Praktik Penggunaan *Software* R Studio dan Visualisasi Data Spasial

# *Tools Software* R Studio

A screenshot of a computer

AI-generated content may be incorrect.

4

3

2

1

# Keterangan :

# Fitur *enviroment* berguna untuk memberikan informasi mengenai hasil dari pemrograman R yang mana memuat informasi data (spasial) yang akan dimodelkan atau divisualkan. Kemudian fitur *history* untuk memeriksa riwayat penggunaan hasil pemrograman. Adapun fitur *connection* yang biasanay untuk menghubungkan data internal maupun eksternal. Kemudian fitur *tutorial* berguna untuk membantu memodelkan proses pelatihan analisis data R (dalam hal ini pemodelan spasial) secara mandiri.

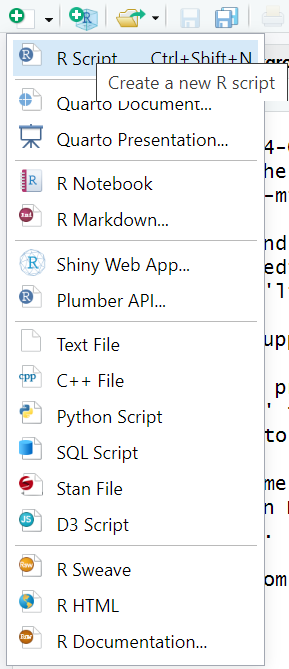
# Terdapat fitur *files* yang mana dapat digunakan untuk menyatukan data yang ada di dalam *folder* komputer. Kemudian fitur *plot* yang berfungsi sebagai hasil akhir dari proses per-*coding*-an pada fitur *script*. Kemudian fitur *packages* bertujuan untuk mengaktifkan berkas atau *library* dalam menjalankan sebuah pemrograman R. Kemudian fitur *help* yang bertujuan mencari bantuan apabila sulit mengatasi kejadian eror pada hasil *run*. Adapaun fitur *viewer* dan *presentation* bergunanya sama seperti *plot* yaitu mempresentasikan secara animasi dan mengulas kembali hasil pemrograman.

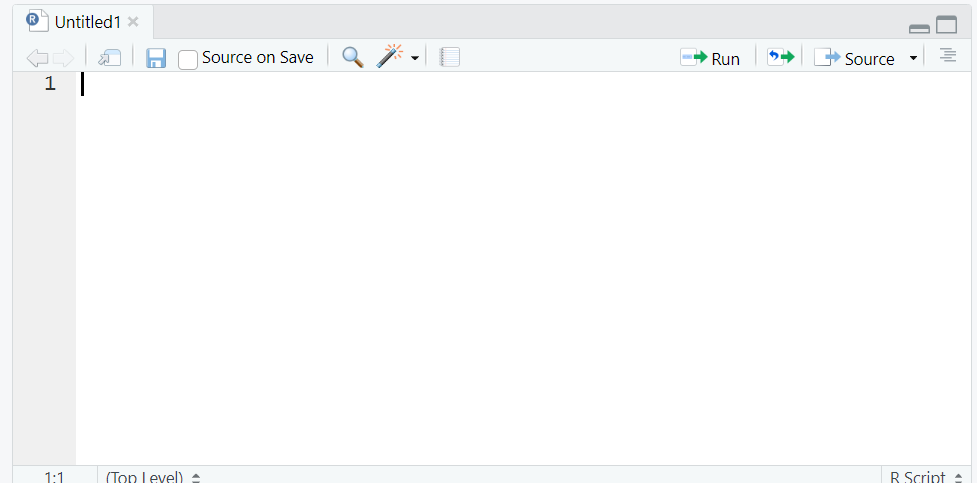
# Fitur *console* berguna untuk memperlihatkan hasil dari *coding* pemrograman R sekaligus proses dari perancangan model. Pada bagian fitur terminal berguna untuk mengintegrasikan *command* pada perangkat komputer serta *background* untuk mengakses *file coding* dengan ukuran yang lebih besar.

# Fitur *toolbars* seperti *new, folder, save,* dll digunakan untuk memulai dan mengakhiri pekerjaan.

# Menjalankan Dasar Pemrograman R Studio dan Visualisasi Data Spasial

# Buka *software* R Studio > klik *new file* > klik R *script*





1. Dasar-dasar fungsi *coding* pada R Studio

|  |  |
| --- | --- |
| Jenis Pemrograman R | Fungsi |
| st\_read | Memanggil data yang akan dimodelkan dan atau divisualkan |
| install.packages | Menginstalasi paket data pemrograman yang akan dirancang dalam memodelkan |
| library | Untuk membuka paket pemrograman R hasil dari pekerjaan instalasi |
| plot | Untuk menampilkan hasil visual data di pemrograman R |
| view | Untuk menampilkan tabel apabila memanggil data berformat .csv atau .shp atau sejenisnya |

1. Praktik Tipe Data dan Operator > rancang/buatkan/salin dan tempel *coding* pada sebagai berikut :

#Tipe Data di R

# Tipe data Logical, (TRUE, FALSE), nilai boolean

Ayam <- TRUE

class(Ayam)

# Tipe data numeric, (16,2, 198, 456), segala jenis angka

X <- 198

class(X)

#Tipe data integer, (567L, 78L, 5L), Bilangan bulat

y <- 5L

class(y)

#Tipe data Character, ("Ayam", "UI", "26")

q <- "Batas Administrasi"

class(q)

#Tipe data factor, (1, 0, "merah"), dapat berupa numerik atau string

U <- "saya"

class(U)

#Tipe data raw

xy <- charToRaw("Hello world")

class(xy)

#OPerator

#Operator penugasan/penjelas, (=, <-, ->)

PenggunaanLahan = "sawah irigasi"

print(PenggunaanLahan)

JumlahKota = 30

print(JumlahKota)

H <- 789

print(H)

"nama" -> J

print(J)

#Operator aritmatik, (+, -, \*, /, ^, %%, %/%)

# + addition

7+8

P = 5+6

print(P)

RK = 8%%3

print(RK)

#Operator pembanding, (==, !=, >, <, >=, <=)

LI <- 23

UI <- 29

LI > UI

LI == UI

#Operator Logika, (&&(dan), || (atau), ! (not))

X = 20

Y = 20

Z = 10

Y && Z -> logika

print(logika)

Y >= 10 || z >= 20 -> logika

print(logika)

# Praktik Percabangan dan Kondisi > rancang/buatkan/salin dan tempel *coding* pada sebagai berikut

#Percabangan atau pengkondisian

#1. Percabangan IF (If tunggal, If ganda, IF majemuk)

#IF Tunggal

nilai = 90

if(nilai>80){

print("Berhasil")

}

#IF ganda

nilai1 = 40

if(nilai>80 && nilai <= 100){

print("Berhasil")}else{print("gagal")}

#IF Majemuk

nilai1 = 70

#Diketahui apabila => 90 - 100 => A, 70 - 89 +> B, 60 - 69 +>

if(nilai>90 && nilai <= 100){

print("nilai kamu A")

}else if(nilai >= 70 && nilai <= 89){

print("Nilai kamu B")}else if(nilai >=60 && nilai<=69){

print("nilai anda C")

}else{

"Jangan Malas"

}

#2. Percabangan swictch case

angka = 1

switch(angka,

{print("saya")},

{print("Mahasiswa")},

{print("STIKES")}

)

string = "4"

switch(string,

"1" = {print("saya")},

"2" = {print("adalah")},

"3" = {print("Mahasiswa")},

"4" = {print("STIKES")},

"5" = {print("D3")},

"6" = {print("Keperawatan")}

)

string2 = "IA"

switch(string2,

"A" = {print("saya")},

"B" = {print("suka")},

"DD" = {print("makan")},

"KL" = {print("nasi")},

"FG" = {print("goreng")},

"IA" = {print("kambing")}

)

# Praktik Teknik Parsing dan Readline

# Teknik Parsing

nilai\_1 <- as.numeric(readline(prompt="Masukkan angka pertama: ")) # Mengambil input dari user

nilai\_1 <- 8 # Overwrite nilai dengan angka 8

print(nilai\_1 - 8) # Mencetak hasil operasi

nilai <- as.numeric(readline(prompt="Masukkan angka kedua: ")) # Perbaikan prompt

print(nilai)

nilai <- as.double(readline(prompt="Masukkan angka ketiga: ")) # Perbaikan prompt

print(nilai)

# Readline (mengambil sebuah nilai inputan)

nama <- readline(prompt="Silahkan masukkan nama ibu kandung Anda: ")

nilai <- as.numeric(readline(prompt="Silahkan masukkan PIN Anda: "))

# Praktik Array dan Vektor

#Array dan Vector

#Array, variabel yang menyimpan lebih dari 1 buah data yang memiliki tipe data yang sama

x <- array(1:12)

#Vector, bisa menyimpan banyak data dengan berbagai tipe data

y = c(1, 2, TRUE, "Nama", 70.6)

length(y)

print(length(y))

#Menambah data di vector 1 data

y[6] = 90

#Menambah data lebih dari satu

y[7:9] = c("Kelas", 21, 53.8)

#mengupdate/merubah vector

y[1] = 67

#menghapus data pada vector

print(y)

y <- y[y!=2]

print(y)

x <- array(c(4,6,8,9,10), dim = c(2,2,3))

print(x)

1. Praktik *for* dan *while*

#Perulangan/looping pada R

#perulangan for

nilaiawal = 1

for (nilaiawal in nilaiawal:20) {

print(nilaiawal)

}

nilaiawal = as.numeric(readline("masukan nilai awal:"))

for (nilaiawal in nilaiawal:20) {

print(nilaiawal)

}

Akhir = as.numeric(readline("masukkan nilai akhir:"))

for(Akhir in 1:Akhir){

print(Akhir)

}

Akhir = as.numeric(readline("masukkan nilai akhir:"))

for(Akhir in Akhir:1){

print(Akhir)

}

Akhir = as.numeric(readline("masukkan nilai akhir:"))

for(Akhir in Akhir:1){

if(Akhir %% 2 == 1)

print(Akhir)

}

Akhir = as.numeric(readline("masukkan nilai akhir:"))

for(Akhir in Akhir:1){

if(Akhir %% 2 == 0)

print(Akhir)

}

#perulangan while

angka <- as.numeric(readline("masukkan angka anda:"))

nilaiawal = 1

while(nilaiawal <= angka){

print(nilaiawal)

nilaiawal = nilaiawal+5

}

angka <- as.numeric(readline("masukkan angka anda:"))

nilaiawal = 1

while(angka >= nilaiawal){

print(angka)

angka = angka - 4

}

1. Praktik Visualisasi Data Spasial, Raster, dan *Leaflet*

install.packages(c("sf", "terra", "raster", "leaflet", "ggplot2", "tidyverse", "broom", "maps", "mapdata", "mapproj"))

# Load library yang diperlukan

library(sf)

library(terra)

library(raster)

library(leaflet)

library(ggplot2)

library(tidyverse)

library(broom)

library(maps)

library(mapproj)

library(mapdata)

# Definisi direktori

dataDir <- "E:/NGI\_Mentor\_Data/Praktikum\_Tipe\_Data\_Operator/SHP\_Depok"

setwd(dataDir)

#cara1

shp <- st\_read("SHP\_Depok.shp")

plot(shp)

#Cara2

shp2 <- st\_read("SHP\_Depok.shp")

plot(shp2)

plot((shp2),col="blue")

library(sf)

library(ggplot2)

shp2 <- st\_read("SHP\_Depok.shp")

# Plot dengan geom\_sf

ggplot() +

geom\_sf(data = shp2, color = "black", fill = "lightgrey", size = 0.1) +

coord\_sf() +

labs(title = "Peta Kota Depok")

#cara memunculkan lefleat berbasis online

library(leaflet)

names(providers)

#memunculkan Kota Depok (Boleh ganti daerah lain)

leaflet()%>%addProviderTiles('Esri.WorldImagery')%>%

setView(lng = 106.80887110868056,lat = -6.414290187138151,

zoom = 15)%>%

addMarkers(lng = 106.80887110868056,lat = -6.414290187138151)

#cara membuat gamaran lokasi tanpa shp dan raster

library(leaflet)

library(tidyverse)

library(ggmap)

library(leaflet.extras)

library(htmltools)

library(ggplot2)

library(maps)

library(mapproj)

library(mapdata)

w <- map\_data('world')

isf <- map\_data('world',

region = c('Iran'))

ggplot(isf, aes(x = long, y = lat, group = group, fill = region))+

geom\_polygon(color = 'black')+

coord\_map('polyconic')

isf <- map\_data('world', region = c('Saudi Arabia'))

ggplot(isf, aes(x = long, y = lat, group = group, fill = region))+

geom\_polygon(color = 'black')+

coord\_map('polyconic')

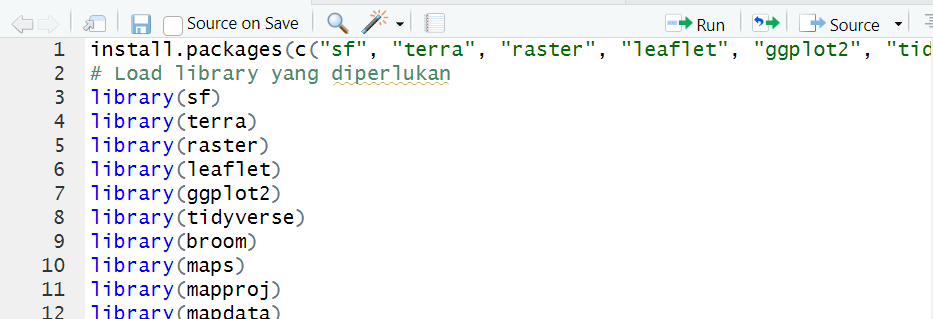
isf <- map\_data('world',

region = c('Palestine'))

ggplot(isf, aes(x = long, y = lat, group = group, fill = region))+

geom\_polygon(color = 'black')+

coord\_map('polyconic')

1. Berdasarkan urutan nomor 1 – 8 setelah dirancang kodingnya klik sebelah kanan *coding* > klik *run* dan perhatikan *console*, *plot*, dan *view*

# Hasil Akhir

# A group of blue and black shapes

A screenshot of a computer screen

AI-generated content may be incorrect.

