

BAB I

PENGANTAR STRUKTUR DATA

A. TUJUAN

1. Mahasiswa mampu mengetahui dan memahami pengantar struktur data.
2. Mahasiswa mampu memahami struktur data linear dan non-linear.
3. Mahasiswa mampu Memahami berbagai bentuk tipe data primitive.
4. Mahasiswa mampu menginstalasi Visual Studio Code dan software pendukungnya.
5. Mahasiswa mampu mengimplementasikan tipe data primitive berdasarkan kasus yang diberikan menggunakan bahasa pemrograman C++.

B. ALAT DAN BAHAN

1. Laptop
2. Mouse
3. IDE C++ (Visual Studio Code)

C. KESEHATAN DAN KESELAMATAN KERJA

1. Hati-hatilah dalam memakai perangkat elektronik
2. Pastikan kabel listrik terpasang dan dalam kondisi baik
3. Lakukan praktikum dalam posisi duduk yang benar
4. Jauhkan kabel listrik dari sentuhan anda
5. Gunakan alas kaki celana Panjang dan kemeja
6. Gunakan kacamata anti radiasi layar

D. TEORI

1. Pendahuluan

Setiap kurikulum ilmu komputer di dunia mencakup kursus tentang struktur data dan algoritma. Struktur data sangat penting; struktur data meningkatkan kualitas hidup kita dan bahkan secara terstruktur membantu kehidupan kita. Banyak perusahaan besar telah dibangun seputar struktur data.

Bagaimana ini bisa terjadi? Jika kita berhenti untuk memikirkannya, kita akan menyadari bahwa kita berinteraksi dengan struktur data secara konstan hamper setiap harinya dan bahkan setiap saat. Perhatikan hal-hal yang pernah anda lakukan seperti berikut ini:

- Membuka file: Struktur data sistem file digunakan untuk mencari bagian-bagian file tersebut pada disk sehingga mereka dapat ditampilkan. Hal ini tidak mudah karena disk sebagai media

penyimpanan berisi ratusan juta blok. Konten dari file yang anda cari berada di salah satu blok tersebut.

- Mencari kontak pada ponsel: Struktur data digunakan untuk mencari nomor telepon dalam daftar kontak Anda berdasarkan informasi parsial bahkan sebelum Anda selesai memanggil / mengetik. Ini tidak mudah; telepon Anda mungkin berisi informasi tentang banyak orang — setiap orang yang pernah Anda hubungi melalui telepon atau email — dan ponsel Anda tidak memiliki prosesor yang sangat cepat atau banyak memori.
- Log in ke media social: Server jaringan menggunakan informasi login Anda untuk mencari informasi akun Anda. Ini tidak mudah; jejaring sosial paling populer memiliki ratusan juta pengguna aktif.
- Pencarian web: Mesin pencari menggunakan struktur data untuk menemukan halaman web yang mengandung istilah pencarian Anda. Ini tidak mudah; ada lebih dari 8,5 miliar halaman web di Internet dan setiap halaman berisi banyak istilah pencarian potensial.

2. Kebutuhan Efisiensi

Di bagian selanjutnya, kita melihat operasi yang didukung oleh struktur data yang paling umum digunakan. Siapa pun yang memiliki sedikit pengalaman pemrograman akan melihat bahwa operasi ini tidak sulit untuk diimplementasikan dengan benar.

Kita dapat menyimpan data dalam array atau linked list dan setiap operasi dapat diimplementasikan dengan mengulangi semua elemen dari array atau list dan mungkin juga menambahkan atau menghapus elemen.

Implementasi semacam ini mudah, tetapi tidak terlalu efisien. Apakah ini penting? Komputer menjadi lebih cepat dan lebih cepat. Mungkin implementasi yang jelas sudah cukup baik. Mari kita lakukan perhitungan kasar untuk mengetahuinya.

- Jumlah operasi: Bayangkan aplikasi dengan kumpulan data berukuran sedang, misalkan satu juta (10^6) item. Masuk akal, dalam sebagian besar aplikasi, untuk menganggap bahwa aplikasi akan ingin mencari setiap item setidaknya sekali. Ini berarti kami dapat berharap untuk melakukan setidaknya satu juta (10^6) pencarian dalam data ini. Jika masing-masing dari 10^6 pencarian memeriksa masing-masing dari 10^6 item, ini memberikan total $10^6 \times 10^6 = 10^{12}$ (seribu miliar) inspeksi.
- Kecepatan prosesor: Pada saat penulisan, bahkan komputer desktop yang sangat cepat tidak dapat melakukan lebih dari satu miliar (10^9) operasi per detik. Berarti aplikasi ini akan memakan waktu setidaknya $10^{12} / 10^9 = 1000$ detik, atau kira-kira 16 menit dan 40 detik.
- Kumpulan data yang lebih besar: Sekarang pertimbangkan sebuah perusahaan seperti Google, yang mengindeks lebih dari 8,5 miliar halaman web. Dengan perhitungan, melakukan segala

jenis pertanyaan atas data ini akan memakan waktu setidaknya 8,5 detik. Kita sudah tahu bahwa ini bukan masalahnya; pencarian web selesai dalam waktu kurang dari 8,5 detik, dan mereka melakukan pertanyaan yang jauh lebih rumit daripada hanya menanyakan apakah halaman tertentu ada dalam daftar halaman yang diindeks. Pada saat penulisan, Google menerima sekitar 4.500 permintaan per detik, yang berarti bahwa mereka akan membutuhkan setidaknya $4500 \times 8,5 = 38,250$ detik, server yang sangat cepat hanya untuk mengatasi masalah pencarian.

3. Defenisi Struktur Data

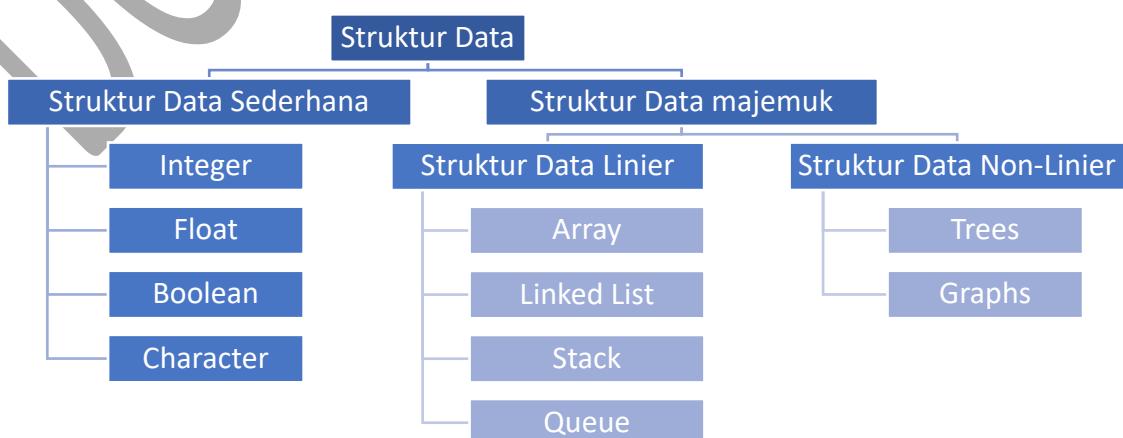
Struktur data terdiri atas dua kata yakni "struktur" dan "data". Data adalah catatan atas kumpulan fakta. Data merupakan bentuk jamak dari datum, berasal dari bahasa Latin yang berarti "sesuatu yang diberikan". Dalam penggunaan sehari-hari data berarti suatu pernyataan yang diterima secara apa adanya. Pernyataan ini adalah hasil pengukuran atau pengamatan suatu variabel yang bentuknya dapat berupa angka, kata-kata, atau citra.

Struktur adalah cara sesuatu disusun atau dibangun; susunan; bangunan; Struktur juga bisa berarti sesuatu yang disusun dengan pola tertentu; pengaturan unsur atau bagian suatu benda; ketentuan unsur-unsur dari suatu benda; atau pengaturan pola dalam bahasa secara sintagmatis;

Jadi struktur data secara sederhana dapat didefinisikan sebagai sebuah skema organisasi, yang diterapkan pada data sehingga data dapat diinterpretasikan dan sehingga operasi-operasi spesifik dapat dilaksanakan pada data tersebut.

Lebih sederhana lagi dalam konteks pemrograman, struktur data dapat didefinisikan sebagai cara terprogram untuk menyimpan data sehingga data dapat digunakan secara efisien. Pemakai struktur data yang tepat didalam proses pemrograman akan menghasilkan algoritme yang lebih jelas dan tepat, sehingga menjadikan program secara keseluruhan lebih efisien dan sederhana.

4. Peta Konsep Struktur Data



Gambar 1.1. Jenis struktur data

Seperti nampak pada Gambar 1, struktur data terdiri atas 2 jenis, yaitu:

- Struktur data sederhana
 - Integer
 - Float
 - Boolean
 - Character
- Struktur data majemuk
 - Struktur data liner
 - Array
 - Linked list
 - Stack
 - Queue
 - Struktur data non-liner
 - Trees
 - Graphs

Struktur data primitive/sederhana atau bisa disebut dengan tipe data primitif merupakan wadah data yang paling sederhana. Struktur data primitive hanya mampu menyimpan satu nilai tiap satu variabelnya. Oleh karena itu disebut juga sebagai tipe data primitive.

Struktur data linear adalah kumpulan komponen-komponen yang tersusun membentuk satu garis linear (berurutan). Bila komponen-komponen ditambahkan (atau dikurangi), maka struktur-struktur tersebut berkembang (atau menyusut). Pemakaian struktur data yang tepat di dalam proses pemrograman akan menghasilkan algoritma yang lebih jelas dan tepat, sehingga menjadikan program secara keseluruhan lebih efisien dan sederhana. Sedangkan Struktur data Non-Linier adalah adalah struktur data yang tersusun tidak secara berurutan.

5. Perbedaan Struktur Data Linear dan Non-Linear

No	Linear	Non-Linear
1.	Dalam struktur data linier, elemen data disusun dalam urutan linier dimana setiap elemen melekat pada yang berdekatan sebelumnya dan berikutnya.	Dalam struktur data non-linear, elemen data dilampirkan secara hierarkis.
2.	Dalam struktur data linear hanya terdapat satu level data	Dalam struktur data non-linear, bisa terdapat beberapa atau banyak level data

3.	Implementasinya mudah dibandingkan dengan struktur data non-linear.	Sedangkan implementasinya kompleks dibandingkan dengan struktur data linier.
4.	Dalam struktur data linier, elemen data dapat dilalui dalam sekali jalan saja.	Sedangkan pada struktur data non-linier, elemen data tidak dapat dilalui dalam sekali jalan saja.
5.	Dalam struktur data linier, memori tidak digunakan secara efisien.	Sementara dalam struktur data non-linear, memori digunakan secara efisien.
6.	Contohnya adalah: array, stack, queue, linked list, dll.	Sedangkan contohnya adalah: pohon dan grafik.
7.	Aplikasi struktur data linier terutama dalam pengembangan perangkat lunak aplikasi.	Aplikasi struktur data non-linier adalah pada Artificial Intelligence dan pengolahan citra.

6. Jenis/ Tipe Data

Tipe data berfungsi untuk merepresentasikan jenis dari nilai yang terdapat dalam program. Sebagai contoh kita memiliki suatu data dengan nilai 2, maka 2 termasuk ke dalam tipe bilangan bulat. Begitu juga dengan data yang bernilai “**Saya mahasiswa Tekom**” maka data tersebut dikategorikan bertipe string. Dalam C++ pada saat mendeklarasikan suatu variable, kita perlu menyebutkan tipe data dari variable tersebut. Dalam C++ terdapat beberapa tipe data dasar yang telah didefinisikan yaitu tipe data bilangan bulat (integer), bilangan riil (Floating-point), tipe logika (Boolean), tipe karakter (char). Tipe-tipe data tersebut adalah tipe data yang siap digunakan tanpa harus didefinisikan terlebih dahulu.

6.1. Tipe Bilangan Bulat

Tipe data integer atau tipe data bilangan bulat digunakan untuk merepresentasikan data numerik yang berupa angka bilangan bulat, yaitu angka yang tidak mengandung angka di belakang koma. Contoh bilangan bulat yakni 1, 5, 11, 99, dan sebagainya. Adapun data yang termasuk ke dalam tipe ini adalah dapat dilihat pada tabel di bawah ini:

Tabel 1.1. Tipe data bilangan bulat

Tipe Data	Ukuran (dalam bit)	Rentang
int	16 atau 32	-32.768 sampai 32.767 atau -2.147.483.648 sampai 2.147.483.647

unsigned int	16 atau 32	0 sampai 65.535 atau 0 sampai 4.294.967.295
signed int	16 atau 32	Sama seperti int
short int	16	-32.768 sampai 32.767
unsigned short int	16	0 sampai 65.535
signed short int	16	Sama seperti short int
long int	32	-2.147.483.648 sampai 2.147.483.647
signed long int	32	Sama seperti long int
unsigned long int	32	0 sampai 4.294.967.295

6.2. Tipe bilangan ril

Tipe ini adalah tipe yang merepresentasikan data-data bilangan yang mengandung angka di belakang koma, misalnya, 1.2, 3.5, 90.5, dan sebagainya. Adapun tipe yang termasuk ke dalam kategori ini dapat dilihat pada tabel berikut:

Tabel 1.2. Tipe data bilangan ril

Tipe Data	Ukuran (dalam bit)	Rentang	Presisi
float	32	1.2E-38 sampai 3.4E+38	6 digit
double	64	2.3E-308 sampai 1.7E+308 (15 Digit)	15 digit
long double	80	3.4E-4932 sampai 1.1e+4932 (19 digit)	19 Digit

6.3. Tipe logika

Tipe ini adalah tipe yang merepresentasikan data-data yang hanya mengandung dua buah nilai, yaitu nilai logika (Boolean). Nilai logika sendiri hanya terdiri dari nilai benar (1) dan salah (0) dalam C++ standar, tipe ini dinyatakan dalam bool, yang nilainya berupa true atau false.

6.4. Tipe karakter

Tipe ini digunakan untuk merpresentasikan data bertipe karakter, seperti ‘A’, ‘a’, ‘9’, ‘&’, dan sebagainya. Dalam C++ tipe karakter dinyatakan dalam char. Sedangkan untuk karakter Unicode (wide character) dinyatakan dengan wchar_t. Adapun yang termasuk ke dalam tipe data ini dapat dilihat pada tabel berikut:

Tabel 1.3. Tipe data karakter

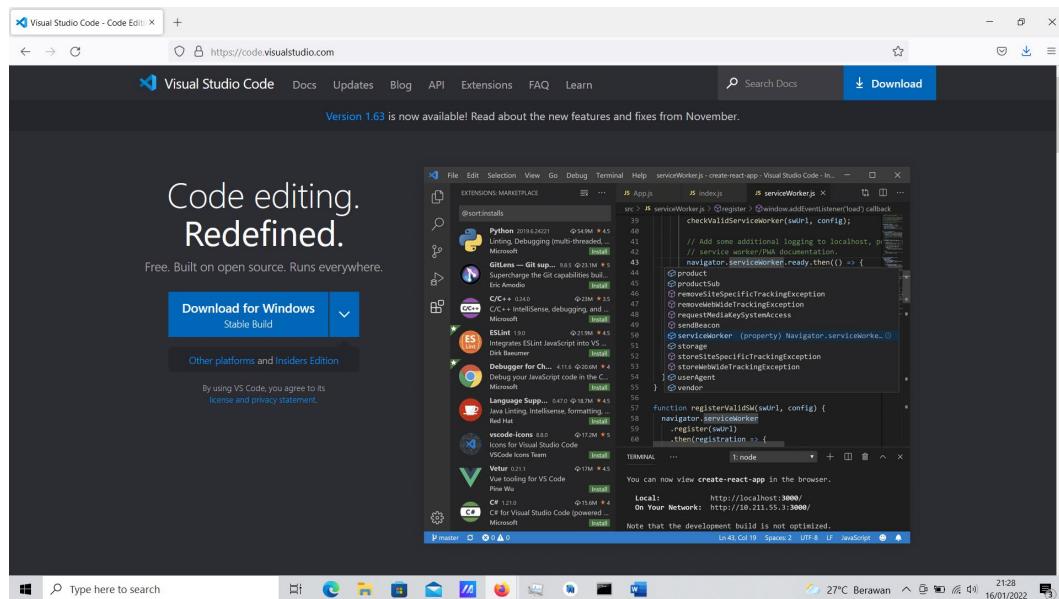
Tipe Data	Ukuran (dalam bit)	Rentang
char	8	1.2E-38 sampai 3.4E+38 (6 Digit)

donot copy

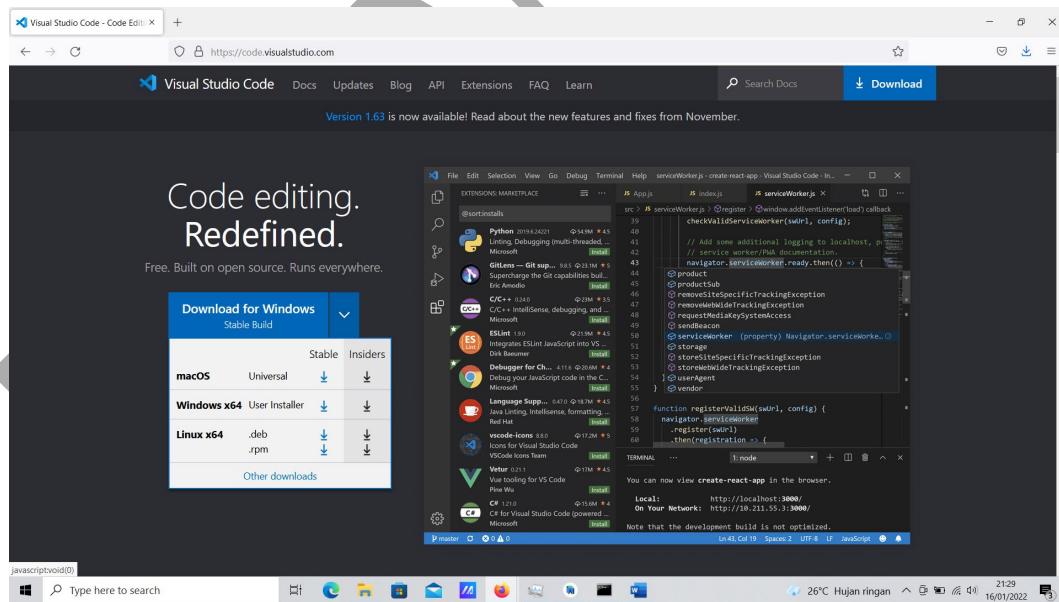
E. PRAKTIKUM

Praktikum 1.1. Instalasi Visual Studio Code

1. Kunjungi laman “Visual Studio Code”

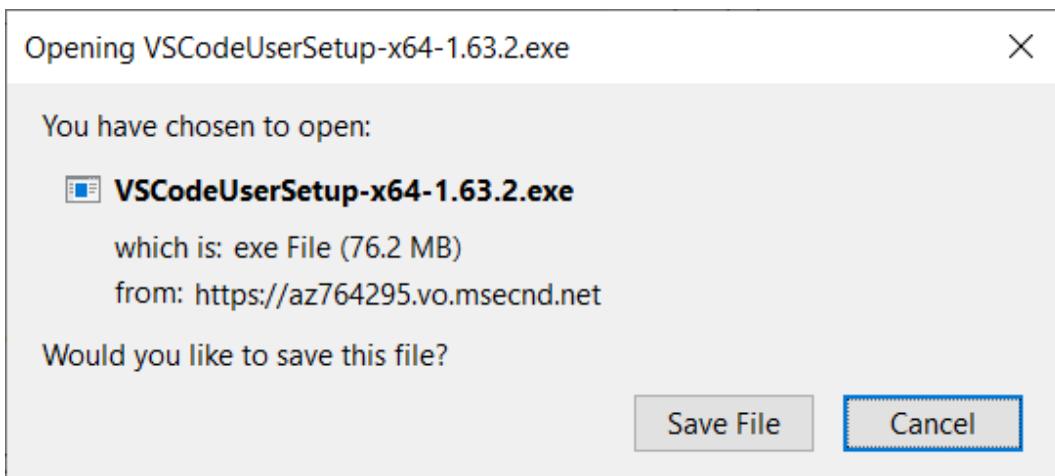


2. Klik menu drop down “Download for Windows”

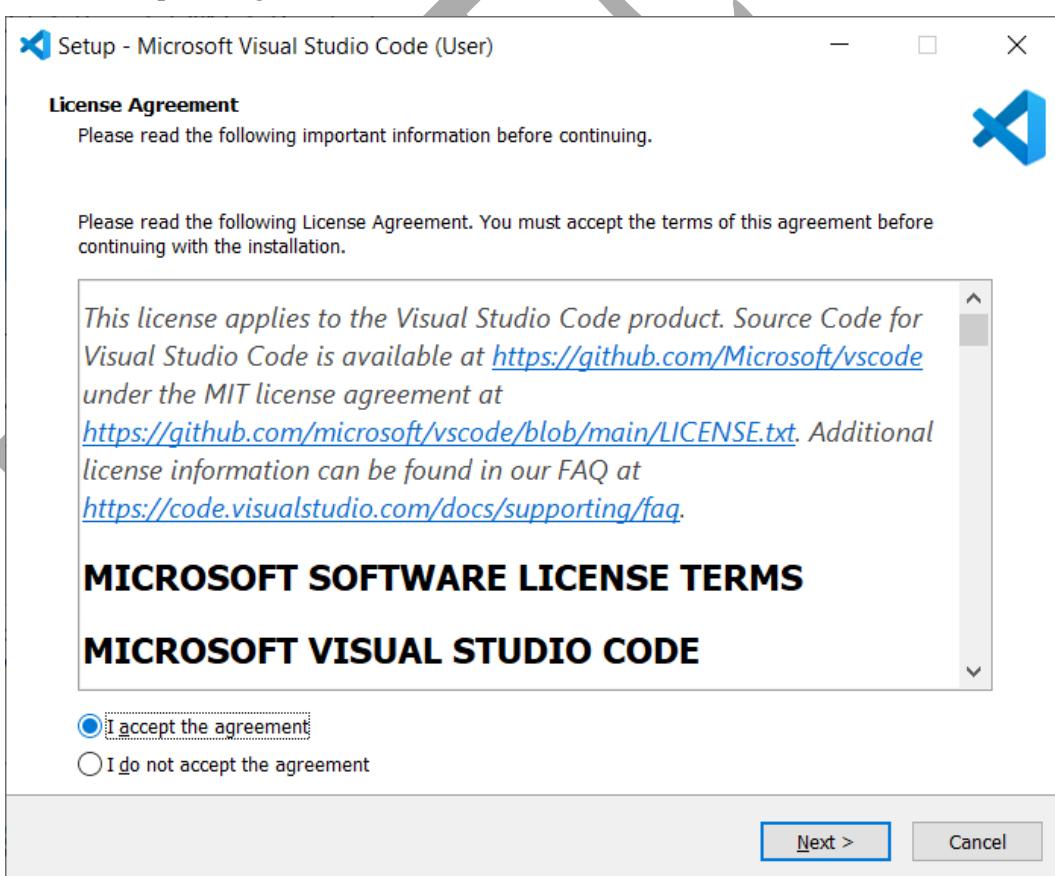


3. Pilih dan klik tombol download pada bagian stable untuk windows x64 (atau sesuai OS yang anda gunakan)

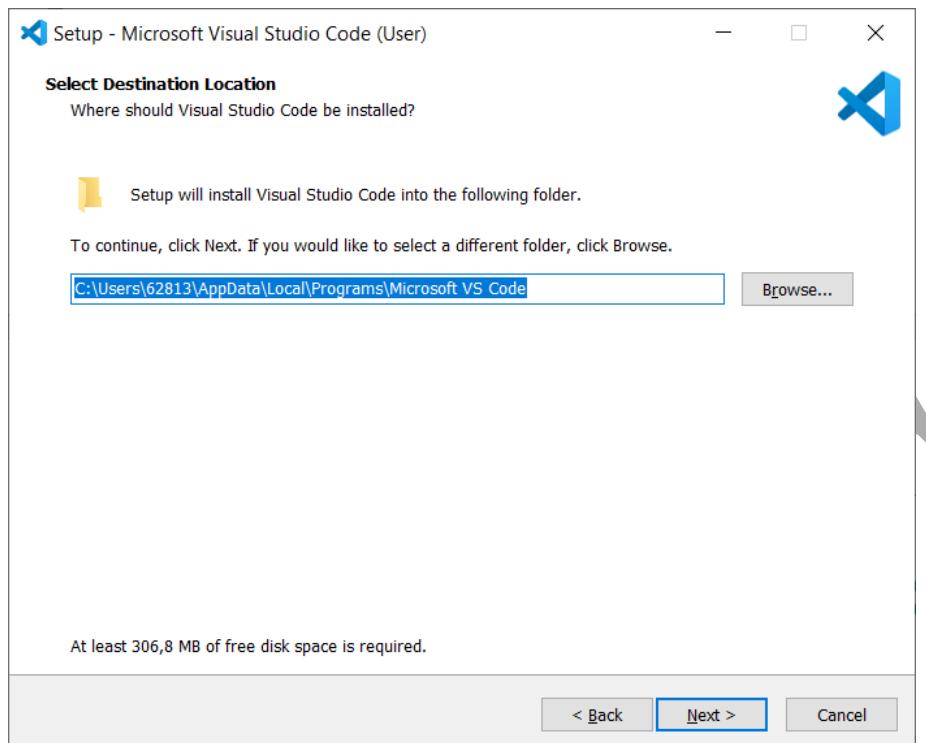
-
4. Pada kotak dialog yang muncul, klik “Save”.



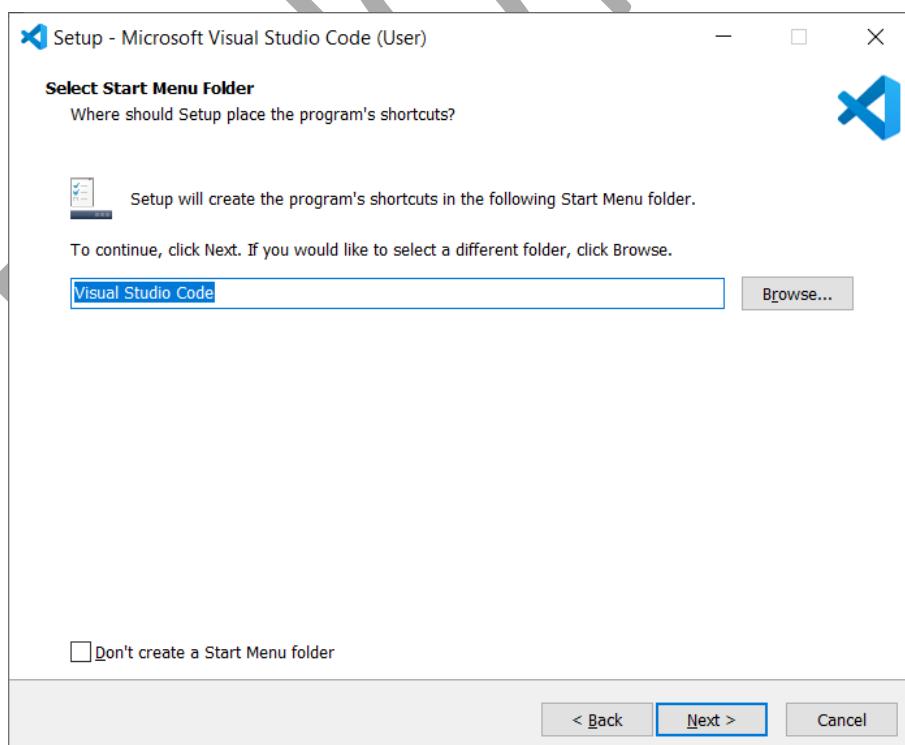
5. Setelah proses download selesai, klik dua kali untuk jalankan file “VSCodeUserSetup-x64-1.63.2” tersebut.
6. Pilih “I Accept the Agreement”.



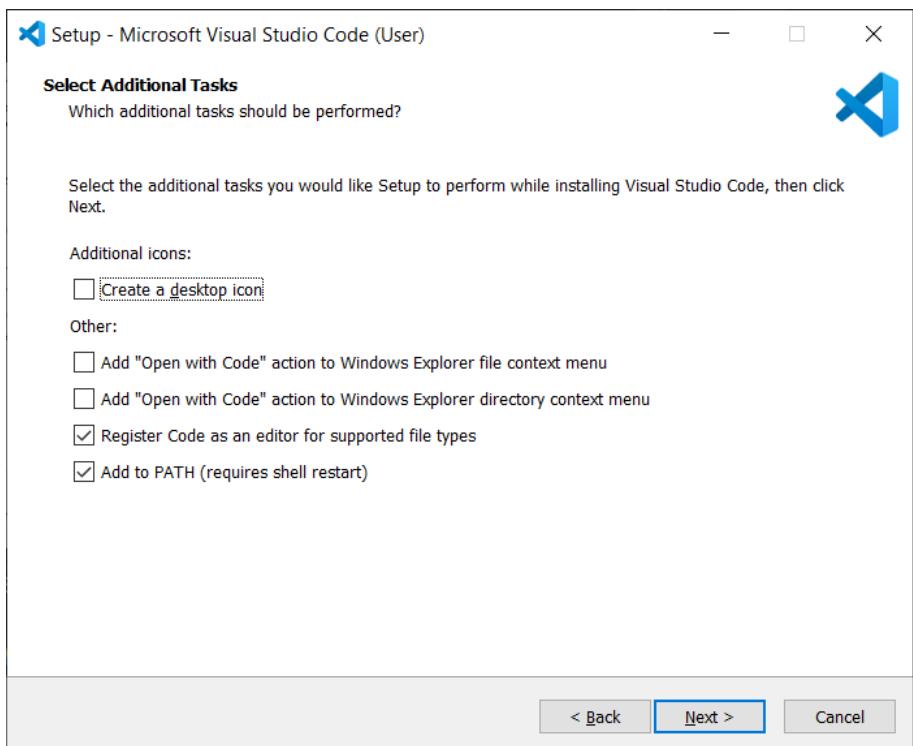
7. Pada jendela “Select Destination Location”, klik “Next”.



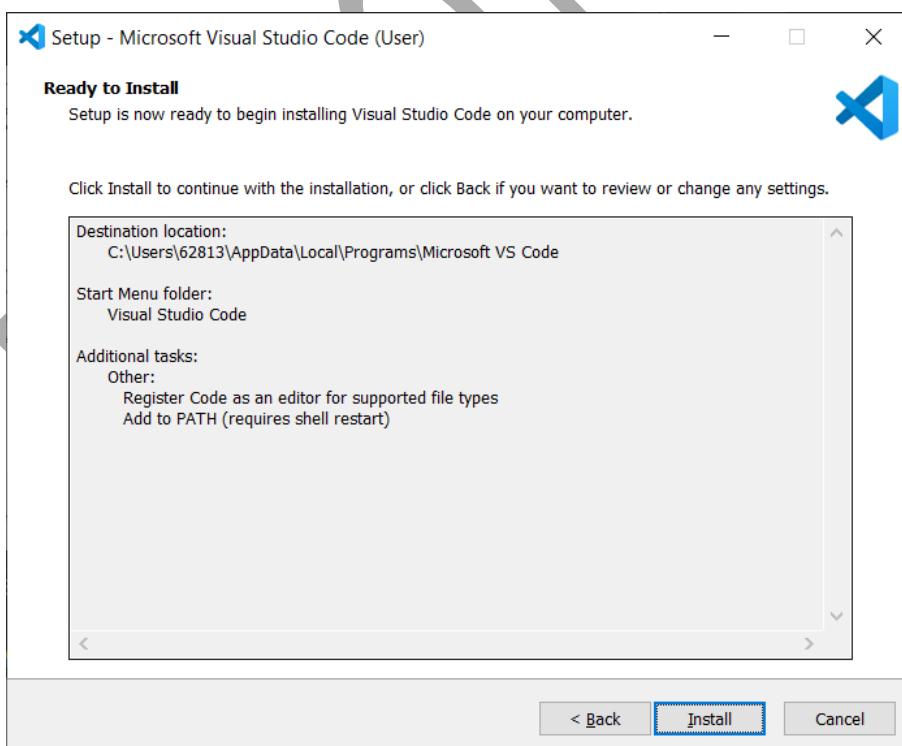
8. Pada jendela “Select Start Menu Folder”, klik “Next”.



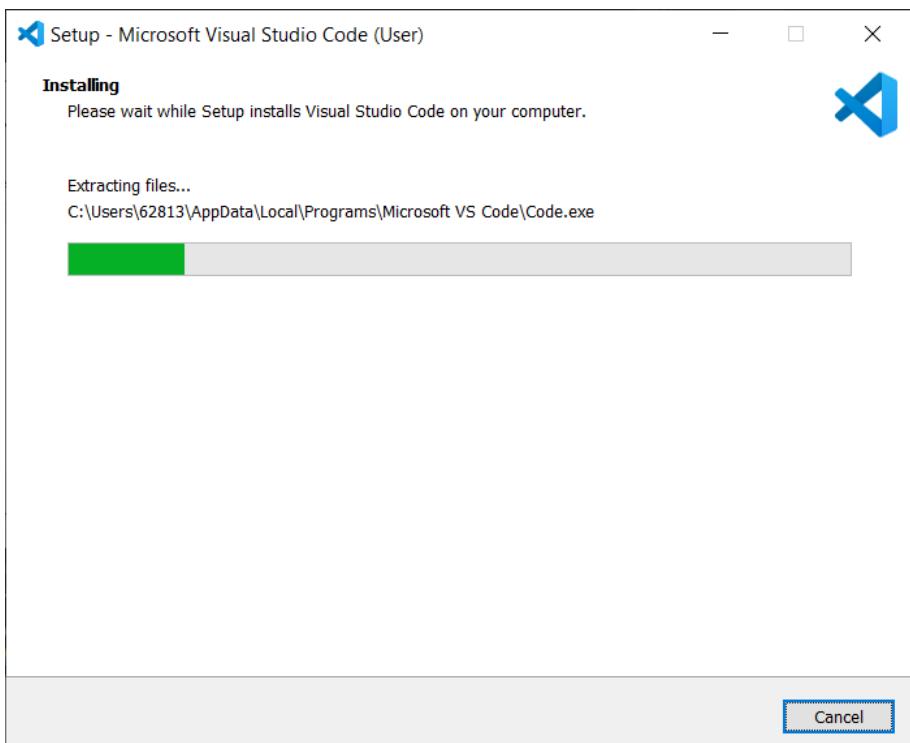
-
9. Pada jendela “Select additional task”, klik “Next”.



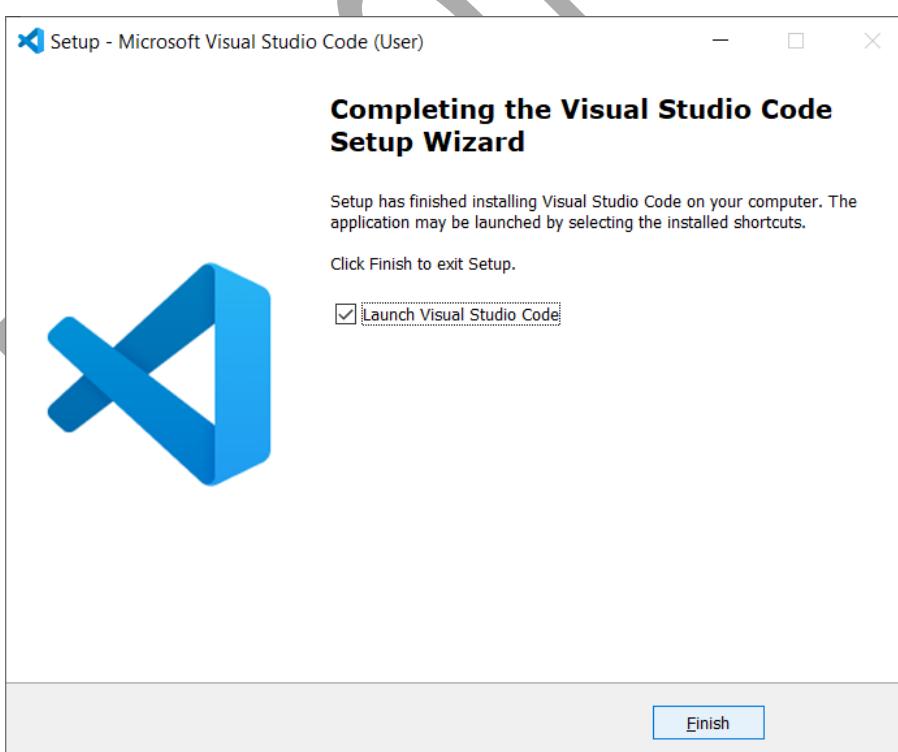
10. Pada jendela “Ready to Install” pastikan semua pengaturan benar, lalu klik “Next”.



11. Tunggu hingga proses penginstalan selesai.

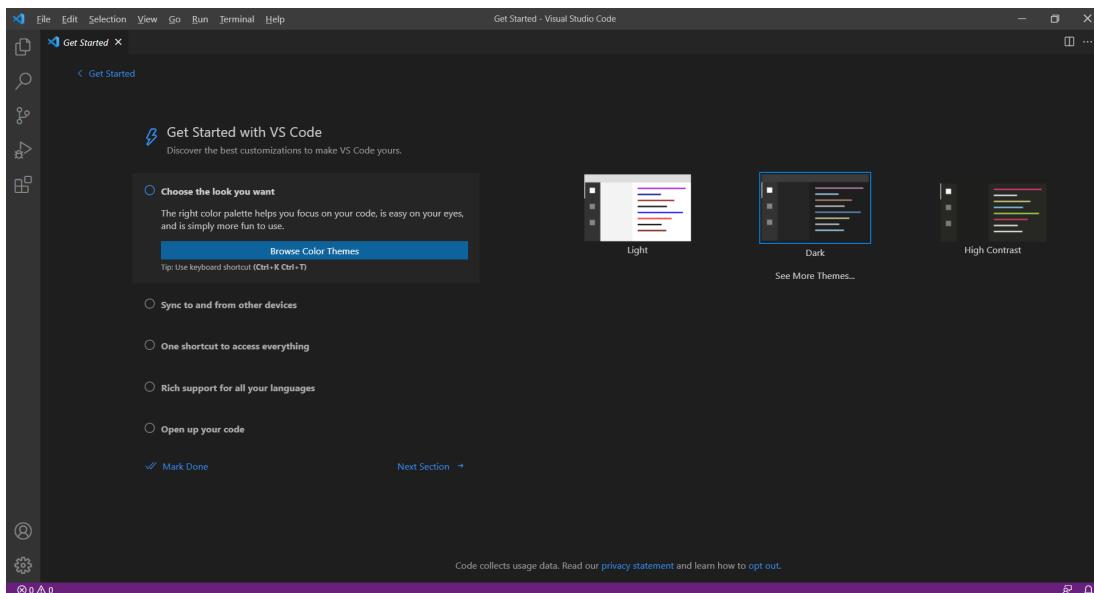


12. Jika telah selesai, klik "Finish".



13. Buka Visual Studio Code yang telah anda instal. Tampilan awal kurang lebih seperti berikut.

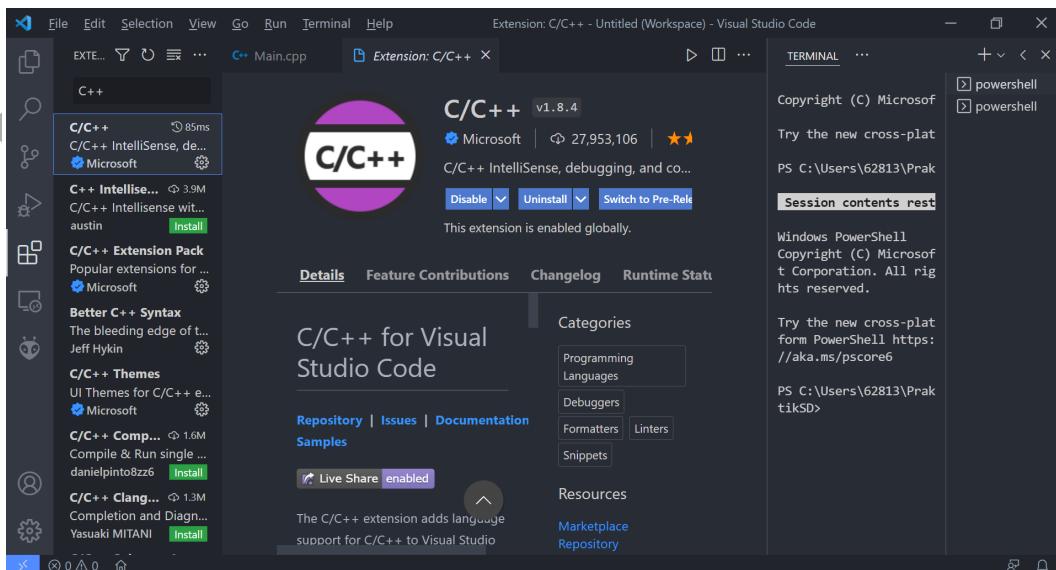
Silahkan pilih mode yang ingin anda gunakan.



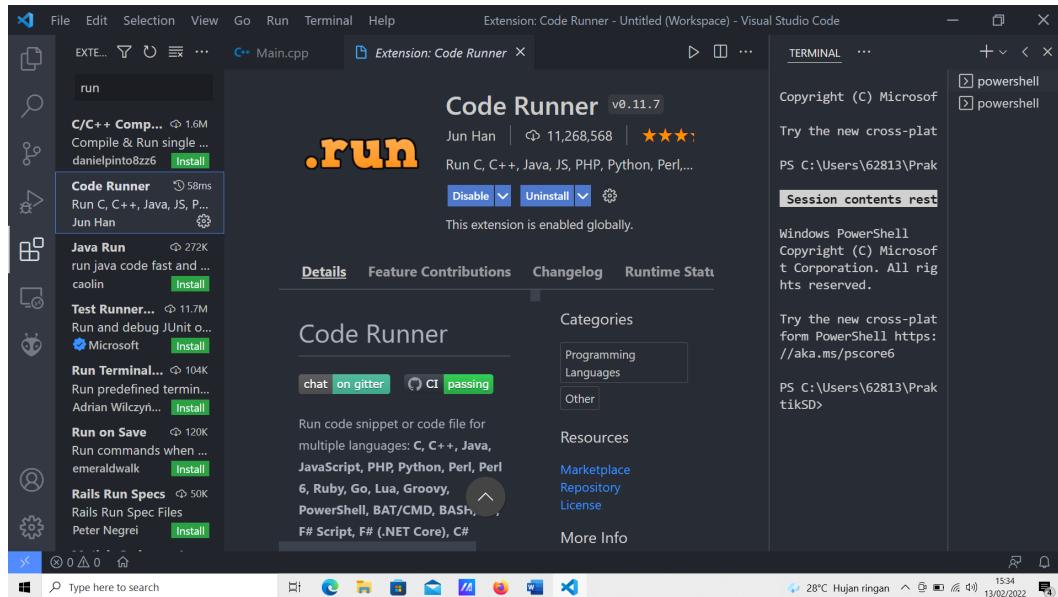
14. Visual Studio Code berhasil di instal.

Praktikum 1.2. Menginstal ekstensi

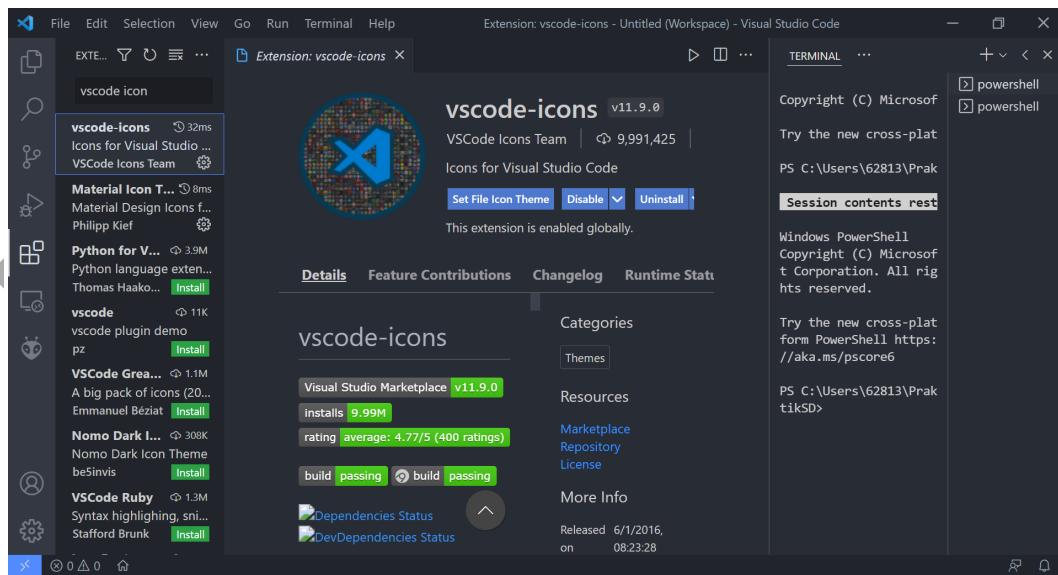
- Pilih dan klik “Extensions”.
- Pada bagian search, ketikkan “C++”.
- Pilih dan klik C/C++, lalu klik “Install”.



-
4. Extensi C/C++ berhasil anda install.
 5. Selanjutnya pada kolom search ketikkan “Code Runner”.
 6. Klik “Install”.



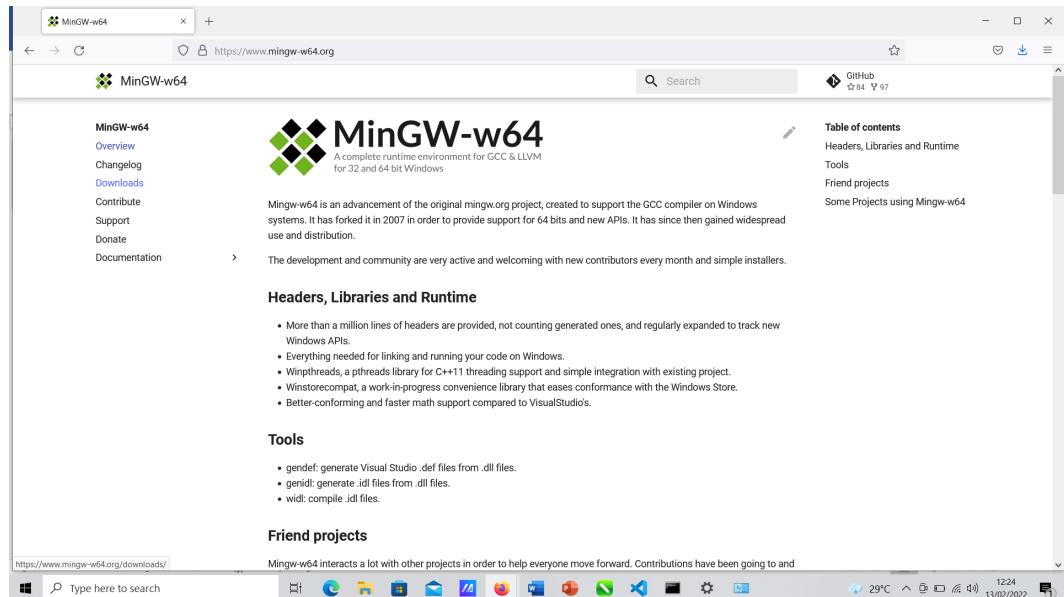
7. Ekstensi Code Runner berhasil anda install.
8. Masih pada kolom search Extensions, cari “vscode icon”.
9. Klik “Install”.



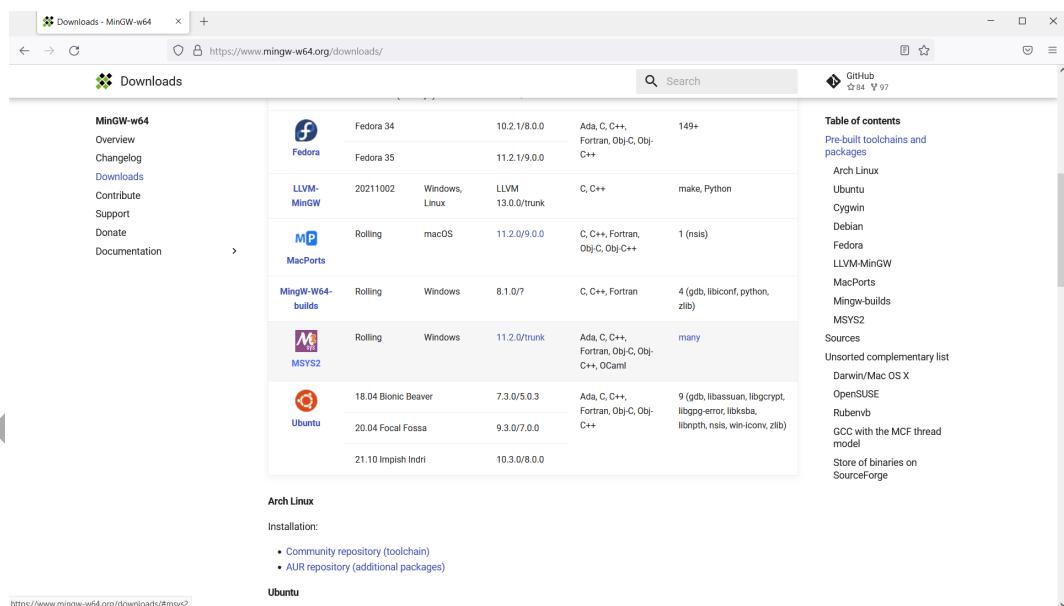
10. Ekstensi vscode icon berhasil diinstall
-

Praktikum 1.3. Menginstal Compiler C++

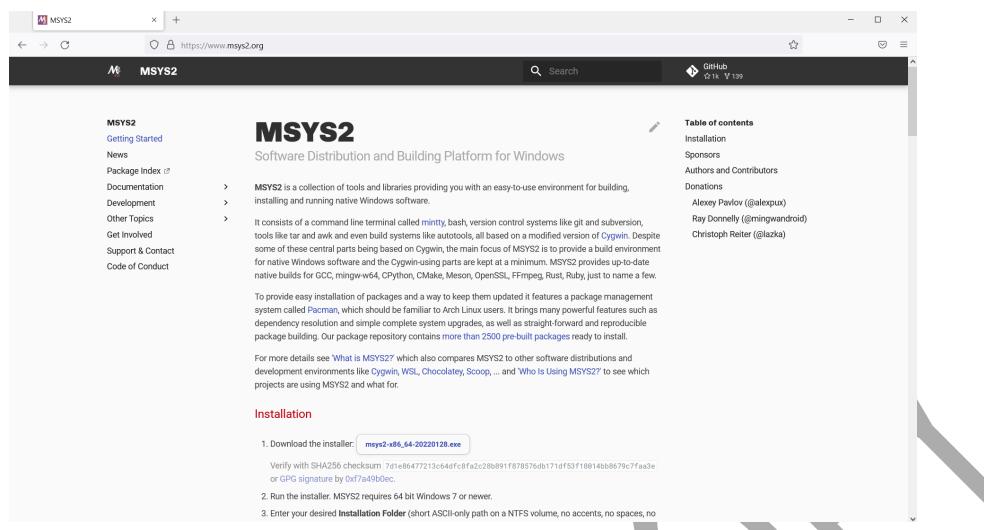
1. Kunjungi laman web <https://www.mingw-w64.org> lalu pilih dan klik “download”.



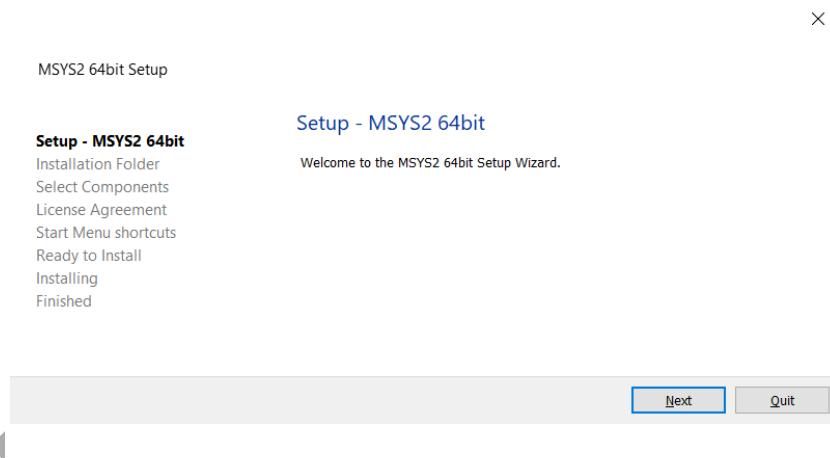
2. Pada halaman yang tampil, pilih dan klik “MSYS2”.



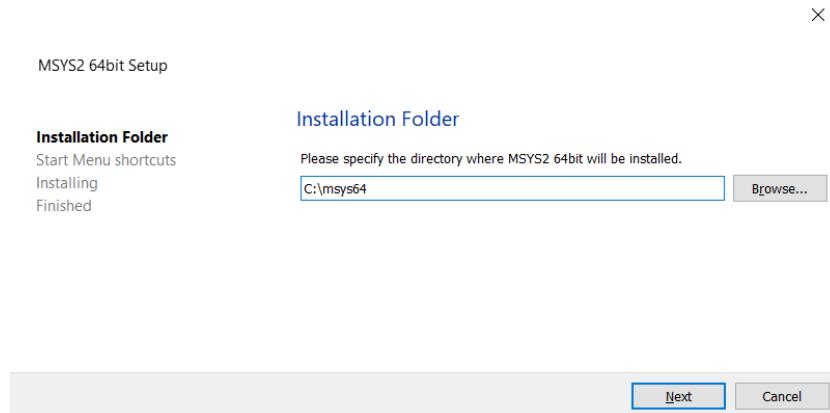
3. Pada halaman MYSYS2 yang tampil, Pilih dan klik download installer “[msys2-x86_64-20220128.exe](#)”



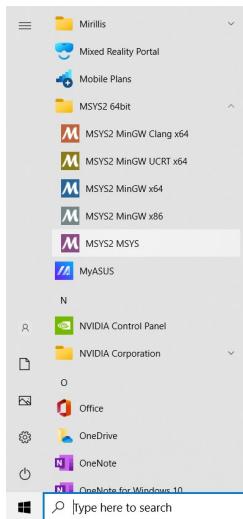
4. Tunggu hingga proses download selesai.
5. Klik 2 kali file instalasi maka akan mucul jendela berikut, lalu klik “Next”.



6. Biarkan tempat instalasi secara default, tidak perlu diubah. Klik “Next”.



-
7. Tunggu hingga proses instalasi selesai, lalu klik “Finish”.
 8. Pada menu “Start”, pilih dan klik “MSYS2 MSYS” (icon berwarna ungu).



9. Pada “Command Window” yang terbuka, ketikkan “pacman -Syu”, kemudian tekan “Enter” pada keyboard anda.

```
62813@LAPTOP-4K50002L MSYS ~
$ pacman -Syu
:: Synchronizing package databases...
mingw32           15.7 KiB  0.00  B/s --:-- [-----]  1%
mingw64           15.7 KiB  0.00  B/s --:-- [-----]  1%
ucrt64          352.0 KiB  380 KiB/s 00:03 [####-----] 22%
clang64          15.7 KiB  0.00  B/s --:-- [-----]  1%
msys            111.9 KiB  50.5 KiB/s 00:05 [#####-----] 29%
```

10. Ketikkan “Y”, lalu tekan “Enter” dan tunggu hingga proses instalasi selesai.

```
62813@LAPTOP-4K50002L MSYS ~
$ pacman -Syu
warning: terminate other MSYS2 programs before proceeding
resolving dependencies...
looking for conflicting packages...

Packages (3) filesystem-2022.01-2 msys2-runtime-3.3.4-1
                                         pacman-mirrors-20220126-1

Total Download Size:  3.31 MiB
Total Installed Size: 11.80 MiB
Net Upgrade Size:    0.00 MiB

:: Proceed with installation? [Y/n] Y
:: Retrieving packages...
pacman-mirrors-2-...   3.7 KiB  1823  B/s 00:02 [#####-----] 100%
filesystem-2022....   64.0 KiB  49.4 KiB/s 00:00 [#####-----] 60%
msys2-runtime-3....   155.5 KiB  72.4 KiB/s 00:43 [-----]  4%
Total (1/3)        223.2 KiB  127 KiB/s 00:24 [-----]  6%
```

11. Masih pada “Command Window” yang terbuka ketikkan “pacman -Su”, kemudian tekan “Enter”.



```
M ~
62813@LAPTOP-4K50002L MSYS ~
$ pacman -Su
:: Starting core system upgrade...
there is nothing to do
:: Starting full system upgrade...
resolving dependencies...
looking for conflicting packages...

Packages (9) bsdtar-3.5.3-1 libexpat-2.4.4-1 libgnutls-3.7.3-1
                   libgpg-error-1.44-1 libgpgme-1.17.0-1 libp11-kit-0.24.1-1
                   libzstd-1.5.2-1 p11-kit-0.24.1-1 zstd-1.5.2-1

Total Download Size:    4.16 MiB
Total Installed Size:  15.02 MiB
Net Upgrade Size:      -0.55 MiB

:: Proceed with installation? [Y/n] Y
```

12. Ketik “Y” lalu tekan “Enter”. Tunggu hingga proses instalasi selesai. (jika gagal ulangi perintah kembali dan pastikan koneksi internet memadai)

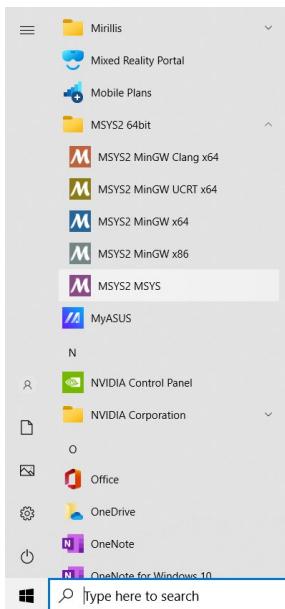


```
M ~
libgpg-error-1.4... 183.2 KiB 125 KiB/s 00:01 [#####] 100%
libgnutls-3.7.3... 1380.0 KiB 191 KiB/s 00:07 [#####] 100%
libexpat-2.4.4-1... 61.9 KiB 69.7 KiB/s 00:01 [#####] 100%
Total (9/9)        4.2 MiB 550 KiB/s 00:08 [#####] 100%
(9/9) checking keys in keyring [#####] 100%
(9/9) checking package integrity [#####] 100%
(9/9) loading package files [#####] 100%
(9/9) checking for file conflicts [#####] 100%
(9/9) checking available disk space [#####] 100%
:: Processing package changes...
(1/9) upgrading libexpat [#####] 100%
(2/9) upgrading libzstd [#####] 100%
(3/9) upgrading bsdtar [#####] 100%
(4/9) upgrading libgpg-error [#####] 100%
(5/9) upgrading libp11-kit [#####] 100%
(6/9) upgrading libgnutls [#####] 100%
(7/9) upgrading p11-kit [#####] 100%
(8/9) upgrading libgpgme [#####] 100%
(9/9) upgrading zstd [#####] 100%
:: Running post-transaction hooks...
(1/1) Updating the info directory file...

62813@LAPTOP-4K50002L MSYS ~
$ |
```

13. Close “Command window”.

-
14. Kembali pada menu “Start”, pilih dan klik “MSYS2 MinGW x64”



15. Pada “Command Window” yang terbuka ketikkan “pacman -Ss gcc”, kemudian tekan “Enter”.

Two screenshots of a terminal window titled "MINGW64".

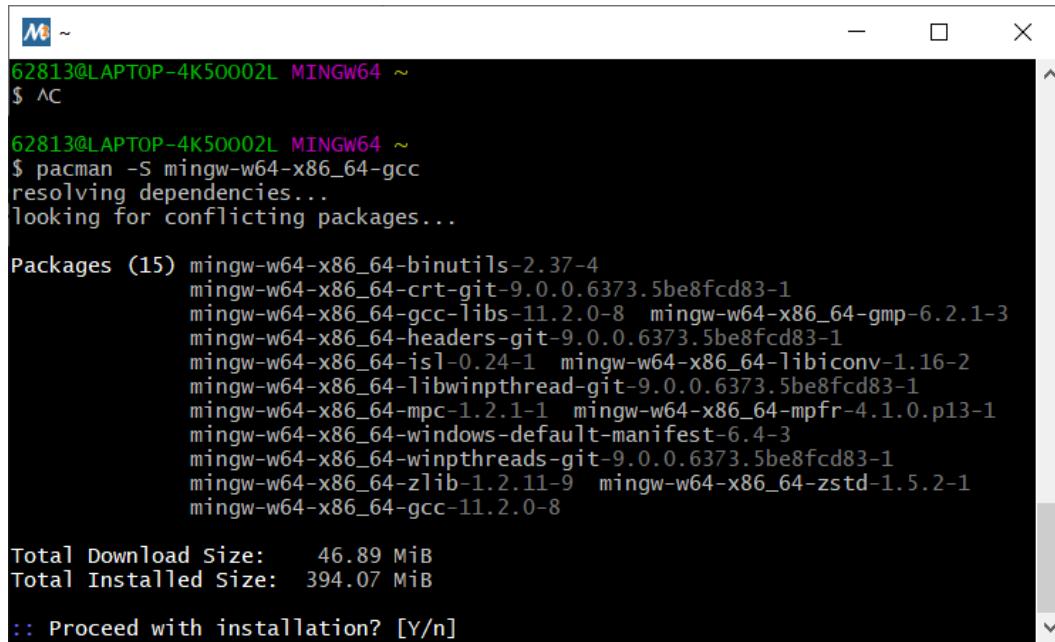
The top screenshot shows the command being typed:

```
62813@LAPTOP-4K50002L MINGW64 ~
$ pacman -Ss gcc|
```

The bottom screenshot shows the results of the command, displaying various packages related to the GNU Compiler Collection (GCC) and its tools:

```
GNU Compiler Collection (ObjC,Obj-C++) for MinGW-w64
ucrt64/mingw-w64-ucrt-x86_64-libgccjit 11.2.0-8 (mingw-w64-ucrt-x86_64-toolchain)
)
GNU Compiler Collection (libgccjit) for MinGW-w64
ucrt64/mingw-w64-ucrt-x86_64-perl 5.32.1-4
A highly capable, feature-rich programming language (mingw-w64)
clang64/mingw-w64-clang-x86_64-libc++ 13.0.1-1 (mingw-w64-clang-x86_64-toolchain)
)
C++ Standard Library (mingw-w64)
clang64/mingw-w64-clang-x86_64-perl 5.32.1-4
A highly capable, feature-rich programming language (mingw-w64)
msys/gcc 11.2.0-3
The GNU Compiler Collection - C and C++ frontends
msys/gcc-fortran 11.2.0-3
Fortran front-end for GCC
msys/gcc-libs 11.2.0-3 [installed]
Runtime Libraries shipped by GCC
msys/lcov 1.15-1
front-end for GCC's coverage testing tool gcov
msys/mingw-w64-cross-gcc 10.2.0-1 (mingw-w64-cross-toolchain mingw-w64-cross)
Cross GCC for the MinGW-w64
62813@LAPTOP-4K50002L MINGW64 ~
$ |
```

16. Pada “Command Window” yang terbuka ketikkan “pacman -S mingw-w64-x86_64-gcc”, kemudian tekan “Enter”.
17. Ketik “Y” lalu tekan “Enter” dan tunggu hingga proses instalasi selesai.



```

62813@LAPTOP-4K50002L MINGW64 ~
$ ^C

62813@LAPTOP-4K50002L MINGW64 ~
$ pacman -S mingw-w64-x86_64-gcc
resolving dependencies...
looking for conflicting packages...

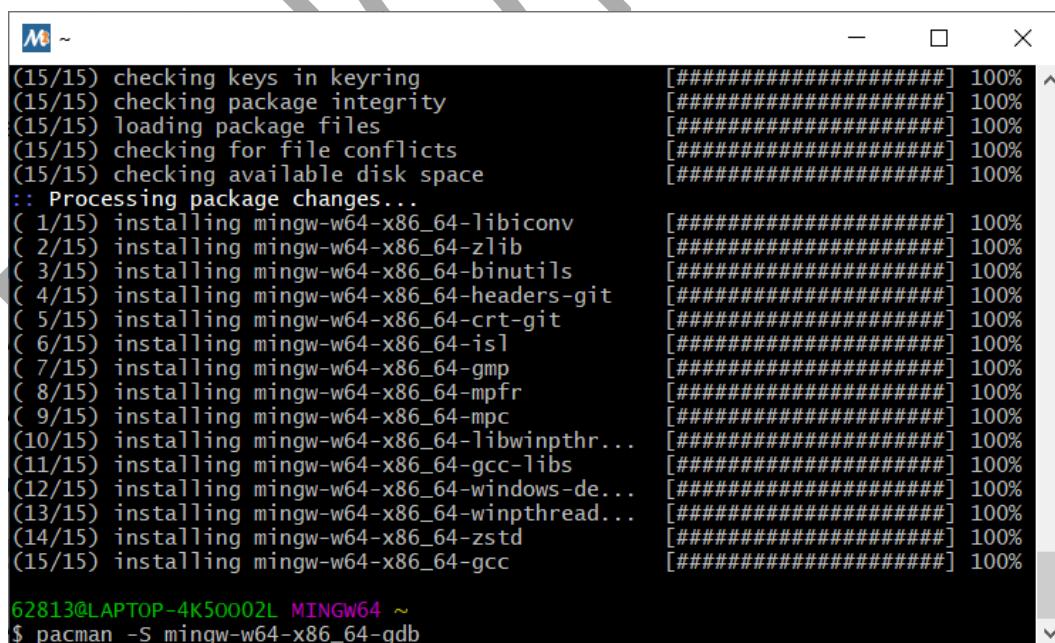
Packages (15) mingw-w64-x86_64-binutils-2.37-4
                  mingw-w64-x86_64-crt-git-9.0.0.6373.5be8fc83-1
                  mingw-w64-x86_64-gcc-libs-11.2.0-8  mingw-w64-x86_64-gmp-6.2.1-3
                  mingw-w64-x86_64-headers-git-9.0.0.6373.5be8fc83-1
                  mingw-w64-x86_64-is1-0.24-1  mingw-w64-x86_64-libiconv-1.16-2
                  mingw-w64-x86_64-libwinpthread-git-9.0.0.6373.5be8fc83-1
                  mingw-w64-x86_64-mpc-1.2.1-1  mingw-w64-x86_64-mpfr-4.1.0.p13-1
                  mingw-w64-x86_64-windows-default-manifest-6.4-3
                  mingw-w64-x86_64-winpthreads-git-9.0.0.6373.5be8fc83-1
                  mingw-w64-x86_64-zlib-1.2.11-9  mingw-w64-x86_64-zstd-1.5.2-1
                  mingw-w64-x86_64-gcc-11.2.0-8

Total Download Size:    46.89 MiB
Total Installed Size:  394.07 MiB

:: Proceed with installation? [Y/n]

```

18. Pada “Command Window” yang terbuka ketikkan “pacman -S mingw-w64-x86_64-gdb”, kemudian tekan “Enter”.



```

62813@LAPTOP-4K50002L MINGW64 ~
$ pacman -S mingw-w64-x86_64-gdb
(15/15) checking keys in keyring [#####
(15/15) checking package integrity [#####
(15/15) loading package files [#####
(15/15) checking for file conflicts [#####
(15/15) checking available disk space [#####
:: Processing package changes...
( 1/15) installing mingw-w64-x86_64-libiconv [#####
( 2/15) installing mingw-w64-x86_64-zlib [#####
( 3/15) installing mingw-w64-x86_64-binutils [#####
( 4/15) installing mingw-w64-x86_64-headers-git [#####
( 5/15) installing mingw-w64-x86_64-crt-git [#####
( 6/15) installing mingw-w64-x86_64-is1 [#####
( 7/15) installing mingw-w64-x86_64-gmp [#####
( 8/15) installing mingw-w64-x86_64-mpfr [#####
( 9/15) installing mingw-w64-x86_64-mpc [#####
(10/15) installing mingw-w64-x86_64-libwinpthr... [#####
(11/15) installing mingw-w64-x86_64-gcc-libs [#####
(12/15) installing mingw-w64-x86_64-windows-de... [#####
(13/15) installing mingw-w64-x86_64-winpthread... [#####
(14/15) installing mingw-w64-x86_64-zstd [#####
(15/15) installing mingw-w64-x86_64-gcc [#####

62813@LAPTOP-4K50002L MINGW64 ~
$ pacman -S mingw-w64-x86_64-gdb

```

19. Ketik “Y” lalu tekan “Enter” dan tunggu hingga proses instalasi selesai.

```

mingw-w64-x86_64-libtre-git-r128.6fb7206-2
mingw-w64-x86_64-mpdecimal-2.5.1-1
mingw-w64-x86_64-ncurses-6.3-3
mingw-w64-x86_64-openssl-1.1.1.m-1
mingw-w64-x86_64-p11-kit-0.24.1-1
mingw-w64-x86_64-python-3.9.10-2
mingw-w64-x86_64-readline-8.1.001-1
mingw-w64-x86_64-sqlite3-3.37.2-1 mingw-w64-x86_64-tcl-8.6.11-5
mingw-w64-x86_64-termcap-1.3.1-6 mingw-w64-x86_64-tk-8.6.11.1-2
mingw-w64-x86_64-tzdata-2021e-1 mingw-w64-x86_64-xxhash-0.8.1-1
mingw-w64-x86_64-xz-5.2.5-2 mingw-w64-x86_64-gdb-11.2-1

Total Download Size: 44.35 MiB
Total Installed Size: 310.53 MiB

:: Proceed with installation? [Y/n] Y
:: Retrieving packages...
mingw-w64-x86_64... 687.9 KiB 228 KiB/s 01:12 [-----] 4%
mingw-w64-x86_64... 1168.0 KiB 403 KiB/s 00:14 [###-----] 16%
mingw-w64-x86_64... 544.0 KiB 177 KiB/s 00:24 [##-----] 11%
mingw-w64-x86_64... 1120.0 KiB 381 KiB/s 00:08 [#####---] 25%
mingw-w64-x86_64... 1664.0 KiB 432 KiB/s 00:03 [#####----] 52%
Total ( 0/22) 5.0 MiB 1602 KiB/s 00:25 [#-----] 11%

```

20. Close semua “Command Window”.

21. Buka “Command Prompt” lalu uji perintah “gcc –version, g++ --version, dan gdb –version” seperti pada gambar yang tampil di bawah.

```

Microsoft Windows [Version 10.0.19044.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\62813>gcc --version
gcc (Rev8, Built by MSYS2 project) 11.2.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

C:\Users\62813>g++ --version
g++ (Rev8, Built by MSYS2 project) 11.2.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

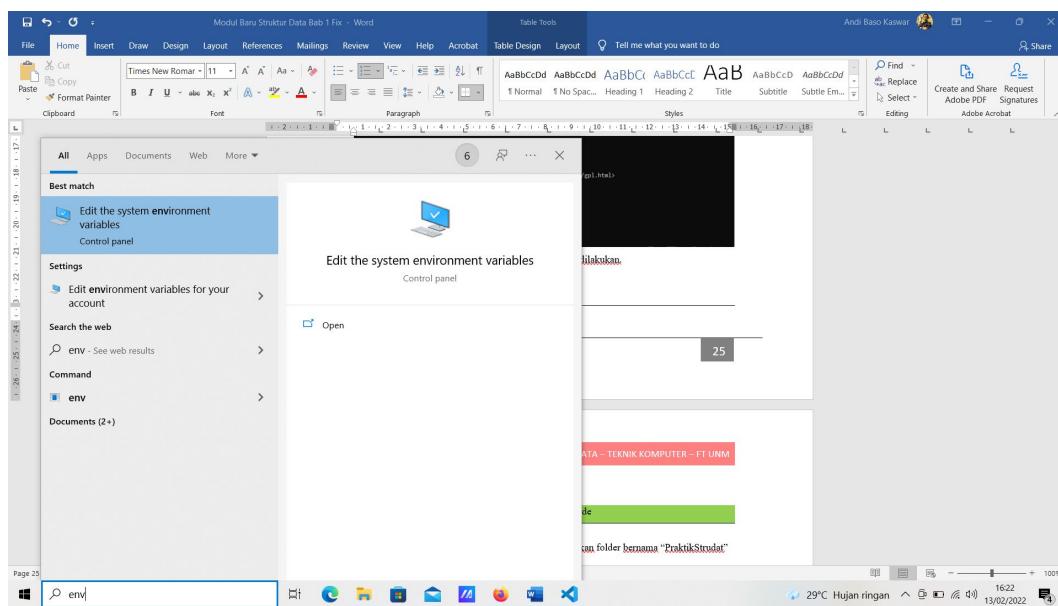
C:\Users\62813>gdb --version
GNU gdb (GDB) 11.2
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

C:\Users\62813>

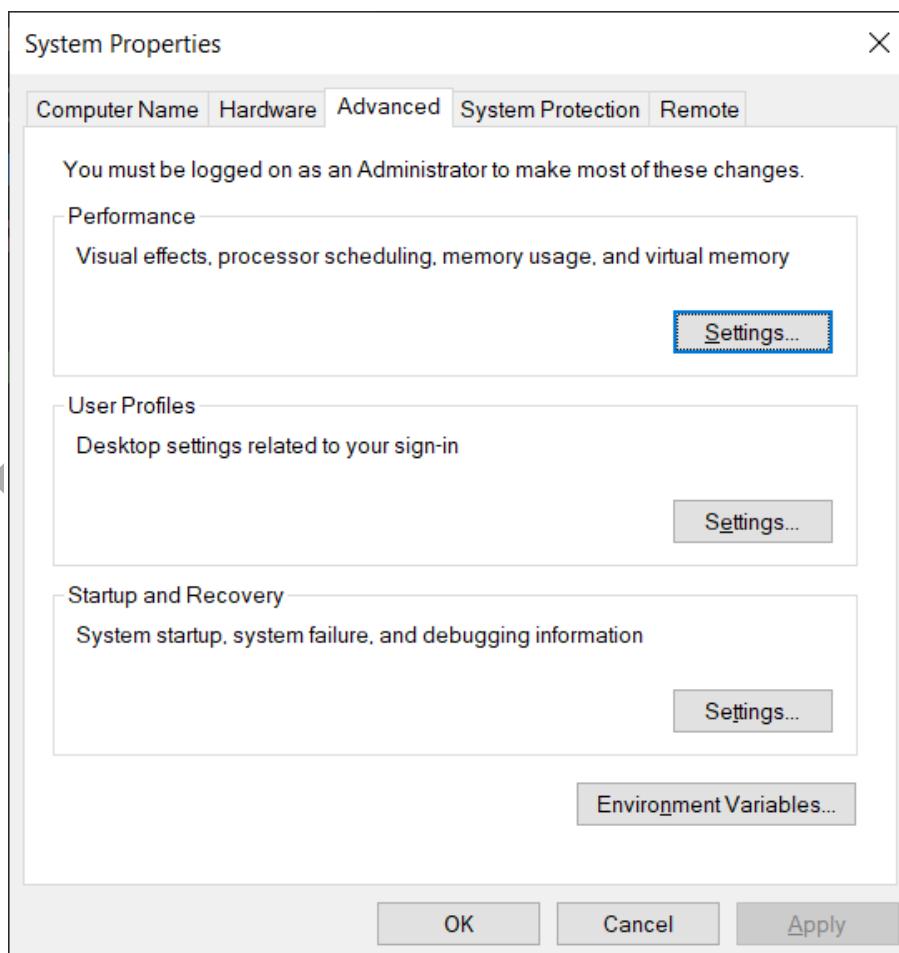
```

22. Jika output seperti pada gambar maka instalasi berhasil dilakukan.

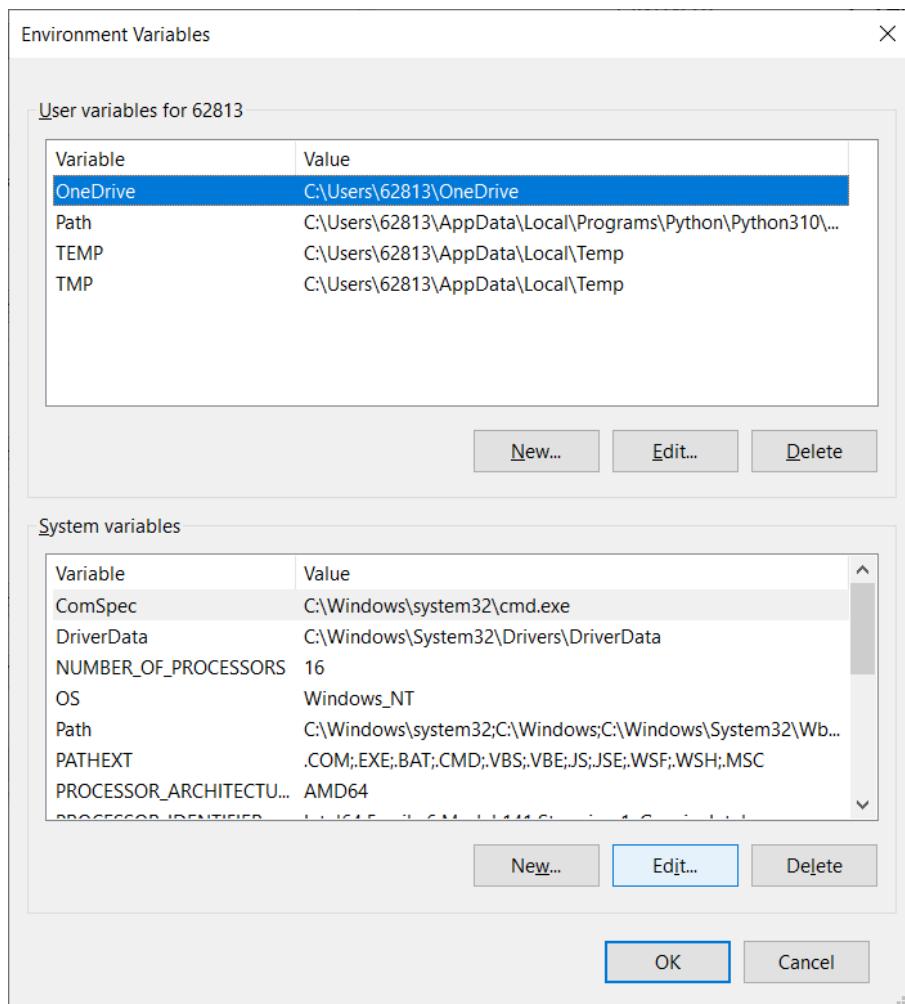
23. Selanjutnya, pada kolom search cari “Edit The System Environment Variable”.



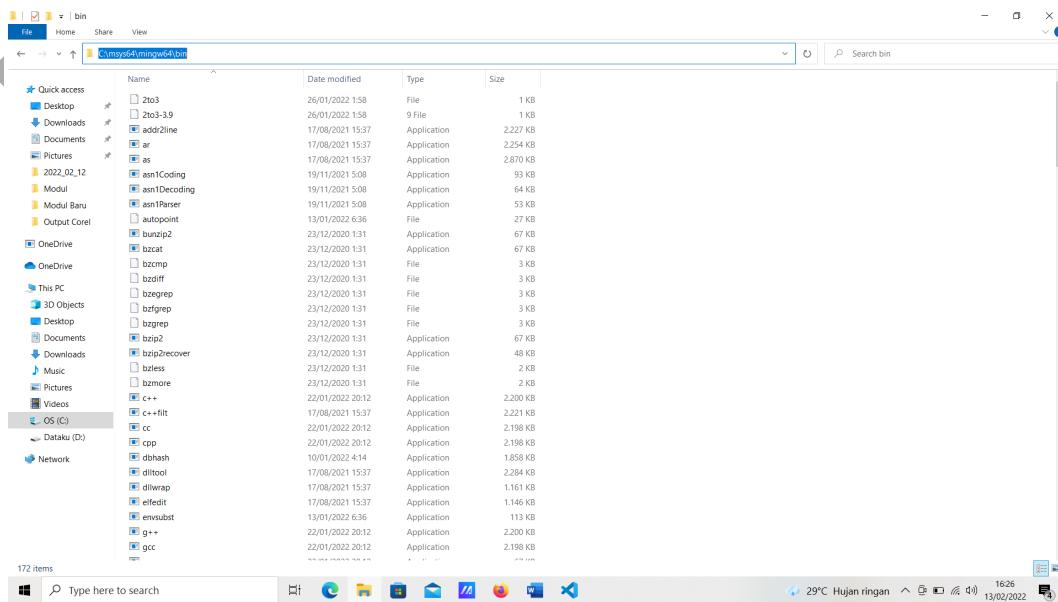
24. Pada jendela “System Properties” yang muncul klik tombol “Environment Variables”.



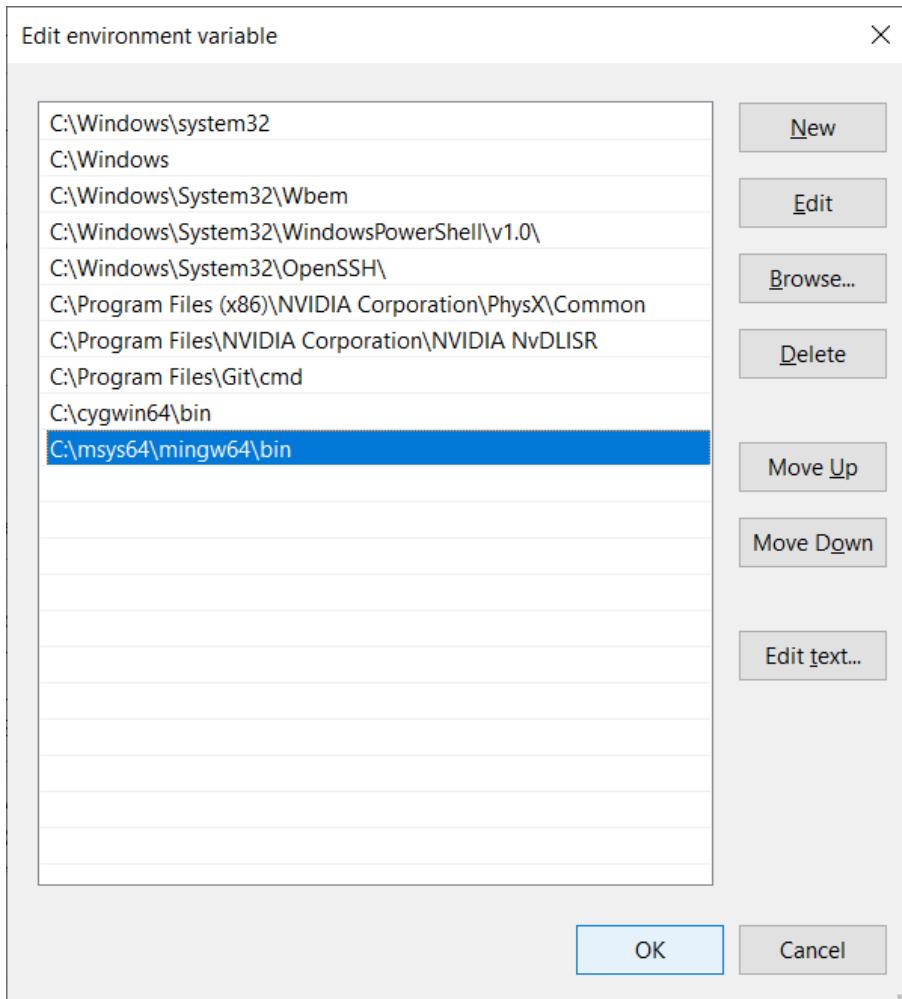
25. Pada jendela yang muncul, pada “System Variables”, pilih “path” lalu klik “Edit”.



26. Copy lokasi direktori mingw64 anda.



-
27. Pada jendela yang muncul klik “New” lalu paste lokasi directory minGW anda. Dalam contoh ini “C:\msys64\mingw64\bin”. Lalu klik “OK”.



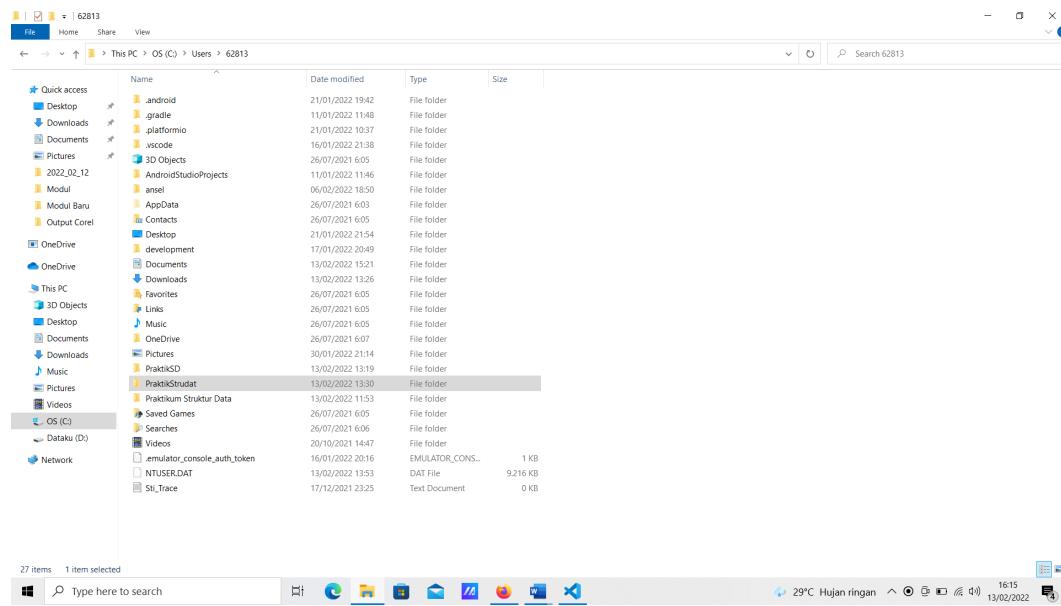
28. Klik “OK” hingga semua jendela tertutup.
-



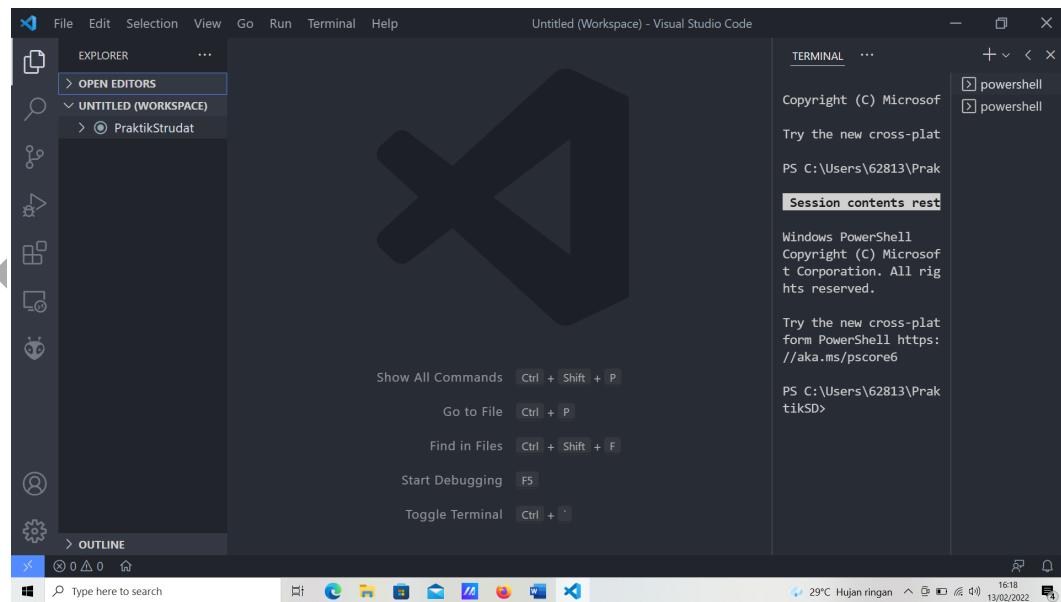
Praktikum 1.4. Mengatur lingkungan kerja Visual Studio Code

1. Pada direktori “C:\Users\”nama user laptopmu” tambahkan folder bernama “PraktikStrudat”

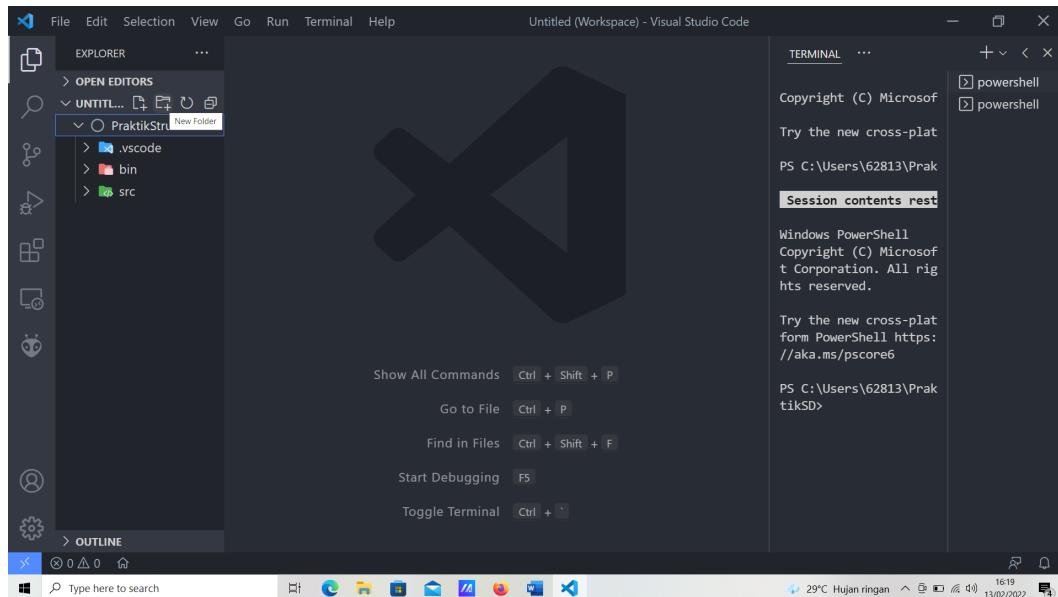
Dalam contoh ini “C:\Users\62813”



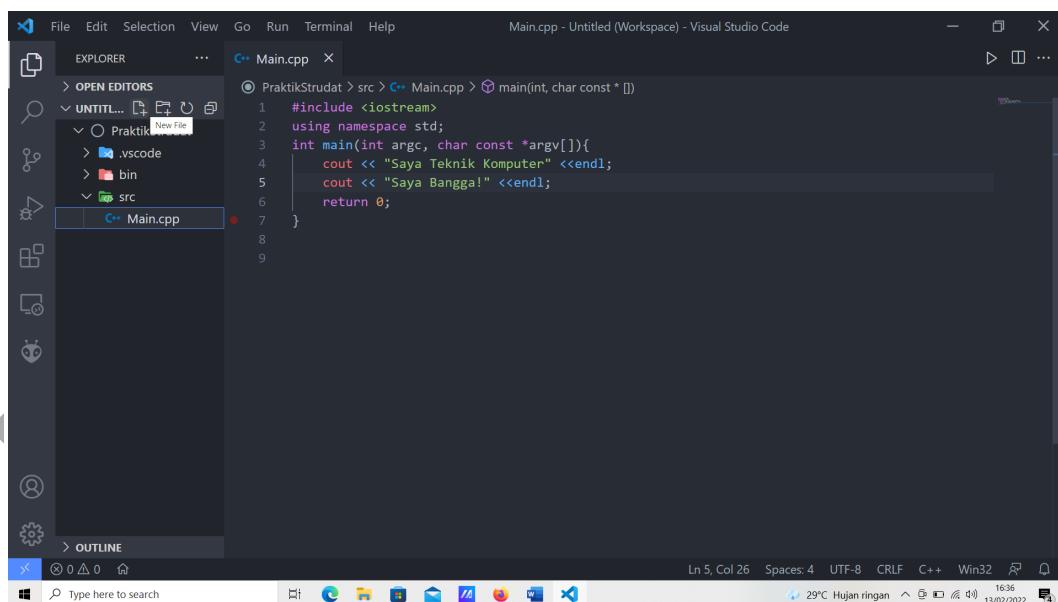
2. Pada Visual Studio Code, pilih file lalu klik “Add Folder To Workspace”. Lalu “add folder” yang telah dibuat tadi sehingga akan tampil seperti ini.



3. Pada folder “PraktikStrudat” tambah folder bernama “bin” dan “src”. Abaikan dulu folder “.vscode” pada gambar. Foldere ini akan tampil otomatis kemudian.



4. Pada folder “src”, tambahkan file baru dan beri nama “Main.cpp”



5. Ketikkan kode program berikut:

```

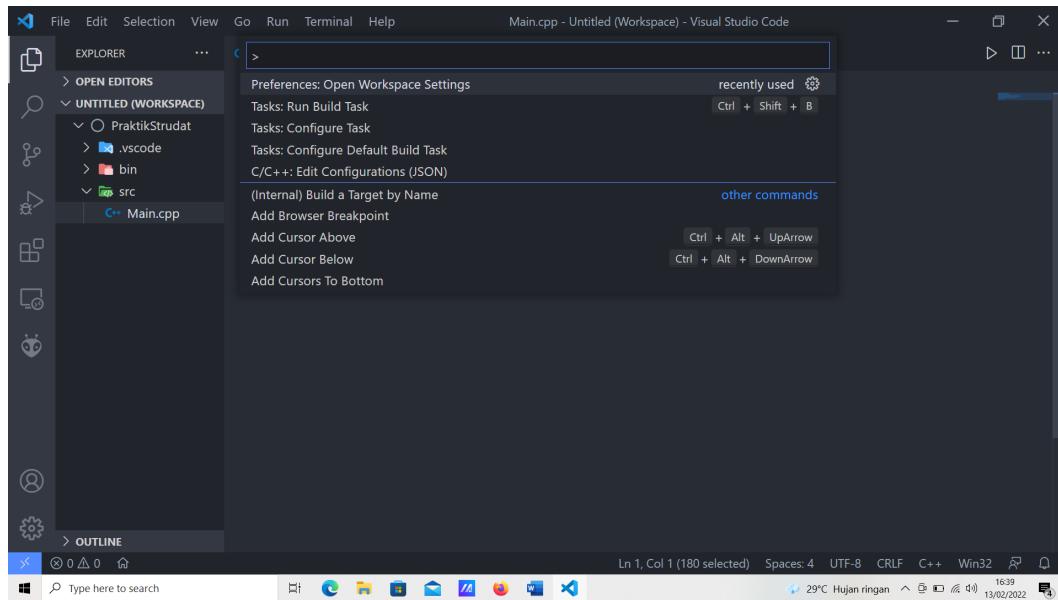
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]){
    cout << "Saya Teknik Komputer" << endl;
    cout << "Saya Bangga!" << endl;
  
```

```

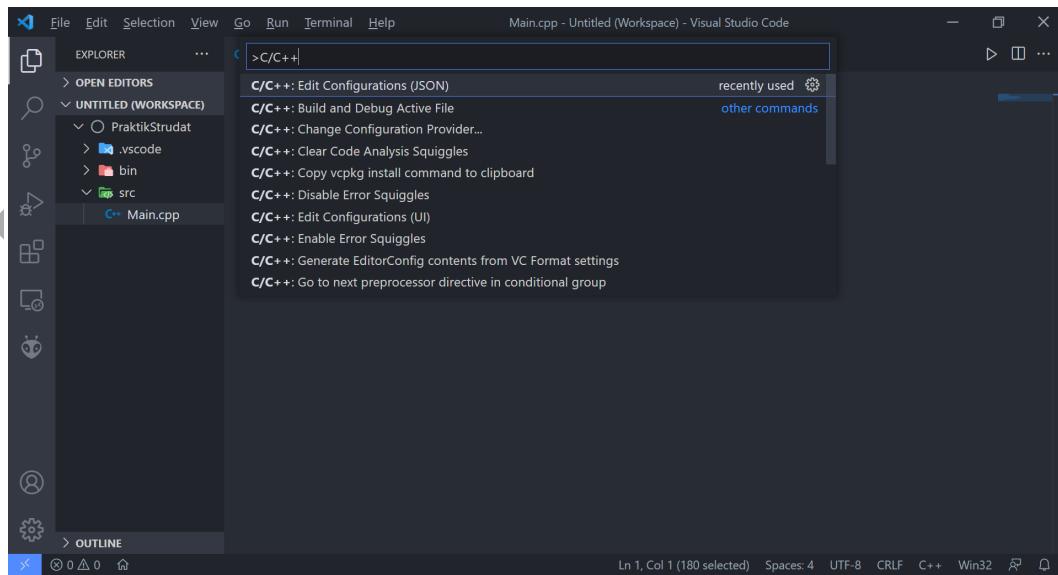
    return 0;
}

```

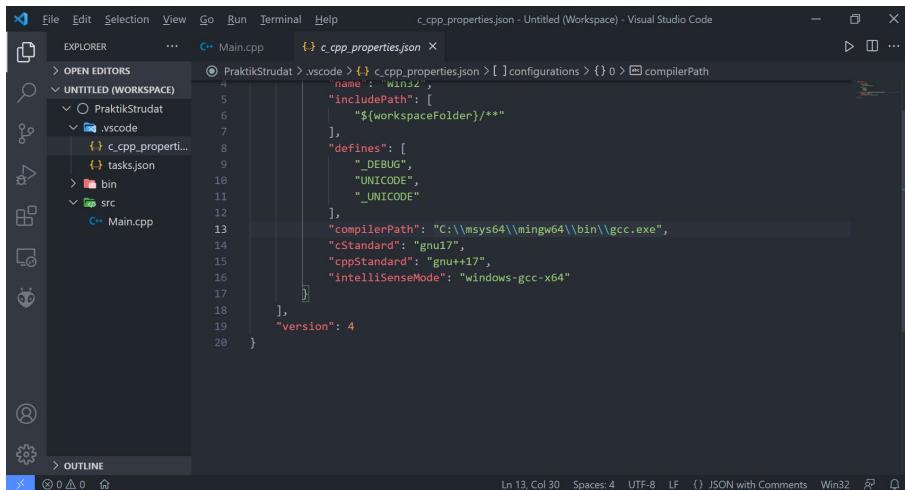
6. Klik “Save”.
7. Sekarang, tekan tombol “Ctrl +Shift+P” pada keyboard anda. Maka akan muncul jendela seperti berikut ini



8. Ketikkan C/C++ lalu pilih pilihan “Edit Configurations (JSON)”.



9. Pada jendela yang muncul, sesuaikan kode di dalam file tersebut dengan kode pada gambar berikut. Tidak perlu merubah apapun.

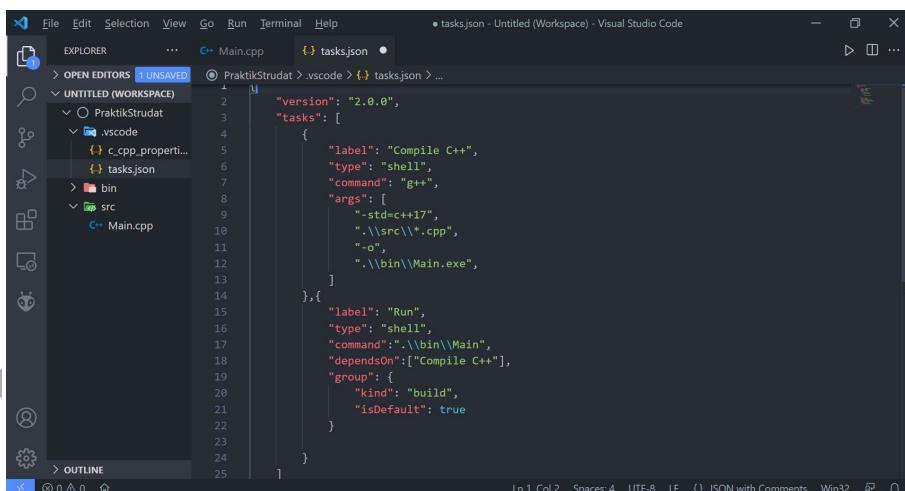


```

{
    "configurations": [
        {
            "name": "Windows",
            "includePath": [
                "${workspaceFolder}/**"
            ],
            "defines": [
                "_DEBUG",
                "UNICODE",
                "_UNICODE"
            ],
            "compilerPath": "C:\\msys64\\mingw64\\bin\\gcc.exe",
            "cStandard": "gnu17",
            "cppStandard": "gnu++17",
            "intelliSenseMode": "windows-gcc-x64"
        }
    ],
    "version": 4
}

```

10. “compilerPath” akan menyesuaikan dengan direktori path yang telah diatur sebelumnya.
 11. Sekarang, tekan tombol “Ctrl +Sift+P” pada keyboard anda. Lalu cari “tasks” kemudianilih menu “Create tasks.json file from template” yang muncul.
 12. Pada file json yang tampil, pastikan kodennya seperti berikut ini.



```

{
    "version": "2.0.0",
    "tasks": [
        {
            "label": "Compile C++",
            "type": "shell",
            "command": "g++",
            "args": [
                "-std=c++17",
                "\\\\src\\\\*.cpp",
                "-o",
                "\\\\bin\\\\Main.exe"
            ]
        },
        {
            "label": "Run",
            "type": "shell",
            "command": "\\\\bin\\\\Main",
            "dependsOn": ["Compile C++"],
            "group": {
                "kind": "build",
                "isDefault": true
            }
        }
    ]
}

```

13. Kodennya seperti berikut

```
{
    "version": "2.0.0",
    "tasks": [
        {
            "label": "Compile C++",
            "type": "shell",

```

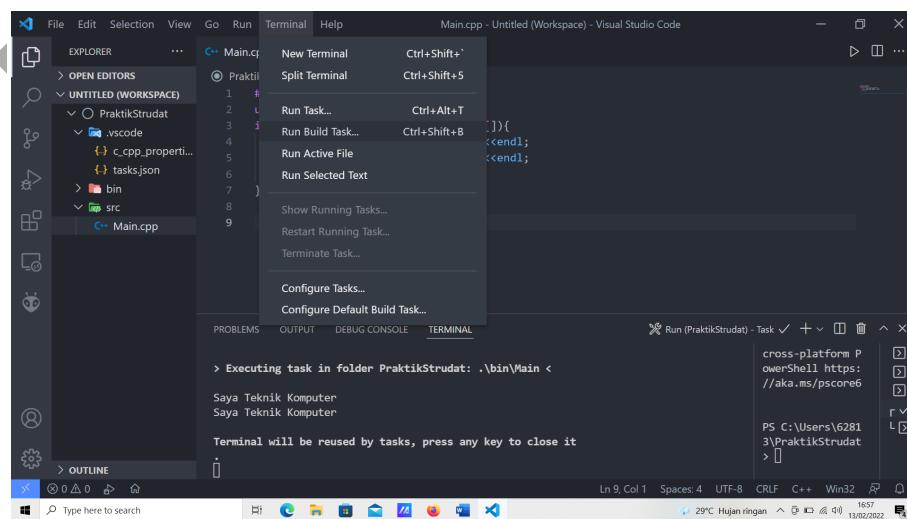
```

    "command": "g++",
    "args": [
        "-std=c++17",
        ".\\src\\*.cpp",
        "-o",
        ".\\bin\\Main.exe",
    ]
}, {
    "label": "Run",
    "type": "shell",
    "command": ".\\bin\\Main",
    "dependsOn": ["Compile C++"],
    "group": {
        "kind": "build",
        "isDefault": true
    }
}
]
}
}

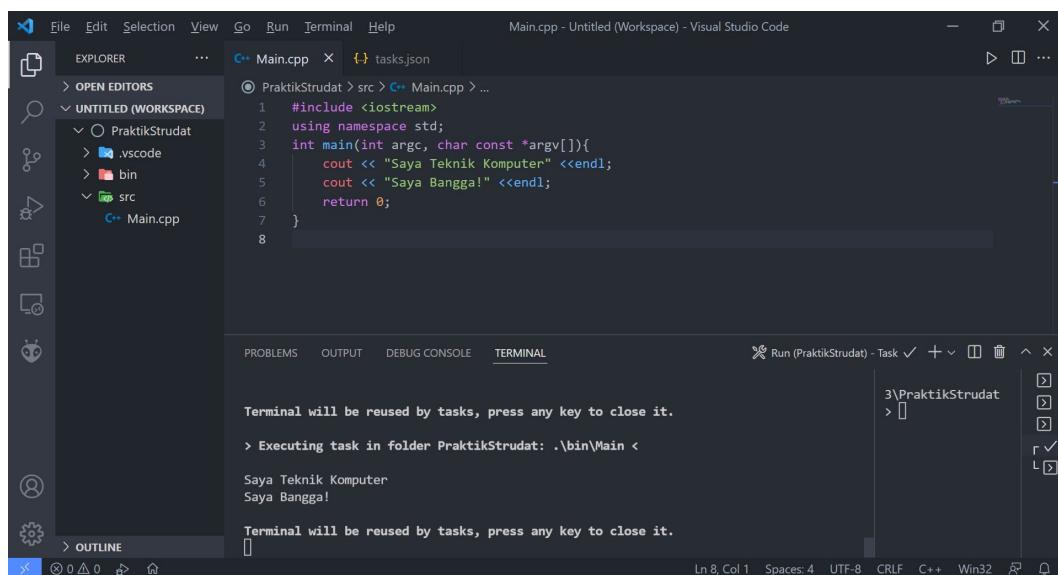
```

14. Klik “Save”.

15. Coba jalankan kode program dengan memilih menu “Terminal” lalu pilih dan klik “Run Build Task”.



16. Jika berhasil maka output akan tampil pada terminal.



```
#include <iostream>
using namespace std;

int main(int argc, char const *argv[]){
    cout << "Saya Teknik Komputer" << endl;
    cout << "Saya Bangga!" << endl;
    return 0;
}
```

The terminal output shows:

```
Saya Teknik Komputer
Saya Bangga!
```

17. Visual Studio Code siap digunakan.

Praktikum 1.5. Menggunakan tipe data string dan integer

```
#include <iostream>
using namespace std;

void tampilData (string nama, int hargaBarang){
    cout <<"Nama barang : "<< nama << endl;
    cout <<"Harga barang : "<< hargaBarang << " Rupiah" << endl;
}

int main()
{
    string nama;
    int harga;
    nama = "Yamaha F1ZR";
    harga = 12000000;
    tampilData(nama, harga);
    return 0;
}
```

Praktikum 1.6. Menggunakan tipe data float

```
#include <iostream>
using namespace std;
float luas;

float cariLuas(float tinggi, float alas) {
    float luas = (tinggi * alas)/2;
    return luas;
}

int main() {
    float tinggiSegitiga;
    float alasSegitiga;
    float luasSegitiga;

    cout << "Masukkan tinggi segitiga" << endl;
    cin >> tinggiSegitiga;

    cout << "Masukkan tinggi segitiga" << endl;
    cin >> alasSegitiga;

    luasSegitiga = cariLuas(tinggiSegitiga, alasSegitiga);
    cout << "Luas segitiga adalah " << luasSegitiga << endl;

    return 0;
}
```

Praktikum 1.7. Menggunakan tipe data boolean

```
#include <iostream>
using namespace std;
float volume;

float cariVolume(float s) {
    float volume = s*s*s;
    cout << volume << endl;
    return volume;
```

```
}
```

```
int main() {
    float panjangKubus1;
    float panjangKubus2;

    cout << "Masukkan panjang rusuk kubus pertama: "<<endl;
    cin >> panjangKubus1;

    cout << "Masukkan panjang rusuk kubus pertama: "<<endl;
    cin >> panjangKubus2;

    if (cariVolume(panjangKubus1)>cariVolume(panjangKubus2)){
        cout << "Kubus 1 lebih besar daripada kubus 2" << endl;
    } else if (cariVolume(panjangKubus1)==cariVolume(panjangKubus2)){
        cout << "Kubus 2 sama besarnya dengan kubus 1" << endl;
    } else {
        cout << "Kubus 2 lebih besar daripada kubus 1" << endl;
    }
    return 0;
}
```



CATATAN

donotcopy