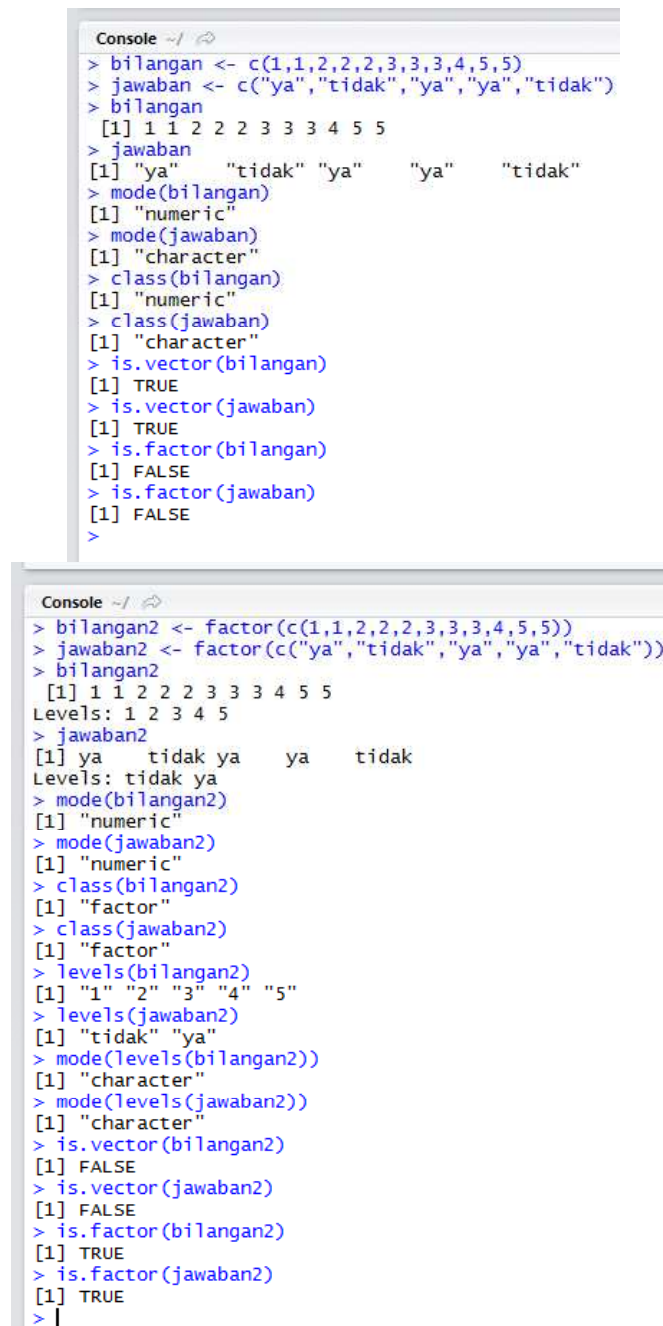


BAB 8

FACTOR

8.1 Factor

Gambar 8.1 diberikan ilustrasi dalam R penggunaan fungsi `factor()`.



```
Console ~/ |
> bilangan <- c(1,1,2,2,2,3,3,3,4,5,5)
> jawaban <- c("ya","tidak","ya","ya","tidak")
> bilangan
[1] 1 1 2 2 2 3 3 3 4 5 5
> jawaban
[1] "ya" "tidak" "ya" "ya" "tidak"
> mode(bilangan)
[1] "numeric"
> mode(jawaban)
[1] "character"
> class(bilangan)
[1] "numeric"
> class(jawaban)
[1] "character"
> is.vector(bilangan)
[1] TRUE
> is.vector(jawaban)
[1] TRUE
> is.factor(bilangan)
[1] FALSE
> is.factor(jawaban)
[1] FALSE
>

Console ~/ |
> bilangan2 <- factor(c(1,1,2,2,2,3,3,3,4,5,5))
> jawaban2 <- factor(c("ya","tidak","ya","ya","tidak"))
> bilangan2
[1] 1 1 2 2 2 3 3 3 4 5 5
Levels: 1 2 3 4 5
> jawaban2
[1] ya tidak ya ya tidak
Levels: tidak ya
> mode(bilangan2)
[1] "numeric"
> mode(jawaban2)
[1] "numeric"
> class(bilangan2)
[1] "factor"
> class(jawaban2)
[1] "factor"
> levels(bilangan2)
[1] "1" "2" "3" "4" "5"
> levels(jawaban2)
[1] "tidak" "ya"
> mode(levels(bilangan2))
[1] "character"
> mode(levels(jawaban2))
[1] "character"
> is.vector(bilangan2)
[1] FALSE
> is.vector(jawaban2)
[1] FALSE
> is.factor(bilangan2)
[1] TRUE
> is.factor(jawaban2)
[1] TRUE
> |
```

Gambar 8.1

Berdasarkan Gambar 8.1, dapat ditarik informasi sebagai berikut.

- ⇒ Dibentuk vektor bernama **bilangan** dan **jawaban**. Diketahui *mode* dari vektor **bilangan** adalah *numeric*, sementara *mode* dari vektor **jawaban** adalah *character*. Diketahui juga *class* dari vektor **bilangan** adalah *numeric*, sementara *class* dari vektor **jawaban** adalah *character*.
- ⇒ Selanjutnya digunakan fungsi *factor()* untuk membentuk objek *factor* bernama **bilangan2** dan **jawaban2**.
- ⇒ Perhatikan bahwa *mode* dari *factor* **bilangan2** adalah *numeric*, begitu juga *mode* dari *factor* **jawaban2** adalah *numeric* (meskipun data pada *factor* **jawaban2** adalah huruf/karakter (“ya”, “tidak”)). Diketahui juga *class* dari *factor* **bilangan2** adalah *factor*, dan *class* dari *factor* **jawaban2** juga *factor*.

```

Console ~| ↻
> is.vector(bilangan)
[1] TRUE
> is.vector(jawaban)
[1] TRUE
> is.factor(bilangan)
[1] FALSE
> is.factor(jawaban)
[1] FALSE
> is.vector(bilangan2)
[1] FALSE
> is.vector(jawaban2)
[1] FALSE
> is.factor(bilangan2)
[1] TRUE
> is.factor(jawaban2)
[1] TRUE

```

Gambar 8.2

- ⇒ Berdasarkan Gambar 8.2, perintah R *is.vector(bilangan)* dan *is.vector(jawaban)* menampilkan hasil TRUE, yang berarti **bilangan** dan **jawaban** adalah vektor. Perintah R *is.factor(bilangan)* dan *is.factor(jawaban)* menampilkan hasil FALSE, yang berarti **bilangan** dan **jawaban** bukan *factor*.
- ⇒ Perintah R *is.vector(bilangan2)* dan *is.vector(jawaban2)* menampilkan hasil FALSE, yang berarti **bilangan2** dan **jawaban2** bukan merupakan vektor. Perintah R *is.factor(bilangan2)* dan *is.factor(jawaban2)* menampilkan hasil TRUE, yang berarti **bilangan2** dan **jawaban2** adalah *factor*.

```

Console ~| ↻
> bilangan2
[1] 1 1 2 2 2 3 3 3 4 5 5
Levels: 1 2 3 4 5
> mode(bilangan2)
[1] "numeric"
> mode(levels(bilangan2))
[1] "character"

```

Gambar 8.3

Berdasarkan Gambar 8.3, *mode* dari nilai-nilai pada *factor* **bilangan2** adalah *numeric*, sementara *mode* dari *level-level* atau kategori-kategori pada *factor* **bilangan2** adalah *character*. Sebagaimana yang diutarakan oleh Phil Spector (2008:67) sebagai berikut.

“Factors in R are stored as a vector of integer values with a corresponding set of character values to use when the factor is displayed. The factor function is used to create a factor. The only required argument to factor is a vector of values which will be returned as a vector of factor values. Both numeric and character variables can be made into factors, but factor’s levels will always be character values. You can see the possible levels for a factor by calling the levels function; the nlevels function will return the number of levels of a factor.”

```
> bilangan2
[1] 1 1 2 2 2 3 3 3 4 5 5
Levels: 1 2 3 4 5
> mode(bilangan2)
[1] "numeric"
> mode(levels(bilangan2))
[1] "character"
> jawaban2
[1] ya      tidak ya      ya      tidak
Levels: tidak ya
> mode(jawaban2)
[1] "numeric"
> mode(levels(jawaban2))
[1] "character"
> levels(jawaban2)
[1] "tidak" "ya"
> |
```

Gambar 8.4

Berdasarkan Gambar 8.4, nilai-nilai dari *factor* **jawaban2** adalah ya, tidak, ya, ya, dan tidak, yang mana memiliki *mode numeric*. Sementara *level-level* dari *factor* **jawaban2** adalah ya dan tidak, yang mana berjenis *character*.

8.2 Mengganti Level dari Suatu Factor dengan Fungsi levels()

Perhatikan ilustrasi pada Gambar 8.5. Berdasarkan Gambar 8.5, digunakan fungsi *factor()* untuk membentuk *factor* bernama **fdata**. Diketahui *level* dari *factor* **fdata** pada awalnya adalah 1, 2, dan 3. Kemudian perintah R `levels(fdata) <- c("I","II","III")` mengganti *level* dari *factor* **fdata**, yang sebelumnya 1, 2, dan 3, menjadi I, II, dan III.

```

Console ~/ ↻
> data <- c(1,2,3,3,3,2,1,2,2,3,3,3,2,1,2)
> fdata <- factor(c(1,2,3,3,3,2,1,2,2,3,3,3,2,1,2))
> fdata
[1] 1 2 3 3 3 2 1 2 2 3 3 3 2 1 2
Levels: 1 2 3
> levels(fdata) <- c("I","II","III")
> fdata
[1] I   II  III III III II  I   II  II  III III III II  I   II
Levels: I II III
> |

```

Gambar 8.5

Pada Gambar 8.6, melibatkan argumen *levels* pada pembentukan *factor* yang bernama **fbulan**.

```

Console ~/ ↻
> bulan <- c("Februari","Februari","Maret","Juni","Januari","Januari","S
eptember","Oktober","Oktober","Maret","Desember","Desember","September",
"Desember","Juni","Juli","Mei","Juli")
> fbulan <- factor(bulan)
> table(fbulan)
fbulan
Desember  Februari  Januari      Juli      Juni      Maret
          3          2          2          2          2          2
      Mei  Oktober  September
          1          2          2
> fbulan
[1] Februari Februari Maret      Juni      Januari  Januari
[7] september Oktober  Oktober  Maret    Desember Desember
[13] september Desember Juni      Juli      Mei      Juli
9 Levels: Desember Februari Januari Juli Juni Maret ... september
>
> fbulan <- factor(bulan, levels = c("Januari","Februari","Maret","April
","Mei","Juni","Juli","Agustus","September","Oktober","November","Desemb
er"))
> table(fbulan)
fbulan
Januari  Februari  Maret  April  Mei  Juni
          2          2          2          0          1          2
      Juli  Agustus  September  Oktober  November  Desember
          2          0          2          2          0          3
>

```

Gambar 8.6

BAB 9

FUNGSI TABLE DAN FTABLE

9.1 Membuat Tabel Distribusi Frekuensi dengan Fungsi `table()`

Fungsi `table()` dapat digunakan untuk membuat tabel distribusi frekuensi. Perhatikan Gambar 9.1.

```
Console -/ ↻
> bilangan <- c(1,2,3,2,3,3,4,5,3,2,3,4,5,5,5,5,3,2,1,3)
> length(bilangan)
[1] 20
> table(bilangan)
bilangan
 1  2  3  4  5
 2  4  7  2  5
> |
```

Gambar 9.1

Berdasarkan Gambar 9.1 dibentuk vektor bernama **bilangan** yang terdiri dari 20 bilangan. Kemudian fungsi `table()` digunakan untuk membuat distribusi frekuensi berdasarkan bilangan-bilangan pada vektor **bilangan**. Diketahui nilai 1 terdapat sebanyak 2, nilai 2 terdapat sebanyak 4, nilai 3 terdapat sebanyak 7, dan seterusnya.

```
Console -/ ↻
> jawaban <- c("ya","tidak","tidak","tidak","ya")
> length(jawaban)
[1] 5
> table(jawaban)
jawaban
tidak   ya
      3   2
> |
```

Gambar 9.2

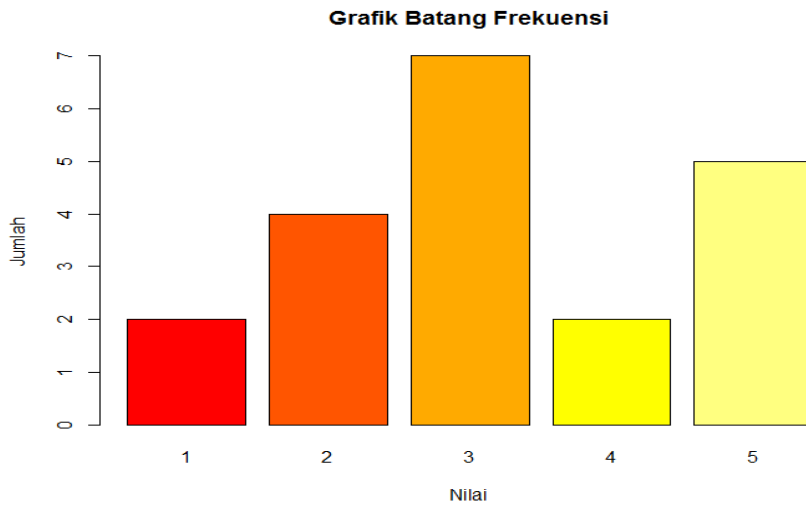
Berdasarkan Gambar 9.2 dibentuk vektor bernama **jawaban** yang terdiri dari 5 elemen. Kemudian fungsi `table()` digunakan untuk membuat distribusi frekuensi berdasarkan elemen-elemen pada vektor **jawaban**. Diketahui jawaban “ya” sebanyak 2, sementara jawaban “tidak” sebanyak 3.

Gambar 9.3 dan 9.4 digunakan fungsi `barplot()` untuk membuat grafik batang frekuensi. Perbedaan grafik batang antara Gambar 9.3 dan Gambar 9.4 adalah pada Gambar 9.3, tidak menyajikan nilai frekuensi pada tiap-tiap batang, sementara pada Gambar 9.4 menyajikan nilai frekuensi pada tiap-tiap batang. Berdasarkan Gambar 9.4, nilai 1 sebanyak 2, nilai 2 sebanyak 4, nilai 3 sebanyak 7, dan seterusnya.

```

Console ~/
> bilangan <- c(1,2,3,2,3,3,4,5,3,2,3,4,5,5,5,3,2,1,3)
> length(bilangan)
[1] 20
> table(bilangan)
bilangan
1 2 3 4 5
2 4 7 2 5
> tabel <- table(bilangan)
> barplot(tabel, col = c(heat.colors(5)), main="Grafik Batang Frekuensi", xlab="Nilai", ylab = "Jumlah")
>

```



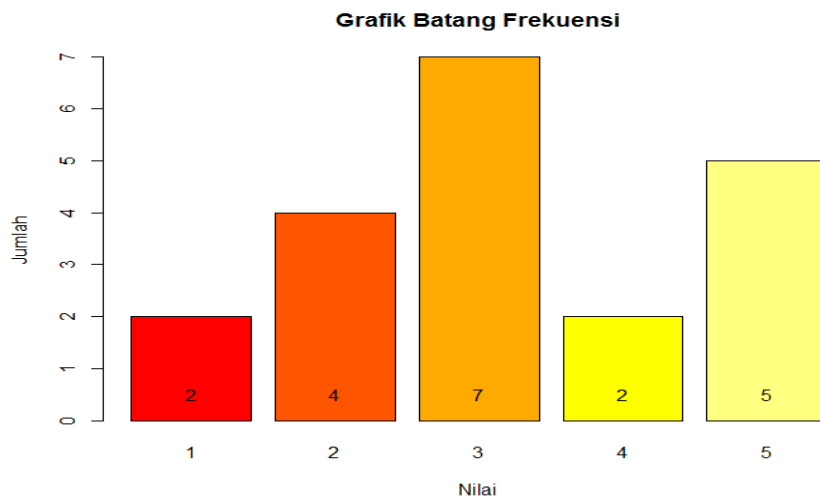
Gambar 9.3

Pada Gambar 9.3, argumen *main* bertujuan untuk memberi judul dari grafik batang, argumen *xlab* untuk memberi nama pada sumbu *x*, dan argumen *ylab* untuk memberi nama pada sumbu *y*. Argumen *col* bertujuan untuk memberi warna pada grafik batang.

```

Console ~/
> A <- barplot(tabel, col = c(heat.colors(5)), main="Grafik Batang Frekuensi", xlab="Nilai", ylab = "Jumlah")
> text(A, 0.5, tabel)
>

```



Gambar 9.4

9.2 Membuat Tabel Distribusi Frekuensi untuk Data Berkelompok

Gambar 9.5 diberikan ilustrasi dalam R membuat tabel distribusi frekuensi untuk data berkelompok. Pada ilustrasi tersebut digunakan fungsi `cut()` untuk membuat suatu interval. Argumen `break` digunakan untuk menentukan batas-batas interval. Diketahui terdapat 7 nilai yang termasuk ke dalam interval $1 \leq \text{bilangan} < 5$, sementara terdapat 9 nilai yang termasuk ke dalam interval $5 \leq \text{bilangan} \leq 10$.

```
Console -1
> bilangan <- c(1,2,3,4,5,6,7,8,9,10,10,9,8,4,3,2)
> distribusi_frekuensi <- cut(bilangan, breaks = c(1,5,10), include.lowest = TRUE, right = FALSE)
> tabel <- table(distribusi_frekuensi)
> tabel
distribusi_frekuensi
 [1,5) [5,10]
      7      9
> names(tabel)
[1] "[1,5)" "[5,10]"
> names(tabel) <- c("1<=x<5", "5<=x<=10")
> tabel
1<=x<5 5<=x<=10
      7      9
.
```

Gambar 9.5

9.3 Membuat Tabel Contingency Dua Arah dengan Fungsi `table()`

Misalkan dibentuk *data frame* dengan nama `data_frame`, seperti pada Gambar 9.6.

```
Console -1
> pendidikan <- c("s1","s1","s1","s1","s1","s2","s2","s2","s2","s2")
> jenis_kelamin <- c("laki-laki","laki-laki","laki-laki","perempuan","perempuan","perempuan","perempuan","perempuan","laki-laki")
> data_frame <- data.frame(pendidikan,jenis_kelamin)
> data_frame
  pendidikan jenis_kelamin
1          s1      laki-laki
2          s1      laki-laki
3          s1      laki-laki
4          s1      perempuan
5          s1      perempuan
6          s2      perempuan
7          s2      perempuan
8          s2      perempuan
9          s2      perempuan
10         s2      laki-laki
> table(data_frame)
      pendidikan jenis_kelamin
      s1 laki-laki perempuan
      s2          3          2
      s2          1          4
> mode(table(data_frame))
[1] "numeric"
> class(table(data_frame))
[1] "table"
> simpan <- table(data_frame)
> simpan
      pendidikan jenis_kelamin
      s1 laki-laki perempuan
      s2          3          2
      s2          1          4
> simpan[,1]
s1 s2
3  1
> simpan[1,]
laki-laki perempuan
      3          2
.
```

Digunakan fungsi `table()` pada *data frame* `data_frame`, sehingga menampilkan tabel *contingency*.

Gambar 9.6

Berdasarkan Gambar 9.6, digunakan fungsi `table()` pada *data frame* `data_frame`, sehingga menampilkan tabel *contingency*. Berdasarkan tabel *contingency* tersebut, diketahui:

- ⇒ Terdapat sebanyak 3 responden yang berpendidikan S1, dengan jenis kelamin laki-laki.
- ⇒ Terdapat sebanyak 2 responden yang berpendidikan S2, dengan jenis kelamin perempuan.
- ⇒ Terdapat sebanyak 1 responden yang berpendidikan S2, dengan jenis kelamin laki-laki.
- ⇒ Terdapat sebanyak 4 responden yang berpendidikan S2, dengan jenis kelamin perempuan.

Berdasarkan Gambar 9.6, perintah R `class(table(data_frame))` memperlihatkan hasil *table*.

Gambar 9.7 hingga Gambar 9.9 diperlihatkan langkah demi langkah untuk menyajikan grafik batang frekuensi. Fungsi `barplot()` digunakan untuk membuat grafik batang frekuensi.

```

Console ~/
> pendidikan <- c("s1", "s1", "s1", "s1", "s1", "s2", "s2", "s2", "s2", "s2")
> jenis_kelamin <- c("laki-laki", "laki-laki", "laki-laki", "perempuan",
"perempuan", "perempuan", "perempuan", "perempuan", "laki-laki")
> data_frame <- data.frame(pendidikan, jenis_kelamin)
> data_frame
  pendidikan jenis_kelamin
1         s1     laki-laki
2         s1     laki-laki
3         s1     laki-laki
4         s1     perempuan
5         s1     perempuan
6         s2     perempuan
7         s2     perempuan
8         s2     perempuan
9         s2     perempuan
10        s2     laki-laki
> tabel_kontingensi <- table(data_frame)
> tabel_kontingensi
      jenis_kelamin
pendidikan laki-laki perempuan
s1          3          2
s2          1          4
> |

```

```

Console ~/
> tabel_kontingensi
      jenis_kelamin
pendidikan laki-laki perempuan
s1          3          2
s2          1          4
> as.data.frame(tabel_kontingensi)
  pendidikan jenis_kelamin Freq
1         s1     laki-laki    3
2         s2     laki-laki    1
3         s1     perempuan    2
4         s2     perempuan    4
> frekuensi <- as.data.frame(tabel_kontingensi)$Freq
> frekuensi
[1] 3 1 2 4
> |

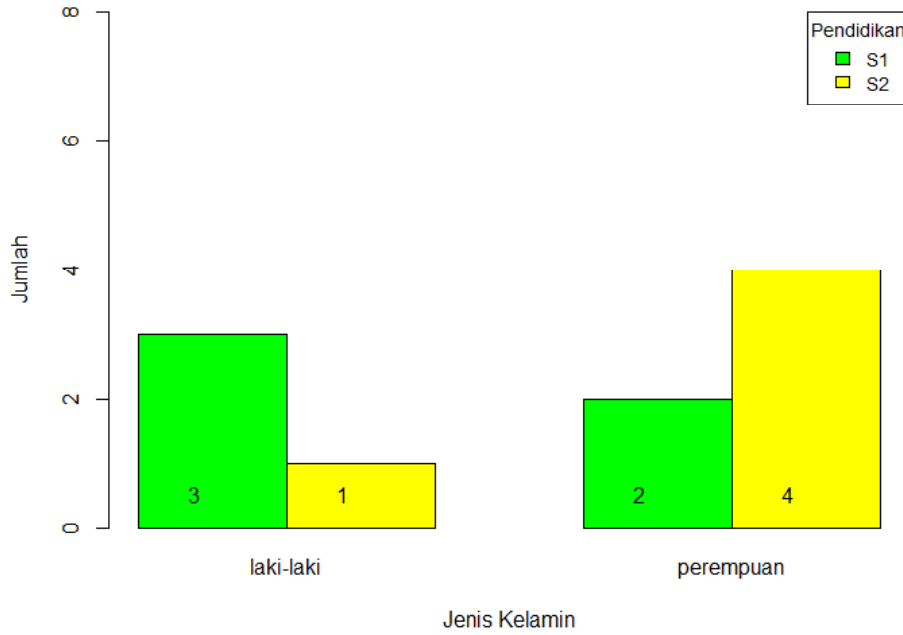
```

```

Console ~/
> A <- barplot(tabel_kontingensi, main="Grafik Batang Frekuensi berdasarkan P
endidikan dan Jenis Kelamin", xlab="Jenis Kelamin", ylab="Jumlah", col = c("g
reen", "yellow"), legend=row.names(tabel_kontingensi), args.legend = list(cex=.
9, x="topright", title="Pendidikan"), ylim=c(0,8), beside = TRUE )
> text(A,0.5, frekuensi, cex=1, pos=2)
> |

```

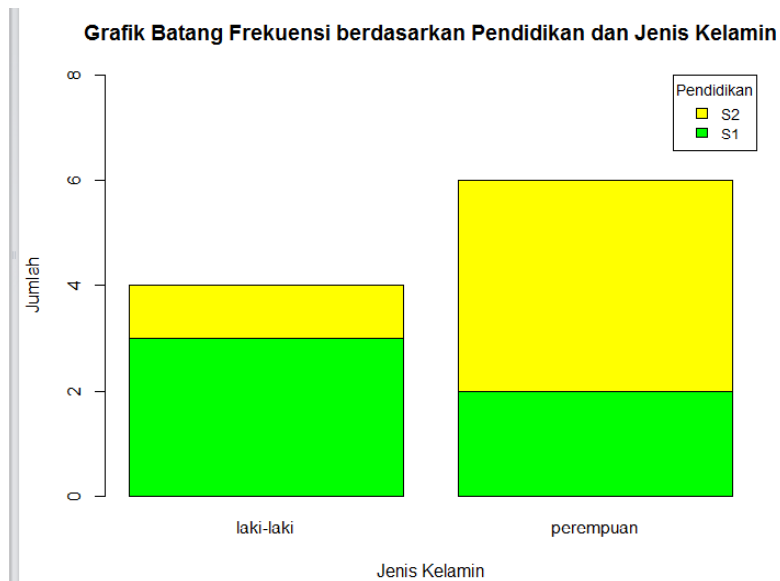

Grafik Batang Frekuensi berdasarkan Pendidikan dan Jenis Kelamin



Gambar 9.7

```

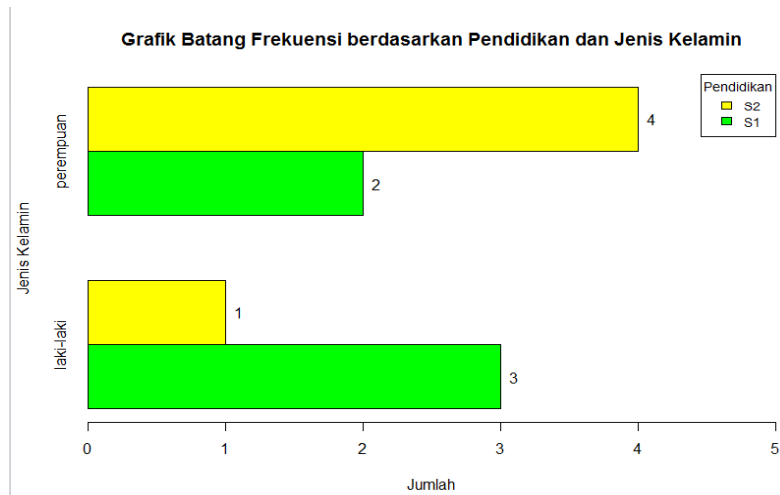
Console ~1
> barplot(tabel_kontingensi, main="Grafik Batang Frekuensi berdasarkan Pendidikan dan Jenis Kelamin", xlab="Jenis Kelamin", ylab="Jumlah", col = c("green", "yellow"), legend=rownames(tabel_kontingensi), args.legend = list(cex=.9, x="topright", title="Pendidikan"), ylim=c(0,8) )
>
    
```



Gambar 9.8

```

Console ~1
> A <- barplot(tabel_kontingensi, main="Grafik Batang Frekuensi berdasarkan Pendidikan dan Jenis kelamin", xlab="Jumlah", ylab="Jenis Kelamin", col=c("green","yellow"), legend=rownames(tabel_kontingensi), args.legend=list(cex=.9, x="topright", title="Pendidikan"), xlim=c(0,5), beside = TRUE, horiz = TRUE)
> text(x=tabel_kontingensi+0.1, y=A, labels = as.character(tabel_kontingensi))
>
    
```



Gambar 9.9

9.4 Membuat Tabel *Contingency* Tiga Arah dengan Fungsi *table()*

Perhatikan Gambar 9.10. Pada Gambar 9.10 digunakan fungsi *ftable()* untuk membentuk tabel *contingency* tiga arah dari *data frame* `data_frame`. Berdasarkan tabel *contingency* tiga arah tersebut, diketahui:

- ⇒ Terdapat sebanyak 1 responden dengan pendidikan S1, jenis kelamin laki-laki, dan belum menikah.
- ⇒ Terdapat sebanyak 2 responden dengan pendidikan S1, jenis kelamin laki-laki, dan sudah menikah, dan seterusnya.

```

Console - / ?
> data_frame
  pendidikan jenis_kelamin
1          S1 laki-laki
2          S1 laki-laki
3          S1 laki-laki
4          S1 perempuan
5          S1 perempuan
6          S2 perempuan
7          S2 perempuan
8          S2 perempuan
9          S2 perempuan
10         S2 laki-laki
> status <- c("sudah menikah","sudah menikah","belum menikah","sudah meni
kah","sudah menikah","belum menikah","sudah menikah","belum menikah","bel
um menikah","sudah menikah")
> data_frame <- data.frame(data_frame,status)
> data_frame
  pendidikan jenis_kelamin      status
1          S1 laki-laki sudah menikah
2          S1 laki-laki sudah menikah
3          S1 laki-laki belum menikah
4          S1 perempuan sudah menikah
5          S1 perempuan sudah menikah
6          S2 perempuan belum menikah
7          S2 perempuan sudah menikah
8          S2 perempuan belum menikah
9          S2 perempuan belum menikah
10         S2 laki-laki sudah menikah
> ftable(data_frame)
      status belum menikah sudah menikah
pendidikan jenis_kelamin
S1 laki-laki          1          2
   perempuan          0          2
S2 laki-laki          0          1
   perempuan          3          1
> |

```

Gambar 9.10

9.5 Membuat Tabel *Contingency* Empat Arah dengan Fungsi *table()*

Perhatikan Gambar 9.11. Pada Gambar 9.11 digunakan fungsi *ftable()* untuk membentuk tabel *contingency* empat arah dari *data frame* `data_frame`. Berdasarkan tabel *contingency* tiga empat tersebut, diketahui:

- ⇒ Tidak terdapat responden dengan jenis kelamin laki-laki, pendidikan S1, status belum menikah, dan hobi memasak.
- ⇒ Terdapat satu responden dengan jenis kelamin laki-laki, pendidikan S1, status belum menikah, dan hobi membaca, dan seterusnya.

```
Console -/
> pendidikan <- c("s1","s1","s1","s1","s1","s2","s2","s2","s2","s2")
> hobi <- c("membaca","membaca","membaca","membaca","memasak","membaca","membaca","memasak","membaca","memasak")
> jenis_kelamin <- c("laki-laki","laki-laki","laki-laki","perempuan","perempuan","perempuan","perempuan","perempuan","laki-laki")
> status <- c("sudah menikah","sudah menikah","belum menikah","sudah menikah","sudah menikah","belum menikah","sudah menikah","belum menikah","sudah menikah","sudah menikah")
>
> data_frame <- data.frame(jenis_kelamin,pendidikan,status,hobi)
> data_frame
  jenis_kelamin pendidikan      status      hobi
1     laki-laki         S1 sudah menikah membaca
2     laki-laki         S1 sudah menikah membaca
3     laki-laki         S1 belum menikah membaca
4     perempuan         S1 sudah menikah membaca
5     perempuan         S1 sudah menikah memasak
6     perempuan         S2 belum menikah membaca
7     perempuan         S2 sudah menikah membaca
8     perempuan         S2 belum menikah memasak
9     perempuan         S2 belum menikah membaca
10    laki-laki         S2 sudah menikah memasak
> ftable(data_frame)
      hobi memasak membaca
jenis_kelamin pendidikan status
laki-laki      S1      belum menikah      0      1
                S1      sudah menikah      0      2
                S2      belum menikah      0      0
                S2      sudah menikah      1      0
perempuan     S1      belum menikah      0      0
                S1      sudah menikah      1      1
                S2      belum menikah      1      2
                S2      sudah menikah      0      1
> |
```

Gambar 9.11

9.6 Menghitung beberapa Ukuran Statistik berdasarkan Kelompok

Gambar 9.12 diberikan ilustrasi dalam R untuk menghitung nilai rata-rata, minimum, dan maksimum dari data gaji, berdasarkan tingkat pendidikan. Berdasarkan Gambar 9.12 digunakan fungsi *subset()* untuk melakukan perhitungan berdasarkan kelompok. Diketahui rata-rata gaji pada kelompok responden dengan pendidikan S1 adalah 4,74 juta, sementara rata-rata gaji pada kelompok responden dengan pendidikan S2 adalah 7,5 juta. Diketahui juga gaji minimum pada responden dengan pendidikan S1 adalah 3,2 juta, sementara gaji maksimum pada responden dengan pendidikan S2 adalah 9.9 juta.

```

> data_frame
pendidikan gaji
1          S1  4.6
2          S1  5.5
3          S1  3.2
4          S1  5.5
5          S1  4.9
6          S2  5.9
7          S2  8.7
8          S2  6.5
9          S2  6.5
10         S2  9.9
> mean(subset(data_frame, pendidikan=="S1")$gaji )
[1] 4.74
> mean(subset(data_frame, pendidikan=="S2")$gaji )
[1] 7.5
> min(subset(data_frame, pendidikan=="S1")$gaji )
[1] 3.2
> max(subset(data_frame, pendidikan=="S2")$gaji )
[1] 9.9

```

Gambar 9.12

9.7 Menghitung beberapa Ukuran Statistik berdasarkan Kelompok dengan *Package dplyr*

Gambar 9.13 diberikan ilustrasi dalam R untuk menghitung nilai rata-rata, minimum, dan maksimum dari variabel gaji dan pengeluaran berdasarkan variabel kategori jenis kelamin dan pendidikan. **Perhatikan bahwa** perhitungan tersebut menggunakan fungsi *summarise()* yang mana terlebih dahulu memasang (*attach*) *package dplyr*.

```

> data_frame
pendidikan jenis_kelamin gaji pengeluaran
1          S1 laki-laki 4.6 3.3
2          S1 laki-laki 5.5 3.5
3          S1 laki-laki 3.2 2.8
4          S1 perempuan 5.5 3.1
5          S1 perempuan 4.9 3.3
6          S2 perempuan 5.9 4.0
7          S2 perempuan 8.7 4.1
8          S2 perempuan 6.5 3.9
9          S2 laki-laki 6.5 4.4
10         S2 laki-laki 9.9 3.5
> library(dplyr)
> library(dplyr)
> grup <- group_by(data_frame, pendidikan, jenis_kelamin)
> grup %>% summarise(
+   minimum_gaji = min(gaji),
+   minimum_pengeluaran = min(pengeluaran),
+   maksimum_gaji = max(gaji),
+   maksimum_pengeluaran = max(pengeluaran),
+   rata_rata_gaji = mean(gaji),
+   rata_rata_pengeluaran = mean(pengeluaran))
Source: local data frame [5 x 8]
Groups: pendidikan [?]
  pendidikan jenis_kelamin minimum_gaji minimum_pengeluaran maksimum_gaji maksimum_pengeluaran rata_rata_gaji rata_rata_pengeluaran
  <fctr>      <fctr>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
1          S1 laki-laki 3.2 2.8 5.5 3.5 4.433333 3.333333
2          S1 perempuan 4.9 3.1 5.5 3.3 5.200000 3.300000
3          S2 laki-laki 9.9 3.5 9.9 3.5 9.900000 3.500000
4          S2 laki-laki 6.5 4.4 6.5 4.4 6.500000 4.400000
5          S2 perempuan 5.9 3.9 8.7 4.1 7.033333 4.033333

```

Perhatikan bahwa perhitungan tersebut menggunakan fungsi *summarise()* yang mana terlebih dahulu memasang (*attach*) *package dplyr*

Gambar 9.13