD-00055    696193
94          47          KD-00056    876551
95          47          KD-00111    876551
96          48          KD-00059          -
97          48          KD-00122    986455
98          49          KD-00061    896113
99          50          KD-00062    487451
100         51          KD-00064    987451
101         52          KD-00065    967222
102         53          KD-00067    967223
103         54          KD-00069    349981
```

```
104       54       KD-00114   349981
105       55       KD-00073   876512
106       56       KD-00077   987601
107       56       KD-00085   987601
108       57       KD-00078   817324
109       58       KD-00079   986456
110       59       KD-00081   967229
111       59       KD-00109   967229
112       60       KD-00082   967221
113       60       KD-00097   567120
114       60       KD-00150   967221
115       61       KD-00084   811613
116       61       KD-00104   811613
117       62       KD-00088   633429
118       62       KD-00132   633429
119       63       KD-00089   696193
120       64       KD-00090   511431
121       65       KD-00092   696193
122       66       KD-00096   633431
123       66       KD-00119   633430
124       67       KD-00098   696193
125       68       KD-00099   712984
126       68       KD-00192   712984
127       69       KD-00100   896549
128       70       KD-00102   666122
129       71       KD-00105   321321
130       72       KD-00106   896555
131       72       KD-00136   896555
132       73       KD-00107   893422
133       74       KD-00108   768035
134       75       KD-00112   696193
135       76       KD-00115   986455
136       76       KD-00128   986455
137       77       KD-00116   986455
138       77       KD-00144   986455
139       78       KD-00118   696193
140       79       KD-00120   567120
141       80       KD-00123   813442
142       81       KD-00124   321321
143       82       KD-00129   986454
144       83       KD-00130   876614
145       84       KD-00131   567151
146       85       KD-00134   986456
147       86       KD-00135   876612
148       87       KD-00139   511431
149       88       KD-00142   986455
150       89       KD-00145   896555
151       90       KD-00146   666123
152       91       KD-00147   967224
153       92       KD-00149   764450
154       92        KD-0047   764450
155       93       KD-00151   876612
                                                 alamat jumlah_record
1           Jalan Pulo Bambu No. 15, Kota Tenggara Lama              4
2                 Pulo Bambu No. 15, Kota Tenggara Lama              4
```

3	Pulo Bambu No. 57, Kota Tenggara Lama	4
4	Jalan Pulau Bambu No. 15 - Kota Tenggara Lama	4
5	Taman Vivo Indah, Blok AA No. 7	2
6	Taman Vivo Indah, Blok AA No. 7	2
7	Meta Residences, No. 32C	2
8	Meta Residences, No. 1A	2
9	Gang Bulan Desember III, No. 9	3
10	Gang Bulan Desember III, No. 155	3
11	Gang Bulan Desember III, No. 145	3
12	Jalan Tegal Sari Indah, No. D87 -- Kota H	2
13	Jalan Tegal Sari Indah, No. D77 -- Kota H	2
14	Perum Pluto, Blok C No. 1	1
15	Apartemen Kecapi Indah, Lt. 16 No. 1610	2
16	Apartemen Kecapi Indah, Lt. 18 No. 1801	2
17	Kali Mars Cluster, No. 24C	2
18	Indah Mars Cluster, No. 22F	2
19	Jalan Kebon Jahe, No. F16 - Kota E	1
20	Perum Venus, Gg. Harimau No. 1A	3
21	Perum Venus, Gang. Kelinci No. 12	3
22	Perum Venus, Gang. Harimau No. 4A	3
23	Cluster Ikan Mas, Taman Intan No. 2	2
24	Cluster Ikan Mas, Taman Baru No. 96	2
25	Jalan Hang Tuah, No. 11, Kota DM	2
26	Jalan Hang Tuah, No. 31, Kota DM	2
27	Boulevard Raya Residences, Blok AA2 No. 88	2
28	Boulevard Raya Residences, Blok AB2 No. 102	2
29	Jalan Puri Arteri Raya, No. 88 - Kota T	2
30	Jalan Puri Arteri Raya, No. 88 - Kota T	2
31	Jalan Pahlawan, No. 69CCD	2
32	Jalan Pahlawan, No. 69FFF	2
33	Asrama Pelajar No. 22 A - Pondok Bima Sakti	2
34	Asrama Pelajar No. 11 B - Pondok Bima Sakti	2
35	Jalan Bintang Supernova, No. 78	2
36	Jalan Bintang Supernova, No. 78	2
37	Jalan Wisma Tenteram Saja, No. A22	3
38	Jalan Wisma Tenteram Saja, No. A31	3
39	Jalan Wisma Tenteram Saja No. B-01	3
40	Jalan Manggis II, Gang Buntu No. 1	2
41	Jalan Manggis II - Gang Buntu No. 4	2
42	Puspa Loka, No. 98B, Kota Y	2
43	Puspa Loka, No. 98F, Kota Y	2
44	Asrama Perawat IV, No. 1 - Kota D	2
45	Asrama Perawat IV, No. 2 - Kota D	2
46	Jalan Macan Buntung, No. 1F	3
47	Jalan Macan Buntung, No. 4F	3
48	Jalan Macan Buntung, No. 1F - Kota D	3
49	Perum Maju Permai Persada Indah, Gang Kenari No. 3	1
50	Kampoeng Harimau, No. 81 - Kota K	2
51	Kampung Harimau, No. 88, Kota K	2
52	Jalan Kebon Jahe, Kota EntahDimana	1
53	Vila Bukit Sagitarius, Blok A1 No. 1	1
54	Jalan Kp. Kijang, Blok A1 - No. 2F	1
55	Pondok Bima Sakti, Jalan Asrama Pelajar No. 11FF	1
56	Gang Tupai, No. 7 - Desa CL	2
57	Gang Piranha, No. 3 - Desa BT	2

58	Vila Sempilan, No. 67 - Kota B	3
59	Vila Sempilan, No. 11 - Kota B	3
60	Vila Sempilan, No. 1 - Kota B	3
61	Perum Kali Meksiko, No. 8C	3
62	Perum Kali Meksiko, No. D22	3
63	Perum Kali Meksiko, No. 8F	3
64	Taman Bunga Langit, Jalan Timur No. 1	4
65	Taman Bunga Langit, Jalan Utara No. 3	4
66	Taman Bunga Langit, Jalan Selatan No. 12	4
67	Taman Bunga Langit, Jalan Utara No. 3	4
68	Vila Gunung Seribu, Blok O1 - No. 1	2
69	Vila Gunung Seribu, Blok F4 - No. 8	2
70	Perumahan Bina Andromeda, Jalan Teri No. 4	2
71	Perumahan Bina Andromeda, Jalan Salmon No. 22	2
72	Perum Indah Supernova II, No. 9	2
73	Perum Indah Supernova, No. 1	2
74	Gang Samun Saja No. 132, Kode Pos A99222	1
75	Jalan Pulau Sentosa No. 133	4
76	Jalan Pulau Sentosa No. 133	4
77	Jalan Pulau Sentosa No. 133	4
78	Jalan Pulau Sentosa No. 1335	4
79	Jalan Raya Hang Lekir, Kota Z, No. 62	1
80	Kompleks Pelaut Tangguh, No. 5A	2
81	Kompleks Pelaut Tangguh, No. 5A	2
82	Vila Bukit Sagitarius, Gang Kelapa No. 6	2
83	Vila Bukit Sagitarius, Gang. Sawit No. 3	2
84	Kompleks Permai Angkasa, Blok M No. 10	2
85	Kompleks Permai Angkasa, Blok J No. 09	2
86	Kompleks Selatan-Selatan, No. 121	2
87	Kompleks Selatan-Selatan, No. 111	2
88	Jalan Binjai 200, Kota L	1
89	Jalan Ring Road Neolitik, No. 1 RT 5	1
90	Jalan Gula Pahit, No. 001	3
91	Jalan Gula Pahit, No. 015	3
92	Jalan Gula Pahit, No. 081	3
93	Jalan Raya Jupiter Titan, No. 55	1
94	Vila Permata Intan Berkilau, Blok C5-7	2
95	Vila Permata Intan Berkilau, Blok A1/2	2
96	Perumahan Sektor Bougenville, Jalan Karet No. 7P	2
97	Perumahan Sektor Bougenville, Jalan Sawit No. 8A	2
98	Griya Asri Mawar Harum, Blok G No. 1	1
99	Perum Sektor 50, Gang Permai No. 5	1
100	Perumahan Catalina, Jalan Kereta Api No. 77	1
101	Corina Residences Apartment, No. 0612	1
102	Condominium Pesona Indah, No. 0708	1
103	Perum Titan, Jalan Trobos No. 8	2
104	Perum Titan, Jalan Kelinci No. 12	2
105	Jalan Puri Indah Menawan, No. 818 - Kota T	1
106	Jalan Sutomo Baru 21 - Kota M	2
107	Jalan Sutomo Baru No. 21 - Kota M	2
108	Blok C 2/4, Bukit Vivo Indah	1
109	Perumahan Duku Satu, Gang Merpati - No. 41	1
110	Bukit Vivo Indah, Blok C 2/4	2
111	Bukit Vivo Indah, Blok C 2/4	2
112	Gang Arwana, No. 6 - Kota S	3

```
113                       Gang Kelinci, No. 666 - Kota B             3
114                           Gang Arwana No. 12, Kota S             3
115           Perum Maju Permai P.I., Gang Kesturi No. 5             2
116           Perum Maju Permai P.I., Gang Kesturi No. 5             2
117                     Rusun Kerinci Indah, Lt. 5 No. 6             2
118                     Rusun Kerinci Indah, Lt. 6 No. 1             2
119                Jalan Raya Hang Lekir, No. 62 - Kota Z             1
120                           Ruko Almond Manis, Blok C7/8             1
121                       Jalan Bukit Tol Km. 3, No. 971             1
122                    Rumah Susun Eunos, Lantai 2 No. 2             2
123                  Rumah Susun Gelora, Lantai 1 No. 12             2
124   Jalan Pesisir No. 5, Kampoeng Maju Surya Gemilang              1
125           Taman Bunga Langit, Jalan Barat Laut No. 6             2
126           Taman Bunga Langit, Jalan Barat Laut No. 6             2
127           Jalan Asia No. 55, Kompleks Pelajar Kota C             1
128               Jalan Kangguru No. 92, RT 005 - kota R             1
129         Kompleks Akademi Perawat, Gang Farmasi No. 3             1
130               Jalan Kemenangan Besar, Blok C8 No. 22             2
131         Jalan Kemenangan Besar, Blok C8 No. 22 RT 02             2
132                     Kampung Kijang, Blok D3 - No. 12             1
133                  Apartement Clifften, Lantai 12 No. 3             1
134                    Jalan Raya Andromeda, Blok D No. 3             1
135       Apartemen Bukit Merah Annex Plaza, Lt 3 No. A1             2
136      Apartemen Bukit Merah, Annex Plaza, Lt 3 No. A1             2
137                Apartemen Lucky Beruntung, Lt. 5 No. 4             2
138                Apartemen Lucky Beruntung, Lt. 3 No. 4             2
139              Jalan Semantik Semut Berjalan, No. 3333              1
140                       Jalan Ring Road Konstan, No. 5             1
141                                Kampung Harimau, No. 3             1
142             Kompleks Nelayan Permai, Blok DD - 98/99             1
143 Perumahan Sektor Telekomunikasi, Jalan Afrika No. 3              1
144               Jalan Raya Griya Barbarosa, Blok AF 789            1
145                         Gang Piranha, No. 13 - Desa BT           1
146              Perumahan Duku Lima, Gang Perkutut No. 1            1
147                Kota T,  Jalan Taman Kencana No. 11112            1
148                          Ruko Azalea, No. 3 RT 001/002           1
149           Apartemen Bukit Baru, Dahlia Tower, No. A3             1
150                Jalan Kampung Kijang, Blok C5 - No. 9             1
151               Jalan G. Asri Mawar Harum Blok G No. 9             1
152               Cluster Griya Bima Sakti, Blok A No. 1             1
153                  Perum Bimasakti Raya, Blok A No. 10             2
154                  Perum Bimasakti Raya, Blok A No. 10             2
155                 Jalan Taman Kencana No. 11112, Kota T            1

> summary(hasil.akhir$kode_pos)
     - 321321 349922 349981 476511 476533 487451 487851 511431 511432 567120
     5      8      1      2      2      2      1      1      2      2      3
567130 567151 633429 633430 633431 666122 666123 696193 712983 712984 764449
     3      2      2      1      1      3      1      9      2      6      2
764450 764550 768031 768034 768035 768091 811613 813442 813444 817321 817324
     2      2      2      2      1      2      3      3      1      1      1
866162 876511 876512 876551 876552 876612 876614 877521 877613 877614 877615
     1      6      1      2      2      2      1      3      1      1      1
893422 896112 896113 896114 896115 896549 896550 896555 896566 967220 967221
     1      1      1      1      1      8      2      7      1      3      2
```

```
967222 967223 967224 967229 986454 986455 986456 987451 987452 987453 987601
     1      3      1      2      1      8      2      1      2      2      2

> #----------------- 2. MENGISI KODE POS YANG KOSONG --------------------
> #Inisialisasi kolom baru kode pos
> hasil.akhir$kode_pos_enrich <-hasil.akhir$kode_pos

> #Mengambil nomor grouping kode pos yang kosong
> kode_pos_kosong <- hasil.akhir[hasil.akhir$kode_pos == "-",]

> while(length(kode_pos_kosong$kode_pos)>0)
+ {
+       grouping_no <- kode_pos_kosong$grouping[1]
+       #Membuat variable filter
+       filter.data <- hasil.akhir$grouping == grouping_no & hasil.akhir$kode_pos !=
"-"
+       #Mengambil data pertama dari hasil filter dengan function head
+       temp.data <- head(hasil.akhir[filter.data,],1)
+       #Mengisi kolom kode_pos_enrich dengan kolom kode_pos yang ditemukan
+ hasil.akhir[hasil.akhir$grouping == grouping_no,]$kode_pos_enrich <- temp.data$kode
_pos
+
+     #Menghapus row pertama dari variable kode_pos_kosong
+     kode_pos_kosong <- kode_pos_kosong[-1,]
+ }

> hasil.akhir
    grouping kode_pelanggan kode_pos
1          1       KD-00001   876511
2          1       KD-00012   876511
3          1       KD-00045   876511
4          1       KD-00778   876511
5          2       KD-00002   712983
6          2       KD-00075   712983
7          3       KD-00003   764550
8          3       KD-00043   764550
9          4       KD-00004   967220
10         4       KD-00071   967220
11         4       KD-00093   967220
12         5       KD-00005   476511
13         5       KD-00101   476511
14         6       KD-00006   487851
15         7       KD-00007   986455
16         7       KD-00060   986455
17         8       KD-00008   813444
18         8       KD-00121   896112
19         9       KD-00009   896555
20        10       KD-00010   987453
21        10       KD-00028   987453
22        10       KD-00125        -
23        11       KD-00011   967223
24        11       KD-00091   967223
25        12       KD-00013   666122
26        12       KD-00033   666122
27        13       KD-00014        -
```

28	13	KD-00072	817321
29	14	KD-00015	876511
30	14	KD-00083	876511
31	15	KD-00016	896550
32	15	KD-00057	896550
33	16	KD-00017	768034
34	16	KD-00037	768034
35	17	KD-00018	896555
36	17	KD-00058	896555
37	18	KD-00019	-
38	18	KD-00048	866162
39	18	KD-00070	696193
40	19	KD-00020	476533
41	19	KD-00080	476533
42	20	KD-00021	511432
43	20	KD-00074	511432
44	21	KD-00022	768031
45	21	KD-00095	768031
46	22	KD-00023	-
47	22	KD-00063	768091
48	22	KD-00148	768091
49	23	KD-00024	811613
50	24	KD-00025	813442
51	24	KD-00086	813442
52	25	KD-00026	896555
53	26	KD-00027	877521
54	27	KD-00029	896566
55	28	KD-00030	349922
56	29	KD-00031	896114
57	29	KD-00068	567151
58	30	KD-00032	567130
59	30	KD-00053	567130
60	30	KD-00133	567130
61	31	KD-00034	877615
62	31	KD-00103	877613
63	31	KD-00143	877614
64	32	KD-00035	712984
65	32	KD-00076	712984
66	32	KD-00113	712984
67	32	KD-00298	712984
68	33	KD-00036	876552
69	33	KD-00126	876552
70	34	KD-00038	987452
71	34	KD-00117	987452
72	35	KD-00039	764449
73	35	KD-00087	764449
74	36	KD-00040	896115
75	37	KD-00041	896549
76	37	KD-00066	896549
77	37	KD-00127	896549
78	37	KD-00140	896549
79	38	KD-00042	696193
80	39	KD-00044	321321
81	39	KD-00492	321321
82	40	KD-00046	877521

```
83        40        KD-00137   877521
84        41        KD-00049   321321
85        41        KD-00141   321321
86        42        KD-00050   321321
87        42        KD-00110   321321
88        43        KD-00051   696193
89        44        KD-00052   567120
90        45        KD-00054   896549
91        45        KD-00094   896549
92        45        KD-00138   896549
93        46        KD-00055   696193
94        47        KD-00056   876551
95        47        KD-00111   876551
96        48        KD-00059        -
97        48        KD-00122   986455
98        49        KD-00061   896113
99        50        KD-00062   487451
100       51        KD-00064   987451
101       52        KD-00065   967222
102       53        KD-00067   967223
103       54        KD-00069   349981
104       54        KD-00114   349981
105       55        KD-00073   876512
106       56        KD-00077   987601
107       56        KD-00085   987601
108       57        KD-00078   817324
109       58        KD-00079   986456
110       59        KD-00081   967229
111       59        KD-00109   967229
112       60        KD-00082   967221
113       60        KD-00097   567120
114       60        KD-00150   967221
115       61        KD-00084   811613
116       61        KD-00104   811613
117       62        KD-00088   633429
118       62        KD-00132   633429
119       63        KD-00089   696193
120       64        KD-00090   511431
121       65        KD-00092   696193
122       66        KD-00096   633431
123       66        KD-00119   633430
124       67        KD-00098   696193
125       68        KD-00099   712984
126       68        KD-00192   712984
127       69        KD-00100   896549
128       70        KD-00102   666122
129       71        KD-00105   321321
130       72        KD-00106   896555
131       72        KD-00136   896555
132       73        KD-00107   893422
133       74        KD-00108   768035
134       75        KD-00112   696193
135       76        KD-00115   986455
136       76        KD-00128   986455
137       77        KD-00116   986455
```

```
138       77       KD-00144   986455
139       78       KD-00118   696193
140       79       KD-00120   567120
141       80       KD-00123   813442
142       81       KD-00124   321321
143       82       KD-00129   986454
144       83       KD-00130   876614
145       84       KD-00131   567151
146       85       KD-00134   986456
147       86       KD-00135   876612
148       87       KD-00139   511431
149       88       KD-00142   986455
150       89       KD-00145   896555
151       90       KD-00146   666123
152       91       KD-00147   967224
153       92       KD-00149   764450
154       92        KD-0047   764450
155       93       KD-00151   876612
                                                  alamat jumlah_record
1           Jalan Pulo Bambu No. 15, Kota Tenggara Lama              4
2                 Pulo Bambu No. 15, Kota Tenggara Lama              4
3                 Pulo Bambu No. 57, Kota Tenggara Lama              4
4         Jalan Pulau Bambu No. 15 - Kota Tenggara Lama              4
5                       Taman Vivo Indah, Blok AA No. 7              2
6                       Taman Vivo Indah, Blok AA No. 7              2
7                              Meta Residences, No. 32C              2
8                               Meta Residences, No. 1A              2
9                        Gang Bulan Desember III, No. 9              3
10                      Gang Bulan Desember III, No. 155             3
11                      Gang Bulan Desember III, No. 145             3
12             Jalan Tegal Sari Indah, No. D87 -- Kota H              2
13             Jalan Tegal Sari Indah, No. D77 -- Kota H              2
14                            Perum Pluto, Blok C No. 1              1
15               Apartemen Kecapi Indah, Lt. 16 No. 1610              2
16               Apartemen Kecapi Indah, Lt. 18 No. 1801              2
17                           Kali Mars Cluster, No. 24C              2
18                          Indah Mars Cluster, No. 22F              2
19                    Jalan Kebon Jahe, No. F16 - Kota E             1
20                       Perum Venus, Gg. Harimau No. 1A             3
21                     Perum Venus, Gang. Kelinci No. 12             3
22                     Perum Venus, Gang. Harimau No. 4A             3
23                   Cluster Ikan Mas, Taman Intan No. 2             2
24                   Cluster Ikan Mas, Taman Baru No. 96             2
25                      Jalan Hang Tuah, No. 11, Kota DM             2
26                      Jalan Hang Tuah, No. 31, Kota DM             2
27           Boulevard Raya Residences, Blok AA2 No. 88              2
28           Boulevard Raya Residences, Blok AB2 No. 102             2
29               Jalan Puri Arteri Raya, No. 88 - Kota T             2
30               Jalan Puri Arteri Raya, No. 88 - Kota T             2
31                           Jalan Pahlawan, No. 69CCD               2
32                            Jalan Pahlawan, No. 69FFF              2
33           Asrama Pelajar No. 22 A - Pondok Bima Sakti             2
34           Asrama Pelajar No. 11 B - Pondok Bima Sakti             2
35                      Jalan Bintang Supernova, No. 78              2
36                      Jalan Bintang Supernova, No. 78              2
```

37	Jalan Wisma Tenteram Saja, No. A22	3
38	Jalan Wisma Tenteram Saja, No. A31	3
39	Jalan Wisma Tenteram Saja No. B-01	3
40	Jalan Manggis II, Gang Buntu No. 1	2
41	Jalan Manggis II - Gang Buntu No. 4	2
42	Puspa Loka, No. 98B, Kota Y	2
43	Puspa Loka, No. 98F, Kota Y	2
44	Asrama Perawat IV, No. 1 - Kota D	2
45	Asrama Perawat IV, No. 2 - Kota D	2
46	Jalan Macan Buntung, No. 1F	3
47	Jalan Macan Buntung, No. 4F	3
48	Jalan Macan Buntung, No. 1F - Kota D	3
49	Perum Maju Permai Persada Indah, Gang Kenari No. 3	1
50	Kampoeng Harimau, No. 81 - Kota K	2
51	Kampung Harimau, No. 88, Kota K	2
52	Jalan Kebon Jahe, Kota EntahDimana	1
53	Vila Bukit Sagitarius, Blok A1 No. 1	1
54	Jalan Kp. Kijang, Blok A1 - No. 2F	1
55	Pondok Bima Sakti, Jalan Asrama Pelajar No. 11FF	1
56	Gang Tupai, No. 7 - Desa CL	2
57	Gang Piranha, No. 3 - Desa BT	2
58	Vila Sempilan, No. 67 - Kota B	3
59	Vila Sempilan, No. 11 - Kota B	3
60	Vila Sempilan, No. 1 - Kota B	3
61	Perum Kali Meksiko, No. 8C	3
62	Perum Kali Meksiko, No. D22	3
63	Perum Kali Meksiko, No. 8F	3
64	Taman Bunga Langit, Jalan Timur No. 1	4
65	Taman Bunga Langit, Jalan Utara No. 3	4
66	Taman Bunga Langit, Jalan Selatan No. 12	4
67	Taman Bunga Langit, Jalan Utara No. 3	4
68	Vila Gunung Seribu, Blok O1 - No. 1	2
69	Vila Gunung Seribu, Blok F4 - No. 8	2
70	Perumahan Bina Andromeda, Jalan Teri No. 4	2
71	Perumahan Bina Andromeda, Jalan Salmon No. 22	2
72	Perum Indah Supernova II, No. 9	2
73	Perum Indah Supernova, No. 1	2
74	Gang Samun Saja No. 132, Kode Pos A99222	1
75	Jalan Pulau Sentosa No. 133	4
76	Jalan Pulau Sentosa No. 133	4
77	Jalan Pulau Sentosa No. 133	4
78	Jalan Pulau Sentosa No. 1335	4
79	Jalan Raya Hang Lekir, Kota Z, No. 62	1
80	Kompleks Pelaut Tangguh, No. 5A	2
81	Kompleks Pelaut Tangguh, No. 5A	2
82	Vila Bukit Sagitarius, Gang Kelapa No. 6	2
83	Vila Bukit Sagitarius, Gang. Sawit No. 3	2
84	Kompleks Permai Angkasa, Blok M No. 10	2
85	Kompleks Permai Angkasa, Blok J No. 09	2
86	Kompleks Selatan-Selatan, No. 121	2
87	Kompleks Selatan-Selatan, No. 111	2
88	Jalan Binjai 200, Kota L	1
89	Jalan Ring Road Neolitik, No. 1 RT 5	1
90	Jalan Gula Pahit, No. 001	3
91	Jalan Gula Pahit, No. 015	3

92	Jalan Gula Pahit, No. 081	3
93	Jalan Raya Jupiter Titan, No. 55	1
94	Vila Permata Intan Berkilau, Blok C5-7	2
95	Vila Permata Intan Berkilau, Blok A1/2	2
96	Perumahan Sektor Bougenville, Jalan Karet No. 7P	2
97	Perumahan Sektor Bougenville, Jalan Sawit No. 8A	2
98	Griya Asri Mawar Harum, Blok G No. 1	1
99	Perum Sektor 50, Gang Permai No. 5	1
100	Perumahan Catalina, Jalan Kereta Api No. 77	1
101	Corina Residences Apartment, No. 0612	1
102	Condominium Pesona Indah, No. 0708	1
103	Perum Titan, Jalan Trobos No. 8	2
104	Perum Titan, Jalan Kelinci No. 12	2
105	Jalan Puri Indah Menawan, No. 818 - Kota T	1
106	Jalan Sutomo Baru 21 - Kota M	2
107	Jalan Sutomo Baru No. 21 - Kota M	2
108	Blok C 2/4, Bukit Vivo Indah	1
109	Perumahan Duku Satu, Gang Merpati - No. 41	1
110	Bukit Vivo Indah, Blok C 2/4	2
111	Bukit Vivo Indah, Blok C 2/4	2
112	Gang Arwana, No. 6 - Kota S	3
113	Gang Kelinci, No. 666 - Kota B	3
114	Gang Arwana No. 12, Kota S	3
115	Perum Maju Permai P.I., Gang Kesturi No. 5	2
116	Perum Maju Permai P.I., Gang Kesturi No. 5	2
117	Rusun Kerinci Indah, Lt. 5 No. 6	2
118	Rusun Kerinci Indah, Lt. 6 No. 1	2
119	Jalan Raya Hang Lekir, No. 62 - Kota Z	1
120	Ruko Almond Manis, Blok C7/8	1
121	Jalan Bukit Tol Km. 3, No. 971	1
122	Rumah Susun Eunos, Lantai 2 No. 2	2
123	Rumah Susun Gelora, Lantai 1 No. 12	2
124	Jalan Pesisir No. 5, Kampoeng Maju Surya Gemilang	1
125	Taman Bunga Langit, Jalan Barat Laut No. 6	2
126	Taman Bunga Langit, Jalan Barat Laut No. 6	2
127	Jalan Asia No. 55, Kompleks Pelajar Kota C	1
128	Jalan Kangguru No. 92, RT 005 - kota R	1
129	Kompleks Akademi Perawat, Gang Farmasi No. 3	1
130	Jalan Kemenangan Besar, Blok C8 No. 22	2
131	Jalan Kemenangan Besar, Blok C8 No. 22 RT 02	2
132	Kampung Kijang, Blok D3 - No. 12	1
133	Apartement Clifften, Lantai 12 No. 3	1
134	Jalan Raya Andromeda, Blok D No. 3	1
135	Apartemen Bukit Merah Annex Plaza, Lt 3 No. A1	2
136	Apartemen Bukit Merah, Annex Plaza, Lt 3 No. A1	2
137	Apartemen Lucky Beruntung, Lt. 5 No. 4	2
138	Apartemen Lucky Beruntung, Lt. 3 No. 4	2
139	Jalan Semantik Semut Berjalan, No. 3333	1
140	Jalan Ring Road Konstan, No. 5	1
141	Kampung Harimau, No. 3	1
142	Kompleks Nelayan Permai, Blok DD - 98/99	1
143	Perumahan Sektor Telekomunikasi, Jalan Afrika No. 3	1
144	Jalan Raya Griya Barbarosa, Blok AF 789	1
145	Gang Piranha, No. 13 - Desa BT	1
146	Perumahan Duku Lima, Gang Perkutut No. 1	1

```
147              Kota T,  Jalan Taman Kencana No. 11112             1
148                       Ruko Azalea, No. 3 RT 001/002             1
149          Apartemen Bukit Baru, Dahlia Tower, No. A3             1
150               Jalan Kampung Kijang, Blok C5 - No. 9             1
151              Jalan G. Asri Mawar Harum Blok G No. 9             1
152              Cluster Griya Bima Sakti, Blok A No. 1             1
153                 Perum Bimasakti Raya, Blok A No. 10             2
154                 Perum Bimasakti Raya, Blok A No. 10             2
155               Jalan Taman Kencana No. 11112, Kota T             1
    kode_pos_enrich
1            876511
2            876511
3            876511
4            876511
5            712983
6            712983
7            764550
8            764550
9            967220
10           967220
11           967220
12           476511
13           476511
14           487851
15           986455
16           986455
17           813444
18           896112
19           896555
20           987453
21           987453
22           987453
23           967223
24           967223
25           666122
26           666122
27           817321
28           817321
29           876511
30           876511
31           896550
32           896550
33           768034
34           768034
35           896555
36           896555
37           866162
38           866162
39           866162
40           476533
41           476533
42           511432
43           511432
44           768031
45           768031
```

```
46          768091
47          768091
48          768091
49          811613
50          813442
51          813442
52          896555
53          877521
54          896566
55          349922
56          896114
57          567151
58          567130
59          567130
60          567130
61          877615
62          877613
63          877614
64          712984
65          712984
66          712984
67          712984
68          876552
69          876552
70          987452
71          987452
72          764449
73          764449
74          896115
75          896549
76          896549
77          896549
78          896549
79          696193
80          321321
81          321321
82          877521
83          877521
84          321321
85          321321
86          321321
87          321321
88          696193
89          567120
90          896549
91          896549
92          896549
93          696193
94          876551
95          876551
96          986455
97          986455
98          896113
99          487451
100         987451
```

```
101         967222
102         967223
103         349981
104         349981
105         876512
106         987601
107         987601
108         817324
109         986456
110         967229
111         967229
112         967221
113         567120
114         967221
115         811613
116         811613
117         633429
118         633429
119         696193
120         511431
121         696193
122         633431
123         633430
124         696193
125         712984
126         712984
127         896549
128         666122
129         321321
130         896555
131         896555
132         893422
133         768035
134         696193
135         986455
136         986455
137         986455
138         986455
139         696193
140         567120
141         813442
142         321321
143         986454
144         876614
145         567151
146         986456
147         876612
148         511431
149         986455
150         896555
151         666123
152         967224
153         764450
154         764450
155         876612
```

```
> summary(hasil.akhir$kode_pos_enrich)
     - 321321 349922 349981 476511 476533 487451 487851 511431 511432 567120
     0      8      1      2      2      2      1      1      2      2      3
567130 567151 633429 633430 633431 666122 666123 696193 712983 712984 764449
     3      2      2      1      1      3      1      8      2      6      2
764450 764550 768031 768034 768035 768091 811613 813442 813444 817321 817324
     2      2      2      2      1      3      3      3      1      2      1
866162 876511 876512 876551 876552 876612 876614 877521 877613 877614 877615
     3      6      1      2      2      2      1      3      1      1      1
893422 896112 896113 896114 896115 896549 896550 896555 896566 967220 967221
     1      1      1      1      1      8      2      7      1      3      2
967222 967223 967224 967229 986454 986455 986456 987451 987452 987453 987601
     1      3      1      2      1      9      2      1      2      3      2

> #Menulis hasil ke file staging.enrichment.kode_pos.xlsx
> write.xlsx(hasil.akhir, file="staging.enrichment.kode_pos.xlsx")
```

Konsolidasi Data Akhir

Tiba saatnya kita menggabungkan seluruh hasil berikut. Skenario konsolidasi sumber data kita adalah sebagai berikut:

- Hasil enrichment (di dalamnya sudah ada informasi grouping duplikat).
  - Enrichment untuk nilai belanja setahun dengan metode mean, yang median tidak kita ambil – file dengan nama **enrichment.mean.xlsx**.
  - Enrichment untuk nilai kode pos – file dengan nama **enrichment.kode_pos.xlsx**.
- Hasil standarisasi – file dengan nama **final.xlsx**.

Kesemua file tersebut memiliki kolom kode_pelanggan yang menjadi key atau kunci untuk menggabungkan seluruh file tersebut dengan function **merge**.

Berbeda dengan sebelumnya, function merge kali ini ada yang mengambil field-field tertentu saja.

Tugas Praktek

Lengkapi code editor dengan menggantikan bagian […1…] dan […3…] dengan nama file yang sesuai dan […4…] dengan function **merge**.

Jika berjalan dengan lancar, maka outputnya adalah satu file bernama "**hasil.final.xlsx**" yang jika dibuka dengan Excel tampilannya adalah sebagai berikut.

	A	B	C	D	E	F	G	H	I	J	K	L
1	kode_pelanggan	nama	alamat	no_telepon	anomali_no_telepon	kode_pos	tanggal_lahir	umur	umur_valid	nilai_belanja_setahun	kode_pos_enrich	grouping
2	KD-00001	Agus Cahyonos	Jalan Pulo Bambu No. 15, Kota Tenggara Lama	08298911112222	TRUE	876511	08-02-1967	51.24931507	TRUE	1082900	876511	1
3	KD-00002	Khairul Nissa	Taman Vivo Indah, Blok AA No. 7	+6287132221371404	FALSE	712983	23-10-1991	26.52876712	TRUE	1336200	712983	2
4	KD-00003	Slamet Wiyanto	Meta Residences, No. 32C	+6285725955303368	FALSE	764550	23-11-1962	55.4630137	TRUE	601700	764550	3
5	KD-00004	DRS. Maria Simangunsong	Gang Bulan Desember III, No. 9	+6283376770990635	FALSE	967220	17-02-2097	-78.8657534	TRUE	451500	967220	4
6	KD-00005	Prihatin Setyonugroho	Jalan Tegal Sari Indah, No. D87 -- Kota H	+6286843623971825	FALSE	476511	19-08-1986	31.70958904	TRUE	1488900	476511	5
7	KD-00006	DR. Candra Wijaya	Perum Pluto, Blok C No. 1	+6284063423953696	FALSE	487851	05-09-1990	27.66027397	TRUE	257100	487851	6
8	KD-00007	Indra Kurniawan, ST	Apartemen Kecapi Indah, Lt. 16 No. 1610	+6283840529196797	FALSE	986455	23-10-1979	38.5369863	TRUE	805900	986455	7
9	KD-00008	Willy Sanjaya	Kali Mars Cluster, No. 24C	+6285312577710538	FALSE	813444	22-07-1973	44.79452055	TRUE	879800	813444	8
10	KD-00009	Antonius Winarta	Jalan Kebon Jahe, No. F16 - Kota E	+6282722234294686	FALSE	896555				272600	896555	9
11	KD-00010	Sri Wahyuni, Ir	Perum Venus, Gg. Harimau No. 1A	+6284079659289143	FALSE	987453	23-10-1991	26.52876712	TRUE	389400	987453	10
12	KD-00011	Rosalina Kurnia	Cluster Ikan Mas, Taman Intan No. 2	+6288339032314103	FALSE	967223	12-01-1969	49.32054795	TRUE	1497900	967223	11
13	KD-00012	Cahyono, Agus	Pulo Bambu No. 15, Kota Tenggara Lama	+62829891111222	TRUE	876511	08-02-1967	51.24931507	TRUE	488400	876511	1
14	KD-00013	Danang Santosa	Jalan Hang Tuah, No. 11, Kota DM	+6282672925000608	FALSE	666122	22-04-1933	85.07123288	FALSE	1138400	666122	12
15	KD-00014	Elisabeth Suryadinata, SKOM, ST	Boulevard Raya Residences, Blok AA2 No. 88	+6285455084014504	FALSE	-	23-10-1995	22.5260274	TRUE	474600	817321	13
16	KD-00015	Mario Setiawan	Jalan Puri Arteri Raya, No. 88 - Kota T	+6282989111122220	FALSE	876511	09-08-1972	45.74520548	TRUE	893600	876511	14
17	KD-00016	Indra K.	Jalan Pahlawan, No. 69CCD	+6289222405928430	FALSE	896550	28-05-1969	48.94794521	TRUE	925200	896550	15
18	KD-00017	Irfan Putra Wijaya	Asrama Pelajar No. 22 A - Pondok Bima Sakti	+6289984358708389	FALSE	768034	23-11-1962	55.4630137	TRUE	525300	768034	16
19	KD-00018	Sudirman Kartono	Jalan Bintang Supernova, No. 78	+6282283957103749	FALSE	896555	19-03-1950	68.15342466	TRUE	1117000	896555	17
20	KD-00019	Maria Yuniarti	Jalan Wisma Tenteram Saja, No. A22	+6289317147992822	FALSE	-	23-11-1962	55.4630137	TRUE	733500	866162	18
21	KD-00020	Hendri Winarto	Jalan Manggis II, Gang Buntu No. 1	+6287384329533477	FALSE	476533	13-11-1962	55.49041096	TRUE	990900	476533	19
22	KD-00021	Paulus Angkasa Putra	Puspa Loka, No. 98B, Kota Y	+6285991672131933	FALSE	511432	14-03-1979	39.14794521	TRUE	825500	511432	20
23	KD-00022	Mbak Dian Sukowati	Asrama Perawat IV, No. 1 - Kota D	+6285796817992325	FALSE	768031	25-07-1974	43.78630137	TRUE	1527400	768031	21
24	KD-00023	Ir. Yahya Permata	Jalan Macan Buntung, No. 1F	+6287660464098623	FALSE	-	12-01-1971	47.32054795	TRUE	1206000	768091	22
25	KD-00024	Solihin Chaerul	Perum Maju Permai Persada Indah, Gang Kenari No.	+6281718632538241	FALSE	811613	12-07-1977	40.81917808	TRUE	1471900	811613	23
26	KD-00025	DRG. Euis Rosidawati	Kampoeng Harimau, No. 81 - Kota K	+6286035230854391	FALSE	813442	19-03-1950	68.15342466	TRUE	543000	813442	24
27	KD-00026	Anton Winarta	Jalan Kebon Jahe, Kota EntahDimana	+6284204043307629	FALSE	896555	12-01-1969	49.32054795	TRUE	459200	896555	25
28	KD-00027	Djoko Wardoyo, Drs.	Vila Bukit Sagitarius, Blok A1 No. 1	+6284871003581659	FALSE	877521	23-11-1962	55.4630137	TRUE	536000	877521	26
29	KD-00028	Aman Pakpahan	Perum Venus, Gang. Kelinci No. 12	+6289311313046417	FALSE	987453	23-10-1991	26.52876712	TRUE	361100	987453	10
30	KD-00029	Sri Rahayu	Jalan Kp. Kijang, Blok A1 - No. 2F	+6283177123456315	FALSE	896566	13-11-1962	55.49041096	TRUE	1293600	896566	27
31	KD-00030	Hendi	Pondok Bima Sakti, Jalan Asrama Pelajar No. 11FF	+6282261101749552	FALSE	349922	28-02-1969	49.19178082	TRUE	308800	349922	28
32	KD-00031	Risman Suyanyo Permata	Gang Tunai, No. 7 - Desa CI	+6287382247209814	FALSE	896114	27-02-1976	42.19178082	TRUE	857226.4901	896114	29

Terlihat data jauh lebih rapi dengan lebih banyak informasi seperti infomasi anomali no telepon, umur dan apakah umur valid atau ga, kemudian informasi duplikat, dan missing value yang telah disii untuk kolom kode pos dan nilai belanja setahun.

Code Editor

```
library(openxlsx)


#Membaca file staging.enrichment.mean.xlsx dan menyimpannnya dalam variable staging.enrichment.mean

staging.enrichment.mean <- read.xlsx("staging.enrichment.mean.xlsx")


#Membaca file staging.enrichment.kode_pos.xlsx dan menyimpannnya dalam variable staging.enrichment.kode_pos

staging.enrichment.kode_pos <- read.xlsx("staging.enrichment.kode_pos.xlsx")


#Membaca file staging.final.xlsx dan menyimpannnya dalam variable staging.final

staging.final <- read.xlsx("staging.final.xlsx")


staging.enrichment.mean


#Membaca file staging.enrichment.kode_pos.xlsx dan menyimpannnya dalam variable staging.enrichment.kode_pos

staging.enrichment.kode_pos <- read.xlsx("staging.enrichment.kode_pos.xlsx")

#Ambil field, kode_pelanggan, dan kode_pos_enrich saja

staging.enrichment.kode_pos <- staging.enrichment.kode_pos[,c("kode_pelanggan", "kode_pos_enrich", "grouping")]

staging.enrichment.kode_pos


#Membaca file staging.final.xlsx dan menyimpannnya dalam variable staging.final

staging.final <- read.xlsx("staging.final.xlsx")


#Menggabungkan variable staging.enrichment.mean dengan staging.enrichment.kode_pos melalui kolom kode_pelanggan
```

hasil.final <- merge(x=staging.enrichment.mean, y=staging.enrichment.kode_pos, by.x = "kode_pelanggan", by.y = "kode_pelanggan", all = TRUE)

#Menggabungkan variable staging.final dengan hasil.final melalui kolom kode_pelanggan

hasil.final <- merge(x=staging.final, y=hasil.final, by.x = "kode_pelanggan", by.y = "kode_pelanggan", all = TRUE)

hasil.final

#Menulis hasil ke file staging.final.xlsx

write.xlsx(hasil.final, file="hasil.final.xlsx")

Console

```
> library(openxlsx)

> #Membaca file staging.enrichment.mean.xlsx dan menyimpannnya dalam variable staging
.enrichment.mean
> staging.enrichment.mean <- read.xlsx("staging.enrichment.mean.xlsx")

> #Membaca file staging.enrichment.kode_pos.xlsx dan menyimpannnya dalam variable sta
ging.enrichment.kode_pos
> staging.enrichment.kode_pos <- read.xlsx("staging.enrichment.kode_pos.xlsx")

> #Membaca file staging.final.xlsx dan menyimpannnya dalam variable staging.final
> staging.final <- read.xlsx("staging.final.xlsx")

> staging.enrichment.mean
    kode_pelanggan nilai_belanja_setahun
1         KD-00032             1275600.0
2         KD-00053              317800.0
3         KD-00133             1537200.0
4         KD-00056             1524700.0
5         KD-00111              655400.0
6         KD-00036             1444400.0
7         KD-00126              350400.0
8         KD-00137              354600.0
9         KD-00046              541300.0
10        KD-00027              536000.0
11        KD-00002             1336200.0
12        KD-00075             1316500.0
13        KD-00076              725600.0
14        KD-00035              398200.0
15        KD-00113              311000.0
16        KD-00099             1491900.0
```

17	KD-00132	538400.0
18	KD-00088	558000.0
19	KD-00119	286200.0
20	KD-00096	1034600.0
21	KD-00139	1128000.0
22	KD-00090	530600.0
23	KD-00074	1452900.0
24	KD-00021	825500.0
25	KD-00045	437200.0
26	KD-00012	488400.0
27	KD-00030	308800.0
28	KD-00129	992100.0
29	KD-00122	732800.0
30	KD-00059	1490800.0
31	KD-00079	489900.0
32	KD-00134	554300.0
33	KD-00064	661500.0
34	KD-00038	588300.0
35	KD-00117	854400.0
36	KD-00010	389400.0
37	KD-00028	361100.0
38	KD-00125	1339400.0
39	KD-00069	1147500.0
40	KD-00114	613600.0
41	KD-00062	471300.0
42	KD-00006	257100.0
43	KD-00024	1471900.0
44	KD-00084	560900.0
45	KD-00104	973700.0
46	KD-00103	1276600.0
47	KD-00143	1221900.0
48	KD-00034	1190900.0
49	KD-00087	959200.0
50	KD-00039	1086300.0
51	KD-0047	950200.0
52	KD-00149	413100.0
53	KD-00003	601700.0
54	KD-00043	237400.0
55	KD-00135	399900.0
56	KD-00050	453800.0
57	KD-00110	857226.5
58	KD-00049	1135500.0
59	KD-00141	1362200.0
60	KD-00044	904900.0
61	KD-00124	1419000.0
62	KD-00105	341900.0
63	KD-00107	1526400.0
64	KD-00086	293800.0
65	KD-00123	326300.0
66	KD-00025	543000.0
67	KD-00008	879800.0
68	KD-00005	1488900.0
69	KD-00101	770300.0
70	KD-00001	1082900.0
71	KD-00020	990900.0

72	KD-00080	489000.0
73	KD-00102	514700.0
74	KD-00146	1437400.0
75	KD-00048	494900.0
76	KD-00019	733500.0
77	KD-00151	1437600.0
78	KD-00130	1503700.0
79	KD-00073	538800.0
80	KD-00778	907500.0
81	KD-00066	253900.0
82	KD-00041	265900.0
83	KD-00140	912100.0
84	KD-00116	679800.0
85	KD-00127	997500.0
86	KD-00057	323100.0
87	KD-00016	925200.0
88	KD-00063	1100200.0
89	KD-00148	857226.5
90	KD-00023	1206000.0
91	KD-00029	1293600.0
92	KD-00136	1151000.0
93	KD-00106	974100.0
94	KD-00026	459200.0
95	KD-00145	1350600.0
96	KD-00018	1117000.0
97	KD-00058	1216800.0
98	KD-00051	1168700.0
99	KD-00144	577600.0
100	KD-00128	657300.0
101	KD-00115	998300.0
102	KD-00009	272600.0
103	KD-00092	756300.0
104	KD-00070	379700.0
105	KD-00118	1060600.0
106	KD-00052	700500.0
107	KD-00120	273400.0
108	KD-00055	280000.0
109	KD-00089	1276800.0
110	KD-00042	1127000.0
111	KD-00112	1353300.0
112	KD-00098	1358600.0
113	KD-00033	1149300.0
114	KD-00013	1138400.0
115	KD-00138	625600.0
116	KD-00094	651600.0
117	KD-00054	835100.0
118	KD-00100	395800.0
119	KD-00121	1232900.0
120	KD-00061	350300.0
121	KD-00031	857226.5
122	KD-00040	1276600.0
123	KD-00068	349200.0
124	KD-00131	612100.0
125	KD-00097	1144100.0
126	KD-00004	451500.0

```
127       KD-00071             1447700.0
128       KD-00093              362100.0
129       KD-00082              811500.0
130       KD-00150             1217300.0
131       KD-00065              613100.0
132       KD-00067              314100.0
133       KD-00011             1497900.0
134       KD-00091              974300.0
135       KD-00147             1122800.0
136       KD-00081              588300.0
137       KD-00109             1160300.0
138       KD-00072             1435600.0
139       KD-00014              474600.0
140       KD-00078              736100.0
141       KD-00095              779900.0
142       KD-00022             1527400.0
143       KD-00017              525300.0
144       KD-00037              857226.5
145       KD-00108              851600.0
146       KD-00015              893600.0
147       KD-00083             1412900.0
148       KD-00060              972700.0
149       KD-00007              805900.0
150       KD-00077             1167800.0
151       KD-00085              999000.0
152       KD-00142             1378500.0
153       KD-00192             1491900.0
154       KD-00298              725600.0
155       KD-00492              904900.0

> #Membaca file staging.enrichment.kode_pos.xlsx dan menyimpannnya dalam variable sta
ging.enrichment.kode_pos
> staging.enrichment.kode_pos <- read.xlsx("staging.enrichment.kode_pos.xlsx")

> #Ambil field, kode_pelanggan, dan kode_pos_enrich saja
> staging.enrichment.kode_pos <- staging.enrichment.kode_pos[,c("kode_pelanggan", "ko
de_pos_enrich", "grouping")]

> staging.enrichment.kode_pos
    kode_pelanggan kode_pos_enrich grouping
1         KD-00001          876511        1
2         KD-00012          876511        1
3         KD-00045          876511        1
4         KD-00778          876511        1
5         KD-00002          712983        2
6         KD-00075          712983        2
7         KD-00003          764550        3
8         KD-00043          764550        3
9         KD-00004          967220        4
10        KD-00071          967220        4
11        KD-00093          967220        4
12        KD-00005          476511        5
13        KD-00101          476511        5
14        KD-00006          487851        6
15        KD-00007          986455        7
```

16	KD-00060	986455	7
17	KD-00008	813444	8
18	KD-00121	896112	8
19	KD-00009	896555	9
20	KD-00010	987453	10
21	KD-00028	987453	10
22	KD-00125	987453	10
23	KD-00011	967223	11
24	KD-00091	967223	11
25	KD-00013	666122	12
26	KD-00033	666122	12
27	KD-00014	817321	13
28	KD-00072	817321	13
29	KD-00015	876511	14
30	KD-00083	876511	14
31	KD-00016	896550	15
32	KD-00057	896550	15
33	KD-00017	768034	16
34	KD-00037	768034	16
35	KD-00018	896555	17
36	KD-00058	896555	17
37	KD-00019	866162	18
38	KD-00048	866162	18
39	KD-00070	866162	18
40	KD-00020	476533	19
41	KD-00080	476533	19
42	KD-00021	511432	20
43	KD-00074	511432	20
44	KD-00022	768031	21
45	KD-00095	768031	21
46	KD-00023	768091	22
47	KD-00063	768091	22
48	KD-00148	768091	22
49	KD-00024	811613	23
50	KD-00025	813442	24
51	KD-00086	813442	24
52	KD-00026	896555	25
53	KD-00027	877521	26
54	KD-00029	896566	27
55	KD-00030	349922	28
56	KD-00031	896114	29
57	KD-00068	567151	29
58	KD-00032	567130	30
59	KD-00053	567130	30
60	KD-00133	567130	30
61	KD-00034	877615	31
62	KD-00103	877613	31
63	KD-00143	877614	31
64	KD-00035	712984	32
65	KD-00076	712984	32
66	KD-00113	712984	32
67	KD-00298	712984	32
68	KD-00036	876552	33
69	KD-00126	876552	33
70	KD-00038	987452	34

71	KD-00117	987452	34
72	KD-00039	764449	35
73	KD-00087	764449	35
74	KD-00040	896115	36
75	KD-00041	896549	37
76	KD-00066	896549	37
77	KD-00127	896549	37
78	KD-00140	896549	37
79	KD-00042	696193	38
80	KD-00044	321321	39
81	KD-00492	321321	39
82	KD-00046	877521	40
83	KD-00137	877521	40
84	KD-00049	321321	41
85	KD-00141	321321	41
86	KD-00050	321321	42
87	KD-00110	321321	42
88	KD-00051	696193	43
89	KD-00052	567120	44
90	KD-00054	896549	45
91	KD-00094	896549	45
92	KD-00138	896549	45
93	KD-00055	696193	46
94	KD-00056	876551	47
95	KD-00111	876551	47
96	KD-00059	986455	48
97	KD-00122	986455	48
98	KD-00061	896113	49
99	KD-00062	487451	50
100	KD-00064	987451	51
101	KD-00065	967222	52
102	KD-00067	967223	53
103	KD-00069	349981	54
104	KD-00114	349981	54
105	KD-00073	876512	55
106	KD-00077	987601	56
107	KD-00085	987601	56
108	KD-00078	817324	57
109	KD-00079	986456	58
110	KD-00081	967229	59
111	KD-00109	967229	59
112	KD-00082	967221	60
113	KD-00097	567120	60
114	KD-00150	967221	60
115	KD-00084	811613	61
116	KD-00104	811613	61
117	KD-00088	633429	62
118	KD-00132	633429	62
119	KD-00089	696193	63
120	KD-00090	511431	64
121	KD-00092	696193	65
122	KD-00096	633431	66
123	KD-00119	633430	66
124	KD-00098	696193	67
125	KD-00099	712984	68

```
126       KD-00192          712984       68
127       KD-00100          896549       69
128       KD-00102          666122       70
129       KD-00105          321321       71
130       KD-00106          896555       72
131       KD-00136          896555       72
132       KD-00107          893422       73
133       KD-00108          768035       74
134       KD-00112          696193       75
135       KD-00115          986455       76
136       KD-00128          986455       76
137       KD-00116          986455       77
138       KD-00144          986455       77
139       KD-00118          696193       78
140       KD-00120          567120       79
141       KD-00123          813442       80
142       KD-00124          321321       81
143       KD-00129          986454       82
144       KD-00130          876614       83
145       KD-00131          567151       84
146       KD-00134          986456       85
147       KD-00135          876612       86
148       KD-00139          511431       87
149       KD-00142          986455       88
150       KD-00145          896555       89
151       KD-00146          666123       90
152       KD-00147          967224       91
153       KD-00149          764450       92
154        KD-0047          764450       92
155       KD-00151          876612       93

> #Membaca file staging.final.xlsx dan menyimpannnya dalam variable staging.final
> staging.final <- read.xlsx("staging.final.xlsx")

> #Menggabungkan variable staging.enrichment.mean dengan staging.enrichment.kode_pos
melalui kolom kode_pelanggan
> hasil.final <- merge(x=staging.enrichment.mean, y=staging.enrichment.kode_pos, by.x
= "kode_pelanggan", by.y = "kode_pelanggan", all = TRUE)

> #Menggabungkan variable staging.final dengan hasil.final melalui kolom kode_pelangg
an
> hasil.final <- merge(x=staging.final, y=hasil.final, by.x = "kode_pelanggan", by.y
= "kode_pelanggan", all = TRUE)

> hasil.final
    kode_pelanggan                              nama
1         KD-00001                     Agus Cahyonos
2         KD-00002                     Khairul Nissa
3         KD-00003                    Slamet Wiyanto
4         KD-00004           DRS. Maria Simangunsong
5         KD-00005             Prihatin Setyonugroho
6         KD-00006                 DR. Candra Wijaya
7         KD-00007               Indra Kurniawan, ST
8         KD-00008                     Willy Sanjaya
9         KD-00009                   Antonius Winarta
```

```
10        KD-00010                   Sri Wahyuni, Ir
11        KD-00011                   Rosalina Kurnia
12        KD-00012                     Cahyono, Agus
13        KD-00013                    Danang Santosa
14        KD-00014   Elisabeth Suryadinata, SKOM, ST
15        KD-00015                   Mario Setiawan
16        KD-00016                         Indra K.
17        KD-00017              Irfan Putra Wijaya
18        KD-00018                Sudirman Kartono
19        KD-00019                   Maria Yuniarti
20        KD-00020                   Hendri Winarto
21        KD-00021             Paulus Angkasa Putra
22        KD-00022               Mbak Dian Sukowati
23        KD-00023                Ir. Yahya Permata
24        KD-00024                  Solihin Chaerul
25        KD-00025             DRG. Euis Rosidawati
26        KD-00026                    Anton Winarta
27        KD-00027               Djoko Wardoyo, Drs.
28        KD-00028                    Aman Pakpahan
29        KD-00029                       Sri Rahayu
30        KD-00030                            Hendi
31        KD-00031           Risman Suparyo Permata
32        KD-00032               Eva Novianti, S.H.
33        KD-00033                    Citra Permana
34        KD-00034              Rita Meutia Latief
35        KD-00035                   Sidharta Paul
36        KD-00036                  Irwan Setianto
37        KD-00037                    Cynthia Agus
38        KD-00038                     Putri Utomo
39        KD-00039 Joko Wiryanto Abadi Pelanggan OKE
40        KD-00040                       Sri Utami
41        KD-00041                   Poernomo Hadi
42        KD-00042                   Ahmad Junaidi
43        KD-00043                   Suharno Jamar
44        KD-00044              dr. Yati Octavianus
45        KD-00045                  Usman Pandajaya
46        KD-00046                  Ir. Ita Nugraha
47        KD-00048                        Lilis Ong
48        KD-00049                  Dianto Laksmana
49        KD-00050                Intan Tri Wahyuni
50        KD-00051                     Abdul Kadir
51        KD-00052                         Iriawan
52        KD-00053                       Heidi Goh
53        KD-00054                  Yudistira Utomo
54        KD-00055                   Maria Wiryawan
55        KD-00056                Jokolono Sukarman
56        KD-00057                    Sumardi Utomo
57        KD-00058                Fineli Rahmadianto
58        KD-00059           Prof Dr. Sadli Masikun
59        KD-00060                 Sulaiman Baskara
60        KD-00061           Tjipto Kesuma Wardhaya
61        KD-00062                  Zulkifli Kirana
62        KD-00063               Widianto Nuryajaya
63        KD-00064                      Fauzan Amir
64        KD-00065              Civara Intan Wahyudi
```

65	KD-00066	Purnomo Hadi
66	KD-00067	Niken Sri Utami
67	KD-00068	Miliana
68	KD-00069	Syarifuddin Mahmud
69	KD-00070	I Made Mulyana
70	KD-00071	Suparta
71	KD-00072	Harry Widijanto
72	KD-00073	Takashi Yudistira Arief
73	KD-00074	Taka Teguh
74	KD-00075	Kaka Ari Lima
75	KD-00076	Safira Hana Sahrani
76	KD-00077	Frenki Pranata
77	KD-00078	Gugun Gunawan Wijaya
78	KD-00079	Meiti Kuswara
79	KD-00080	Cristian Pakpahan Winarno
80	KD-00081	Andy Gunawan
81	KD-00082	Darmadi
82	KD-00083	Setiawan Mario
83	KD-00084	Surya
84	KD-00085	Frenki P.
85	KD-00086	Sisilia Lai
86	KD-00087	Budi Setiawan
87	KD-00088	Ayu
88	KD-00089	Acmad Junaidi
89	KD-00090	Andreas Sutanto
90	KD-00091	Indri Nourina Marthia
91	KD-00092	M Hasbi
92	KD-00093	Partono
93	KD-00094	Sri Utami
94	KD-00095	Sri Resti Agung
95	KD-00096	Rahmat Chandra
96	KD-00097	Frenkie Pranata
97	KD-00098	B. Sulaiman
98	KD-00099	Sanjaya Priyantoro
99	KD-00100	Rahayu Sri Asih
100	KD-00101	Fera Kurniawan
101	KD-00102	Leny Sarmini
102	KD-00103	Yonathan Bagus
103	KD-00104	Iqbal Setiawan
104	KD-00105	Urip Chandra Effendi
105	KD-00106	Budi Yahya
106	KD-00107	Rachmat Chandra
107	KD-00108	Jujur Suwito
108	KD-00109	Purwadianto Hadi
109	KD-00110	Sumartono Salim
110	KD-00111	Tommy Sinaga
111	KD-00112	Ari Masbun
112	KD-00113	Edi Alexander
113	KD-00114	Tri Iskandar
114	KD-00115	Teddy Rahmanto
115	KD-00116	Risma Sihombing
116	KD-00117	Florensia Novianti
117	KD-00118	Abdul Kadir
118	KD-00119	Tri Sulistianti
119	KD-00120	Dewi Sryani

```
120       KD-00121                    Diana Sumirah
121       KD-00122                 Christine Angkasa
122       KD-00123                   Rakhmat Chandra
123       KD-00124                         Yakob Tan
124       KD-00125                       Tedi Halim
125       KD-00126                     Agus Cahyono
126       KD-00127                    Herdi Rivanto
127       KD-00128                    Tedi Rahmanto
128       KD-00129                     Edward Salim
129       KD-00130                     Jujur Suwito
130       KD-00131                     Dewi Pratiwi
131       KD-00132                  Rachmat Chandra
132       KD-00133                    Unang Handoko
133       KD-00134                       Budi Yahya
134       KD-00135                       Tiah Feris
135       KD-00136                     Joko Wibawa
136       KD-00137                    Maria Sirait
137       KD-00138                    Teddja Yanto
138       KD-00139                      Agnes Rita
139       KD-00140                   Leonardo Tedja
140       KD-00141                     Edi Sumantri
141       KD-00142                    Tedi Rahmanto
142       KD-00143                      Hari Wibowo
143       KD-00144                  Risma Sihombing
144       KD-00145                      Lilis Kasim
145       KD-00146                    Roger Sirait
146       KD-00147                    Budi Setiawan
147       KD-00148                         Kuswanto
148       KD-00149                  Chandra Rachmat
149       KD-00150                      Maria Utami
150       KD-00151                       Ferry Thia
151       KD-00192                Sanjaya Priyantoro
152       KD-00298               Safira Hana Sahrani
153        KD-0047                     Puspita Citra
154       KD-00492               dr. Yati Octavianus
155       KD-00778                   Cahyono Agus H.
                                              alamat        no_telepon
1          Jalan Pulo Bambu No. 15, Kota Tenggara Lama    08298911112222
2                      Taman Vivo Indah, Blok AA No. 7 +6287132221371404
3                             Meta Residences, No. 32C +6285725955303368
4                        Gang Bulan Desember III, No. 9 +6283376770990635
5            Jalan Tegal Sari Indah, No. D87 -- Kota H +6286843623971825
6                           Perum Pluto, Blok C No. 1 +6284063423953696
7              Apartemen Kecapi Indah, Lt. 16 No. 1610 +6283840529196797
8                         Kali Mars Cluster, No. 24C +6285312577710538
9                Jalan Kebon Jahe, No. F16 - Kota E +6282722234294686
10                  Perum Venus, Gg. Harimau No. 1A +6284079659289143
11              Cluster Ikan Mas, Taman Intan No. 2 +6288339032314103
12             Pulo Bambu No. 15, Kota Tenggara Lama   +62829891111222
13                  Jalan Hang Tuah, No. 11, Kota DM +6282672925000608
14         Boulevard Raya Residences, Blok AA2 No. 88  +6285455084014504
15             Jalan Puri Arteri Raya, No. 88 - Kota T +6282989111122220
16                        Jalan Pahlawan, No. 69CCD  +6289222405928430
17         Asrama Pelajar No. 22 A - Pondok Bima Sakti +6289984358708389
18                      Jalan Bintang Supernova, No. 78 +6282283957103749
```

19	Jalan Wisma Tenteram Saja, No. A22	+6289317147992822
20	Jalan Manggis II, Gang Buntu No. 1	+6287384329533477
21	Puspa Loka, No. 98B, Kota Y	+6285991672131933
22	Asrama Perawat IV, No. 1 - Kota D	+6285796817992325
23	Jalan Macan Buntung, No. 1F	+6287660464098623
24	Perum Maju Permai Persada Indah, Gang Kenari No. 3	+6281718632538241
25	Kampoeng Harimau, No. 81 - Kota K	+6286035230854391
26	Jalan Kebon Jahe, Kota EntahDimana	+6284204043307629
27	Vila Bukit Sagitarius, Blok A1 No. 1	+6284871003581659
28	Perum Venus, Gang. Kelinci No. 12	+6289311313046417
29	Jalan Kp. Kijang, Blok A1 - No. 2F	+6283177123456315
30	Pondok Bima Sakti, Jalan Asrama Pelajar No. 11FF	+6282261101749552
31	Gang Tupai, No. 7 - Desa CL	+6287382247200814
32	Vila Sempilan, No. 67 - Kota B	+6285419651438216
33	Jalan Hang Tuah, No. 31, Kota DM	+6286734992308497
34	Perum Kali Meksiko, No. 8C	+6284588563149814
35	Taman Bunga Langit, Jalan Timur No. 1	+6286725681847845
36	Vila Gunung Seribu, Blok O1 - No. 1	+6285842418573681
37	Asrama Pelajar No. 11 B - Pondok Bima Sakti	+6283155468652762
38	Perumahan Bina Andromeda, Jalan Teri No. 4	+6286621940809359
39	Perum Indah Supernova II, No. 9	+6289122766908102
40	Gang Samun Saja No. 132, Kode Pos A99222	+6287263432705516
41	Jalan Pulau Sentosa No. 133	08763322558899
42	Jalan Raya Hang Lekir, Kota Z, No. 62	+6284399241602502
43	Meta Residences, No. 1A	+6285158186394886
44	Kompleks Pelaut Tangguh, No. 5A	+6285879131063825
45	Pulo Bambu No. 57, Kota Tenggara Lama	+6282607473168157
46	Vila Bukit Sagitarius, Gang Kelapa No. 6	+6288267903981205
47	Jalan Wisma Tenteram Saja, No. A31	+6285317681095918
48	Kompleks Permai Angkasa, Blok M No. 10	+6284311691840121
49	Kompleks Selatan-Selatan, No. 121	+6283594524411404
50	Jalan Binjai 200, Kota L	+6283835679381969
51	Jalan Ring Road Neolitik, No. 1 RT 5	+6282695676827512
52	Vila Sempilan, No. 11 - Kota B	+6282189517223455
53	Jalan Gula Pahit, No. 001	+6288743246116630
54	Jalan Raya Jupiter Titan, No. 55	+6288385590443770
55	Vila Permata Intan Berkilau, Blok C5-7	+6289278629437370
56	Jalan Pahlawan, No. 69FFF	+6286996345317721
57	Jalan Bintang Supernova, No. 78	+6289503422652894
58	Perumahan Sektor Bougenville, Jalan Karet No. 7P	+6283468728620812
59	Apartemen Kecapi Indah, Lt. 18 No. 1801	+6286106166597558
60	Griya Asri Mawar Harum, Blok G No. 1	+6283534357190274
61	Perum Sektor 50, Gang Permai No. 5	+6286916223612856
62	Jalan Macan Buntung, No. 4F	+6285463027900499
63	Perumahan Catalina, Jalan Kereta Api No. 77	+6285526151431004
64	Corina Residences Apartment, No. 0612	+6287500842511771
65	Jalan Pulau Sentosa No. 133	-
66	Condominium Pesona Indah, No. 0708	+6286546368604671
67	Gang Piranha, No. 3 - Desa BT	+6284941004806026
68	Perum Titan, Jalan Trobos No. 8	+6281298730359784
69	Jalan Wisma Tenteram Saja No. B-01	+6281950071656111
70	Gang Bulan Desember III, No. 155	+6285361733615048
71	Boulevard Raya Residences, Blok AB2 No. 102	+6288942438259785
72	Jalan Puri Indah Menawan, No. 818 - Kota T	+6281859313870200
73	Puspa Loka, No. 98F, Kota Y	+6281902807450191

```
74                      Taman Vivo Indah, Blok AA No. 7 +6283309536733507
75                Taman Bunga Langit, Jalan Utara No. 3 +6286815308308264
76                        Jalan Sutomo Baru 21 - Kota M +6283957775331152
77                        Blok C 2/4, Bukit Vivo Indah +6283670227924527
78           Perumahan Duku Satu, Gang Merpati - No. 41 +6284927709580269
79                  Jalan Manggis II - Gang Buntu No. 4 +6284032125604618
80                        Bukit Vivo Indah, Blok C 2/4 +6288590906353243
81                         Gang Arwana, No. 6 - Kota S +6284338493742386
82              Jalan Puri Arteri Raya, No. 88 - Kota T +6282989111122220
83           Perum Maju Permai P.I., Gang Kesturi No. 5 +6286837329291803
84                  Jalan Sutomo Baru No. 21 - Kota  M +6289781665737911
85                     Kampung Harimau, No. 88, Kota K +6281334304509664
86                        Perum Indah Supernova, No. 1 +6285318844151067
87                    Rusun Kerinci Indah, Lt. 5 No. 6 +6283203183708137
88                Jalan Raya Hang Lekir, No. 62 - Kota Z +6281550391417945
89                        Ruko Almond Manis, Blok C7/8 +6287066745737382
90                  Cluster Ikan Mas, Taman Baru No. 96 +6288718681168878
91                       Jalan Bukit Tol Km. 3, No. 971 +6284298240961859
92                    Gang Bulan Desember III, No. 145 +6287029784792141
93                             Jalan Gula Pahit, No. 015 +6284941125391866
94                    Asrama Perawat IV, No. 2 - Kota D +6285736296760607
95                    Rumah Susun Eunos, Lantai 2 No. 2 +6286210781145764
96                       Gang Kelinci, No. 666 - Kota B +6282055715061873
97    Jalan Pesisir No. 5, Kampoeng Maju Surya Gemilang +6283382626807712
98           Taman Bunga Langit, Jalan Barat Laut No. 6 +6281729600654645
99           Jalan Asia No. 55, Kompleks Pelajar Kota C +6282208807303229
100           Jalan Tegal Sari Indah, No. D77 -- Kota H +6285375019511143
101              Jalan Kangguru No. 92, RT 005 - kota R +6281941958971086
102                           Perum Kali Meksiko, No. D22 +6283481690089399
103          Perum Maju Permai P.I., Gang Kesturi No. 5 +6286401899308998
104        Kompleks Akademi Perawat, Gang Farmasi No. 3 +6288507258756263
105              Jalan Kemenangan Besar, Blok C8 No. 22 +6283460823430150
106                    Kampung Kijang, Blok D3 - No. 12 +6282792175097533
107                Apartement Clifften, Lantai 12 No. 3 +6284037884325249
108                       Bukit Vivo Indah, Blok C 2/4 +6286240577462157
109                    Kompleks Selatan-Selatan, No. 111 +6288942588082822
110              Vila Permata Intan Berkilau, Blok A1/2 +6284384621977881
111                  Jalan Raya Andromeda, Blok D No. 3 +6285734298900666
112            Taman Bunga Langit, Jalan Selatan No. 12 +6281413705348345
113                   Perum Titan, Jalan Kelinci No. 12 +6284122970381517
114      Apartemen Bukit Merah Annex Plaza, Lt 3 No. A1    08765439876543
115              Apartemen Lucky Beruntung, Lt. 5 No. 4 +6287642929298977
116       Perumahan Bina Andromeda, Jalan Salmon No. 22 +6283166638654813
117             Jalan Semantik Semut Berjalan, No. 3333 +6281693345459608
118                 Rumah Susun Gelora, Lantai 1 No. 12 +6289176501199576
119                      Jalan Ring Road Konstan, No. 5 +6285239934324639
120                         Indah Mars Cluster, No. 22F +6288508083942658
121    Perumahan Sektor Bougenville, Jalan Sawit No. 8A +6286663398617904
122                               Kampung Harimau, No. 3 +6286051245623557
123            Kompleks Nelayan Permai, Blok DD - 98/99 +6284366427534780
124                   Perum Venus, Gang. Harimau No. 4A +6286353637542265
125                 Vila Gunung Seribu, Blok F4 - No. 8 +6289522699290044
126                         Jalan Pulau Sentosa No. 133 +6284991627085550
127     Apartemen Bukit Merah, Annex Plaza, Lt 3 No. A1     0898198765432
128 Perumahan Sektor Telekomunikasi, Jalan Afrika No. 3 +6289323214692782
```

```
129              Jalan Raya Griya Barbarosa, Blok AF 789 +6282833816760984
130                       Gang Piranha, No. 13 - Desa BT +6284939933374036
131                     Rusun Kerinci Indah, Lt. 6 No. 1 +6282352225142570
132                        Vila Sempilan, No. 1 - Kota B +6282952955586979
133             Perumahan Duku Lima, Gang Perkutut No. 1 +6284094392278758
134               Kota T,  Jalan Taman Kencana No. 11112 +6283674655321990
135         Jalan Kemenangan Besar, Blok C8 No. 22 RT 02 +6288841308560422
136             Vila Bukit Sagitarius, Gang. Sawit No. 3 +6288389541238485
137                            Jalan Gula Pahit, No. 081 +6286357357965169
138                        Ruko Azalea, No. 3 RT 001/002 +6285986817540683
139                         Jalan Pulau Sentosa No. 1335 +6289699357035892
140               Kompleks Permai Angkasa, Blok J No. 09 +6286730629494828
141           Apartemen Bukit Baru, Dahlia Tower, No. A3 +6289859935888974
142                           Perum Kali Meksiko, No. 8F +6281672571203724
143               Apartemen Lucky Beruntung, Lt. 3 No. 4 +6287642929298977
144                Jalan Kampung Kijang, Blok C5 - No. 9 +6281980423349356
145               Jalan G. Asri Mawar Harum Blok G No. 9 +6288888862370254
146               Cluster Griya Bima Sakti, Blok A No. 1 +6282891052016637
147                 Jalan Macan Buntung, No. 1F - Kota D +6289756523291187
148                  Perum Bimasakti Raya, Blok A No. 10 +6289337617505007
149                            Gang Arwana No. 12, Kota S +6287188198226353
150                Jalan Taman Kencana No. 11112, Kota T +6287896807815060
151           Taman Bunga Langit, Jalan Barat Laut No. 6 +6281729600654645
152                Taman Bunga Langit, Jalan Utara No. 3 +6286815308308264
153                  Perum Bimasakti Raya, Blok A No. 10 +6282793268821143
154                       Kompleks Pelaut Tangguh, No. 5A +6285879131063825
155        Jalan Pulau Bambu No. 15 - Kota Tenggara Lama   +62829891112222
    anomali_no_telepon kode_pos tanggal_lahir       umur umur_valid
1                 TRUE   876511    08-02-1967  51.249315       TRUE
2                FALSE   712983    23-10-1991  26.528767       TRUE
3                FALSE   764550    23-11-1962  55.463014       TRUE
4                FALSE   967220    17-02-2097 -78.865753       TRUE
5                FALSE   476511    19-08-1986  31.709589       TRUE
6                FALSE   487851    05-09-1990  27.660274       TRUE
7                FALSE   986455    23-10-1979  38.536986       TRUE
8                FALSE   813444    22-07-1973  44.794521       TRUE
9                FALSE   896555          <NA>         NA         NA
10               FALSE   987453    23-10-1991  26.528767       TRUE
11               FALSE   967223    12-01-1969  49.320548       TRUE
12                TRUE   876511    08-02-1967  51.249315       TRUE
13               FALSE   666122    22-04-1933  85.071233      FALSE
14               FALSE        -    23-10-1995  22.526027       TRUE
15               FALSE   876511    09-08-1972  45.745205       TRUE
16               FALSE   896550    28-05-1969  48.947945       TRUE
17               FALSE   768034    23-11-1962  55.463014       TRUE
18               FALSE   896555    19-03-1950  68.153425       TRUE
19               FALSE        -    23-11-1962  55.463014       TRUE
20               FALSE   476533    13-11-1962  55.490411       TRUE
21               FALSE   511432    14-03-1979  39.147945       TRUE
22               FALSE   768031    25-07-1974  43.786301       TRUE
23               FALSE        -    12-01-1971  47.320548       TRUE
24               FALSE   811613    12-07-1977  40.819178       TRUE
25               FALSE   813442    19-03-1950  68.153425       TRUE
26               FALSE   896555    12-01-1969  49.320548       TRUE
27               FALSE   877521    23-11-1962  55.463014       TRUE
```

28	FALSE	987453	23-10-1991	26.528767	TRUE
29	FALSE	896566	13-11-1962	55.490411	TRUE
30	FALSE	349922	28-02-1969	49.191781	TRUE
31	FALSE	896114	27-02-1976	42.191781	TRUE
32	FALSE	567130	01-04-2028	-9.936986	TRUE
33	FALSE	666122	21-05-1981	36.958904	TRUE
34	FALSE	877615	12-01-1972	46.320548	TRUE
35	FALSE	712984	24-01-1952	66.301370	TRUE
36	FALSE	876552	20-02-1970	48.213699	TRUE
37	FALSE	768034	03-10-1988	29.583562	TRUE
38	FALSE	987452	12-07-1977	40.819178	TRUE
39	FALSE	764449	05-09-1990	27.660274	TRUE
40	FALSE	896115	12-01-1971	47.320548	TRUE
41	TRUE	896549	19-03-1950	68.153425	TRUE
42	FALSE	696193	17-09-1982	35.632877	TRUE
43	FALSE	764550	25-07-1974	43.786301	TRUE
44	FALSE	321321	21-05-1980	37.958904	TRUE
45	FALSE	876511	07-07-1977	40.832877	TRUE
46	FALSE	877521	14-03-1879	139.213699	FALSE
47	FALSE	866162	<NA>	NA	NA
48	FALSE	321321	28-02-1969	49.191781	TRUE
49	FALSE	321321	05-09-1990	27.660274	TRUE
50	FALSE	696193	17-09-1982	35.632877	TRUE
51	FALSE	567120	15-02-1997	21.208219	TRUE
52	FALSE	567130	19-08-1986	31.709589	TRUE
53	FALSE	896549	01-01-1982	36.342466	TRUE
54	FALSE	696193	29-02-1976	42.186301	TRUE
55	FALSE	876551	13-10-1979	38.564384	TRUE
56	FALSE	896550	<NA>	NA	NA
57	FALSE	896555	23-12-1968	49.375342	TRUE
58	FALSE	-	05-07-1987	30.832877	TRUE
59	FALSE	986455	24-09-1990	27.608219	TRUE
60	FALSE	896113	30-11-1954	63.449315	TRUE
61	FALSE	487451	28-02-1969	49.191781	TRUE
62	FALSE	768091	<NA>	NA	NA
63	FALSE	987451	14-11-1987	30.471233	TRUE
64	FALSE	967222	14-03-1879	139.213699	FALSE
65	TRUE	896549	19-03-1905	113.183562	FALSE
66	FALSE	967223	15-02-1997	21.208219	TRUE
67	FALSE	567151	05-06-1979	38.920548	TRUE
68	FALSE	349981	24-06-1992	25.857534	TRUE
69	FALSE	696193	10-10-1982	35.569863	TRUE
70	FALSE	967220	29-12-1963	54.364384	TRUE
71	FALSE	817321	20-11-1987	30.454795	TRUE
72	FALSE	876512	26-01-1979	39.276712	TRUE
73	FALSE	511432	01-12-1964	53.438356	TRUE
74	FALSE	712983	28-02-1969	49.191781	TRUE
75	FALSE	712984	20-02-1970	48.213699	TRUE
76	FALSE	987601	07-07-1968	49.838356	TRUE
77	FALSE	817324	26-11-1983	34.441096	TRUE
78	FALSE	986456	05-12-1979	38.419178	TRUE
79	FALSE	476533	13-11-1962	55.490411	TRUE
80	FALSE	967229	20-10-1987	30.539726	TRUE
81	FALSE	967221	26-11-1983	34.441096	TRUE
82	FALSE	876511	19-03-1950	68.153425	TRUE

83	FALSE	811613	25-02-1987	31.189041	TRUE
84	FALSE	987601	07-07-1968	49.838356	TRUE
85	FALSE	813442	13-11-1962	55.490411	TRUE
86	FALSE	764449	25-06-1987	30.860274	TRUE
87	FALSE	633429	01-01-2001	17.328767	TRUE
88	FALSE	696193	30-11-1967	50.441096	TRUE
89	FALSE	511431	17-07-1987	30.800000	TRUE
90	FALSE	967223	01-01-2001	17.328767	TRUE
91	FALSE	696193	22-11-1979	38.454795	TRUE
92	FALSE	967220	30-11-1954	63.449315	TRUE
93	FALSE	896549	16-06-1975	42.893151	TRUE
94	FALSE	768031	01-12-1964	53.438356	TRUE
95	FALSE	633431	26-08-1983	34.693151	TRUE
96	FALSE	567120	23-06-1968	49.876712	TRUE
97	FALSE	696193	07-07-1968	49.838356	TRUE
98	FALSE	712984	26-08-1983	34.693151	TRUE
99	FALSE	896549	20-10-1987	30.539726	TRUE
100	FALSE	476511	19-08-1950	67.734247	TRUE
101	FALSE	666122	01-12-1964	53.438356	TRUE
102	FALSE	877613	30-11-1954	63.449315	TRUE
103	FALSE	811613	17-08-1986	31.715068	TRUE
104	FALSE	321321	18-08-1988	29.709589	TRUE
105	FALSE	896555	30-11-1954	63.449315	TRUE
106	FALSE	893422	01-01-2001	17.328767	TRUE
107	FALSE	768035	28-02-1969	49.191781	TRUE
108	FALSE	967229	17-08-1986	31.715068	TRUE
109	FALSE	321321	12-12-1950	67.419178	TRUE
110	FALSE	876551	24-03-1976	42.120548	TRUE
111	FALSE	696193	07-07-1968	49.838356	TRUE
112	FALSE	712984	22-02-2000	18.189041	TRUE
113	FALSE	349981	28-02-1969	49.191781	TRUE
114	TRUE	986455	08-03-1955	63.180822	TRUE
115	FALSE	986455	20-12-1977	40.378082	TRUE
116	FALSE	987452	<NA>	NA	NA
117	FALSE	696193	04-07-1987	30.835616	TRUE
118	FALSE	633430	01-01-2001	17.328767	TRUE
119	FALSE	567120	29-11-1967	50.443836	TRUE
120	FALSE	896112	14-11-1987	30.471233	TRUE
121	FALSE	986455	20-06-2001	16.863014	TRUE
122	FALSE	813442	13-11-1963	54.490411	TRUE
123	FALSE	321321	14-01-1988	30.304110	TRUE
124	FALSE	-	31-01-2001	17.246575	TRUE
125	FALSE	876552	14-11-1987	30.471233	TRUE
126	FALSE	896549	24-02-1978	40.197260	TRUE
127	TRUE	986455	08-03-1955	63.180822	TRUE
128	FALSE	986454	23-04-1978	40.038356	TRUE
129	FALSE	876614	24-09-1990	27.608219	TRUE
130	FALSE	567151	21-01-2001	17.273973	TRUE
131	FALSE	633429	24-01-1987	31.276712	TRUE
132	FALSE	567130	11-07-1981	36.819178	TRUE
133	FALSE	986456	14-07-1977	40.813699	TRUE
134	FALSE	876612	08-03-1955	63.180822	TRUE
135	FALSE	896555	31-01-2001	17.246575	TRUE
136	FALSE	877521	12-01-1968	50.323288	TRUE
137	FALSE	896549	12-12-1987	30.394521	TRUE

```
138              FALSE   511431    21-05-1980  37.958904        TRUE
139              FALSE   896549    22-12-1993  24.361644        TRUE
140              FALSE   321321    30-11-1954  63.449315        TRUE
141              FALSE   986455    14-12-2003  14.378082        TRUE
142              FALSE   877614    17-07-1987  30.800000        TRUE
143              FALSE   986455    20-12-1977  40.378082        TRUE
144              FALSE   896555    08-08-2008   9.723288        TRUE
145              FALSE   666123    31-01-2001  17.246575        TRUE
146              FALSE   967224    29-12-1967  50.361644        TRUE
147              FALSE   768091    17-08-1986  31.715068        TRUE
148              FALSE   764450    12-01-1968  50.323288        TRUE
149              FALSE   967221    20-06-2001  16.863014        TRUE
150              FALSE   876612    29-03-1967  51.115068        TRUE
151              FALSE   712984    26-08-1983  34.693151        TRUE
152              FALSE   712984    20-02-1970  48.213699        TRUE
153              FALSE   764450    19-03-1950  68.153425        TRUE
154              FALSE   321321    21-05-1980  37.958904        TRUE
155               TRUE   876511    08-02-1967  51.249315        TRUE
    nilai_belanja_setahun kode_pos_enrich grouping
1               1082900.0          876511        1
2               1336200.0          712983        2
3                601700.0          764550        3
4                451500.0          967220        4
5               1488900.0          476511        5
6                257100.0          487851        6
7                805900.0          986455        7
8                879800.0          813444        8
9                272600.0          896555        9
10               389400.0          987453       10
11              1497900.0          967223       11
12               488400.0          876511        1
13              1138400.0          666122       12
14               474600.0          817321       13
15               893600.0          876511       14
16               925200.0          896550       15
17               525300.0          768034       16
18              1117000.0          896555       17
19               733500.0          866162       18
20               990900.0          476533       19
21               825500.0          511432       20
22              1527400.0          768031       21
23              1206000.0          768091       22
24              1471900.0          811613       23
25               543000.0          813442       24
26               459200.0          896555       25
27               536000.0          877521       26
28               361100.0          987453       10
29              1293600.0          896566       27
30               308800.0          349922       28
31               857226.5          896114       29
32              1275600.0          567130       30
33              1149300.0          666122       12
34              1190900.0          877615       31
35               398200.0          712984       32
36              1444400.0          876552       33
```

37	857226.5	768034	16
38	588300.0	987452	34
39	1086300.0	764449	35
40	1276600.0	896115	36
41	265900.0	896549	37
42	1127000.0	696193	38
43	237400.0	764550	3
44	904900.0	321321	39
45	437200.0	876511	1
46	541300.0	877521	40
47	494900.0	866162	18
48	1135500.0	321321	41
49	453800.0	321321	42
50	1168700.0	696193	43
51	700500.0	567120	44
52	317800.0	567130	30
53	835100.0	896549	45
54	280000.0	696193	46
55	1524700.0	876551	47
56	323100.0	896550	15
57	1216800.0	896555	17
58	1490800.0	986455	48
59	972700.0	986455	7
60	350300.0	896113	49
61	471300.0	487451	50
62	1100200.0	768091	22
63	661500.0	987451	51
64	613100.0	967222	52
65	253900.0	896549	37
66	314100.0	967223	53
67	349200.0	567151	29
68	1147500.0	349981	54
69	379700.0	866162	18
70	1447700.0	967220	4
71	1435600.0	817321	13
72	538800.0	876512	55
73	1452900.0	511432	20
74	1316500.0	712983	2
75	725600.0	712984	32
76	1167800.0	987601	56
77	736100.0	817324	57
78	489900.0	986456	58
79	489000.0	476533	19
80	588300.0	967229	59
81	811500.0	967221	60
82	1412900.0	876511	14
83	560900.0	811613	61
84	999000.0	987601	56
85	293800.0	813442	24
86	959200.0	764449	35
87	558000.0	633429	62
88	1276800.0	696193	63
89	530600.0	511431	64
90	974300.0	967223	11
91	756300.0	696193	65

92	362100.0	967220	4
93	651600.0	896549	45
94	779900.0	768031	21
95	1034600.0	633431	66
96	1144100.0	567120	60
97	1358600.0	696193	67
98	1491900.0	712984	68
99	395800.0	896549	69
100	770300.0	476511	5
101	514700.0	666122	70
102	1276600.0	877613	31
103	973700.0	811613	61
104	341900.0	321321	71
105	974100.0	896555	72
106	1526400.0	893422	73
107	851600.0	768035	74
108	1160300.0	967229	59
109	857226.5	321321	42
110	655400.0	876551	47
111	1353300.0	696193	75
112	311000.0	712984	32
113	613600.0	349981	54
114	998300.0	986455	76
115	679800.0	986455	77
116	854400.0	987452	34
117	1060600.0	696193	78
118	286200.0	633430	66
119	273400.0	567120	79
120	1232900.0	896112	8
121	732800.0	986455	48
122	326300.0	813442	80
123	1419000.0	321321	81
124	1339400.0	987453	10
125	350400.0	876552	33
126	997500.0	896549	37
127	657300.0	986455	76
128	992100.0	986454	82
129	1503700.0	876614	83
130	612100.0	567151	84
131	538400.0	633429	62
132	1537200.0	567130	30
133	554300.0	986456	85
134	399900.0	876612	86
135	1151000.0	896555	72
136	354600.0	877521	40
137	625600.0	896549	45
138	1128000.0	511431	87
139	912100.0	896549	37
140	1362200.0	321321	41
141	1378500.0	986455	88
142	1221900.0	877614	31
143	577600.0	986455	77
144	1350600.0	896555	89
145	1437400.0	666123	90
146	1122800.0	967224	91

```
147              857226.5          768091       22
148              413100.0          764450       92
149             1217300.0          967221       60
150             1437600.0          876612       93
151             1491900.0          712984       68
152              725600.0          712984       32
153              950200.0          764450       92
154              904900.0          321321       39
155              907500.0          876511        1

> #Menulis hasil ke file staging.final.xlsx
> write.xlsx(hasil.final, file="hasil.final.xlsx")
```

Kesimpulan

Data enrichment adalah proses pengisian data yang hilang atau menambah data baik dari sumber internal maupun eksternal dengan cara mengkorelasikan berdasarkan beberapa kolom tertentu sehingga analisa data lebih tajam.

Pendekatan pada bab ini menunjukkan bagaimana menggunakan teknik grouping duplikat juga memungkinkan kita melakukan enrichment.

Kita telah melakukan hal-hal berikut pada bab ini:

- Kolom **nilai_belanja_setahun** yang kosong diisi dengan perhitungan nilai rata-rata menggunakan function **mean** dan atau nilai tengah menggunakan function **median**.
- Data **kodepos** yang kosong diisi dengan langkah melakukan grouping duplikat dari kolom alamat.

Dan pada bagian terakhir kita juga mengkonsolidasikan seluruh data mulai dari hasil standarisasi, grouping duplikat dan pengisian missing value.

Hasil konsolidasi terlihat jauh lebih rapi dengan lebih banyak informasi seperti infomasi anomali no telepon, umur dan apakah umur valid atau ga, kemudian informasi duplikat, dan missing value yang telah disii untuk kolom kode pos dan nilai belanja setahun.

Hasil ini tentunya akan membantu para analis mengolah data dengan tingkat kepercayaan diri lebih tinggi.

Klik tombol Next untuk melanjutkan.

Bab Penutup

Selamat, Anda telah menyelesaikan course Data Wrangling with R – Part 2. Salah satu course terpenting di DQLab. Dikatakan demikian karena fokus bab ini yang ke arah profiling, cleansing, deduplikasi dan enrichment – adalah proses yang paling banyak dibutuhkan sebelum analis dapat melakukan tugasnya.

Oleh sebab itu DQLab memfokuskan bab ini dengan contoh yang sangat intensif sehingga Anda mendapatkan Skillset untuk melakukan 80 persen problem di data analytic ini.

Sejauh ini Anda telah melakukan hal berikut:

- **Contoh Dataset "Kotor":** Perkenalan contoh dataset master pelanggan yang sengaja dirancang dengan "kotor" atau mengandung isi yang tidak standar – menyerupai kondisi riil yang banyak ditemukan oleh tim DQLab selama terlibat dalam proyek-proyek pengolahan data di Indonesia.
- **Profiling**: Bagaimana mengidentifikasi pola dataset kita sebelum tau apa yang perlu dibersihkan atau dirapikan dengan menggunakan package **bpa**.
- **Membaca Database Relasional**: Bagaimana mengakses dari sistem database dengan memperkenalkan objek-objek database dan bahasa SQL (Structured Query Language) dengan menggunakan package RMySQL.
- **Data Cleansing – Standarisasi**: Bagaimana melakukan perapian isi berbagai tipe data dengan menggunakan fungsi-fungsi transformasi data dengan menggunakan gabungan berbagai function seperti gsub, trimws, colsplit dibantu dengan penerapan Regular Expression (regex).
- **Data Cleansing – Missing Value**: Bagaimana mengisi *missing value* pada kolom numerik dengan menggunakan metode pengisian nilai rata-rata (*mean*) dan nilai tengah (*median*).
- **Data Cleansing – Deduplikasi**: Menemukan data yang redundancy dan melakukan grouping terhadap data yang redundan.
- **Data Enrichment**: Bagaimana melengkapi data kosong dengan melengkapi data kosong dengan melakukan lookup dari internal data.

Klik tombol Next untuk melanjutkan.

What Next?

Data wrangling mencakup area yang sangat luas. Dari sisi sumber data yang perlu dikonsumsi seperti:

- file dbf – merupakan file sistem dbase, clipper, foxpro yang masih banyak digunakan di Indonesia.
- artikel web – crawling.
- email – teks dan gambar.
- gambar / foto yang berisi teks.
- dan lain-lain.

Selain itu, algoritma yang digunakan pada contoh R ini cocok untuk banyak kasus. Namun dengan makin kompleksnya data dari sisi jumlah dan kecepatan penambahan data, maka kemampuan penanganan ini perlu otomatisasi atau kalau tidak akan memakan waktu perusahaan atau organisasi sehingga tujuan akhir – berupa informasi yang diceritakan oleh data – tidak dapat tercapai.

Untuk mengatasinya, sistem dengan pengembangan produk dan template manajemen data yang baik sudah tidak bisa ditawar.

DQLab secara berkala akan posting update mengenai manajemen data ini di Facebook group kami di https://www.facebook.com/groups/DQLab. Kami sarankan untuk bergabung dan mengikuti update kami sehingga produktivitas dan efisiensi organisasi Anda meningkat.