# Unsupervised Learning

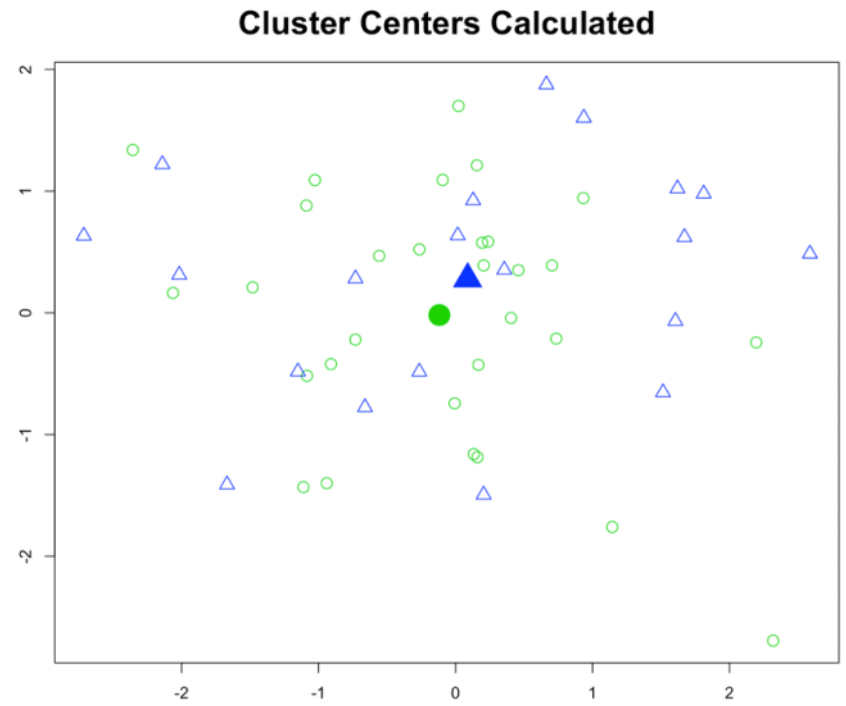
Saturday, November 14, 2020 2:17 PM

Unsupervised learning adalah proses machine learning dimana tidak ada referensi untuk hasilnya.

K-Means Clustering
Salah satu algoritma machine learning adalah k-means clustering. Kmeans clustering bekerja dengan cara meng assign seluruh data ke jumlah cluster yang didefinisikan sebelumnya secara random. Lalu menghitung titik tengah dari cluster2 tersebut. Kemudian menghitung jarak tiap datapoint ke tiap center cluster dan mengassign datapoint tersebut ke cluster yang centernya lebih dekat. Center cluster jadi bergeser karena adanya pergeseran anggota cluster. Sebelum menggunakan kmeans clustering sebaiknya semua datanya di normalisasi sehingga semua satuannya sama untuk menghindari efek yg tidak diinginkan akibat pada saat clustering.

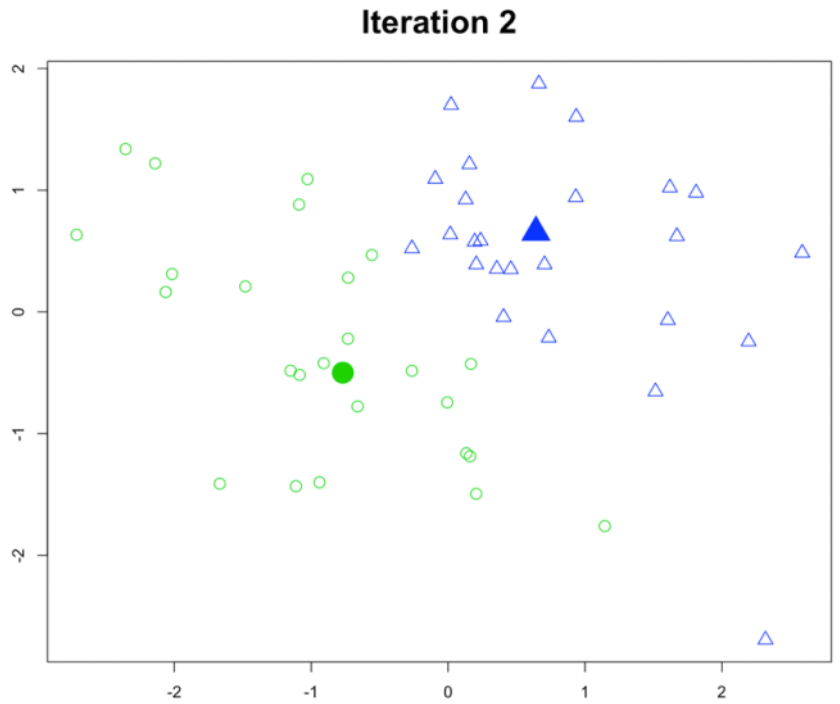
Proses ini dilakukan berulang-ulang hingga antar iterasi tidak ada datapoint yang berubah cluster. Namun tetap bisa menggunakan criteria berhenti lainnya seperti jumlah iterasi maks. Atau jika cluster center bergerak kurang dari jarak yang ditentukan.Hasil terbaik ditentukan dari total sum square dalam cluster yang terkecil.


Screen clipping taken: 11/14/2020 2:29 PM

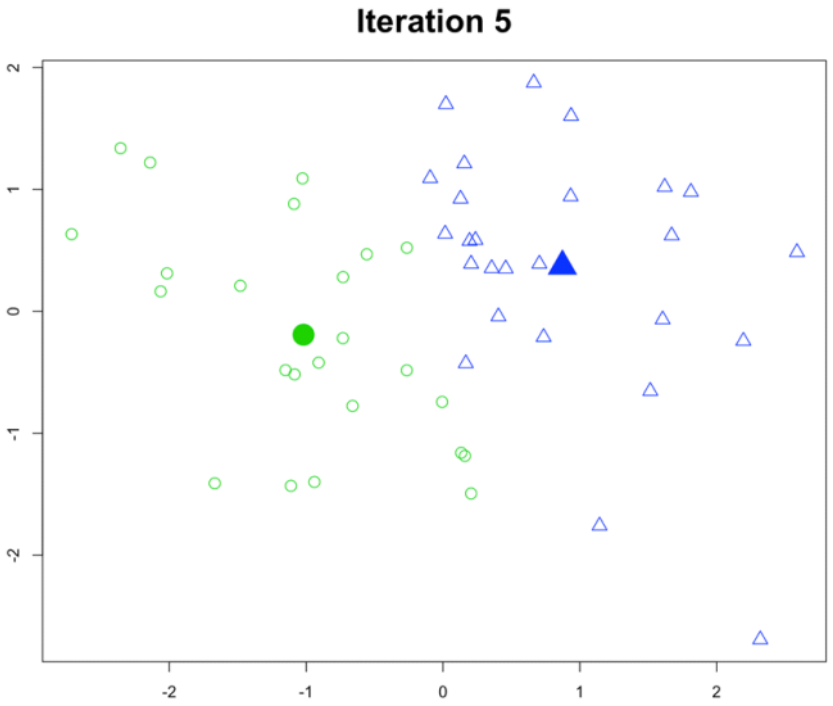



Screen clipping taken: 11/14/2020 2:29 PM


Screen clipping taken: 11/14/2020 2:29 PM




Screen clipping taken: 11/14/2020 2:29 PM

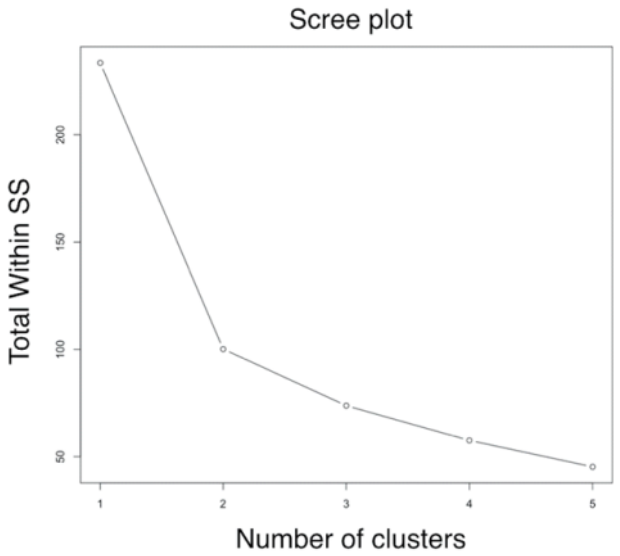



Screen clipping taken: 11/14/2020 2:30 PM

Penggunaan k-means clustering
Kmeans(df,center = jumlah_cluster,nstart = jumlah_model_dirun)

Fungsi kmeans akan memilih model yang total within ss nya paling kecil diantara jumlah run tadi. Untuk menentukan jumlah cluster dapat menggunakan scree plot. Dimana penuruna total witihn ss tidak sebanyak sebelumnya dengan penambahan cluster. Titik dimana penurunan total wss setelahnya lebih

sedikit adalah elbow.




Screen clipping taken: 11/14/2020 2:51 PM

Cara Mendapatkan screeplot

```
#bikin df kosong
wss <- 0

# Untuk 1 - 15 cluster
for (i in 1:15) {
  km.out <- kmeans(df, centers = i, nstart=20)
  # simpan nilai total withinss
  wss[i] <- km.out$tot.withinss
}
# Plot total within sum of squares vs. number of clusters
plot(1:15, wss, type = "b",
     xlab = "Number of Clusters",
     ylab = "Within groups sum of squares")
```

Hierarchical Clustering

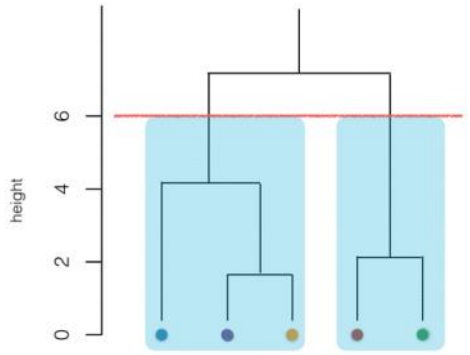
Pada hierarchical clustering tidak diperlukan jumlah cluster, ada dua pendekatan di hiearchical clustering, yaitu top down dan bottom up.

Penggunaan hierarchical clustering :

```
#menentukan jarak antar titik observasi di df
Dist_matrix <- dist(df)

#membuat model hclust
Hclust_model <- hclust(d = dist_matrix,method = metode_linkage)

#plot hclust dalam bentuk dendogram
Plot(hclust_model)
Abline(h=6,col="red")
```




Screen clipping taken: 11/14/2020 3:49 PM

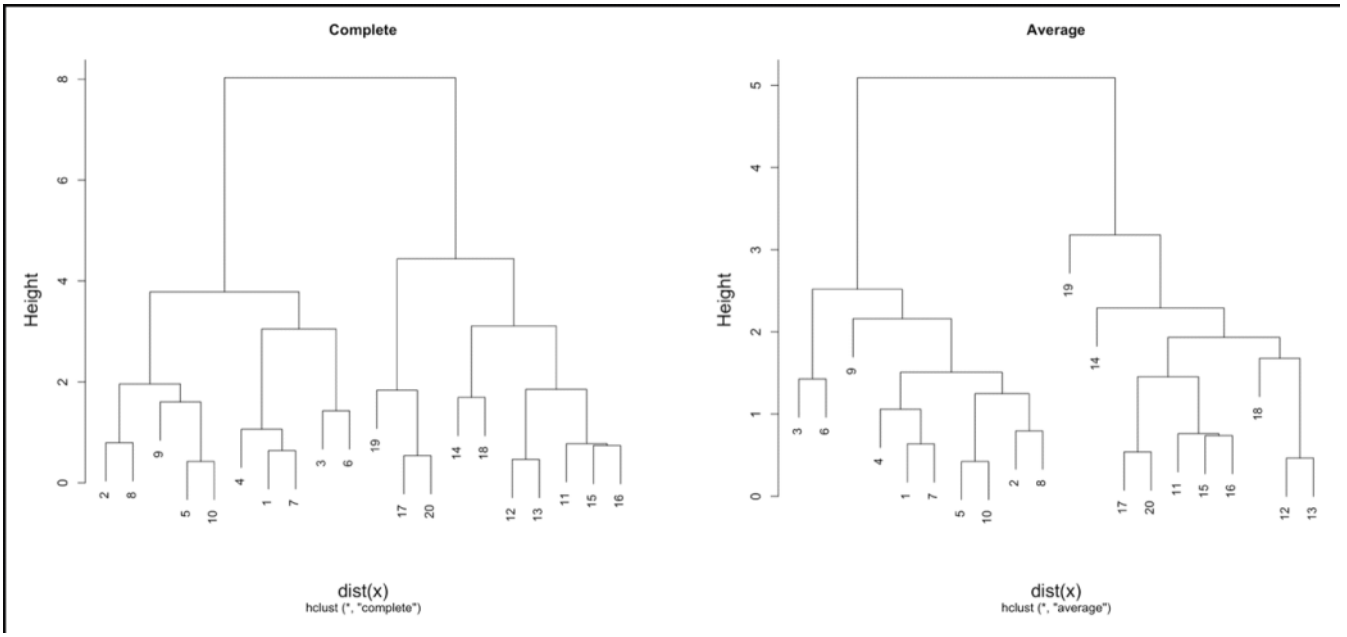
Memotong tree hclust
#memotong tree berdasarkan height
Cutree(hclust_model, h = 6)

#memotong tree berdasarkan jumlah cluster
Cutree(hclust_model,k=2)

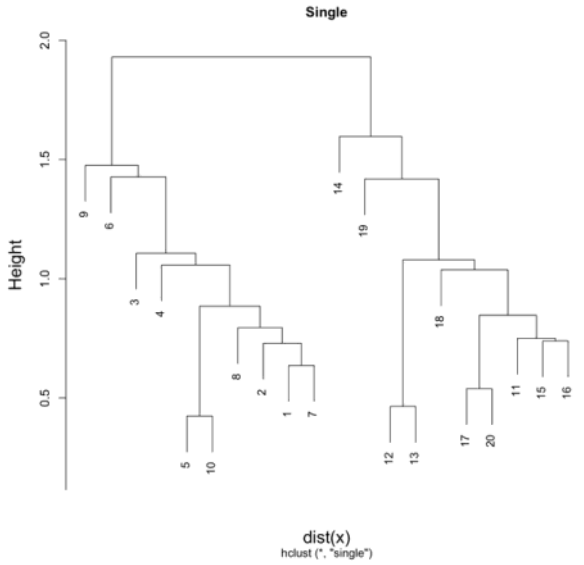
Pengclusteran point data bisa berbeda-beda tergantung metode pengukuran jarak antar cluster. Dalam hclust() ada empat metode memasangkan antar cluster :

1. Complete(default) : memasangkan similiarity dalam semua observasi cluster A dan B lalu ditentukan berdasarkan similiarities terbesar antar cluster
2. Single : sama dengan diatas namun berdasarkan similiarities terkecil antar cluster
3. Average : sama dengan diatas namun berdasarkan rata-rata similiarity antar cluster
4. Centroid : sama dengan diatas namun berdasarkan titik tengah antar cluster

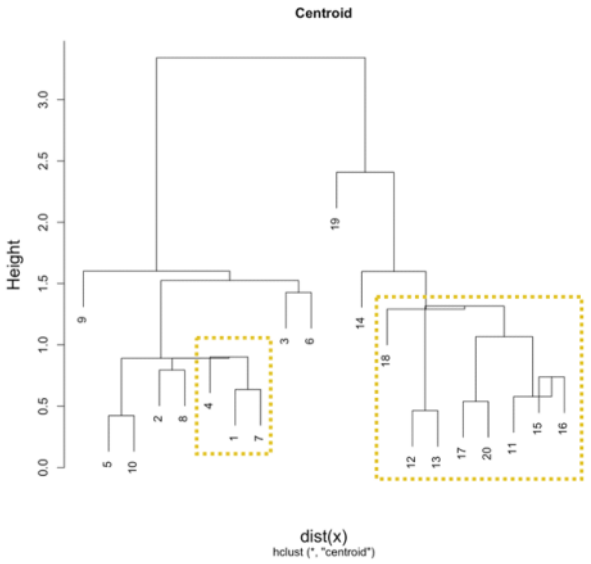
Untuk menghasilkan dendogram yang balance gunakan metode complete atau average, untuk dendogram yang unalance gunakan metode single atau centroid. Tree yang balance berguna jika jumlah data per cluster rata. Tree yang unbalance berguna untuk mendeteksi outlier.




Screen clipping taken: 11/14/2020 5:45 PM




Screen clipping taken: 11/14/2020 5:46 PM

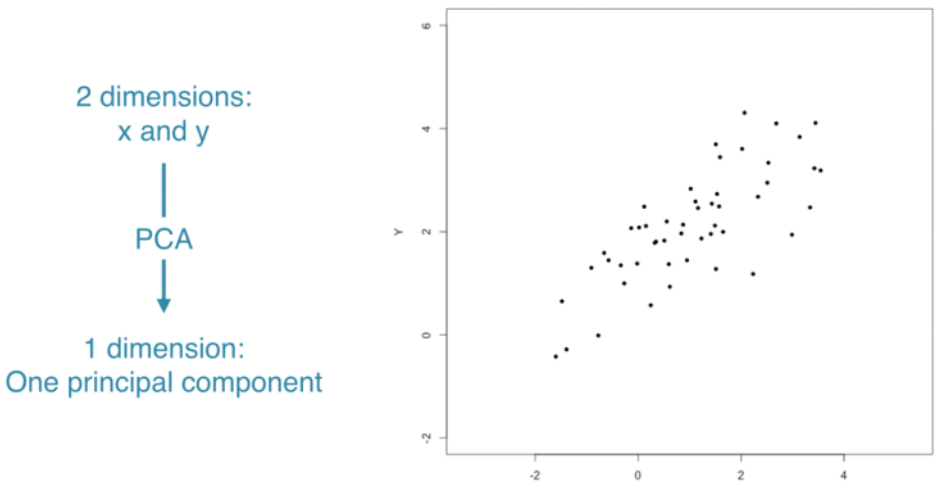



Screen clipping taken: 11/14/2020 5:46 PM

Dimensionality Reduction
Dimensionality reduction adalah penggunaan unsupervised learning dengan data continuous.
Dimensionality reduction berguna untuk menemukan struktur dalam fitur serta membantu visualisasi.

PCA (Principal Component Analysis)
PCA adalah salah satu dimensionality reduction yg populer. Ada 3 tujuan saat mencari dimensional representation yg lebih rendah :

1. Menemukan kombinasi linear dari variabel2 untuk membentuk principal component
2. Menjaga varian yang ada dalam data aslinya
3. Membuat principal component saling uncorrelated.



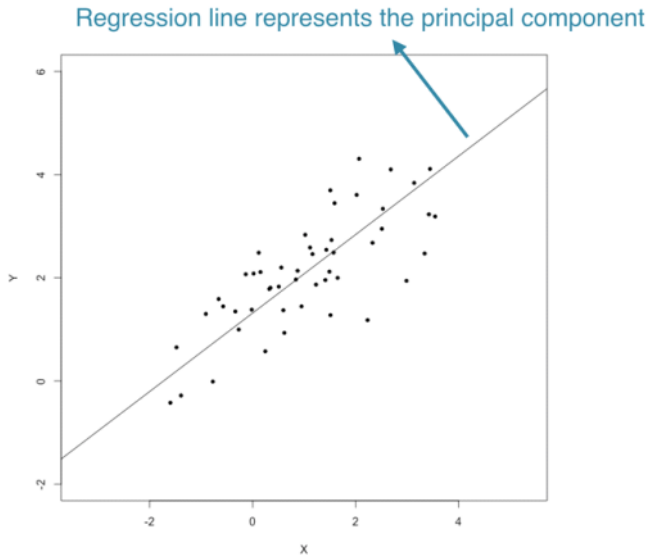

Screen clipping taken: 11/14/2020 6:22 PM

Cara kerja PCA :

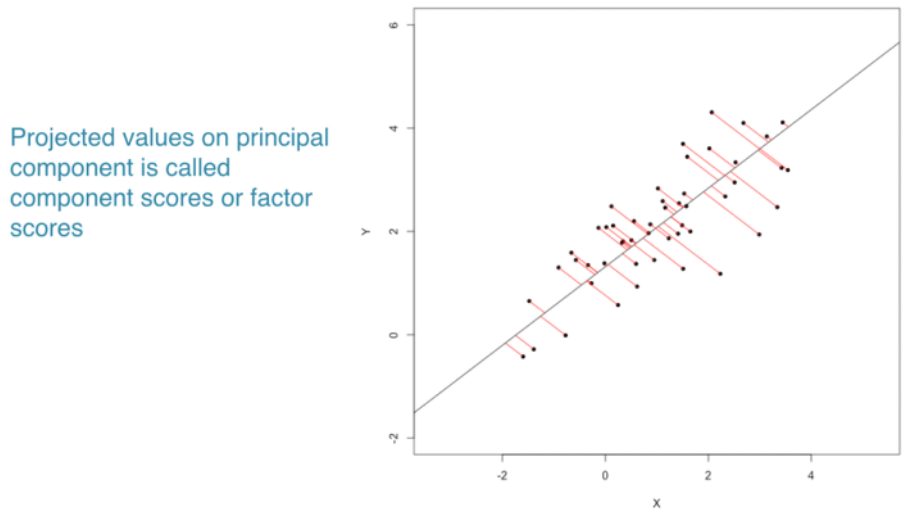
1. Membuat garis regresi yang merepresentasikan principal component dengan residual eror

seminimal mungkin.

2. Proyeksikan tiap datapoint ke princiapl component, hasil proyeksi disebut component score atau factor score

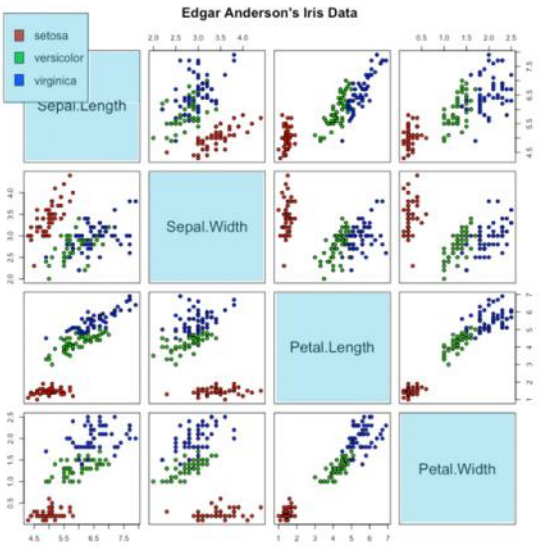


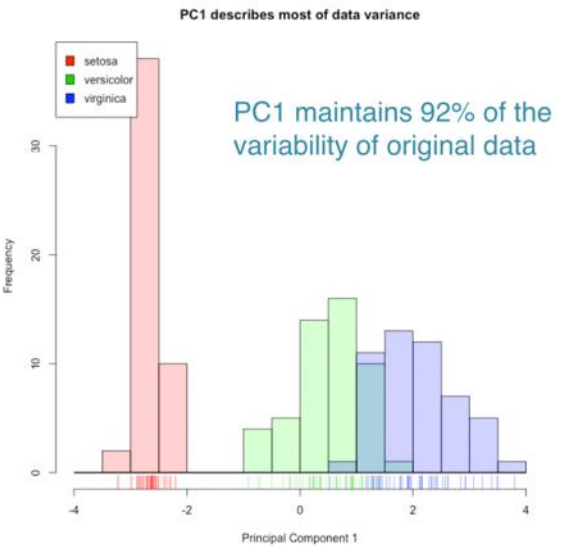

Screen clipping taken: 11/14/2020 6:24 PM




Screen clipping taken: 11/14/2020 6:27 PM

# Visualization




Screen clipping taken: 11/14/2020 6:29 PM

Cara membuat pc
Pc <- prcomp(x=df,scale=FALSE/TRUE,center=FALSE/TRUE)

Summary(pc)

Argument scale untuk menormalisasi data atau tidak sebelum pca, argumen center untuk ngecenter data-data ke 0 sebelum pca.

# Additional results of PCA

PCA models in R produce additional diagnostic and output components:

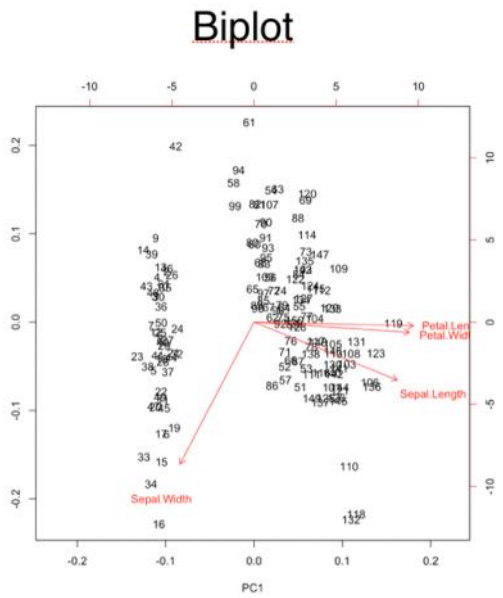
- `center` : the column means used to center to the data, or `FALSE` if the data weren't centered
- `scale` : the column standard deviations used to scale the data, or `FALSE` if the data weren't scaled
- `rotation` : the directions of the principal component vectors in terms of the original features/variables. This information allows you to define new data in terms of the original principal components
- `x` : the value of each observation in the original dataset projected to the principal components

Screen clipping taken: 11/14/2020 6:38 PM

Plotting pca
Memvisualisasikan pca bisa dengan biplot atau dengan scree plot,dr biplot kita dapat melihat garis regresi principal component dan arahnya. Jika arah 2 principal component sama berarti ada korelasi

antara keduanya. Dari scree plot kita bisa melihat kontribusi varian sebuah principal component dari data awal.

Cara membuat biplot
Biplot(pc)

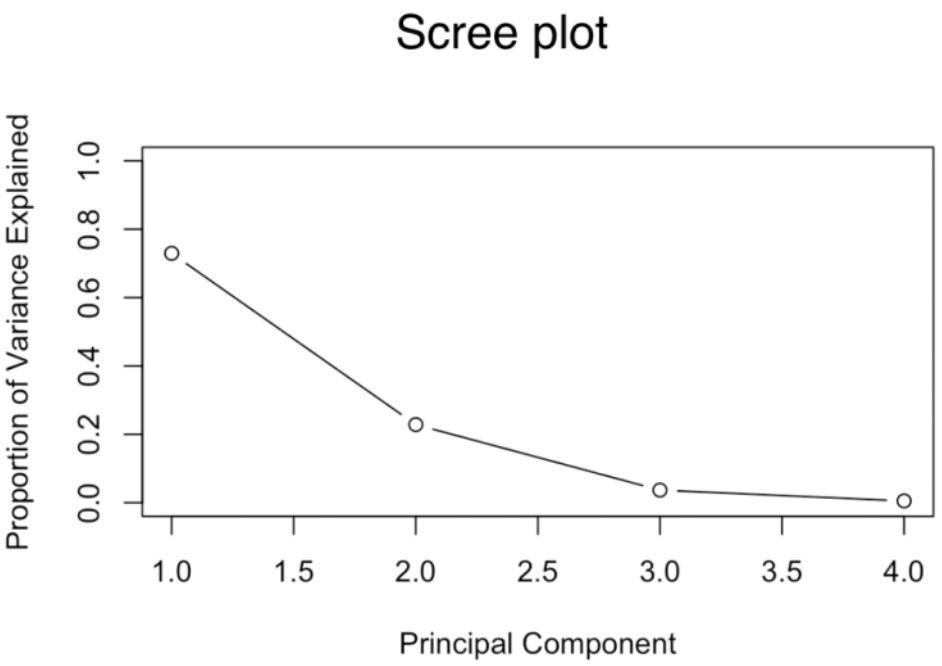



Screen clipping taken: 11/14/2020 6:44 PM

Cara membuat screeplot

```
#Mendapatkan proporsi varian utk screeplot
Pc.var <- pc$sdev^2
Pve <- pc.var/sum(pc.var)

#plot varian untuk tiap principal component
Plot(pve,xlab="Principal Component",
        Ylab="Proportion of Variance Explained",
        Ylim=c(0,1),type="b".)
```




Screen clipping taken: 11/14/2020 6:48 PM