



List dalam
Mathematica

N. K. Kutha Ardana

*Departemen Matematika
FMIPA – IPB University*

List dalam *Mathematica*

Ngakan Komang Kutha Ardana

Departemen Matematika, FMIPA - IPB University

Juni 2023

Ringkasan

List merupakan salah satu struktur data utama dalam Mathematica yang terdiri dari kumpulan elemen terurut. Masing-masing elemen dapat berupa angka, himpunan bagian list, dll. List biasanya dilambangkan dengan $\{a_1, a_2, \dots, a_n\}$ yang dapat dipandang sebagai vektor (khususnya, n -vektor) atau n -tuple. Berbagai konsep dasar list dipaparkan pada artikel ini. Pemahaman yang baik tentang list amat menunjang implementasi Mathematica sebagai mesin komputasi, khususnya dalam pengembangan pemrograman fungsional.

Pendahuluan

OpenAI yang dikenal sebagai pionir pengembang ChatGPT dalam versi Pro telah menggandeng WolframAlpha sebagai *plugins* komputasinya sehingga akan menghasilkan tanggapan yang makin akurat dalam hal melakukan komputasi, baik secara numerik maupun simbolik. Wolfram Alpha adalah mesin komputasi *online* yang menggunakan teknologi *Mathematica* untuk memungkinkan semua pengetahuan sistematis dapat dihitung. Salah satu struktur data dasar yang amat penting pada Mathematica adalah pemahaman tentang list.

Pada uraian berikutnya akan dipaparkan berbagai konsep penting list seperti: pengertian, pembangkitan serta berbagai manipulasinya. Contoh-contoh lainnya dapat mengacu misalnya pada (Ruskeepää, 2009), (Abel dan Braselton, 2022). Mathematica sendiri juga menyediakan Wolfram Documentation pada menu Help, terutama pada bagian List. Perintah-perintah pada artikel ini telah dicoba pada Mathematica versi 13.2.

Pengertian List

List merupakan struktur data dasar dalam *Mathematica* yang berbentuk:

$$\{\text{elemen-1}, \text{elemen-2}, \dots, \text{elemen-n}\}$$

Elemen suatu list dapat berupa bilangan, variabel, string, fungsi, ekspresi, atau list lainnya. Perhatikan bahwa elemen-elemen list dipisahkan oleh koma dan diapit dengan $\{\dots\}$, seperti penulisan himpunan, namun suatu list dapat mempunyai lebih dari satu elemen yang sama. Sebuah list dapat dipandang sebagai suatu vektor (list berdimensi satu), matriks (list berdimensi dua), atau tensor (list berdimensi banyak).

Pembangkitan List

Sebuah list dapat dibangkitkan langsung dengan menyenaraikan elemen-elemennya.

Berikut adalah sebuah list dengan tiga elemen:

```
In[1]:= list1 = {N[E], b, Sqrt[25]}
```

```
Out[1]= {2.71828, b, 5}
```

Elemen sebuah list dapat berupa list lainnya:

```
In[2]:= list2 = {list1, b, {c, d}}
```

```
Out[2]= {{2.71828, b, 5}, b, {c, d}}
```

Fungsi-fungsi listable

Pada dasarnya sebagian besar fungsi *built-in Mathematica* merupakan suatu fungsi yang *listable*. Artinya, fungsi-fungsi tersebut dapat langsung dioperasikan terhadap elemen-elemen suatu list. Hal ini akan sangat memudahkan operasi-operasi fungsi tersebut terhadap list.

Perintah berikut langsung mengevaluasi akar kuadrat dan logaritma asli dari elemen list

```
In[3]:= Sqrt[{a, b, c, d}]
```

```
Log[{a, b, c, d}]
```

```
Out[3]= {√a, √b, √c, √d}
```

```
Out[4]= {Log[a], Log[b], Log[c], Log[d]}
```

Perintah berikut langsung mengevaluasi operasi aritmatik pada elemen list

```
In[5]:= {a, b, c} ^ d
```

```
a / {b, c, d}
```

```
Out[5]= {ad, bd, cd}
```

```
Out[6]= { $\frac{a}{b}$ ,  $\frac{a}{c}$ ,  $\frac{a}{d}$ }
```

Table

Selain dengan cara menyenaraikan elemen-elemennya, list dapat juga dibangkitkan melalui beberapa perintah otomatis, seperti perintah **Table**, **Range**, **Array**, **SparseArray**.

Table merupakan perintah umum untuk menghasilkan list dengan sintaks sbb.

Sintaks

- `Table[expr, {imax}]` menghasilkan list *expr* sebanyak *imax*.
- `Table[expr, {i, imax}]` menghasilkan list *expr* dari 1 sampai *imax* step 1.
- `Table[expr, {i, imin, imax}]` mulai *imin* hingga *imax*, step 1.
- `Table[expr, {i, imin, imax, di}]` step *di*.
- `Table[expr, {i, imin, imax}, {j, jmin, jmax}, ...]` list tersarang.

Perhatikan contoh-contoh list berdimensi satu berikut:

```
In[7]:= Table[k2, {10}]
Table[k2, {k, 10}]
Table[k2, {k, 2, 10}]
Table[k2, {k, 2, 10, 2}]

Out[7]= {k2, k2, k2, k2, k2, k2, k2, k2, k2, k2}

Out[8]= {1, 4, 9, 16, 25, 36, 49, 64, 81, 100}

Out[9]= {4, 9, 16, 25, 36, 49, 64, 81, 100}

Out[10]= {4, 16, 36, 64, 100}
```

Berikut adalah list berdimensi 2 yang dapat dipandang sebagai matriks berukuran 2 x 3

```
In[11]:= Table[i - j, {i, 2}, {j, 3}]

Out[11]= {{0, -1, -2}, {1, 0, -1}}
```

Untuk menampilkan list dalam bentuk matriks, gunakan perintah `List//MatrixForm` atau `MatrixForm[List]`

```
In[12]:= MatrixForm[%]

Out[12]//MatrixForm=

$$\begin{pmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \end{pmatrix}$$

```

Berikut adalah list pasangan enam bilangan prima ke-2015 hingga bilangan prima ke-2025

```
In[13]:= prima = Table[{i, Prime[i]}, {i, 2015, 2025}]

Out[13]= {{2015, 17509}, {2016, 17519}, {2017, 17539}, {2018, 17551}, {2019, 17569}, {2020, 17573},
{2021, 17579}, {2022, 17581}, {2023, 17597}, {2024, 17599}, {2025, 17609}}
```

Perintah `List//TableForm` atau `TableForm[List]` menampilkan `List` dalam bentuk tabular

```
In[14]:= prima // TableForm

Out[14]//TableForm=
2015    17509
2016    17519
2017    17539
2018    17551
2019    17569
2020    17573
2021    17579
2022    17581
2023    17597
2024    17599
2025    17609
```

Perintah `Table` berikut masing-masing menghasilkan list lima bilangan acak bulat dan real dari -10 sampai 10

```
In[15]:= Table[Random[Integer, {-10, 10}], {5}]
Table[Random[Real, {-10, 10}], {5}]

Out[15]= {7, -6, -4, 6, -7}

Out[16]= {3.92993, -9.34788, 9.9607, -9.55781, -7.73227}
```

Argumen *expr* pada `Table` juga dapat berupa pernyataan kondisional seperti **If** atau **Which**. Elemen matriks berikut bernilai 1 jika indeks i dan j memenuhi kondisi $|i - j| \leq 2$, dan bernilai 0 untuk kondisi lainnya.

```
In[17]:= matrix1 = Table[If[Abs[i - j] ≤ 2, 1, 0], {i, 5}, {j, 5}];
MatrixForm[matrix1]
```

```
Out[18]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

```

Suatu list tidak harus berbentuk persegi panjang. Pada contoh berikut, batas atas loop dalam j bergantung pada nilai loop luar i

```
In[19]:= Table[i + j, {i, 1, 4}, {j, 1, i}]

Out[19]= {{2}, {3, 4}, {4, 5, 6}, {5, 6, 7, 8}}
```

sehingga menghasilkan suatu list segitiga.

```
In[20]:= TableForm[%]
```

```
Out[20]//TableForm=
2
3 4
4 5 6
5 6 7 8
```

Tak ada batas banyaknya dimensi pada suatu list. Berikut diberikan tensor berpangkat 4 (berdimensi 4) dengan menggunakan empat iterator pada perintah `Table`.

```
In[21]:= tensor = Table[i + j + k - l, {i, 2}, {j, 3}, {k, 4}, {l, 5}];
```

```
In[22]:= TableForm[tensor]
```

```
Out[22]//TableForm=
2 1 0 -1 -2    3 2 1 0 -1    4 3 2 1 0
3 2 1 0 -1    4 3 2 1 0    5 4 3 2 1
4 3 2 1 0     5 4 3 2 1    6 5 4 3 2
5 4 3 2 1     6 5 4 3 2    7 6 5 4 3
3 2 1 0 -1    4 3 2 1 0    5 4 3 2 1
4 3 2 1 0     5 4 3 2 1    6 5 4 3 2
5 4 3 2 1     6 5 4 3 2    7 6 5 4 3
6 5 4 3 2     7 6 5 4 3    8 7 6 5 4
```

Range

Cara lain untuk membuat list adalah dengan perintah **Range**.

Sintaks

- `Range[imax]` menghasilkan list $\{1, 2, \dots, imax\}$.
- `Range[imin, imax]` list $\{imin, \dots, imax\}$.
- `Range[imin, imax, di]` step di .

Perhatikan contoh-contoh berikut:

```
In[23]:= Range[5]
         Range[x, x + 5]
         Range[0, 1, 0.3]
         Sqrt[Range[5]] // N (* list akar lima bilangan asli pertama *)
         Null

Out[23]= {1, 2, 3, 4, 5}

Out[24]= {x, 1 + x, 2 + x, 3 + x, 4 + x, 5 + x}

Out[25]= {0., 0.3, 0.6, 0.9}

Out[26]= {1., 1.41421, 1.73205, 2., 2.23607}
```

Array

Sintaks

- `Array[f, n]` akan menghasilkan list dengan elemen $f[i]$, $i = 1, 2, \dots, n$.
- `Array[f, {n1, n2, ...}]` menghasilkan list tersarang $n_1 \times n_2 \times \dots$, dengan elemen $f[i_1, i_2, \dots]$.

Perhatikan contoh-contoh berikut:

```
In[28]:= Array[f, 3]
         Array[a, {3, 4}] // MatrixForm

Out[28]= {f[1], f[2], f[3]}

Out[29]//MatrixForm=

$$\begin{pmatrix} a[1, 1] & a[1, 2] & a[1, 3] & a[1, 4] \\ a[2, 1] & a[2, 2] & a[2, 3] & a[2, 4] \\ a[3, 1] & a[3, 2] & a[3, 3] & a[3, 4] \end{pmatrix}$$

```

Catatan:

- Saat ini Mathematica telah dilengkapi dengan struktur data terkini seperti **Dataset**, **Association** yang amat efisien digunakan dalam analisis data statistik.
- Berbagai perintah pertukaran data tersedia (**Export**, **Import**). Salah satu perintah “**SemanticImport**” dapat digunakan untuk mengimpor file secara semantik untuk memberikan objek **Dataset**.

In[30]:= **SemanticImport**["ExampleData/RetailSales.tsv"]

Out[30]=

Date	City	Sales
Wed 1 Jan 2014	Boston	198
Wed 1 Jan 2014	New York City	220
Wed 1 Jan 2014	Paris	215
Wed 1 Jan 2014	London	225
Wed 1 Jan 2014	Shanghai	241
Wed 1 Jan 2014	Tokyo	218
Thu 2 Jan 2014	Boston	189
Thu 2 Jan 2014	New York City	232
Thu 2 Jan 2014	Paris	211
Thu 2 Jan 2014	London	228
Thu 2 Jan 2014	Shanghai	242
Thu 2 Jan 2014	Tokyo	229
Fri 3 Jan 2014	Boston	196
Fri 3 Jan 2014	New York City	235
Fri 3 Jan 2014	Paris	221
Fri 3 Jan 2014	London	229
Fri 3 Jan 2014	Shanghai	238
Fri 3 Jan 2014	Tokyo	236
Sat 4 Jan 2014	Boston	194
Sat 4 Jan 2014	New York City	237


rows 1-20 of 600

SparseArray

Ada kalanya kita ingin membangkitkan array yang sebagian besar elemennya memiliki nilai tertentu, misalnya 0. Hanya sebagian kecil elemen memiliki nilai yang tak beraturan. Gunakan `SparseArray` untuk kasus seperti ini.

Berikut adalah vektor dengan 10 elemen, elemen ke-3 dan 10 masing-masing 2 dan 4, selainnya 0.

In[31]:= **SparseArray**[{3 → 2, 10 → 4}]

Out[31]= **SparseArray** [ Specified elements: 2
Dimensions: {10}]

Perintah **Normal** digunakan untuk melihat keseluruhan list hasil dari **SparseArray** sebelumnya (%)

```
In[32]:= Normal[%]
```

```
Out[32]= {0, 0, 2, 0, 0, 0, 0, 0, 0, 4}
```

Berikut adalah **SparseArray** berukuran 15, dengan unsur default x

```
In[33]:= Normal[SparseArray[{1→a, 14→b}, 15, x]]
```

```
Out[33]= {a, x, x, x, x, x, x, x, x, x, x, x, x, b, x}
```

Perintah berikut menghasilkan matriks 4 × 4, dengan unsur bilangan acak pada diagonal utama.

```
In[34]:= Normal[SparseArray[{{i_, i_} :> Random[]}, {4, 4}]]//MatrixForm
```

```
Out[34]//MatrixForm=
```

$$\begin{pmatrix} 0.211901 & 0 & 0 & 0 \\ 0 & 0.916806 & 0 & 0 \\ 0 & 0 & 0.38027 & 0 \\ 0 & 0 & 0 & 0.44641 \end{pmatrix}$$

Pembangkit matriks

Selain perintah-perintah **Table**, **Range**, atau **Array**, *Mathematica* menyediakan banyak pembangkit matriks khusus, seperti **IdentityMatrix** dan **DiagonalMatrix**, **UpperTriangularMatrix**, **HilbertMatrix**.

Perintah berikut membangkitkan matriks identitas 3 x 3

```
In[35]:= id3 = IdentityMatrix[5]
id3 // MatrixForm
```

```
Out[35]= {{1, 0, 0, 0, 0}, {0, 1, 0, 0, 0}, {0, 0, 1, 0, 0}, {0, 0, 0, 1, 0}, {0, 0, 0, 0, 1}}
```

```
Out[36]//MatrixForm=
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Perintah berikut membangkitkan matriks diagonal dengan elemen diagonal utama a, b, c, d

```
In[37]:= diag = DiagonalMatrix[{a, b, c, d}]
diag // MatrixForm
```


```
Out[37]= {{a, 0, 0, 0}, {0, b, 0, 0}, {0, 0, c, 0}, {0, 0, 0, d}}
```

```
Out[38]//MatrixForm=
```

$$\begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d \end{pmatrix}$$

Perintah berikut membangkitkan matriks segitiga atas berukuran 3 x 3

```
In[39]:= ut = UpperTriangularMatrix[{{1, 2, 0}, {0, 0, -1}, {0, 0, 2}}]
ut // MatrixForm
```

```
Out[39]= UpperTriangularMatrix[ Dimensions: {3, 3}]
```

```
Out[40]//MatrixForm=
```

$$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 2 \end{pmatrix}$$

Karakteristik List

Panjang list

Panjang list menunjukkan banyaknya elemen list tersebut, dinyatakan dengan **Length**

Perhatikan kembali **list1** dan **list2** pada contoh sebelumnya. Keduanya memiliki panjang yang sama.

```
In[41]:= list1
list2
{Length[list1], Length[list2]}
```

```
Out[41]= {2.71828, b, 5}
```

```
Out[42]= {{2.71828, b, 5}, b, {c, d}}
```

```
Out[43]= {3, 3}
```

Panjang list berdimensi dua (matriks) sama dengan banyaknya baris pada matriks tersebut. Perhatikan kembali **matrix1** pada contoh sebelumnya

```
In[44]:= matrix1
matrix1 // MatrixForm
Length[matrix1]
```

```
Out[44]= {{1, 1, 1, 0, 0}, {1, 1, 1, 1, 0}, {1, 1, 1, 1, 1}, {0, 1, 1, 1, 1}, {0, 0, 1, 1, 1}}
```

```
Out[45]//MatrixForm=
```

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

```
Out[46]= 5
```

Dimensi list

Dimensi vektor seperti **list1** dan **list2**, sama dengan panjang list tersebut, sedangkan dimensi matriks menyatakan banyaknya baris dan kolom matriks tersebut

```
In[47]:= Dimensions[list1]
Dimensions[list2]
Dimensions[matrix1] (* matrix1 adalah matriks berukuran 5 x 5 *)

Out[47]= {3}

Out[48]= {3}

Out[49]= {5, 5}
```

Posisi elemen tertentu

Posisi elemen tertentu pada suatu list dapat diperoleh dengan perintah

Position[list, elemen]

Posisi "b" pada list2 terdapat pada elemen ke-1 posisi ke-2, dan elemen ke-2

```
In[50]:= list2
Position[list2, b]

Out[50]= {{2.71828, b, 5}, b, {c, d}}

Out[51]= {{1, 2}, {2}}
```

Banyaknya elemen tertentu

Banyaknya elemen tertentu pada suatu list dapat diperoleh dengan perintah

Count[list, elemen]

Hanya ada satu elemen **b** pada **list2**, bukan 2

```
In[52]:= Count[list2, b]

Out[52]= 1
```

list2 tidak mengandung elemen **c**

```
In[53]:= Count[list2, c]

Out[53]= 0
```

Seleksi elemen tertentu

Seleksi elemen tertentu pada suatu list dapat diperoleh dengan perintah

Select[list, kriteria]

kriteria seringkali memerlukan suatu "predikat"

Predikat adalah suatu fungsi untuk menguji apakah suatu ekspresi memiliki sifat tertentu, yang dinyatakan dengan True atau False. *Mathematica* menyediakan banyak fungsi predikat yang umumnya berakhir dengan "Q", yang dapat ditampilkan dengan perintah "?*Q"

Perintah berikut memilih elemen list yang berupa bilangan ganjil

```
In[54]:= Select[{a, 23, 12, 0, 3.5}, OddQ]

Out[54]= {23}
```

Perintah berikut memilih elemen list yang berupa bilangan bulat

```
In[55]:= Select[{a, 23, 12, 0, 3.5}, IntegerQ]
```

```
Out[55]= {23, 12, 0}
```

Perintah berikut memilih elemen list yang > 50 (lihat topik "Pemrograman" tentang penggunaan simbol # dan &)

```
In[56]:= Select[{23, 6, 2, 62, 673}, #1 > 50 &]
```

```
Out[56]= {62, 673}
```

Manipulasi List

3.4.1 List Berdimensi Satu

Misalkan kita mempunyai list berdimensi satu dengan nama *namaList*

$$namaList = \{elemen[[1]], elemen[[2]], \dots, elemen[[n]]\}$$

- Untuk mengekstrak elemen ke-*i* sesuai dengan posisinya, gunakan perintah

$$namaList[[i]] \text{ atau } Part[namaList, i]$$

Ada perintah khusus untuk $i = 1$ dan $i = n$. `namaList[[1]]` juga dapat dituliskan dengan `First[namaList]`, `namaList[[n]]` sama dengan `Last[namaList]`.

Selain itu, tanda minus pada `namaList[[-i]]` menghasilkan elemen ke-*i* dari belakang.

- Untuk mengekstrak beberapa elemen, gunakan perintah **Take**

Perhatikan contoh-contoh berikut. Pertama dibangkitkan sepuluh data acak berupa bilangan bulat dari 0 sampai 100.

```
In[57]:= dataAcak = Table[Random[Integer, {0, 100}], {10}]
```

```
Out[57]= {68, 55, 43, 33, 26, 42, 69, 85, 44, 76}
```

```
In[58]:= dataAcak[[2]]
dataAcak[[-2]]
```

```
Out[58]= 55
```

```
Out[59]= 44
```

```
In[60]:= Take[dataAcak, 2] (* Ekstrak 2 elemen pertama *)
Take[dataAcak, -2] (* Ekstrak 2 elemen dari belakang *)
Take[dataAcak, {2, 6}]
(* Ekstrak elemen ke-2 sampai ke-6 *)
```

```
Out[60]= {68, 55}
```

```
Out[61]= {44, 76}
```

```
Out[62]= {55, 43, 33, 26, 42}
```

- Untuk menghilangkan beberapa elemen, gunakan perintah **Drop**

Cara kerja perintah **Drop** serupa dengan **Take**. Perhatikan bahwa **Drop** tidak akan mempengaruhi struktur data awal.

```
In[63]:= Drop[dataAcak, 2] (* Hilangkan 2 elemen pertama *)
Drop[dataAcak, -2] (* Hilangkan 2 elemen dari belakang *)
Drop[dataAcak, {2, 6}] (* Hilangkan elemen ke-2 sampai ke-6 *)
dataAcak
```

```
Out[63]= {43, 33, 26, 42, 69, 85, 44, 76}
```

```
Out[64]= {68, 55, 43, 33, 26, 42, 69, 85}
```

```
Out[65]= {68, 69, 85, 44, 76}
```

```
Out[66]= {68, 55, 43, 33, 26, 42, 69, 85, 44, 76}
```

- Elemen list dapat ditambah, dikurangi, atau disisipi elemen lain dengan perintah berturut-turut: **Append**, **Prepend**, **Insert**. Perhatikan contoh-contoh berikut

```
In[67]:= list3 = {1, 2, 3, 4, 5};
```

```
In[68]:= Append[list3, 10]
```

```
Out[68]= {1, 2, 3, 4, 5, 10}
```

```
In[69]:= Prepend[%, {a, b, c}]
```

```
Out[69]= {{a, b, c}, 1, 2, 3, 4, 5, 10}
```

List sebelumnya akan disisipi sebuah elemen pada posisi ke-3

```
In[70]:= Insert[%, N[Pi], 3]
```

```
Out[70]= {{a, b, c}, 1, 3.14159, 2, 3, 4, 5, 10}
```

Ketiga perintah ini pun tidak mempengaruhi struktur data awal.

```
In[71]:= list3
```

```
Out[71]= {1, 2, 3, 4, 5}
```

- Untuk memodifikasi struktur data awal, gunakan **AppendTo** atau **PrependTo**

```
In[72]:= list4 = list3
```

```
Out[72]= {1, 2, 3, 4, 5}
```

```
In[73]:= PrependTo[list4, {a, b, c}]
```

```
Out[73]= {{a, b, c}, 1, 2, 3, 4, 5}
```

list4 kini telah berubah

```
In[74]:= list4
```

```
Out[74]= {{a, b, c}, 1, 2, 3, 4, 5}
```

- Untuk mengganti elemen pada posisi tertentu dengan elemen baru, gunakan

ReplacePart[List,elem-baru, posisi]

In[75]:= **list4**

Out[75]= { {a, b, c}, 1, 2, 3, 4, 5 }

In[76]:= **ReplacePart**[list4, new, 3]

Out[76]= { {a, b, c}, 1, new, 3, 4, 5 }

- Untuk mengatur kembali posisi elemen suatu list, gunakan **Reverse**, **RotateLeft**, **RotateRight**, atau **Sort**

In[77]:= **list5 = Table**[Random[Integer, {0, 100}], {5}]

Out[77]= {33, 87, 56, 51, 100}

In[78]:= **Reverse**[list5]

Sort[list5]

Out[78]= {100, 51, 56, 87, 33}

Out[79]= {33, 51, 56, 87, 100}

In[80]:= **RotateLeft**[list5, 1], **RotateRight**[list5, 1]

Out[80]= { {87, 56, 51, 100, 33}, {100, 33, 87, 56, 51} }

- Perintah **Flatten** dapat digunakan untuk menghilangkan tanda kurung suatu output.

In[81]:= **Flatten**[{ {a, b}, {c, d}, { {e} }]

Out[81]= {a, b, c, d, e}

Perintah berikut hanya menghilangkan satu pasang { } dari tiap elemen

In[82]:= **Flatten**[{ {a, b}, {c, d}, { {e} }], 1]

Out[82]= {a, b, c, d, {e} }

- List juga dapat dipandang sebagai suatu himpunan. *Mathematica* menyediakan berbagai perintah untuk operasi himpunan

In[83]:= **Union**[{b, d, c}, {d, e}]

Out[83]= {b, c, d, e}

In[84]:= **Intersection**[{b, d, c}, {d, e}]

Out[84]= {d}

In[85]:= **Complement**[{b, d, c}, {d, e}]

Out[85]= {b, c}

- Perintah yang mirip dengan **Union** adalah **Join**. Bedanya, **Join** semata-mata menggabungkan elemen-elemen list dan tidak melakukan proses sortir.

```
In[86]:= Join[{b, d, c}, {d, e}]
```

```
Out[86]= {b, d, c, d, e}
```

3.4.2 List Berdimensi Banyak

Pada list berdimensi banyak, elemen suatu list dapat berupa list lainnya.

Sebagai contoh, berikut ini adalah list sepuluh data nama angin topan beserta kecepatan angin (knot) dan tekanan udaranya (milibar)

```
In[87]:= topan =
  {"Allison", 65, 987},
  {"Barry", 50, 995},
  {"Chantal", 60, 992},
  {"Dean", 40, 999},
  {"Erin", 75, 975},
  {"Felix", 115, 950},
  {"Gabrielle", 60, 998},
  {"Humberto", 90, 970},
  {"Iris", 75, 970},
  {"Jerry", 35, 1003}];
```

Perintah berikut menampilkan data topan dalam bentuk tabular disertai judul *field*, dan nomor rekord ("**Automatic**" pada opsi **TableHeadings**)

```
In[88]:= TableForm[topan, TableHeadings -> {Automatic, {"Angin Topan", "Kec. Angin", "Tekanan"}}]
```

```
Out[88]//TableForm=
```

	Angin Topan	Kec. Angin	Tekanan
1	Allison	65	987
2	Barry	50	995
3	Chantal	60	992
4	Dean	40	999
5	Erin	75	975
6	Felix	115	950
7	Gabrielle	60	998
8	Humberto	90	970
9	Iris	75	970
10	Jerry	35	1003

Untuk mengekstrak suatu rekord (keseluruhan sublist) gunakan **Part** atau **[[]]** seperti pada list berdimensi satu

```
In[89]:= topan[[2]]
```

```
Out[89]= {Barry, 50, 995}
```

Untuk mengekstrak satu elemen, perhatikan koordinatnya (posisi baris dan kolom). Misalnya, untuk mengekstrak kecepatan angin topan Iris (baris 9, kolom 2) tuliskan

```
In[90]:= topan[[9, 2]]
```

```
Out[90]= 75
```

Perhatikan perbedaan perintah di atas dengan perintah berikut (menggunakan `[[{}]]`)

In[91]:= `topan[[{9, 2}]]`

Out[91]= `{{Iris, 75, 970}, {Barry, 50, 995}}`

Perintah tersebut lebih ringkas dibandingkan menuliskan

In[92]:= `{topan[[9]],topan[[2]]}`

Out[92]= `{{Iris, 75, 970}, {Barry, 50, 995}}`

Perintah `TableView` amat praktis dalam menampilkan data tabular berbentuk *spreadsheet*.

In[93]:= `TableView[RandomReal[{0, 100}], {10, 3}]]`

Out[93]=

	1	2	3
1	57.1397	65.2161	65.9961
2	60.1817	29.1227	52.0547
3	19.181	62.8875	11.3182
4	13.9165	38.6679	41.0513
5	48.0841	8.51338	5.23447
6	28.844	69.3819	17.2611
7	93.1978	43.6882	85.1945
8	24.4826	61.3304	3.09524
9	19.2171	17.5075	66.7798
10	11.2502	10.6156	33.266

Pustaka

- Ruskeepää, H. 2009. *Mathematica Navigator - Mathematics, Statistics, and Graph*, 3rd edition. Elsevier.
- Abel, M. L., J. P. Braselton. 2022. *Mathematica by Example*, 6th edition. Elsevier.