

**TUGAS KE-3**  
**“DATA SCRAPING”**  
**MATA KULIAH STATISTIKA KOMPUTASI**  
**KELAS A**



**DISUSUN OLEH:**

Shafira Amanda Putri (NPM. 22083010008)  
Nabilah Selayanti (NPM. 22083010013)

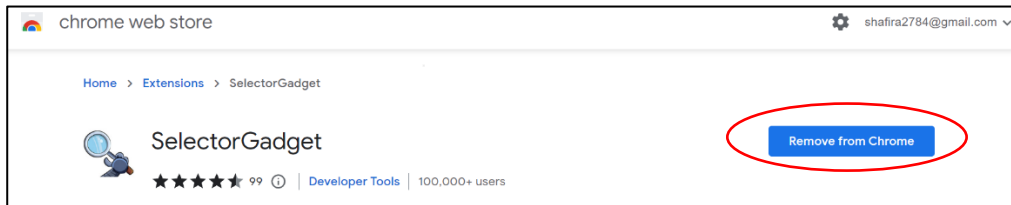
**DOSEN PENGAMPU:**

Trimono, S.Si., M.Si (NIP. 199509082022031003)

**PROGRAM STUDI SAIN DATA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAWA TIMUR**  
**2023**

## A. DATA SCRAPING

### 1. Install SelectorGadget pada Google Chrome

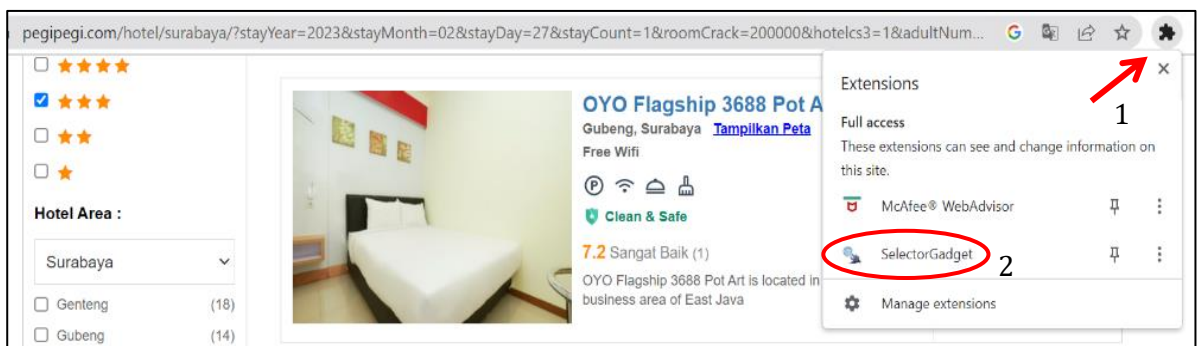


### 2. Mencari web untuk di scraping

Pada modul ini, kita menggunakan web Pegipegi.com untuk di scraping datanya dan dianalisis deskriptif serta divisualisasikan.

<https://www.pegipegi.com/hotel/surabaya/?stayYear=2023&stayMonth=02&stayDay=27&stayCount=1&roomCrack=200000&hotelcs3=1&adultNum=2&lowestPriceTemp=111497&highestPriceTemp=2297591&roomCount=1&activeSort=1>

Lalu mengfilter apa saja yang ingin kita scraping dari web tersebut menggunakan selector gadget



### 3. Install Package di software R-Studio

Langkah pertama yang dilakukan adalah install package **xml2**, **rvest**. Menggunakan fungsi dari sebuah package, yaitu

```
library(xml2)
library(rvest)
```

Setelah kedua package terinstall dan diaktifkan, copy link web Pegipegi untuk mengambil data dari Pegipegi.com hotel di Surabaya ke dalam R dan menggunakan sintaks berikut :

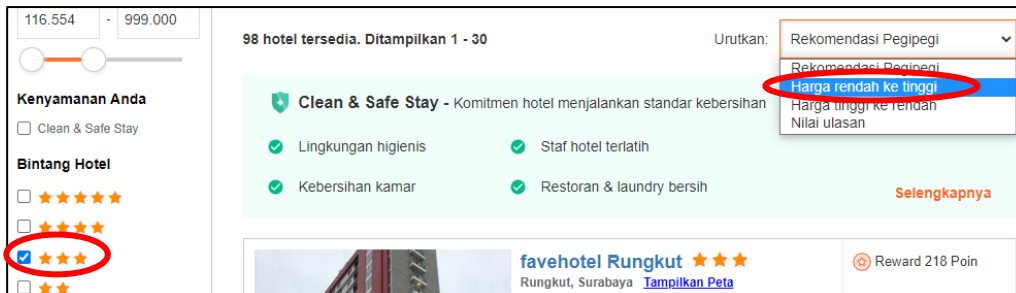
```
5
6 alamatweb = 'https://www.pegipegi.com/hotel/surabaya/?stayYear=2023&stayMonth=02&stayDay=25&stayCount=1&roomCrack=200000&m
7 lamanweb = read_html(alamatweb)
8 lamanweb
```

Dibawah ini merupakan Output dari sintaks diatas :

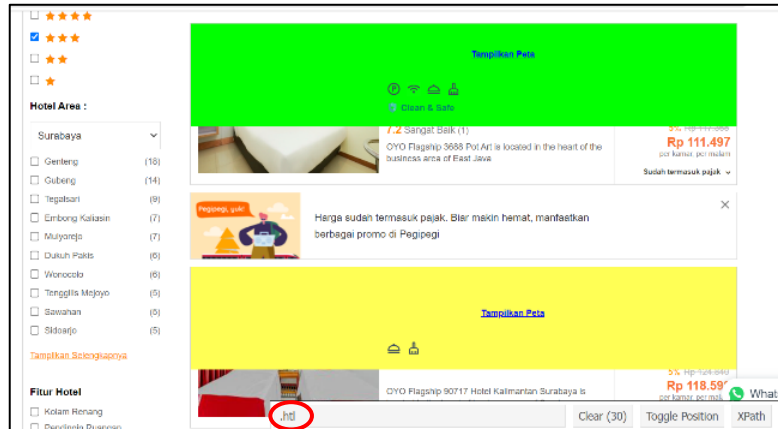
```
> alamatweb = 'https://www.pegipegi.com/hotel/surabaya/?stayYear=2023&stayMonth=02&stayDay=25&stayCount=1&roomCrack=200000&minPrice=138510&maxPrice=999000&hotelcs3=1&reshFlg=1&adultNum=2&lowestPriceTemp=138510&highestPriceTemp=2680521&roomCount=1&activeSort=0'
> lamanweb = read_html(alamatweb)
> lamanweb
{html_document}
<html lang="en">
[1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">\n<meta http-equiv="conte ...
[2] <body mc="3">\n<script type="text/javascript">\nvar dataLayer = [\n{\n  "hotel_id" : ['915511','9427' ...
>
```

#### 4. Scraping data VARIABEL 1 : Nama Hotel

Pada variabel 1, kita ingin scraping “Nama Hotel di Surabaya” yang sudah di filter bintang 3 dan harga rendah ke tinggi.



Selanjutnya, kita akan mengambil data **Nama Hotel** pada web Pegipegi.com pada Hotel di Surabaya yang sudah difilter bintang 3 dan filter harga rendah ke tinggi.



Pada gambar diatas, yang terpilih di web pegipegi.com ada 30 nama hotel. Kemudian, copy nama yang sudah dilingkari dengan warna merah pada gambar diatas dan paste di SourceCode R-Studio. Berikut merupakan sintaks yang digunakan untuk scraping data **Nama Hotel**

- Source Code: Variabel 1

```
8 #VARIABEL 1: Hotel di Surabaya
9 nama_hotel<- html_nodes(lamanweb, '.html')
10 nama_hotel<- html_text(nama_hotel)
11 nama_hotel
```

Tampilan Output :

```

> #VARIABEL 1: Hotel di Surabaya
> nama_hotel<- html_nodes(lamanweb,'.htl')
> nama_hotel<- html_text(nama_hotel)
> nama_hotel
[1] "favehotel Rungkut\n " "Alimar Premier Hotel\n "
[3] "Luminor Hotel Jemursari\n " "Royal Regantris Cendana Formerly Royal Singosari\n "
[5] "Hotel Santika Pandegiling Surabaya\n " "Maumu Hotel and Lounge\n "
[7] "Country Heritage Hotel\n " "Kampi Hotel Tunjungan - Surabaya\n "
[9] "Best Hotel Surabaya\n " "G Suites Hotel by AMITHYA\n "
[11] "Primebiz Hotel Surabaya\n " "Great Diponegoro Hotel Surabaya\n "
[13] "KHAS Surabaya Hotel\n " "Midtown Hotel Surabaya\n "
[15] "Life Emerald Hotel Surabaya\n " "Swiss-Belinn Manyar Surabaya\n "
[17] "Zoom Dharmahusada Hotel\n " "MaxOneHotels at Tidar Surabaya\n "
[19] "Namira Syariah Surabaya Hotel\n " "Regantris Hotel Surabaya\n "
[21] "BATIQA Hotel Darmo - Surabaya\n " "Whiz Prime Hotel Darmo Harapan Surabaya\n "
[23] "Life Hotel Stasiun Kota Surabaya\n " "La Lisa Hotel Surabaya\n "
[25] "The Life Styles Hotel Surabaya\n " "I&M Hotel\n "
[27] "Swiss-Belinn Tunjungan Surabaya\n " "Hotel Dafam Pacific Caesar Surabaya\n "
[29] "PALM PARK Hotel Surabaya\n " "Grand Surabaya Hotel\n "

```

Langkah selanjutnya kita akan menggunakan fungsi untuk menghapus “\n” dengan menggunakan sintaks berikut :

```

12 nama_hotel<-gsub("\n","",nama_hotel)
13 nama_hotel|
14

```

Maka tampilan dari scraping data **Nama Hotel** seperti tampilan berikut :

```

> nama_hotel<-gsub("\n","",nama_hotel)
> nama_hotel
[1] "OYO Flagship 3688 Pot Art " "OYO Flagship 90717 Hotel Kalimantan Surabaya "
[3] "RedDoorz Plus near Galaxy Mall " "I&M Hotel "
[5] "RedDoorz @ Panglima Sudirman " "RedDoorz Plus @ K23 Rungkut Madya "
[7] "Sans Hotel Kupang Indah Surabaya " "OYO Flagship 3869 Metro House Sub "
[9] "RedDoorz @ Siwalankerto Surabaya " "Life Emerald Hotel Surabaya "
[11] "Maumu Hotel and Lounge " "D'Carol Hotel "
[13] "Country Heritage Hotel " "Life Hotel Sudirman Surabaya "
[15] "RedDoorz near Stasiun TVRI Surabaya 2 " "Kokonut Guesthouse "
[17] "Life Hotel Stasiun Kota Surabaya " "Sans Hotel Box Mansion Surabaya "
[19] "Prime Royal Boutique Hotel Surabaya " "Hotel Bed and Breakfast Surabaya "
[21] "Oval Hotel Surabaya " "High Point Serviced Apartment "
[23] "Life Taman Melati Surabaya " "Livinn Taman Melati Surabaya "
[25] "Tab Hotel Darmo Permai Surabaya " "PALM PARK Hotel Surabaya "
[27] "Sans Hotel International Surabaya " "New Grand Park "
[29] "Sans Hotel Rajawali Surabaya " "favehotel Rungkut "

```

## 5. Scraping data VARIABEL 2 : Harga Normal (sebelum diskon)

Pada variabel 2, mengambil data informasi lainnya yaitu variabel **harga normal** dengan menggunakan SelectorGadget.



Langkah selanjutnya, copy nama variabel yang dipilih atau yang dilingkari warna merah pada gambar diatas dan di paste pada Source code RStudio.

- Source Code : Variabel 2

```

10
17 #VARIABEL 2: Harga normal(sebelum diskon)
18 normal_price <- html_nodes(lamanweb,'.normalPrice i')
19 normal_price <- html_text(normal_price)
20 normal_price

```

Tampilan Output :

```
> #VARIABEL 2: Harga normal(sebelum diskon)
> normal_price <- html_nodes(lamanweb,'.normalPrice i')
> normal_price <- html_text(normal_price)
> normal_price
[1] "\n \n"
"
[2] "\n \n"
"
[3] "\n \n"
"
[4] "\n \n"
"
[5] "\n \n"
"
[6] "\n \n"
"
[7] "\n \n"
"
[8] "\n \n"
"
[9] "\n \n"
"
[10] "\n \n"
"
[11] "\n \n"
"
[12] "\n \n"
"
[13] "\n \n"
"
[14] "\n \n"
"
[15] "\n \n"
"
[16] "\n \n"
"
[17] "\n \n"
"
[18] "\n \n"
"
[19] "\n \n"
"
[20] "\n \n"
"
[21] "\n \n"
"
[22] "\n \n"
"
[23] "\n \n"
"
[24] "\n \n"
"
[25] "\n \n"
"
[26] "\n \n"
"
[27] "\n \n"
"
[28] "\n \n"
"
[29] "\n \n"
"
[30] "\n \n"
"
\n Rp 427.035\n \n \n
\n Rp 732.000\n \n \n
\n Rp 484.210\n \n \n
\n Rp 615.789\n \n \n
\n Rp 485.000\n \n \n
\n Rp 266.667\n \n \n
\n Rp 295.955\n \n \n
\n Rp 610.589\n \n \n
\n Rp 406.667\n \n \n
\n Rp 635.769\n \n \n
\n Rp 613.113\n \n \n
\n Rp 723.922\n \n \n
\n Rp 485.000\n \n \n
\n Rp 807.690\n \n \n
\n Rp 373.332\n \n \n
\n Rp 579.241\n \n \n
\n Rp 500.000\n \n \n
\n Rp 400.000\n \n \n
\n Rp 350.853\n \n \n
\n Rp 520.000\n \n \n
\n Rp 559.615\n \n \n
\n Rp 362.000\n \n \n
\n Rp 373.332\n \n \n
\n Rp 590.003\n \n \n
\n Rp 559.999\n \n \n
\n Rp 230.769\n \n \n
\n Rp 883.210\n \n \n
\n Rp 663.684\n \n \n
\n Rp 450.000\n \n \n
\n Rp 526.667\n \n \n
```

Selanjutnya, kita akan menghilangkan “\n”, “Rp”, dan spasi yang masih kosong (“ ”) menggunakan sintaks berikut :

```
21 normal_price<-gsub("\n","",normal_price)
22 normal_price<-gsub("Rp","",normal_price)
23 normal_price<-gsub(" ", "",normal_price)
24 normal_price
```

Tampilan Output seperti berikut :

```
> normal_price<-gsub("\n","",normal_price)
> normal_price<-gsub("Rp","",normal_price)
> normal_price<-gsub(" ","",normal_price)
> normal_price
[1] "427.035" "732.000" "484.210" "615.789" "485.000" "266.667" "295.955" "610.589" "406.667" "635.769" "613.113"
[12] "723.922" "485.000" "807.690" "373.332" "579.241" "500.000" "400.000" "350.853" "520.000" "559.615" "362.000"
[23] "373.332" "590.003" "559.999" "230.769" "883.210" "663.684" "450.000" "526.667"
> |
```

Karena data masih berupa string, maka langkah selanjutnya kita akan mengubah data menjadi bentuk numeric dengan menggunakan sintaks berikut :

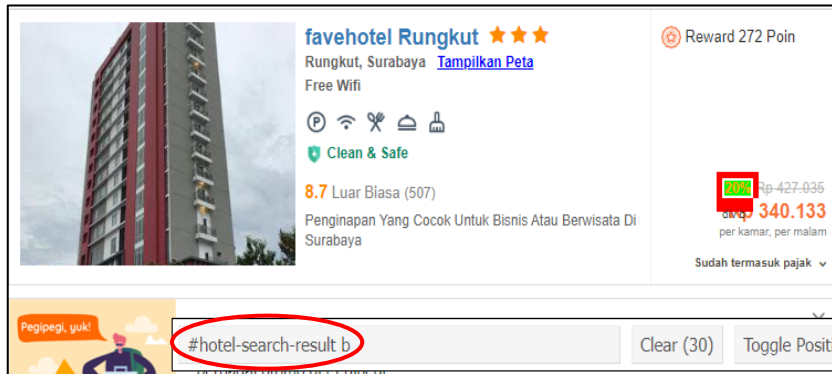
```
25 normal_price<-as.numeric(normal_price)
26 normal_price
27
```

Maka tampilan dari scraping dengan variabel **Harga Normal** seperti tampilan berikut:

```
> normal_price<-as.numeric(normal_price)
> normal_price
[1] 427.035 732.000 484.210 615.789 485.000 266.667 295.955 610.589 406.667 635.769 613.113 723.922 485.000 807.690
[15] 373.332 579.241 500.000 400.000 350.853 520.000 559.615 362.000 373.332 590.003 559.999 230.769 883.210 663.684
[29] 450.000 526.667
> |
```

## 6. Scraping data VARIABEL 3 : Diskon

Variabel selanjutnya kita mengambil variabel **Diskon** dengan menggunakan SelectorGadget.



Copy nama variabel yang telah dipilih yang tertera pada gambar yang ditandai dengan lingkaran warna merah, lalu paste di SourceCode R-Studio dengan menggunakan sintaks berikut :

- Source Code : Variabel 3

```
28 #VARIABEL3: Diskon
29 discount <- html_nodes(lamanweb, '#hotel-search-result b')
30 discount <- html_text(discount)
31 discount
```

Maka tampilan dari scraping dengan variabel **Diskon** seperti tampilan berikut :

```
> #VARIABEL3: Diskon
> discount <- html_nodes(lamanweb, '#hotel-search-result b')
> discount <- html_text(discount)
> discount
[1] "20%" "27%" "5%" "5%" "5%" "25%" "5%" "5%" "25%" "35%" "35%" "25%" "5%" "35%" "25%" "5%" "25%" "19%" "5%"
[20] "5%" "35%" "13%" "25%" "5%" "27%" "35%" "5%" "5%" "5%" "26%"
```

## 7. Scraping data VARIABEL 4 : Harga Setelah Diskon

Pada variabel 4 kita mengambil variabel tentang **Harga Setelah Diskon** dan sudah termasuk pajak dengan menggunakan SelectorGadget.



Langkah selanjutnya, copy nama variabel yang dipilih atau yang dilingkari warna merah pada gambar diatas kemudian di paste pada Source code R-Studio.

- Source Code : Variabel 4

```
32
33 #VARIABEL 4: Harga setelah diskon
34 net_price <- html_nodes(lamanweb, '.diskonPrice')
35 net_price <- html_text(net_price)
36 net_price
```

Tampilan Output menggunakan sintaks diatas seperti berikut :

```
> #VARIABEL 4: Harga setelah diskon
> net_price <- html_nodes(lamanweb, '.diskonPrice')
> net_price <- html_text(net_price)
> net_price
[1] "\n \n\n \n Rp 359.030\n \n \n"
[2] "\n \n\n \n Rp 532.530\n \n \n"
[3] "\n \n\n \n Rp 460.000\n \n \n"
[4] "\n \n\n \n Rp 585.000\n \n \n"
[5] "\n \n\n \n Rp 460.750\n \n \n"
[6] "\n \n\n \n Rp 200.000\n \n \n"
[7] "\n \n\n \n Rp 281.158\n \n \n"
[8] "\n \n\n \n Rp 580.060\n \n \n"
[9] "\n \n\n \n Rp 305.000\n \n \n"
[10] "\n \n\n \n Rp 413.250\n \n \n"
[11] "\n \n\n \n Rp 398.524\n \n \n"
[12] "\n \n\n \n Rp 542.941\n \n \n"
[13] "\n \n\n \n Rp 460.750\n \n \n"
[14] "\n \n\n \n Rp 524.999\n \n \n"
[15] "\n \n\n \n Rp 279.999\n \n \n"
[16] "\n \n\n \n Rp 550.279\n \n \n"
[17] "\n \n\n \n Rp 375.000\n \n \n"
[18] "\n \n\n \n Rp 323.190\n \n \n"
[19] "\n \n\n \n Rp 333.311\n \n \n"
[20] "\n \n\n \n Rp 494.000\n \n \n"
[21] "\n \n\n \n Rp 363.750\n \n \n"
[22] "\n \n\n \n Rp 316.026\n \n \n"
[23] "\n \n\n \n Rp 279.999\n \n \n"
[24] "\n \n\n \n Rp 560.503\n \n \n"
[25] "\n \n\n \n Rp 407.399\n \n \n"
[26] "\n \n\n \n Rp 150.000\n \n \n"
[27] "\n \n\n \n Rp 839.050\n \n \n"
[28] "\n \n\n \n Rp 630.500\n \n \n"
[29] "\n \n\n \n Rp 735.375\n \n \n"
[30] "\n \n\n \n Rp 427.500\n \n \n"
```

Langkah selanjutnya, kita akan menghilangkan “\n”, “Rp”, dan spasi yang masih kosong (“ ”) menggunakan sintaks berikut :

```
37 net_price<-gsub("\n","",net_price)
38 net_price<-gsub(" ","",net_price)
39 net_price<-gsub("Rp","",net_price)
40 net_price
```

Tampilan Output seperti berikut :

```
> net_price<-gsub("\n","",net_price)
> net_price<-gsub("Rp","",net_price)
> net_price<-gsub(" ","",net_price)
> net_price
[1] "359.030" "532.530" "460.000" "585.000" "460.750" "200.000" "281.158" "580.060" "305.000" "413.250" "398.524" "542.941"
[13] "460.750" "524.999" "279.999" "550.279" "375.000" "323.190" "333.311" "494.000" "363.750" "316.026" "279.999" "560.503"
[25] "407.399" "150.000" "839.050" "630.500" "735.375" "427.500"
>
```

Karena data pada gambar diatas masih berupa **string**, maka langkah selanjutnya kita akan mengubah data menjadi bentuk **numeric** dengan menggunakan sintaks berikut :

```
41 net_price<-as.numeric(net_price)
42 net_price
```

Maka tampilan dari scraping dengan variabel **Harga Setelah Diskon** seperti tampilan berikut :

```
> net_price<-as.numeric(net_price)
> net_price
[1] 359.030 532.530 460.000 585.000 460.750 200.000 281.158 580.060 305.000 413.250 398.524 542.941 460.750 524.999 279.999
[16] 550.279 375.000 323.190 333.311 494.000 363.750 316.026 279.999 560.503 407.399 150.000 839.050 630.500 735.375 427.500
>
```

## 8. Scraping data VARIABEL 5 : Rating Hotel

Variabel selanjutnya kita mengambil variabel **Rating Hotel** dengan menggunakan



SelectorGadget.



Kemudian, copy nama variabel yang dipilih atau yang dilingkari dengan warna merah pada gambar diatas lalu di paste pada Source code R-Studio.

- Source Code : Variabel 5

```
44 #VARIABEL5: Rating Hotel
45 rating <- html_nodes(lamanweb, '.ratingRight span:nth-child(1)')
46 rating <- html_text(rating)
47 rating
```

Tampilan Output menggunakan sintaks pada gambar diatas seperti berikut :

```
> #VARIABEL5: Rating Hotel
> rating <- html_nodes(lamanweb, '.ratingRight span:nth-child(1)')
> rating <- html_text(rating)
> rating
[1] "8.7" "8.3" "8.7" "8.8" "8.4" "8.1" "7.9" "9.3" "8.7" "8.4" "9.5" "8.8" "8.9" "8.6" "8.4" "8.5" "8.7" "8.7" "9.2" "8.5"
[21] "9.2" "8.4" "8.2" "8.8" "9.0" "8.4" "9.0" "9.2" "8.2" "9.3"
>
```

Karena data pada gambar diatas masih berupa **string**, maka langkah selanjutnya kita akan mengubah data menjadi bentuk **numeric** dengan menggunakan sintaks berikut :

```
48 rating <- as.numeric(rating)
49 rating
50
```

Maka tampilan dari scraping dengan variabel **Rating Hotel** seperti tampilan berikut :

```
> rating <- as.numeric(rating)
> rating
[1] 8.7 8.3 8.7 8.8 8.4 8.1 7.9 9.3 8.7 8.4 9.5 8.8 8.9 8.6 8.4 8.5 8.7 8.7 9.2 8.5 9.2 8.4 8.2 8.8 9.0 8.4 9.0 9.2 8.2 9.3
>
```

## 9. Scraping data VARIABEL 6 : Keterangan Pajak

Pada variabel 5 kita mengambil variabel **Keterangan Pajak** tentang apakah harga tersebut sudah termasuk pajak dengan menggunakan SelectorGadget.





Langkah selanjutnya, copy nama variabel yang dipilih atau yang dilingkari dengan warna merah pada gambar diatas lalu di paste pada Source Code R-Studio.

- Source Code : Variabel 6

```
51 #VARIABEL 6: Keterangan pajak
52 tax<- html_nodes(lamanweb,'.priceDetailButton')
53 tax <- html_text(tax)
54 tax
```

Tampilan Output menggunakan sintaks pada gambar diatas seperti berikut :

```
> #VARIABEL 6: Keterangan pajak
> tax<- html_nodes(lamanweb,'.priceDetailButton')
> tax <- html_text(tax)
> tax
[1] "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n "
[4] "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n "
[7] "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n "
[10] "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n "
[13] "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n "
[16] "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n "
[19] "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n "
[22] "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n "
[25] "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n "
[28] "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n " "\n\n Sudah termasuk pajak\n\n\n "
```

Selanjutnya, kita akan menghilangkan “\n”, menggunakan sintaks berikut :

```
55 tax<-gsub("\n","",tax)
56 tax
```

Maka tampilan dari scraping dengan variabel **Keterangan Pajak** seperti tampilan berikut :

```
> tax<-gsub("\n","",tax)
> tax
[1] " Sudah termasuk pajak " " Sudah termasuk pajak " " Sudah termasuk pajak "
[4] " Sudah termasuk pajak " " Sudah termasuk pajak " " Sudah termasuk pajak "
[7] " Sudah termasuk pajak " " Sudah termasuk pajak " " Sudah termasuk pajak "
[10] " Sudah termasuk pajak " " Sudah termasuk pajak " " Sudah termasuk pajak "
[13] " Sudah termasuk pajak " " Sudah termasuk pajak " " Sudah termasuk pajak "
[16] " Sudah termasuk pajak " " Sudah termasuk pajak " " Sudah termasuk pajak "
[19] " Sudah termasuk pajak " " Sudah termasuk pajak " " Sudah termasuk pajak "
[22] " Sudah termasuk pajak " " Sudah termasuk pajak " " Sudah termasuk pajak "
[25] " Sudah termasuk pajak " " Sudah termasuk pajak " " Sudah termasuk pajak "
[28] " Sudah termasuk pajak " " Sudah termasuk pajak " " Sudah termasuk pajak "
```

## 10. Scraping data VARIABEL 7 : Pemesanan

Variabel selanjutnya kita mengambil variabel **Pemesanan** di Hotel Surabaya pada Pegipegi.com dengan menggunakan SelectorGadget.



Selanjutnya, copy nama variabel yang dipilih atau yang dilingkari dengan warna merah pada gambar diatas lalu di paste pada Source Code R-Studio.

- Source Code : Variabel 7

```
58
59 #VARIABEL 7: Pemesanan
60 booking<- html_nodes(lamanweb,'.perNight')
61 booking <- html_text(booking)
62 booking
```

Tampilan Output menggunakan sintaks pada gambar diatas seperti berikut :

```
> #VARIABEL 7: Pemesanan
> booking<- html_nodes(lamanweb,'.perNight')
> booking <- html_text(booking)
> booking
[1] "\n per kamar, per malam\n " "\n per kamar, per malam\n " "\n per kamar, per malam\n " "\n per kamar, per malam\n "
[5] "\n per kamar, per malam\n " "\n per kamar, per malam\n " "\n per kamar, per malam\n " "\n per kamar, per malam\n "
[9] "\n per kamar, per malam\n " "\n per kamar, per malam\n " "\n per kamar, per malam\n " "\n per kamar, per malam\n "
[13] "\n per kamar, per malam\n " "\n per kamar, per malam\n " "\n per kamar, per malam\n " "\n per kamar, per malam\n "
[17] "\n per kamar, per malam\n " "\n per kamar, per malam\n " "\n per kamar, per malam\n " "\n per kamar, per malam\n "
[21] "\n per kamar, per malam\n " "\n per kamar, per malam\n " "\n per kamar, per malam\n " "\n per kamar, per malam\n "
[25] "\n per kamar, per malam\n " "\n per kamar, per malam\n " "\n per kamar, per malam\n " "\n per kamar, per malam\n "
[29] "\n per kamar, per malam\n " "\n per kamar, per malam\n "
> |
```

Kemudian, kita akan menghilangkan “\n”, menggunakan sintaks berikut :

```
62 booking<-gsub("\n","",booking)
63 booking
```

Maka tampilan dari scraping dengan variabel **Pemesanan** seperti tampilan berikut :

```
> booking<-gsub("\n","",booking)
> booking
[1] " per kamar, per malam " " per kamar, per malam " " per kamar, per malam " " per kamar, per malam "
[5] " per kamar, per malam " " per kamar, per malam " " per kamar, per malam " " per kamar, per malam "
[9] " per kamar, per malam " " per kamar, per malam " " per kamar, per malam " " per kamar, per malam "
[13] " per kamar, per malam " " per kamar, per malam " " per kamar, per malam " " per kamar, per malam "
[17] " per kamar, per malam " " per kamar, per malam " " per kamar, per malam " " per kamar, per malam "
[21] " per kamar, per malam " " per kamar, per malam " " per kamar, per malam " " per kamar, per malam "
[25] " per kamar, per malam " " per kamar, per malam " " per kamar, per malam " " per kamar, per malam "
[29] " per kamar, per malam " " per kamar, per malam "
```

## 11. Data Frame

Untuk melihat tipe dan struktur objek pada data tersebut maka menggunakan sintaks

str(Hotel\_surabaya)

```
68 #DataFrame
69 Hotel_surabaya <-data.frame(Hotel_Surabaya = nama_hotel, Normal_Price = normal_price, Discount = discount,
70                             Net_Price = net_price, Rating = rating, Tax = tax, Booking = booking)
71
```

Maka tampilan Output dari sintaks gambar diatas seperti berikut :

	Hotel_Surabaya	Normal_Price	Discount	Net_Price	Rating	Tax	Booking
1	favehotel Rungkut	419.427	35%	272.628	8.7	Sudah termasuk pajak	per kamar, per malam
2	Alimar Premier Hotel	670.046	35%	435.530	8.3	Sudah termasuk pajak	per kamar, per malam
3	Luminor Hotel Jemursari	484.210	5%	460.000	8.7	Sudah termasuk pajak	per kamar, per malam
4	Royal Regantris Cendana Formerly Royal Singosari	410.000	5%	389.500	8.8	Sudah termasuk pajak	per kamar, per malam
5	Hotel Santika Pandegiling Surabaya	568.181	27%	413.352	8.4	Sudah termasuk pajak	per kamar, per malam
6	Maumu Hotel and Lounge	284.615	35%	185.000	8.1	Sudah termasuk pajak	per kamar, per malam
7	Country Heritage Hotel	261.302	27%	190.098	7.9	Sudah termasuk pajak	per kamar, per malam
8	Kampi Hotel Tunjungan - Surabaya	610.589	5%	580.060	9.3	Sudah termasuk pajak	per kamar, per malam
9	Best Hotel Surabaya	406.667	32%	275.000	8.7	Sudah termasuk pajak	per kamar, per malam
10	G Suites Hotel by AMITHYA	591.923	35%	384.750	8.4	Sudah termasuk pajak	per kamar, per malam
11	Primebiz Hotel Surabaya	613.113	35%	398.524	9.5	Sudah termasuk pajak	per kamar, per malam
12	Great Diponegoro Hotel Surabaya	515.383	35%	334.999	8.8	Sudah termasuk pajak	per kamar, per malam

13	KHAS Surabaya Hotel	500.000	8%	460.750	8.9	Sudah termasuk pajak	per kamar, per malam
14	Midtown Hotel Surabaya	746.153	35%	485.000	8.6	Sudah termasuk pajak	per kamar, per malam
15	Life Emerald Hotel Surabaya	307.690	35%	199.999	8.4	Sudah termasuk pajak	per kamar, per malam
16	Evora Hotel Surabaya	268.046	5%	254.644	8.8	Sudah termasuk pajak	per kamar, per malam
17	Swiss-Belinn Manyar Surabaya	522.268	5%	496.155	8.5	Sudah termasuk pajak	per kamar, per malam
18	Zoom Dharmahusada Hotel	394.737	5%	375.001	8.7	Sudah termasuk pajak	per kamar, per malam
19	Oval Hotel Surabaya	343.695	35%	223.402	7.4	Sudah termasuk pajak	per kamar, per malam
20	MaxOneHotels at Tidar Surabaya	400.000	23%	307.800	8.7	Sudah termasuk pajak	per kamar, per malam
21	Sans Hotel Box Mansion Surabaya	219.209	5%	208.249	8.8	Sudah termasuk pajak	per kamar, per malam
22	Namira Syariah Surabaya Hotel	350.853	5%	333.311	9.2	Sudah termasuk pajak	per kamar, per malam
23	Regantris Hotel Surabaya	520.000	30%	364.800	8.5	Sudah termasuk pajak	per kamar, per malam
24	BATIQA Hotel Darmo - Surabaya	529.769	35%	344.350	9.2	Sudah termasuk pajak	per kamar, per malam
25	Whiz Prime Hotel Darmo Harapan Surabaya	362.000	24%	276.741	8.4	Sudah termasuk pajak	per kamar, per malam
26	Life Hotel Stasiun Kota Surabaya	307.690	35%	199.999	8.2	Sudah termasuk pajak	per kamar, per malam
27	La Lisa Hotel Surabaya	702.969	35%	456.930	8.8	Sudah termasuk pajak	per kamar, per malam
28	The Life Styles Hotel Surabaya	477.536	35%	310.399	9.0	Sudah termasuk pajak	per kamar, per malam
29	I&M Hotel	230.769	35%	150.000	8.4	Sudah termasuk pajak	per kamar, per malam
30	Swiss-Belinn Tunjungan Surabaya	748.291	5%	710.877	9.0	Sudah termasuk pajak	per kamar, per malam

## 12. Tipe dan struktur objek

Data Frame merupakan data yang telah dihimpun untuk dijadikan sebagai data frame.

- Berikut Source Code Data Frame

```
65 #Melihat tipe dan struktur objek
66 str(Hotel_surabaya)
67
```

Tampilan Output data frame dari data telah dihimpun seperti berikut :

```
> str(Hotel_surabaya)
'data.frame': 30 obs. of 7 variables:
 $ Hotel_Surabaya: chr "favehotel Rungkut " "Alimar Premier Hotel " "Luminor Hotel Jemursari " "Royal Regantris
Cendana Formerly Royal Singosari " ...
 $ Normal_Price : num 427 732 484 616 523 ...
 $ Discount : chr "16%" "27%" "5%" "5%" ...
 $ Net_Price : num 359 533 460 585 497 ...
 $ Rating : num 8.7 8.3 8.7 8.8 8.4 8.1 7.9 9.3 8.7 8.4 ...
 $ Tax : chr "Sudahtermasukpajak" "Sudahtermasukpajak" "Sudahtermasukpajak" ...
 $ Booking : chr "perkamar,permalam" "perkamar,permalam" "perkamar,permalam" "perkamar,permalam" ...
> |
```

Dari hasil diatas dapat diketahui bahwa objek **Hotel\_Surabaya** adalah sebuah data frame berukuran 30 (obs) observasi dan 7 variable. Variabel yang bertipe **char (character)** terdapat pada nama variabel Hotel\_Surabaya, Discount, Tax, Booking. Dan variabel yang bertipe **num (numeric)** terdapat pada nama variabel Normal\_Price, Net\_Price, dan Rating.

## B. ANALISIS DESKRIPTIF

### 1. Mencari Missing Value

Untuk mencari keberadaan missing value, maka menggunakan sintaks `is.na(Hotel_surabaya)`.

```
75 #mencari missing value
76 is.na(Hotel_surabaya)
77
```

Tampilan Output dengan menggunakan sintaks gambar diatas :

```
> #mencari missing value
> is.na(Hotel_surabaya)
      Hotel_surabaya Normal_Price Discount Net_Price Rating   Tax Booking
[1,]          FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[2,]          FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[3,]          FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[4,]          FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[5,]          FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[6,]          FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[7,]          FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[8,]          FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[9,]          FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[10,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[11,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[12,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[13,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[14,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[15,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[16,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[17,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[18,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[19,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[20,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[21,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[22,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[23,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[24,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[25,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[26,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[27,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[28,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[29,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
[30,]         FALSE          FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
> |
```

Dari hasil gambar diatas dapat diketahui bahwa data ini **tidak ditemukan** terjadinya missing value maka semua outputnya bertulisan “FALSE”.

## 2. Mencari jumlah missing value

Untuk memudahkan kita mencari jumlah missing value pada data tersebut, maka kita menggunakan fungsi `sum(is.na(Hotel_surabaya))`

```
77
78 #Mencari jumlah missing value
79 sum(is.na(Hotel_surabaya))
80
```

Tampilan Output dapat kita lihat gambar dibawah bahwa jumlah missing value pada data tersebut yaitu 0

```
> #Mencari jumlah missing value
> sum(is.na(Hotel_surabaya))
[1] 0
> |
```

## 3. Summary

Agar lebih mempermudah membaca data, menggunakan function summary untuk meringkas beberapa nilai data menjadi satu nilai.

```
80
81 #ringkasan data menggunakan summary
82 summary(Hotel_surabaya)
83 |
```

Function summary digunakan untuk mencari nilai statistic diantaranya : Minimum,

Q1, Median, Mean, Q3, Maksimum. Gambar dibawah ini merupakan output dari function summary

```
> summary(Hotel_surabaya)
HotelSurabaya      NormalPrice      Discount      NetPrice
Length:30          Min.   :140.4      Length:30          Min.   :133.4
Class :character    1st Qu.:234.7      Class :character    1st Qu.:186.4
Mode  :character    Median :279.0      Mode  :character    Median :216.3
                        Mean  :274.0                        Mean  :208.6
                        3rd Qu.:316.5                      3rd Qu.:234.9
                        Max.  :407.8                        Max.  :265.1
Tax
Length:30
Class :character
Mode  :character
Booking
Length:30
Class :character
Mode  :character
```

Untuk interpretasi hasil dari Analisis Deskriptif dalam data hotel surabaya ini sebagai berikut :

- Min : menunjukkan banyaknya data minimum atau data paling rendah dalam variable
- 1<sup>st</sup>Qu : menunjukkan bahwa kuartil 1 dalam variabel
- Median : menunjukkan bahwa nilai tengah dalam variabel
- Mean : menunjukkan bahwa rata-rata dalam variabel
- 3<sup>rd</sup>Qu : menunjukkan bahwa kuartil 3 dalam variabel
- Max : menunjukkan bahwa banyaknya data maksimum dalam variable

#### 4. Korelasi antara dua variable

Untuk menghitung nilai koefisien antara dua variable continue pada data tersebut menggunakan fungsi cor (Hotel\_surabaya)

```
84 #korelasi antara dua variabel
85 cor(Hotel_surabaya$Normal_Price, Hotel_surabaya$Net_Price)
86
```

Tampilan Output menggunakan sintaks pada gambar diatas seperti berikut :

```
> #korelasi antara dua variabel
> cor(Hotel_surabaya$Normal_Price, Hotel_surabaya$Net_Price)
[1] 0.8605636
```

Dapat kita lihat pada gambar diatas bahwa hasil perhitungan diperoleh nilai Korelasi sebesar 0.8605636. nilai tersebut menunjukkan Korelasi antara variable Normal Price vs Net Price.

#### 5. Menguji nilai signifikan

Untuk menguji apakah nilai hasil perhitungan Korelasi tersebut signifikan dan sekaligus hasil uji statistic, maka menggunakan sintaks cor.test(Hotel\_surabaya)

```
87 #menghitung nilai koefisien korelasi
88 cor.test(Hotel_surabaya$Normal_Price, Hotel_surabaya$Net_Price)
89
```

Tampilan Output menggunakan sintaks pada gambar diatas seperti berikut :

```
> #menguji nilai signifikan
> cor.test(Hotel_surabaya$Normal_Price, Hotel_surabaya$Net_Price)

Pearson's product-moment correlation

data: Hotel_surabaya$Normal_Price and Hotel_surabaya$Net_Price
t = 8.9403, df = 28, p-value = 1.074e-09
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.7251007 0.9319089
sample estimates:
      cor 
0.8605636
```

Berdasarkan Output yang dihasilkan, metode Pearson menghasilkan Output berupa nilai t uji, derajat kebebasan, nilai p-value, rentang estimasi nilai korelasi berdasarkan tingkat kepercayaan, dan estimasi nilai korelasi. Berdasarkan hasil uji tersebut nilai confidence interval koefisien Korelasi yang memiliki nilai Korelasi positif, maka rentang tersebut menunjukkan **nilai Korelasi yang signifikan**.

## 6. Summarytools

Package **Summarytools** digunakan untuk menganalisis dokumen yang diberikan dan meringkas data sehingga akurat, dan dapat mudah dipahami dari suatu dokumen yang panjang.

**Descr** merupakan function untuk menampilkan banyaknya data(n), rata-rata (mean), median, minimum, maksimum, range, standar deviasi (sd), kurtosis, dan kemiringan (skew), dll.

```
90 #ringkasan data menggunakan summarytools
91 library(summarytools)
92 Hotel_surabaya = as.data.frame(Hotel_surabaya)
93 descr(Hotel_surabaya)
```

Gambar dibawah merupakan tampilan output dari package **summarytools** dan **descr**

```
> #ringkasan data menggunakan summarytools
> library(summarytools)
> Hotel_surabaya = as.data.frame(Hotel_surabaya)
> descr(Hotel_surabaya)
Non-numerical variable(s) ignored: Hotel_Surabaya, Discount, Tax, Booking
Descriptive Statistics
Hotel_surabaya
N: 30
```

	Net_Price	Normal_Price	Rating
Mean	349.26	458.90	8.64
Std.Dev	128.61	152.58	0.43
Min	150.00	219.21	7.40
Q1	254.64	343.70	8.40
Median	339.67	448.48	8.70
Q3	435.53	568.18	8.80
Max	710.88	748.29	9.50
MAD	134.09	166.41	0.44
IQR	170.85	213.09	0.40
CV	0.37	0.33	0.05
Skewness	0.63	0.25	-0.52
SE.Skewness	0.43	0.43	0.43
Kurtosis	0.17	-1.02	0.78
N.Valid	30.00	30.00	30.00
Pct.Valid	100.00	100.00	100.00

Untuk interpretasi hasil dari output gambar diatas sebagai berikut :

- MAD : rata – rata kesalahan mutlak atau absolut. Dari suatu data set adalah jarak rata – rata antara setiap nilai data dan rata – rata.
- IQR : merupakan hasil dari  $Q3 - Q1$
- CV : untuk mengevaluasi kinerja model. Jika koefisien variasi semakin kecil, datanya semakin seragam(homogen), namun jika koefisien variasinya semakin besar, datanya semakin heterogen.
- Skewness : ukuran tidak ketidaksimetrisan dalam penyebaran data. Skewness bernilai negatif berarti ekor distribusi berada disebelah kiri.
- Kurtosis : memberikan gambaran apakah distribusi data cenderung rata atau runcing. Kurtosis negative menandakan distribusi yang relative datar.
- N.Valid : unutm menunjukkan sejauh mana alat ukur yang digunakan dalam suatu mengukur apa yang diukur.

## 7. Mencari Range

Untuk mencari range pada data tersebut dengan variable normal\_price, maka menggunakan sintaks seperti berikut :

```
100 #Mencari Range
101 range.data <- function(normal_price){
102   max(normal_price) - min(normal_price)
103 }
104 range.data(normal_price)
```

Tampilan Output menggunakan sintaks pada gambar diatas seperti berikut :

```
> #Mencari Range
> range.data <- function(normal_price){
+   max(normal_price) - min(normal_price)
+ }
> range.data(normal_price)
[1] 529.082
> |
```

## 8. Mencari frekuensi dan presentase data

Untuk Mencari frekuensi dan presentase data menggunakan sintaks seperti berikut :

```
95 #Mencari frekuensi dan presentase data
96 table(Hotel_surabaya$NetPrice) -> new_table
97 new_table
98 prop.table(new_table)
```

Tampilan Output menggunakan sintaks pada gambar diatas seperti berikut :



```

> #Mencari frekuensi dan presentase data
> table(Hotel_surabaya$Net_Price) -> new_table
> new_table

 150    185 190.098 199.999 208.249 223.402 254.644 272.628    275 276.741    307.8 310.399 333.311 334.999
 1      1      1      2      1      1      1      1      1      1      1      1      1      1
344.35 364.8 375.001 384.75    389.5 398.524 413.352 435.53    456.93    460    460.75    485 496.155    580.06
 1      1      1      1      1      1      1      1      1      1      1      1      1      1
710.877
 1
> prop.table(new_table)

 150    185    190.098    199.999    208.249    223.402    254.644    272.628    275    276.741
0.03333333 0.03333333 0.03333333 0.06666667 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333
 307.8    310.399    333.311    334.999    344.35    364.8    375.001    384.75    389.5    398.524
0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333
 413.352    435.53    456.93    460    460.75    485    496.155    580.06    710.877
0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333

```

## C. VISUALISASI DATA

### 1. Boxplot

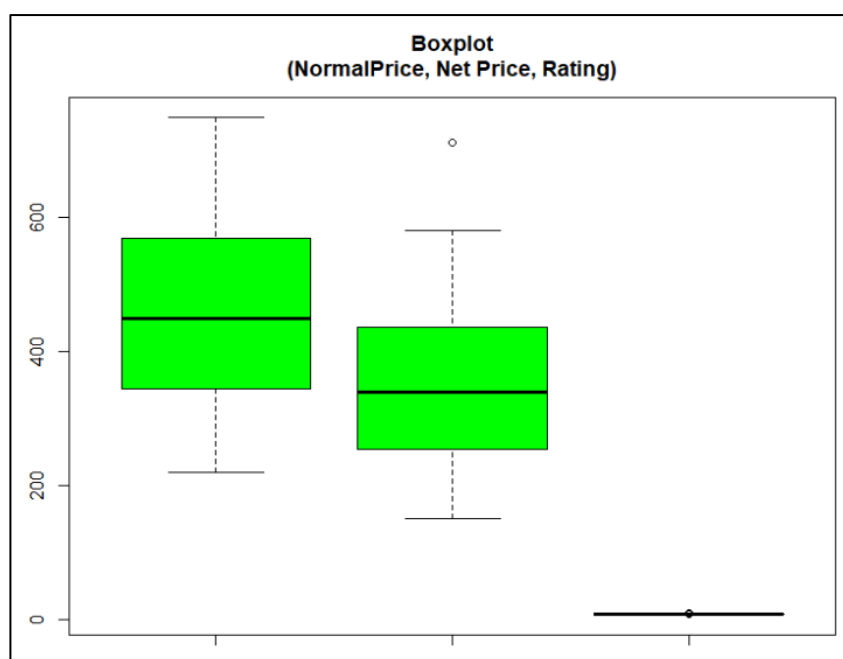
Pada visualisasi data, kita menggunakan function boxplot yang berguna untuk melihat distribusi data, ukuran pemusatan seperti median, kuartil, dan outlier. Disini kami menggunakan function boxplot untuk melihat gambar boxpot dari variabel **Normal\_price**, **Net\_Price**, dan **Rating**.

```

98 #Boxplot
99 boxplot = boxplot(Hotel_surabaya$Normal_Price, Hotel_surabaya$Net_Price,
100                  Hotel_surabaya$Rating, main = "Boxplot\n(NormalPrice, Net Price, Rating)", col = "green")
101

```

Gambar dibawah merupakan output dari boxplot, terlihat pada boxplot dibawah ini. Boxplot sebelah kiri merupakan boxplot dari variable Normal Price, boxplot yang berada di tengah merupakan boxplot dari variable Net Price, boxplot yang berada di sebelah kiri merupakan boxplot dari variable Rating.

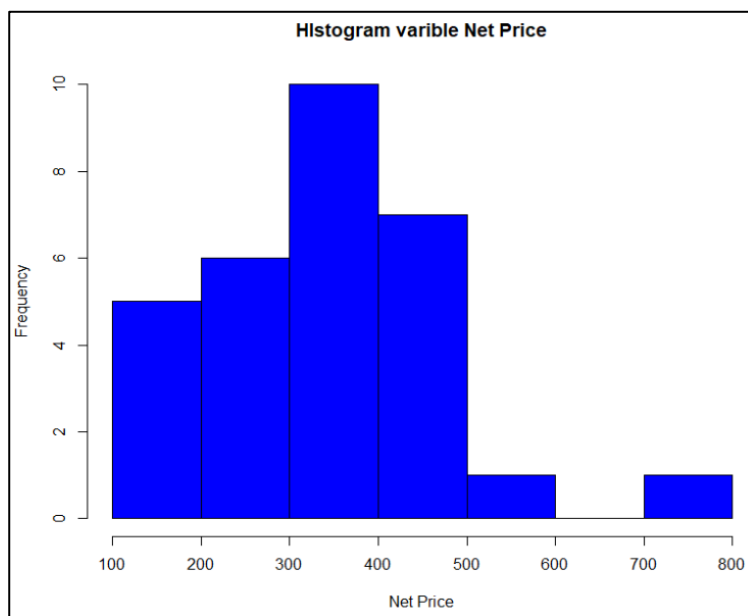


## 2. Histogram

Visualisasi data Histogram merupakan visualisasi table berupa batang dengan ukuran sesuai dengan jumlah numerik tertentu, disini kami menggunakan function histogram untuk melihat histogram dari variabel Net Price.

```
102 #Histogram
103 histogram = hist(x=Hotel_surabaya$Net_Price, xlab = "Net Price",
104                 main = "Histogram variable Net Price", col = "blue")
105
```

Berikut adalah output dari function histogram yang memberikan informasi ukuran pemusatan dan penyebaran data secara ringkas

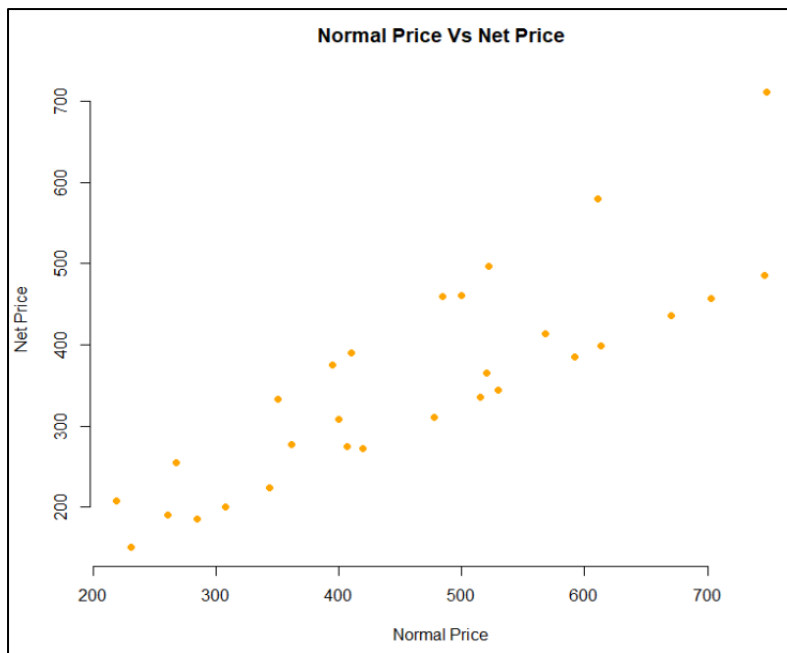


## 3. Plot

Pada visualisasi data selanjutnya kami menggunakan plot yaitu diagram pencar antara variabel Normal\_Price dan Net\_Price dengan menggunakan function `plot()`, function plot ini digunakan untuk mengidentifikasi pola asosiasi antara dua variabel continue

```
106 #plot
107 plot = plot(x=Hotel_surabaya$Normal_Price, y=Hotel_surabaya$Net_Price,
108            pch = 16, frame = FALSE, xlab = "Normal Price", ylab = "Net Price", col = "orange", main = "Normal Price Vs Net Price")
109
```

Output yang dihasilkan ditampilkan pada Gambar dibawah



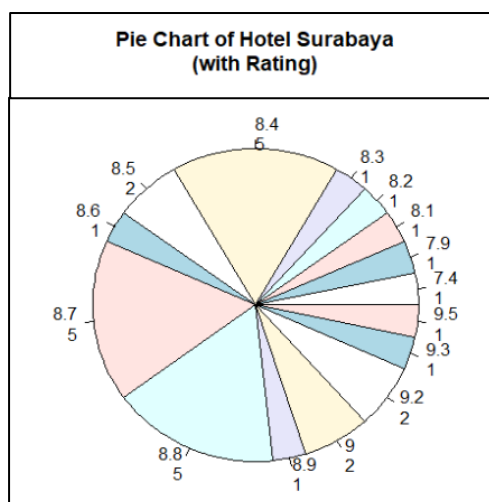
Berdasarkan gambar diatas terlihat bahwa korelasi yang terbentuk linier dan pola yang terbentuk begitu jelas terlihat. Hal ini mengidentifikasi bahwa ada korelasi antara kedua variable tersebut.

#### 4. Pie Chart

Diagram lingkaran (pie chart) merupakan sebuah grafik yang berbentuk lingkaran yang dibagi menjadi irisan-irisan untuk menggambarkan proporsi atau proporsi numerik. Untuk visualisasi Pie Chart ini kami menggunakan variabel Rating dengan menggunakan sintaks berikut :

```
110 #Pie Chart
111 piechart = table(Hotel_surabaya$Rating)
112 lbls = paste(names(piechart), "\n", piechart, sep="")
113 pie(piechart, labels = lbls, main = "Pie Chart of Hotel Surabaya\n(with Rating)")
```

Output yang dihasilkan ditampilkan pada Gambar dibawah

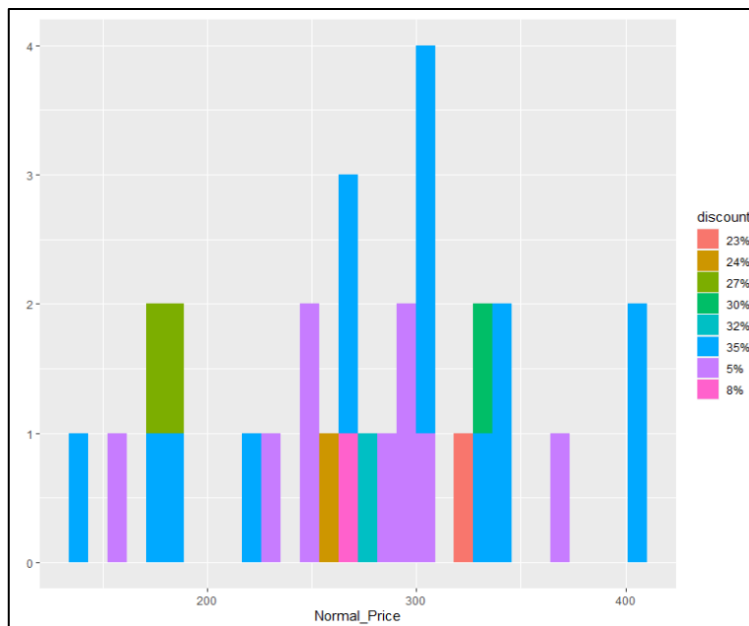


## 5. Qplot

Package ggplot2, menyediakan dua fungsi untuk membuat grammar graphics. qplot dirancang untuk membuat grafik lebih mudah dan cepat. Pada visualisasi qplot kita menggunakan variable Normal Price dengan menggunakan sintaks berikut :

```
115 #qplot
116 library(ggplot2)
117 qplot(data = Hotel_Surabaya, Normal_Price, fill = discount, bins = 30)
118
```

Output yang dihasilkan ditampilkan pada Gambar dibawah

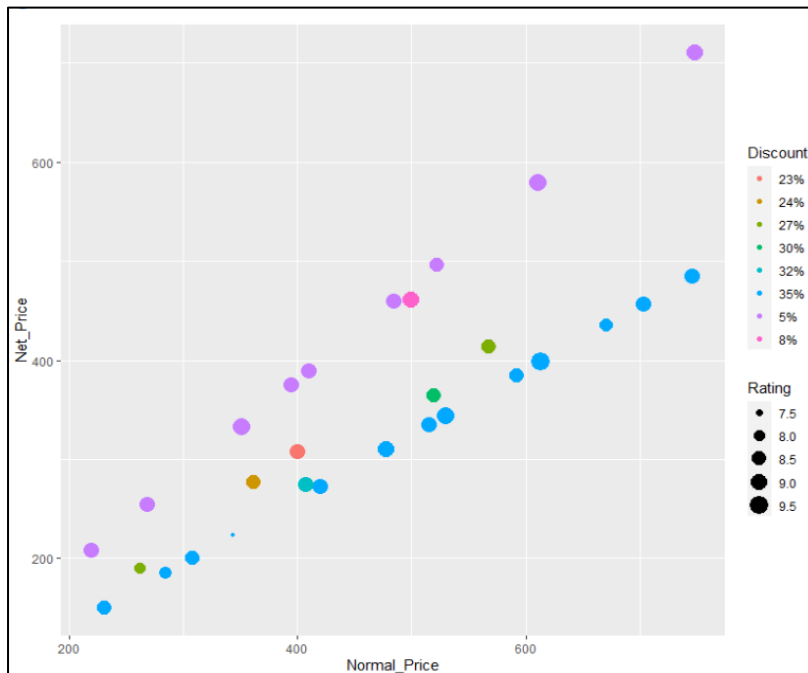


## 6. Ggplot2

Ggplot 2 merupakan package yang dikhususkan untuk visualisasi data. Dengan package ini kita bisa merancang grafik yang menggambarkan data melalui simbol, warna, bentuk, dan sebagainya. Pada visualisasi ggplot2 ini kita menggunakan variable normal price dan net price dengan menginstall package ggplot 2 terlebih dahulu dan menggunakan sintaks seperti berikut :

```
119 library(ggplot2)
120 p = ggplot(Hotel_surabaya, aes(x=Normal_Price, y=Net_Price))
121 p+geom_point(aes(size=Rating, col=Discount))
```

Output yang dihasilkan ditampilkan pada Gambar dibawah



Kita dapat menambahkan garis linier atas loess dengan menambahkan `geom_smooth()` dengan menggunakan sintaks seperti berikut :

```
122 #Menambah garis linier
123 p = ggplot(Hotel_surabaya,aes(x=Normal_Price,y=Net_Price))
124 p+geom_point(aes(size=Rating,col=Discount))+geom_smooth()
```

Output yang didapat ditampilkan seperti gambar dibawah ini

