

Back End Development 1 sesi 3

C# Programming Logic



C# Programming Logic - Sesi 03 INTRODUCTIONS

Logika Pemrogramman dalam C# ini juga dikenal dengan Struktur Pengambilan Keputusan yang mana menyediakan perhitungan logika yang akan eksekusi beberapa bagian kode program.

Contoh.

Masih ingat dengan live code terakhir di sesi 2 ? Dimana kita buat beberapa kondisi : Jika Pengguna umur > 18 maka akan menampilkan text Welcome to the OCBC Dan sebaliknya ...

```
// menggunakan logika AND
if(isAdult && isPasswordValid)

{

Console.WriteLine("WELCOME TO THE CLUB!");

}

else

Console.WriteLine("Sorry, try again!");

}
```

Nah dengan kondisi seperti ini kita bisa menentukan apa yang akan dan harus dikerjakan oleh aplikasi yang kita buat nantinya.



C# Programming Logic - Sesi 03 Statements - IF

Statement If selalu erat kaitannya dengan logika BOOLEAN:

Apabila If menghasilkan nilai true, maka baris kode program yang berkaitan akan di eksekusi. Begitupun sebaliknya: Apabila If menghasilkan nilai false, maka baris kode tersebut akan diabaikan.

```
if (condition) {
    //statements
}
```



Dalam Menggunakan Statement IF kita juga dapat memanfaatkan Operator Relational untuk membangun logika dengan Statement IF.

Misal kita akan membuat logika untuk mengetahui angka mana yang lebih besar. Pada kode Program di atas, IF menjelaskan bahwa Apabila 83 lebih dari 30 maka statement Console.WriteLine akan di jalankan yang mana menampilkan kata "y Lebih Besar daripada x" pada layar.

misal pada program di atas kita ubah operator lebih dari ">" menjadi kurang dari "<" maka kata "y Lebih Besar daripada x" tidak akan dicetak.

Contoh 1:

```
if (10 == 10) {
Console.WriteLine("Benar"); }
// Outputs "Benar"
```

Kode Program di atas, IF menjelaskan bahwa Apabila 10 sama dengan 10 maka statement Console.WriteLine akan di jalankan, yang mana menampilkan kata Benar pada layar.



```
Contoh 2:
```

```
if (10 != 10) {
   Console.WriteLine("Benar");
}
```

pada program di atas operator diubah dari "==" menjadi bukan sama dengan "!=" maka kata benar tidak akan dicetak.

```
using System;
      0 references
      public class Logika1
          0 references
          public static void Main()
                int a = 92;
                int b = 53;
                   if (a > b) {
                        Console.WriteLine("a lebih besar dari b"); }
10
                        // Outputs "a lebih besar dari pada b"
11
12
13
```

Maliks-MacBook-Air:SourceCode swijaya\$
ut:Logika1 && mono Logika1
a lebih besar dari b



C# Programming Logic - Sesi 03 Statement If-Else

Statement Else adalah pengikut dari Statement IF, apabila Statement IF adalah statement Utama maka statement Else adalah statement kedua atau statement Opsional.

Statement Else akan dieksekusi apabila Statement IF tidak mendapatkan hasil atau mempunyai hasil Salah.

```
if (condition) {
   //statements
}
else {
   //statements
}
```

Compiler akan menguji kondisi:

- Apabila statement bernilai true, maka kode di dalam pernyataan IF akan dieksekusi.
- Apabila statement bernilai false, maka kode dalam pernyataan Else akan dieksekusi.



```
using System;
     0 references
     public class Logika2
         0 references
         public static void Main()
              int nilai = 75;
              if (nilai < 60) {
                  Console.WriteLine("Nilai Kamu C" );
11
12
              else if (nilai < 80) {
13
                  Console.WriteLine("Nilai kamu B");
17
              else {
                  Console.WriteLine("Nilai kamu A");
21
```

Maliks-MacBook-Air:Sesi3 swijaya\$ cd "/Users/swijaya/ gika2 && mono Logika2 Nilai kamu B



Task: Buat Fungsi Login sederhana dengan "Username" & "Password" dimana: Jika kondisi benar maka sistem akan menampilkan Anda Berhasil Login,

Apabila salah satu data input salah baik itu username atau password maka sistem akan menampilkan <mark>Username atau Password anda salah.</mark>

Maliks-MacBook-Air:Sesi3 swijaya\$ cd gika3 && mono Logika3 Username: ocbc Password: bootcamp Anda berhasil login

Benar

Maliks-MacBook-Air:Sesi3 swijaya\$ cd 'gika3 && mono Logika3 Username: ocbc Password: BootCamp Username atau Password anda salah _____

Salah



Jawaban:

```
using System;
     0 references
     public class Logika3
         0 references
         public static void Main()
             string Username;
             String Password;
             //digunakan untuk menginput, isikan username= ocbc dan password= bootcamp
10
             Console.Write("Username: ");
11
             Username = Console.ReadLine();
             Console.Write("Password: ");
             Password = Console.ReadLine();
             //logika percabangan jika username dan pass sama maka kondisi pertama akan terpe
17
             if (Username == "ocbc" && Password == "bootcamp")
              Console.WriteLine("Anda berhasil login");
             //jika tidak maka kondisi kedua akan terpenuhi
             else
             Console.WriteLine("Username atau Password anda salah");
             // Console.ReadKey();
```



C# Programming Logic - Sesi 03 STATEMENT if...else...if

Jika percabangan sebelumnya sering digunakan pada sebuah program untuk fungsi login, maka percabangan ini juga tidak kalah penting kegunaanya.

Percabangan IF..ELSE..IF ini dapat digunakan untuk menetapkan output sesuai kondisi.

Contohnya pada sebuah program pendidikan yang mengkategorikan nilai angka menjadi grade. Misalkan nilai murid tersebut 80 maka akan mendapatkan grade B dan begitu seterusnya.

Fungsi tersebut biasanya menggunakan jenis percabangan ini supaya dapat berjalan dengan optimal.



C# Programming Logic - Sesi 03 STATEMENT if...else...if

```
using System;
0 references
public class Logika4
    0 references
    public static void Main()
        double Nilai;
            Console.Write("Nilai: ");
            Nilai = Convert.ToDouble(Console.ReadLine());
            if (Nilai >= 85)
                Console.WriteLine("Kamu mendapat grade A");
            else if (Nilai >= 65)
                Console.WriteLine("Kamu mendapat grade B");
            else if (Nilai >= 45)
                Console.WriteLine("Kamu mendapat grade C");
            else if (Nilai <= 25)
                Console.WriteLine("Kamu mendapat grade D");
```

Maliks-MacBook-Air:Sesi3 swijaya\$ gika4 && mono Logika4 Nilai: 70 Kamu mendapat grade B



C# Programming Logic - Sesi 03 NICE TO KNOW

Apabila terlalu banyak menggunakan pernyataan else if, kode program Anda bisa menjadi terlihat berantakan dan susah dibaca.

Pada skenario seperti ini, solusi yang lebih baik adalah menggunakan pernyataan switch.

Pernyataan switch pada dasarnya hanyalah menggantikan beberapa pernyataan else if



C# Programming Logic - Sesi 03 Statement Switch..Case..

Perintah switch adalah salah satu struktur kode yang pernah kita gunakan. Hampir selalu, struktur if dapat menggantikan struktur switch. Bentuk perintah switch pada C# adalah:

```
switch (varA) {
 case [literal 1]:
    //Blok kode 1
   break:
 case [literal 2]:
    //Blok kode 2
   break:
 case [literal 3]:
    //Blok kode 3
   break;
 default:
    //Jika yang lain tidak
terpenuhi
   break;
```



```
using System;
public class LogikaS
    O references
    public static void Main()
        int nilai;
        nilai = 7;
        switch (milai)
            case 1:{
               Console.WriteLine("Januari");
               break;)
            case 2:{
                Console.WriteLine("Februari");
               break;}
            case 3:{
                Console.WriteLine("Maret");
               break;}
            case 4:{
               Console.WriteLine("April");
               break;}
            case 5:{
                Console.WriteLine("Mei");
                                                                                          Maliks-MacBook-Air:Sesi
Juli
               break;}
            case 6:{
                Console.WriteLine("Juni");
               break;}
            case 7:{
                Console.WriteLine("Juli");
               break;}
            case 8:{
                Console.WriteLine("Agustus");
               breaks)
            case 9:{
                Console.WriteLine("September");
               break;)
            case 18:{
               Console.WriteLine("Oktober");
               breaks)
            case 11:{
               Console.WriteLine("November");
               break;)
            case 12:{
               Console.WriteLine("Desember");
               breaks)
            default:{
                Console.WriteLine("Bulan ?");
               break;}
            Console.ReadLine();
```

C# Programming Logic - Sesi 03 LOOPING

Konsep Dasar : Melakukan suatu hal dengan cara berulang.

Dalam Programming, kita menggunakan looping untuk meng-iterasi item didalam sebuah Collection atau elemen didalam sebuah array.

Dalam hal ini C# menyediakan kontruksi dasar yang dikenal sebagai "loop" untuk kita implementasikan logika iterasi.

Contoh metode looping seperti while, for & do..while

Mari kita bahas 1 per 1



C# Programming Logic - Sesi 03 While

while akan menjalankan atau mengeksekusi kode selama kondisi yang diberikan masih TRUE.

Untuk mendeklarasikan struktur while dapat dilihat pada potongan kode berikut:

```
while(kondisi)
{
    kodeA;
    kodeB;
    kodeC;
}
```

while akan terus mengeksekusi kodeA, kodeB dan kodeC secara terus-menerus hingga kondisi menjadi FALSE. Saat kondisi berubah menjadi FALSE, while akan berheti mengeksekusi kodeA, kodeB dan KodeC.



Contoh penggunaanya dapat dilihat pada potongan kode berikut:

```
using System;
     0 references
     public class Logika6
          O references
          public static void Main()
              // membuat variable
              int a = 10;
              // loop while
              while (a < 20) // menhasilkan TRUE jika variable a lebih kecil dari 20
                  Console.WriteLine("Nilai a = " + a );
                  a = a + 1;
14
17
```

Pada potongan kode tersebut dapat dilihat jika while akan terus melakukan:

- Print nilai variable a
- menambahkan nilai a sebanyak 1

Sehingga, program akan melakukan print dari 10 hingga 19 sebelum berhenti.



C# Programming Logic - Sesi 03 FOR

Berbeda dengan while, loop dengan for dapat dibatasi berapa kali iterasi yang harus dilalui sebelum berhenti.

```
for ( init; condition; increment )
{
   kode;
}
```



Untuk mengisi posisi init, condition dan increment lebih mudah jika langsung melihat contoh penggunaannya.

```
using System;
     0 references
     public class Logika7
         0 references
         public static void Main()
                         condition; increment )
             for (int a = 10; a < 20;
                                            a = a + 1
                Console.WriteLine("Iterasi ke - " + a);
11
12
13
```



Pada posisi init, kita mendeklarasikan variable yang akan digunakan untuk di evaluasi pada posisi condition.

Pada posisi condition, variable yang dideklarasikan pada posisi init akan dievaluasi setiap sebelum iterasi atau loop di jalankan.

Jika mengahasilkan TRUE iterasi akan berlanjut, dan jika FALSE iterasi akan berhenti. Biasanya kondisi berisikan pengecekan variable deklarasi lebih kecil dari nilai stop.

Pada posisi increment, nilai varibale pada posisi init akan ditambahkan sebanyak 1 atau sesuai kebutuhan. increment akan dijalankan setelah iterasi selesai atau sebelum variable deklarasi di evaluasi pada posisi condition.



C# Programming Logic - Sesi 03 Do..while

Pada for dan while kondisi di evaluasi atau dicek diawal atau sebelum iterasi di mulai. do..while melakukan evaluasi kondisi pada akhir iterasi.

do..while sendiri mirip sekali dengan while yang membedakannya hanyalah posisi evaluasi atau pengecekan kondisinya.

Karena do..while melakukan evaluasi pada akhir iterasi atau loop, maka dipastikan pasti akan menjalankan setidaknya satu kali iterasi.

```
do
{
   kode;
} while(condition);
```



```
using System;
      O references
      public class Logika8
          0 references
          public static void Main()
              int i = 0;
              do
                  Console.WriteLine("i = {0}", i);
11
12
                  1++;
13
                  if (i > 5)
                       break;
              } while (i < 10);
17
```

```
Maliks-MacBook-Air:Sesi3 swijaya$ cd "/Users/
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
Maliks-MacBook-Air:Sesi3 swijaya$ [
```



Mengenal CLİ, Git, Github & Remote Repository

GITHUB

Apa itu Version Control System?

Sistem yang mencatat semua perubahan yang dilakukan pada file sehingga semua riwayatnya akan terekam dan bisa dilihat kembali nanti. Saat developer membuat proyek baru, mereka selalu dan akan terus-menerus melakukan pembaruan terhadap kodenya. Bahkan, setelah proyeknya online, developer tetap harus memperbarui versinya, memperbaiki bug, menambahkan fitur baru, dan lain sebagainya.



Apa itu Git dan Kenapa penting bagi seorang programmer?



Git adalah salah satu tool yang sering digunakan dalam proyek pengembangan software.

Version control system membantu developer melacak perubahan yang mereka lakukan terhadap basis kode. Tak hanya itu, sistem ini juga mencatat siapa saja yang membuat perubahan serta memulihkan kode yang telah dihapus atau dimodifikasi.

Karena Git menyimpan banyak salinan kode di repositori, maka tidak ada kode yang saling tertimpa.

REPOSITORY

Repositori atau repo adalah direktori penyimpanan file proyek.

Di sini, Anda bisa menyimpan apa pun yang berkaitan dengan proyek yang sedang Anda buat, misalnya file kode, gambar, atau audio.

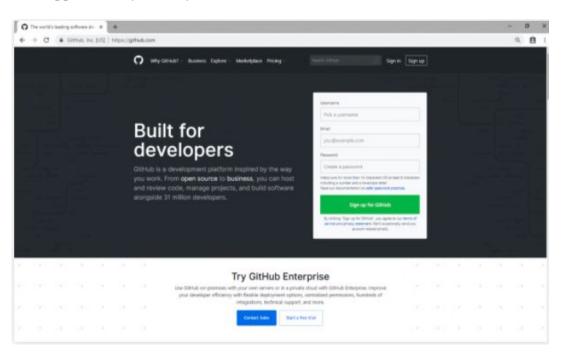
Repo sendiri bertempat di penyimpanan atau storage GitHub atau repositori lokal di komputer Anda.



BRANCH

Branch merupakan salinan dari repositori. Anda bisa menggunakan branch ketika akan melakukan suatu pengembangan atau *development* secara terpisah.

Pekerjaan atau *task* yang Anda kerjakan di branch tidak akan memengaruhi repositori pusat atau branch lainnya. Jika pengembangannya sudah selesai, Anda bisa menggabungkan branch saat ini ke branch lainnya dah juga repositori pusat dengan menggunakan *pull request*.





GitHub adalah manajemen proyek dan sistem *versioning code* sekaligus platform jaringan sosial yang dirancang khusus bagi para developer. Lalu, apa saja fungsi GitHub? Dengan platform ini, Anda bisa bekerja bersama-sama dengan rekan dari berbagai berlahan dunia, merencanakan proyek, dan bahkan *tracking* (melacak) pekerjaan Anda.

Silahkan Lakukan Sign Up/ Register terlebih dahulu.

Buat akun GitHub

- Masuk ke <u>Github</u> dan daftar.
- Tentukan username kamu yang baik, jelas, mudah diingat dan dibaca; dengan huruf kecil (misalnya hacktiv8 atau hacktiv atau Height).
- Konfirmasi email akun GitHub di inbox kamu.
- Kunjungi https://github.com/settings/profile lalu lengkapi profil kamu.
- "Update profile" kamu.
- Jika nanti kamu melihat <u>USERNAME</u>, artinya perlu diganti dengan username-kamu. Misalnya username kamu adalah hacktiv8, berarti github.com/<u>USERNAME</u> menjadi github.com/hacktiv8.



NICE TO KNOW

Git merupakan tool utama yang pasti akan kamu gunakan setiap hari, sepanjang karirmu sebagai developer. Dengan ini kamu dapat tahu apa saja yang kamu lakukan setiap saat dan berkolaborasi dengan rekan lainnya juga. Anggaplah Git sebagai mesin waktu untuk berbagai kerjaan dan file/folder yang kamu kelola.

Sedangkan GitHub merupakan tempat sekaligus komunitas untuk para developer berbagai code dan berkolaborasi dalam berbagai project software. Bahkan modern ini, profil GitHub dapat menggantikan resume/CV untuk melamar pekerjaan!

Kami ingin untuk kamu nyaman menggunakan Git dan GitHub sesegera mungkin. Semoga kamu juga bakal ketagihan dan bahkan nggak bisa bayangin hidup tanpa Git dan GitHub!

Kini kamu akan melalukan instalasi dan konfigurasi Git di komputermu, mampu menjelaskan dan membedakan Git dan GitHub, serta memahami manfaat dan kekuatan version control system (VCS) atau nama lainnya source code management (SCM).



Cara Install Git di Windows

Download Git

Silahkan buka website resminya Git (<u>git-scm.com</u>). Kemudian unduh Git sesuai dengan arsitektur komputer kita. Kalau menggunakan 64bit, unduh yang 64bit. Begitu juga kalau menggunakan 32bit.

Git merupakan tool utama yang pasti akan kamu gunakan setiap hari, sepanjang karirmu sebagai developer. Dengan ini kamu dapat tahu apa saja yang kamu lakukan setiap saat dan berkolaborasi dengan rekan lainnya juga. Anggaplah Git sebagai mesin waktu untuk berbagai kerjaan dan file/folder yang kamu kelola.

Sedangkan GitHub merupakan tempat sekaligus komunitas untuk para developer berbagai code dan berkolaborasi dalam berbagai project software. Bahkan modern ini, profil GitHub dapat menggantikan resume/CV untuk melamar pekerjaan!

Kami ingin untuk kamu nyaman menggunakan Git dan GitHub sesegera mungkin. Semoga kamu juga bakal ketagihan dan bahkan nggak bisa bayangin hidup tanpa Git dan GitHub!

Kini kamu akan melalukan instalasi dan konfigurasi Git di komputermu, mampu menjelaskan dan membedakan Git dan GitHub, serta memahami manfaat dan kekuatan version control system (VCS) atau nama lainnya source code management (SCM).



Q Search entire site...

About

Documentation

Downloads

GUI Clients Logos

Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to **read** online for free. Dead tree versions are available on Amazon.com.

Downloading Git



Your download is starting...

You are downloading the latest (2.30.2) 32-bit version of **Git for Windows**. This is the most recent maintained build. It was released 5 days ago, on 2021-03-09.

Click here to download manually, if your download hasn't started.

Other Git for Windows downloads

Git for Windows Setup 32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

Git for Windows Portable ("thumbdrive edition") 32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

The current source code release is version 2.30.2. If you want the newer version, you can build it from the source code.

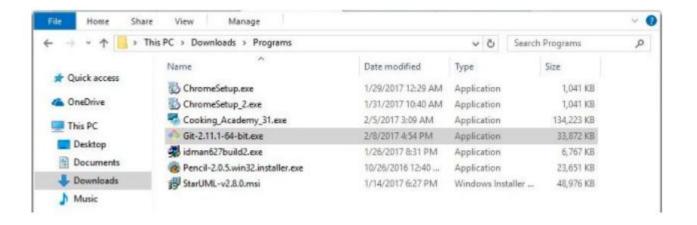
Now What?

Now that you have downloaded Git, it's time to start using it.



Langkah-langkah Install Git di Windows

Baiklah, mari kita mulai cara instalnya. Silahkan klik 2x file instaler Git yang sudah diunduh.



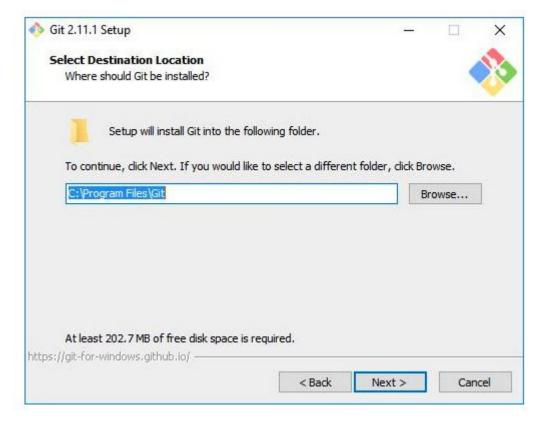


Maka akan muncul informasi lisensi Git, klik Next > untuk melanjutkan.



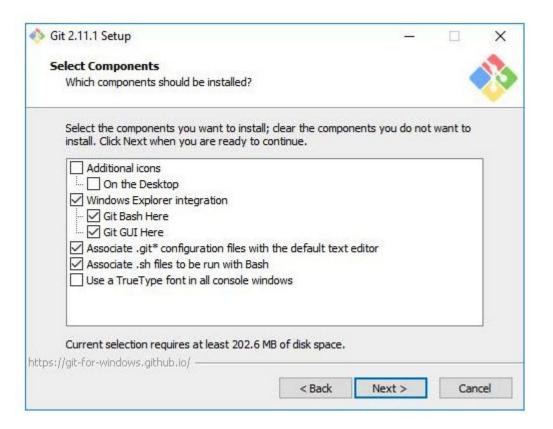


Selanjutnya menentukan lokasi instalasi. Biarkan saja apa adanya, kemudian klik Next >.



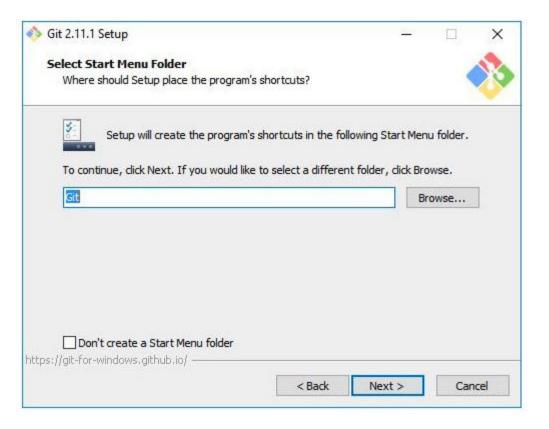


Selanjutnya pemilihan komponen, biarkan saja seperti ini kemudian klik Next >.



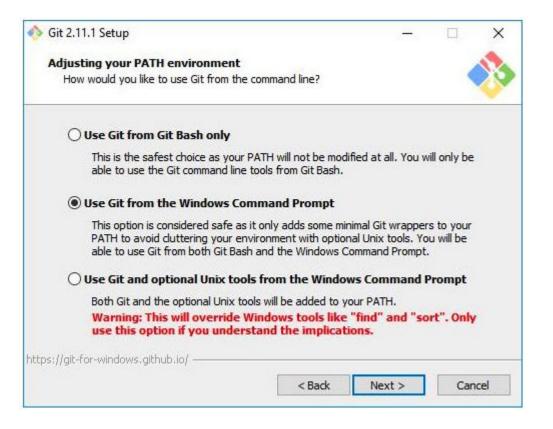


Selanjutnya pemilihan direktori start menu, klik *Next* >.



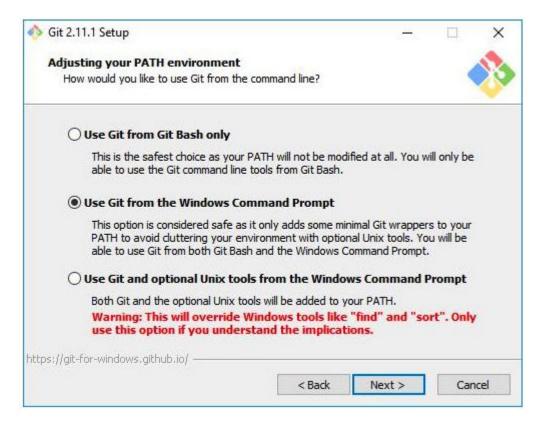


Selanjutnya pengaturan *PATH Environment*. Pilih yang tengah agar perintah git dapat di kenali di Command Prompt (CMD). Setelah itu klik *Next >*.



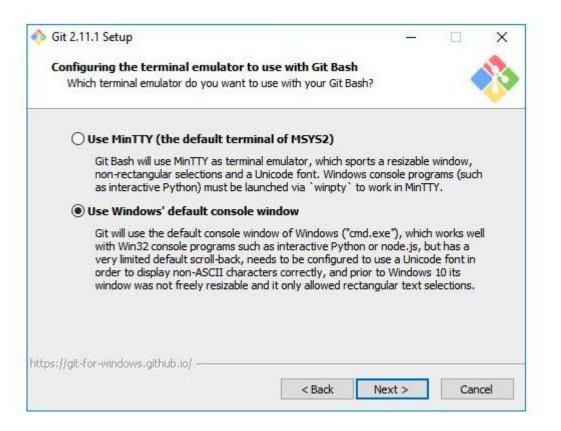


Selanjutnya konversi *line ending*. Biarkan saja seperti ini, kemudian klik *Next >*.



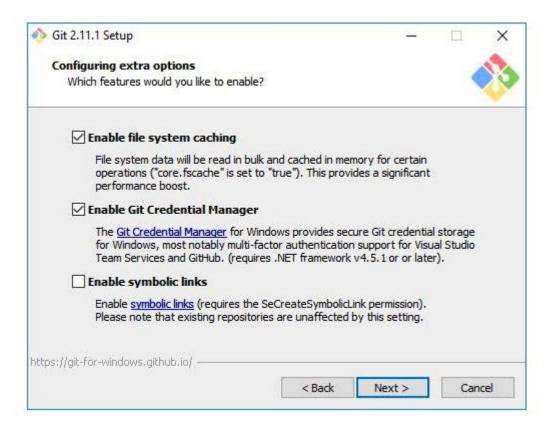


Selanjutnya pemilihan emulator terminal. Pilih saja yang bawah, kemudian klik Next >.



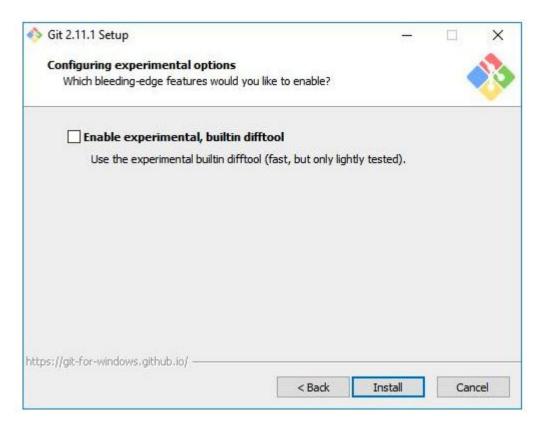


Selanjutnya pemilihan opsi ekstra. Klik saja Next >.



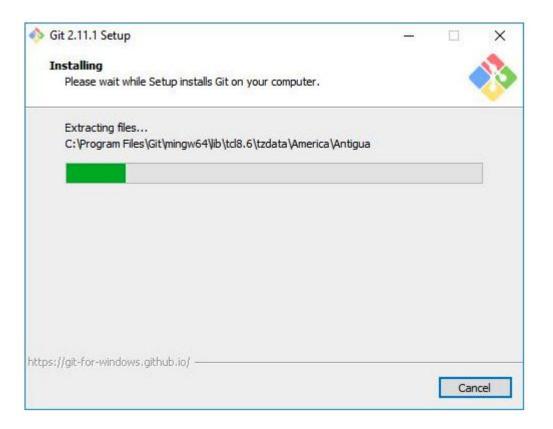


Selanjutnya pemilihan opsi eksperimental, langsung saja klik *Install* untuk memulai instalasi.





Tunggu beberapa saat, instalasi sedang dilakukan.





Setelah selesai, kita bisa langsung klik Finish.





Selamat, Git sudah terinstal di Windows. Untuk mencobanya, silahkan buka CMD atau PowerShell, kemudian ketik perintah git --version.

Konfigurasi Awal yang Harus Dilakukan

Ada beberapa konfigurasi yang harus dipersiapakan sebelum mulai menggunakan Git, seperti *name* dan *email*. Silahkan lakukan konfigurasi dengan perintah berikut ini.

```
git config --global user.name "Nama_Register_Github"
git config --global user.email user@email.com
```

Kemudian periksa konfigurasinya dengan perintah:

```
git config --list
```



Apabila berhasil tampil seperti gambar berikut ini, berarti konfigurasi berhasil.

```
[Maliks-Air:~ swijaya$ git config --list credential.helper=osxkeychain user.name=wijaysali user.email=wijaysali@gmail.com core.autocrlf=input filter.lfs.clean=git-lfs clean -- %f filter.lfs.smudge=git-lfs smudge -- %f filter.lfs.process=git-lfs filter-process filter.lfs.required=true Maliks-Air:~ swijaya$ ■
```

Konfigurasi core.editor bersifat opsional. Sedangkan name dan email wajib.

Jika kamu memiliki akun Github, Gitlab, Bitbucket atau yang lainnya...maka *username* dan *email* harus mengikuti akun tersebut agar mudah diintegrasikan.



Directions

Pastikan kamu siap akan pakai Git dengan command-line

Kamu perlu pengetahuan terminal atau command prompt serta CLI. Kamu harus sudah bisa:

- Navigasi ke berbagai directory/folder
- Membuat file dan folder baru
- Menghapus file dan folder
- Melihat berbagai file yang ada di folder
- Mengetahui lokasi folder di mana kita berada
- Memindahkan file ke berbagai folder
- Menyalin file atau folder ke tempat berbeda
- Membuka atau mengubah file ke dalam editor teks/code

Kamu bisa menggunakan referensi dibawah ini untuk mempelajari command line:

https://lifehacker.com/5633909/who-needs-a-mouse-learn-to-use-the-command-line-for-almost-anything



Best Practice buka terminal/cmd arahkan ke folder projects) terus lakukan: Git init Git add. Git status git commit -m 'your message' git remote add origin 'your_url_name' git push -u origin master kembali browser lagi terus refresh browser untuk cek sudah masuk ke github belum.. kalo sudah masuk berarti sudah berhasil dan selesai.