

Latihan Mengenal dan Membuat Data Sederhana

R bersifat casesensitive, jadi mereka akan membedakan antara huruf besar dan kecil. Tapi jangan khawatir, jika kamu salah mengetik biasanya R akan memberikan informasi/tanda dimana error terjadi (*tapi terkadang informasinya bersifat implisit sehingga kamu harus mencari tahu sendiri*).

Mengatur Tempat Penyimpanan

1. `setwd()` -> ketik `setwd()` -> didalam kurung paste alamat folder yang diinginkan -> ganti “\” menjadi “/” -> ctrl+enter.
2. Session -> pilih session -> pilih set working directory -> pilih lokasi folder penyimpanan yang diinginkan.
3. Help-> pilih “...” untuk memilih folder yang diinginkan -> pilih “more” -> set as working directory.
Perhatian: jangan menghapus apapun dari jendela help, karena tersambung dengan penyimpanan aslinya.

Project vs Script

Project dapat terdiri dari beberapa script, sedangkan script adalah bagian dari project. Seperti pada ms. Excel project adalah file excelnya sedangkan script adalah sheetnya. Script adalah file teks yang berisi sekumpulan perintah dan komentar. Script dapat disimpan dan digunakan nanti untuk menjalankan kembali perintah yang disimpan.

Membuat Project

1. Klik File
2. Klik New Project -> Existing Directory

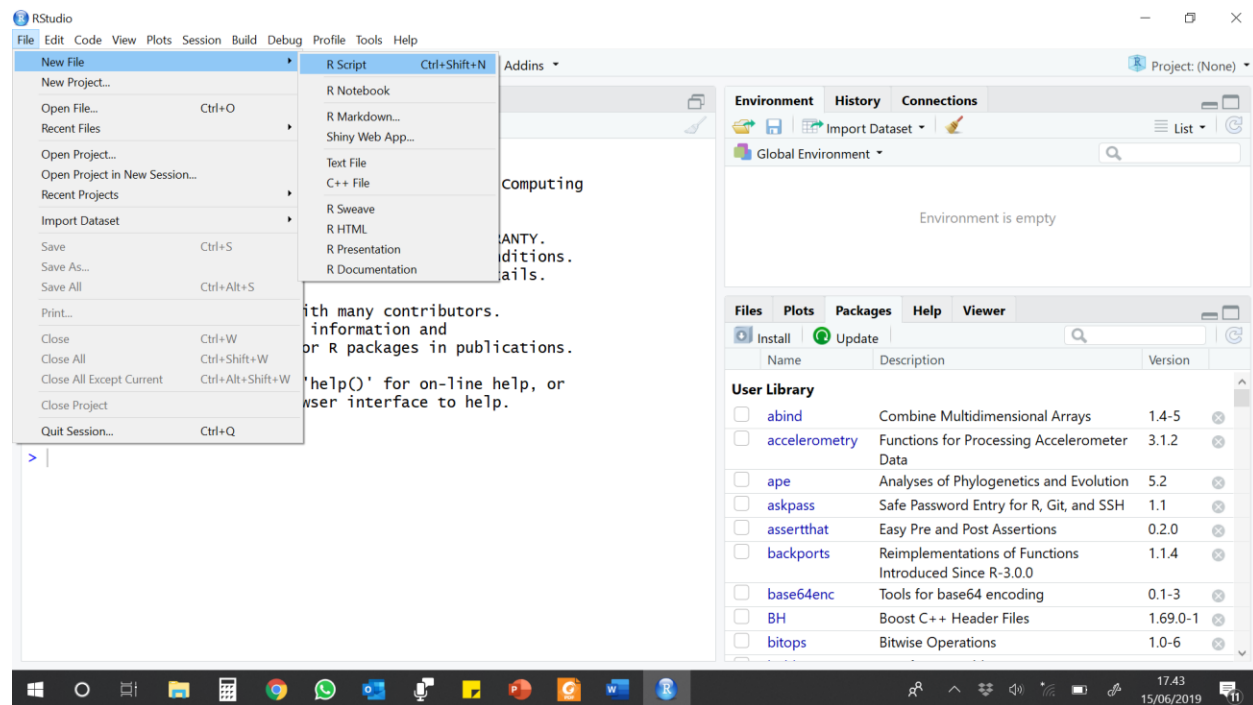
Membuka Script

1. Klik File
2. Pilih New file -> R Script

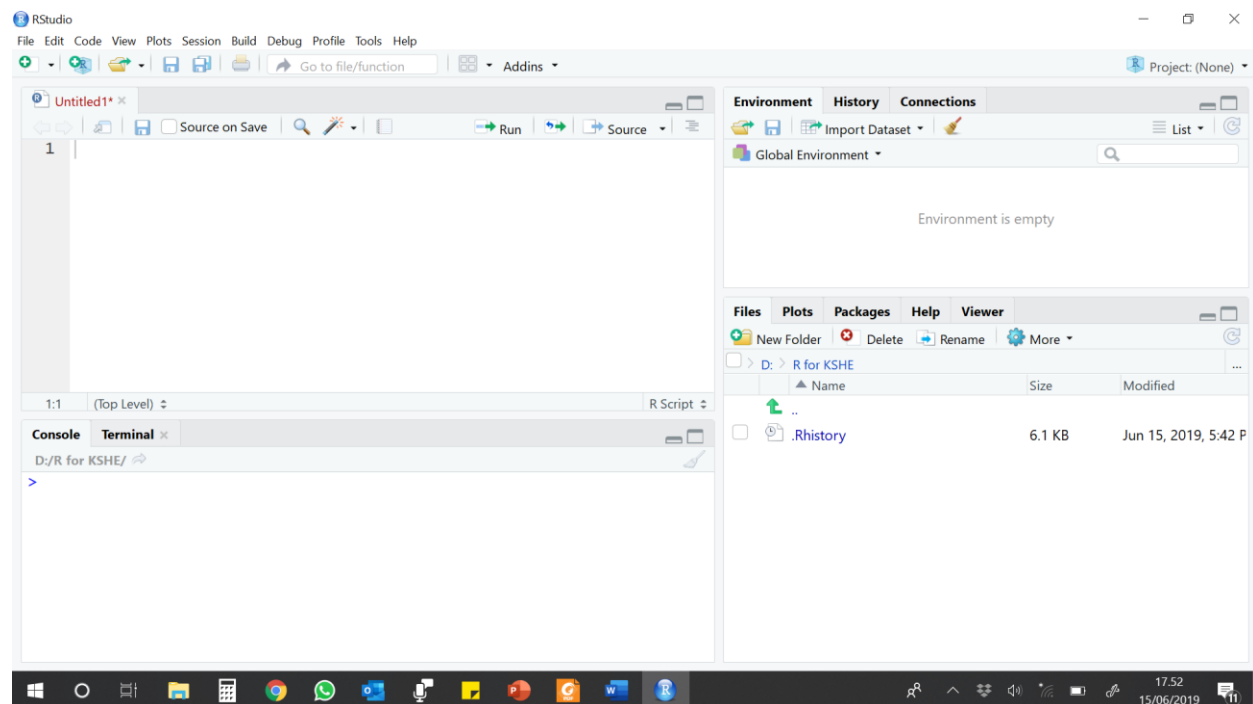
Rizki Amalia Adinda Putri

Ardiantiono

Diterjemahkan dari modul *Online Exploring Data with R* oleh Cedric Tan (WildCRU, University of Oxford)



Nah sekarang tampilan layarmu akan seperti ini.



3. Untuk menyimpan script: Klik File Save Script, jangan simpan di folder yang terlalu dalam.

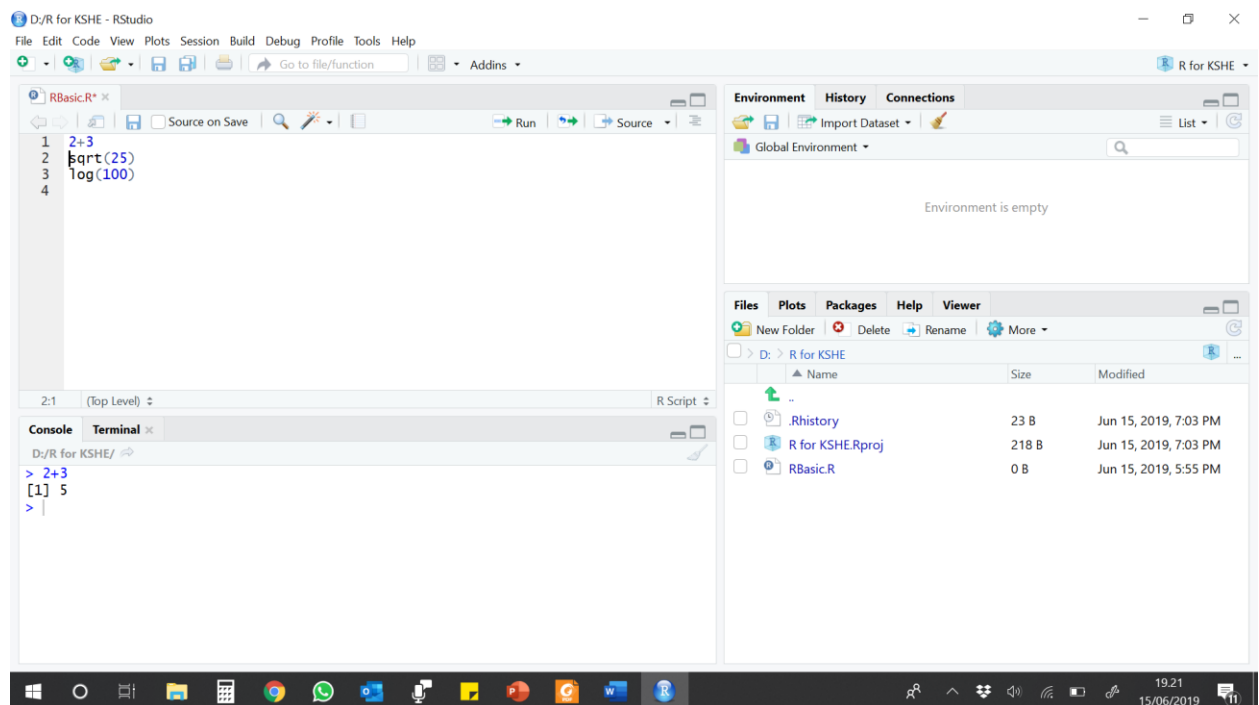
R as Fancy Calculator

- Simbol yang digunakan di R tidak berbeda dengan yang ada di Excel yakni “+” untuk tambah; “-” untuk kurang; “*” untuk kali; “/” untuk bagi; dan “^” untuk pangkat.

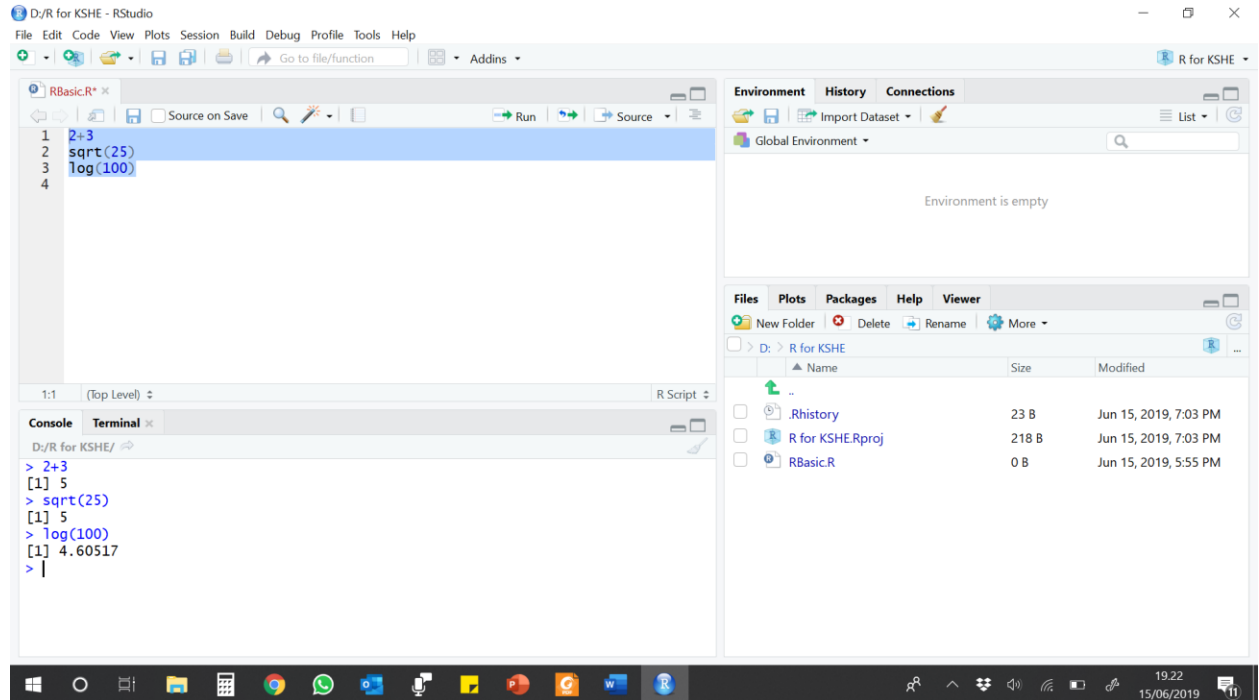
1. Sekarang ketik kode dibawah ini pada jendela script

```
2+3  
sqrt(25)  
log(100)
```

2. Setelah kamu mengetik kode di atas, saatnya menjalankannya di RStudio. Cara paling mudah untuk menjalankan fungsi di R dengan memencet ctrl + enter di baris fungsi yang telah kamu ketik. R akan mengirimkan fungsi tersebut ke jendela R Console dan memprosesnya.



3. Kursor akan otomatis ke baris berikutnya. Jadi kamu bisa menekan Ctrl+R berkali-kali untuk menjalankan beberapa fungsi. Atau kamu bisa men-highlight beberapa baris fungsi dan mengetik Ctrl+enter untuk menjalankannya sekaligus (silahkan dicoba!).



4. Selain `ctrl + enter` kamu juga bisa mengklik “Run”.
5. Perhatikan kalau R akan langsung memberikan jawaban setelah kamu mengetik `Ctrl+enter` pada jendela console.
6. R bersifat **case sensitive**, coba kamu ketik `lo(100)`. Apa yang muncul?
7. R tidak mempermasalahkan jika kamu menggunakan spasi atau tidak. Saya merekomendasikan kamu untuk menggunakan spasi di antara fungsi yang panjang (`*`, `/`, `+`, dll) agar fungsi tidak terlalu padat, tapi tidak di antara tanda kurung “(” dan kode yang diketik setelahnya.
8. Ketik fungsi di bawah dan proses dengan `Ctrl+Enter`. Tanda “`<-`” berfungsi untuk memberikan nilai pada objek di samping kiri, dalam contoh ini adalah “`x`”

```
x<-7*3
```

Kamu juga bisa menggunakan tanda “formula `->` objek”, tapi umumnya kita menggunakan tanda “objek `<-` formula”. Kamu juga sebenarnya bisa menggunakan tanda “`=`” tapi arah dari pemberian nilai ke objek menjadi tidak jelas (*x tidak selalu sama dengan y*).

9. Jika kamu ingin memberikan komentar/catatan terhadap fungsi yang kamu ketik, kamu bisa menggunakan tanda “`#`” dan diikuti dengan komentar. R akan mengabaikan apapun yang kamu ketik setelah tanda `#` ini.
- ```
x <- 7*3 #memberikan nilai kepada x
#menampilkan isi dari x
x
```

10. Ingat kalau R adalah case-sensitive! Coba ketik kode berikut

```
x
```

11. Apabila kamu menekan enter sebelum menyelesaikan fungsi atau kamu mengetik fungsi yang panjang dan membaginya ke beberapa baris, R akan memberikan tanda “+” yang menandakan ia menunggu informasi lebih sebelum menjalankan fungsi.

```
x<-(7*3)/
2
x
```

Jika kamu tersangkut dengan tanda “+” di console, cek apakah formula yang kamu ketik sudah lengkap, dan jika kamu tidak bisa menemukan apa yang salah, ketik “Esc” untuk menghentikan fungsi dan coba lagi.

12. Kamu juga bisa menggunakan objek untuk perhitungan.

```
x <- 7*3
x
y <- x+2743
y
```

Menyimpan informasi di R tidak terbatas hanya pada satu angka saja, kamu juga bisa memasukkan rangkaian angka atau teks sebagai vektor. Ini adalah barisan data satu dimensi, sebagai contoh urutan angka “1,2,3,4,5”; dan ini tidak sama dengan “5,4,3,2,1”.

13. Fungsi `c()` berarti “concatenate” atau menggabungkan nilai-nilai menjadi satu daftar.

```
v<-c(1,2,3,4,5,6)
v
```

14. Sekarang kamu mempunyai vektor “v”. Ketika kamu memasukkannya ke dalam perhitungan, R akan melakukan kalkulasi di setiap nilai di daftar.

```
v2 <- v+2
v2
v3 <- v2*3
v3
vx <- c(v,x)
vx
```

15. Sekarang coba buat vektor “w” yang berisikan enam nilai: nilai empat diulang dua kali dan nilai lima diulang empat kali

```
w<-c(4,4,5,5,5,5)
```

16. Kamu bisa melakukan kalkulasi terhadap vektor-vektor tersebut seakan mereka adalah satu nilai, sebagai contoh dengan menambah sebagai berikut:

```
v*w
```

Perhatikan bahwa nilai pertama dari v ditambah dengan nilai pertama dari w, nilai kedua v dengan w, dan seterusnya. Vektor tidak harus sama panjang untuk digunakan bersama.

17. Buat vektor ketiga "u" yang berisikan nilai satu dan dua.

```
u <- c(1,2)
```

```
V + u
```

Lihat bagaimana vektor yang lebih pendek ini "didaur ulang" hingga mencapai akhir vector yang lebih panjang. R juga memberikan peringatan bahwa kedua vektor tidak sama panjang. Apakah ada peringatan lain lagi?

18. Kamu juga bisa menggunakan nilai teks di R. Jangan lupa untuk menambahkan tanda "".

```
lokasi.utama <- "Bukit Barisan Selatan"
```

19. Kamu juga bisa menggunakan tanda koma "," untuk membuat daftar yang lebih banyak.

```
lokasi.lain <- c("Way Kambas", "Leuser", "Kerinci Seblat")
```

```
lokasi.semua <- c(lokasi.utama, lokasi.lain)
```

```
lokasi.semua
```

20. Sekarang apa yang akan terjadi jika kamu memasukkan nilai angka pada vektor.

```
lokasi.angka <- c(lokasi.semua, v)
```

```
lokasi.angka
```

```
v
```

21. Format nilai angka dalam "lokasi.angka" dan "v" adalah berbeda. R tidak bisa menangani dua tipe data berbeda (numerik dan teks) dalam satu vektor, oleh karena itu dia memperlakukan angka sebagai teks dan menutup mereka dalam tanda kutip ("").

Melakukan kalkulasi pada "lokasi.angka" akan membuat ini lebih jelas. Contohnya:

```
lokasi.angka*2
```

R akan memberikan error message yang menyatakan bahwa kamu mencoba memasukkan nilai non-numerik ke dalam kalkulasi matematik. Mari kita berikan komentar kalau formula ini tidak bisa dilakukan.

```
lokasi.angka*2 #kalkulasi tidak berfungsi: lokasi.angka
mengandung angka.
```

22. Sangat mudah untuk membuat data berpola di R. Sebagai contoh membuat sekuens 1,2,3,4,5,6,7,8,9,10 dapat dilakukan dengan:

```
1:10
```

23. Kamu juga bisa menggunakan fungsi `seq()` untuk membuat sekuens berpola. Untuk tahu lebih banyak tentang fungsi ini kamu bisa gunakan menu `help`

```
?seq
```

Dalam penggunaannya kita akan melihat R meminta informasi `<parameters>` (arguments) untuk memulai proses. Kita akan coba contoh sederhana dengan sekuens dimulai dari 0 (`<from>`) sampai 25 (`<end>`), dengan interval lima setiap kalinya(`<by>`).

```
seq(0, 50, 5)
```

24. Kita juga bisa membuat dataset yang lebih kompleks dengan membuat beberapa vektor dan menggabungkannya bersama. Sebagai contoh kita menduga akan melihat perbedaan home range antara komodo jantan dan betina. Dalam contoh di bawah kita melihat data 20 komodo untuk masingmasing seks.

```
jantan <- rnorm(20, 720, 150) #kita akan melihat fungsi rnorm
di sesi probabilitas.
betina <- rnorm(20, 530, 135)
gabung <- c(jantan, betina)
```

25. Mari sekarang kita buat vektor yang mengkode estimasi home range sebagai data dari jantan dan betina:

```
c(rep("M", 20), rep("F", 20))
```

26. Sekarang kamu mempunyai dua objek vektor yang mewakili kolom seks dan home range. Mari kita gabung keduanya dengan menggunakan fungsi `cbind()`

```
cbind(seks, gabung) #cbind adalah fungsi untuk menggabungkan
kolom
```

Fungsi `rbind()` juga memiliki mekanisme yang sama, dan digunakan untuk menggabungkan baris data.

27. Sekarang simpan kode R kamu dalam R (Script) text file: pergi ke File > Save, dan simpan di dalam folder. R akan menambahkan ekstensi file. R secara otomatis.

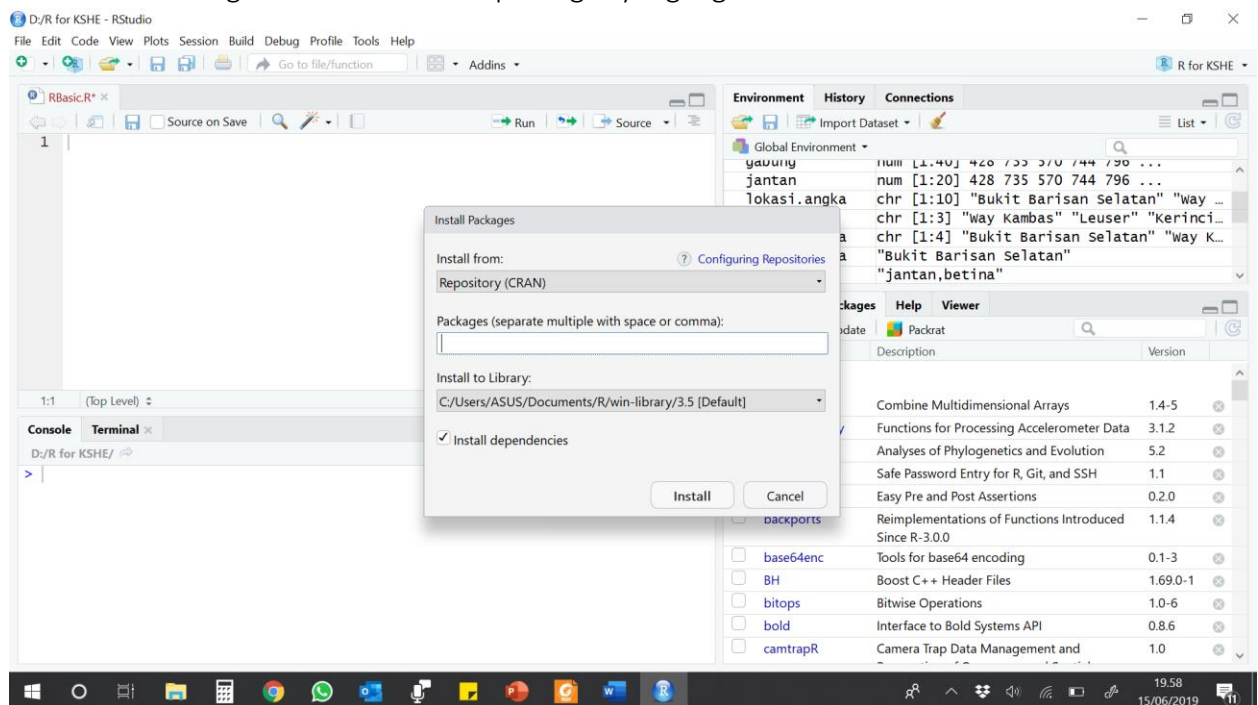
28. Kamu juga sudah mengumpulkan banyak objek (kontainer data) selama latihan ini. Mari kita lihat apa saja:

```
ls() #tampilkan objek di workspace
```

## Menginstall Packages di R

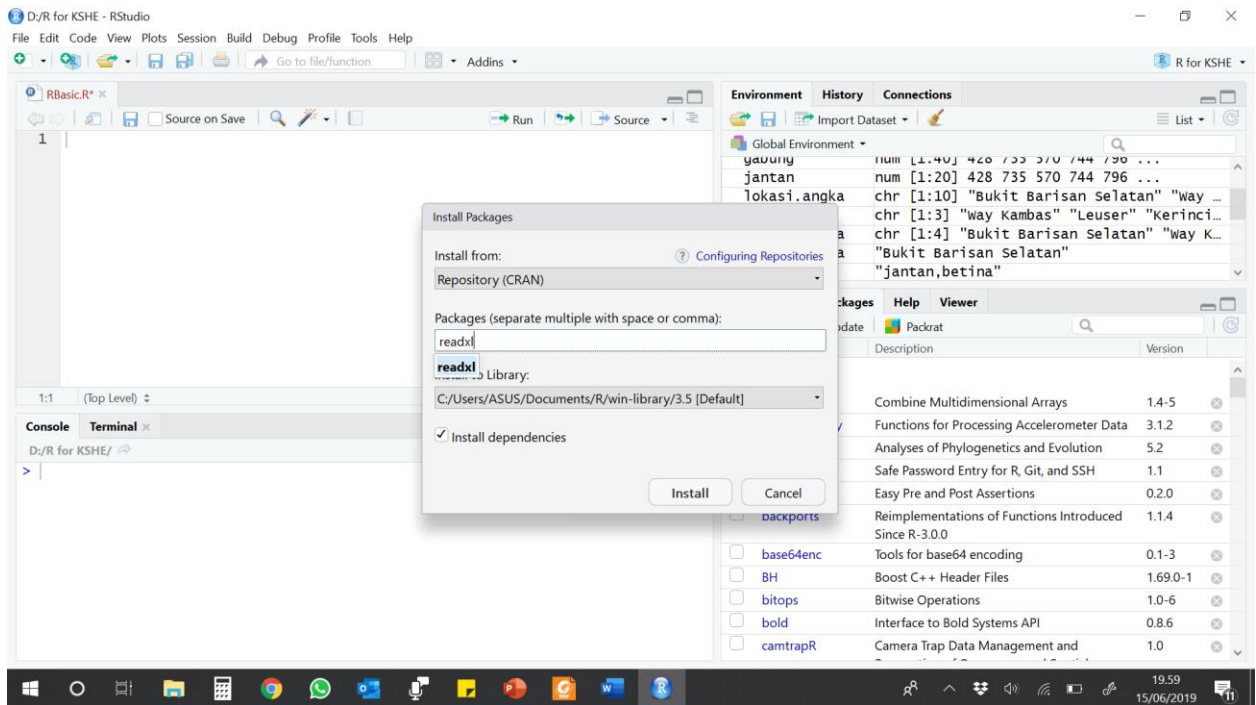
Beberapa fungsi yang ingin kita jalankan umumnya berada dalam package yang belum terdapat di R. Oleh karena itu kita perlu mengunduh dan menginstallnya terlebih dahulu.

29. Pilih menu Packages -> Install -> ketik packages yang ingin di install





30. Ketik nama package yang kamu inginkan, untuk kali ini mari kita coba download dan install package “readxl” (*ingat bahwa kamu butuh koneksi internet untuk tahapan ini!*).



31. Setelah package terinstall, kamu bisa memanggil package tersebut dengan mengetik:
- library(packagename)**