

# Dokumentasi Converter AppSheet

## Divisi Distribution Development Nutrifood Indonesia

Ikang Fadhli  
Market Research Dept.  
Nutrifood Indonesia

## Pendahuluan

### Latar Belakang

Berbeda dengan tim AM yang menggunakan *JotForm* sebagai *platform* pengumpulan data lapangan. Tim *distribution development* di area SUM 1 menggunakan *platform* AppSheet. Salah satu kendala yang dialami sama dengan tim AM yakni struktur datanya tidak bisa langsung diolah. Perlu ada konversi ke struktur yang lebih mudah di-*pivot*.

### Data yang Digunakan

Data hasil survey bulanan tim *distribution development* SUM 1 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())  
library(readxl)  
library(dplyr)  
library(tidyr)  
library(reshape2)
```

### Memanggil Data `.xlsx`

```
# jangan lupa ganti path file  
nama_file_utama = "~/Documents/AppSheet-Sales/Damen/Call & Omzet SETP - Copy.xlsx"  
nama_file_harga = "~/Documents/AppSheet-Sales/Damen/Template Harga.xlsx"  
  
# =====  
# extract data target utama  
data = read_excel(nama_file_utama) %>%  
  janitor::clean_names()  
  
# =====
```

```
# extract database produk
dbase = read_excel(nama_file_harga,
                   col_types = c("text", "text", "numeric")) %>%
  janitor::clean_names() %>%
  mutate(item_standar = janitor::make_clean_names(item),
         brand = ifelse(brand == "TS", "Tropicana Slim", brand),
         brand = ifelse(brand == "NS", "NutriSari", brand))
```

## Mengambil Semua Informasi yang Diperlukan

```
# ambil informasi yang diperlukan
header_data = colnames(data)
nama_item = dbase$item_standar
item_yg_dijual = header_data[header_data %in% nama_item]
```

## Membagi Data Berdasarkan Informasi yang Dimiliki

```
# pertama dari produk
data_1 = data[colnames(data) %in% c("id", item_yg_dijual)]
data = data[!colnames(data) %in% item_yg_dijual]

# kedua dari gimmick
data_2 = data %>% select(id, contains("gimmick"))
data = data %>% select(-contains("gimmick"))

# ketiganya sisanya
data_3 = data %>% select(-starts_with("pg"),
                       -starts_with("sec"),
                       -contains("omzet"),
                       -transaksi_penjualan,
                       -ns_tea_sweet_tea,
                       -ns_wdank_bajigur)
```

## *Splitting Data* Berdasarkan id

```
# proses pengerjaannya mungkin akan rumit. kenapa?
# kalau kita lihat di sheet after,
# banyaknya baris akan tergantung dari banyaknya baris yang ada di
# gimmick dan produk yang terjual.
# maka dari itu, lebih baik semua data dbuat dalam bentuk list saja.
# nanti tinggal ditempel saja ke kanan.
data_1 = data_1 %>% split(., $id)
data_2 = data_2 %>% split(., $id)
data_3 = data_3 %>% split(., $id)
```

## Proses Looping Untuk Menggabungkan

```
# kita siapkan rumahnya dulu
ikanx = vector("list", length(data_1))

# sekarang kita mulai looping dari i = 1 sampai selesai
for(i in 1: length(data_1)){
  temp_1 = data_1[[i]] %>% as.data.frame()
  temp_2 = data_2[[i]] %>% as.data.frame()
  temp_3 = data_3[[i]] %>% as.data.frame()
```

```

# sekarang kita akan kerjakan yang data_2
# kita rapikan gimmick
# rules: saat tidak ada gimmick, maka sisanya dbuat nol alias NA
if(temp_2$pemberian_gimmick == "Ada"){
  temp_2 =
    temp_2 %>%
    melt(id.vars = c("id","pemberian_gimmick")) %>%
    mutate(variable = ifelse(grepl("item",variable),
                              "item_gimmick",
                              "qty_gimmick")
    )
  temp_2_1 =
    temp_2 %>%
    filter(grepl("item",variable)) %>%
    select(-variable) %>%
    rename(item_gimmick = value)
  temp_2_2 =
    temp_2 %>%
    filter(grepl("qty",variable)) %>%
    select(-variable) %>%
    rename(qty_gimmick = value) %>%
    select(qty_gimmick)
  temp_2 = cbind(temp_2_1,temp_2_2)
} else {
  temp_2 = data.frame(
    id = temp_2$id,
    pemberian_gimmick = temp_2$pemberian_gimmick,
    item_gimmick = NA,
    qty_gimmick = NA
  )
}

# sekarang kita akan kerjakan yang data_1
# item penjualan kita buat tabular
temp_1 =
  temp_1 %>%
  melt(id.vars = "id") %>%
  filter(!is.na(value)) %>%
  rename(item_standar = variable) %>%
  merge(dbase) %>%
  mutate(omzet = value*harga) %>%
  select(-item_standar) %>%
  rename(qty_penjualan = value,
         item_penjualan = item) %>%
  relocate(id,item_penjualan,brand,qty_penjualan,harga,omzet)

# sekarang saatnya moment of truth
m1 = nrow(temp_1)
m2 = nrow(temp_2)
m3 = nrow(temp_3)

max_m = max(c(m1,m2,m3))

```

```

if(m1 < max_m){
  temp_1[(m1+1):max_m,] = NA
}
if(m2 < max_m){
  temp_2[(m2+1):max_m,] = NA
}
if(m3 < max_m){
  temp_3[(m3+1):max_m,] = temp_3[1,]
}

temp_1$id = NULL
temp_2$id = NULL
temp_3$id = NULL

final = cbind(temp_3,temp_2,temp_1)
ikanx[[i]] = final
print(i)
}

```

### Proses Penggabungan Data per id

```

# saatnya kita gabung kembali
printed_data = data.frame()
for(i in 1:length(data_1)){
  temp = ikanx[[i]]
  printed_data = rbind(temp,printed_data)
  print(i)
}

```

### *Export* ke Dalam Format .xlsx

```
openxlsx::write.xlsx(printed_data,"hasil_konversi.xlsx")
```

### *Files* yang Dibutuhkan

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