

# Dokumentasi Converter Jotform

Divisi Area Marketing Nutrifood Indonesia

Ikang Fadhli  
Market Research Dept.  
Nutrifood Indonesia

## Pendahuluan

### Latar Belakang

Tim AM pada 2021 menggunakan *JotForm* sebagai *framework* pengumpulan data lapangan. Data yang terkumpul kelak akan digabung secara nasional dan dianalisa di kemudian hari. Untuk melakukan itu, data hasil *JotForm* perlu direkap dengan baik secara struktur data. *Converter* ini digunakan untuk merapikan data sesuai dengan struktur yang diinginkan.

Tim AM memiliki **kustomisasi** survey per departemen. Jadi diharapkan *converter* bisa mengatasi hal tersebut.

### Data yang Digunakan

Data hasil survey harian per departemen AM se-nasional dengan format **.xlsx**.

### Metode

*Data carpentry* dengan prinsip *tidy* menggunakan **dplyr**, **tidyr**, dan **reshape2**. *Converter* disajikan dengan *Shiny App*.

## Penjelasan Algoritma

### Memanggil *Libraries*

```
rm(list=ls())
```

```
# libraries  
library(readxl)  
library(dplyr)  
library(tidytext)  
library(janitor)  
library(tidyr)
```

### Mengambil Data

```
# ambil data  
data =  
  read_excel("tes.xlsx") %>%  
  janitor::clean_names()
```

## Menambah id Baris dan Memecah-Mecah

```
# tambahkan id
# lalu pecah-pecah
data =
  data %>%
  mutate(id = c(1:length(submission_date)),
         tanggal_transaksi = gsub("\\\\", "-", tanggal_transaksi),
         tanggal_transaksi = as.Date(tanggal_transaksi, "%m-%d-%Y"),
         tanggal_transaksi = lubridate::date(tanggal_transaksi),
         submission_date = lubridate::date(submission_date)) %>%
  separate(departemen_area_nama,
         into = c("departemen", "area", "nama"),
         sep = ";") %>%
  separate(jenis_channel_sub_channel_klasifikasi,
         into = c("jenis_channel", "sub_channel", "klasifikasi"),
         sep = ";") %>%
  separate(provinsi_kota_kab_kecamatan_kelurahan,
         into = c("provinsi", "kota_kab", "kecamatan", "kelurahan"),
         sep = ";") %>%
  separate(location_coordinate,
         into = c("longitude", "latitude", "csv"),
         sep = "\\r\\n") %>%
  mutate(departemen = trimws(departemen),
         area = trimws(area),
         nama = trimws(nama),
         jenis_channel = trimws(jenis_channel),
         sub_channel = trimws(sub_channel),
         provinsi = trimws(provinsi),
         kota_kab = trimws(kota_kab),
         kecamatan = trimws(kecamatan),
         kelurahan = trimws(kelurahan),
         longitude = gsub("Longitude: ", "", longitude),
         latitude = gsub("Latitude: ", "", latitude),
         longitude = as.numeric(longitude),
         latitude = as.numeric(latitude),
         csv = gsub("CSV: ", "", csv)
         ) %>%
  mutate(klasifikasi = stringr::str_trim(klasifikasi))
```

## Menghitung penjualan\_products

```
# penjualan products
judul = colnames(data)
judul = ifelse(grepl("penjualan", judul), "penjualan", judul)
colnames(data) = judul
```

## Memecah Data Menjadi Tiga Data

```
# pecah data
data_1 = data %>% select(id, penjualan)
data_2 = data %>% select(id, contains("gimmick"))
data_3 = data %>% select(-penjualan, -contains("gimmick"))
```

## Mengekstrak Dan Mengubah Struktur Data 1

```
# data_1
# pecah produk penjualan
data_all =
  data_1 %>%
  unnest_tokens(out, penjualan, token = "regex", pattern = "\n") %>%
  filter(!grepl("subtotal|tax|total", out, ignore.case = T)) %>%
  separate(out, into = c('produk', 'amount', 'quantity'), sep = "\\:") %>%
  filter(!is.na(amount)) %>%
  mutate(amount = gsub(" idr, quantity", "", amount),
         amount = gsub(".00", "", amount, fixed = T),
         amount = gsub(" idr", "", amount),
         amount = gsub("\\ ", "", amount),
         amount = gsub("\\,", "", amount),
         quantity = gsub("\\)", "", quantity),
         quantity = ifelse(is.na(quantity), 0, quantity),
         amount = as.numeric(amount),
         quantity = as.numeric(quantity),
         produk = gsub(" (amount", "", produk, fixed = T),
         produk = toupper(produk)) %>%
  mutate(brand = case_when(grepl("lokalate", produk, ignore.case = T) ~ "Lokalate",
                             grepl("tropicana|ts|slim", produk, ignore.case = T) ~ "Tropicana Slim",
                             grepl("nutrisari|ns|sari", produk, ignore.case = T) ~ "NutriSari",
                             grepl("diabetamil", produk, ignore.case = T) ~ "Diabetamil",
                             grepl("l-men", produk, ignore.case = T) ~ "L-Men",
                             grepl("hilo", produk, ignore.case = T) ~ "HiLo")
         ) %>%
  rename(price = amount) %>%
  mutate(total_value = price*quantity)
```

## Mengekstrak dan Mengubah Struktur Data 2

```
# data_2
# oprek gimmick
data_2 =
  data_2 %>%
  reshape2::melt(id.vars = "id") %>%
  rename(gimmick = variable) %>%
  mutate(value = as.numeric(value),
         value = ifelse(is.na(value), 0, value)) %>%
  mutate(brand = case_when(grepl("hi_lo", gimmick) ~ "HiLo",
                             grepl("lokalate", gimmick) ~ "Lokalate",
                             grepl("nutrisari|ns", gimmick) ~ "NutriSari",
                             grepl("tropicana|ts", gimmick) ~ "Tropicana Slim")
         ) %>%
  group_by(id, brand) %>%
  summarise(tot_gim = sum(value)) %>%
  ungroup() %>%
  filter(tot_gim > 0)

brand_gimmick = sort(unique(data_2$brand))

for(xx in brand_gimmick){
  temp = data_2 %>% filter(brand == xx & !is.na(tot_gim))
```

```

  colnames(temp)[3] = paste("gimmick",xx,sep = "_")
  data_all = merge(data_all,temp,all = T)
}

```

## Menggabungkan Semua Data Kembali

```

# data_3
data_all = merge(data_3,data_all,all = T) %>% arrange(id,brand)

data_all_1 = data_all %>% select(-contains("gimmick"))

data_all_2 =
  data_all %>%
  group_by(id,brand) %>%
  mutate(penanda = c(1:length(brand))) %>%
  ungroup() %>%
  select(contains("gimmick"),penanda)

data_all_2[data_all_2$penanda>1,] = NA

data_final =
  data.frame(data_all_1,data_all_2) %>%
  mutate(penanda = NULL,
         id = NULL)

tes = colnames(data_final)
tes = gsub("\\\\_", " ",tes)

```

## Membuat *Function* Agar Judul Kolom Sesuai EYD

```

proper <- function(x){
  stringi::stri_trans_general(x,id = "Title")
}

colnames(data_final) = proper(tes)

```

## *Export* ke .xlsx

```

openxlsx::write.xlsx(data_final,"hasil.xlsx")

```

## ***Files* yang Dibutuhkan**

Semua *files* yang dibutuhkan terlampir di *folder* ini.