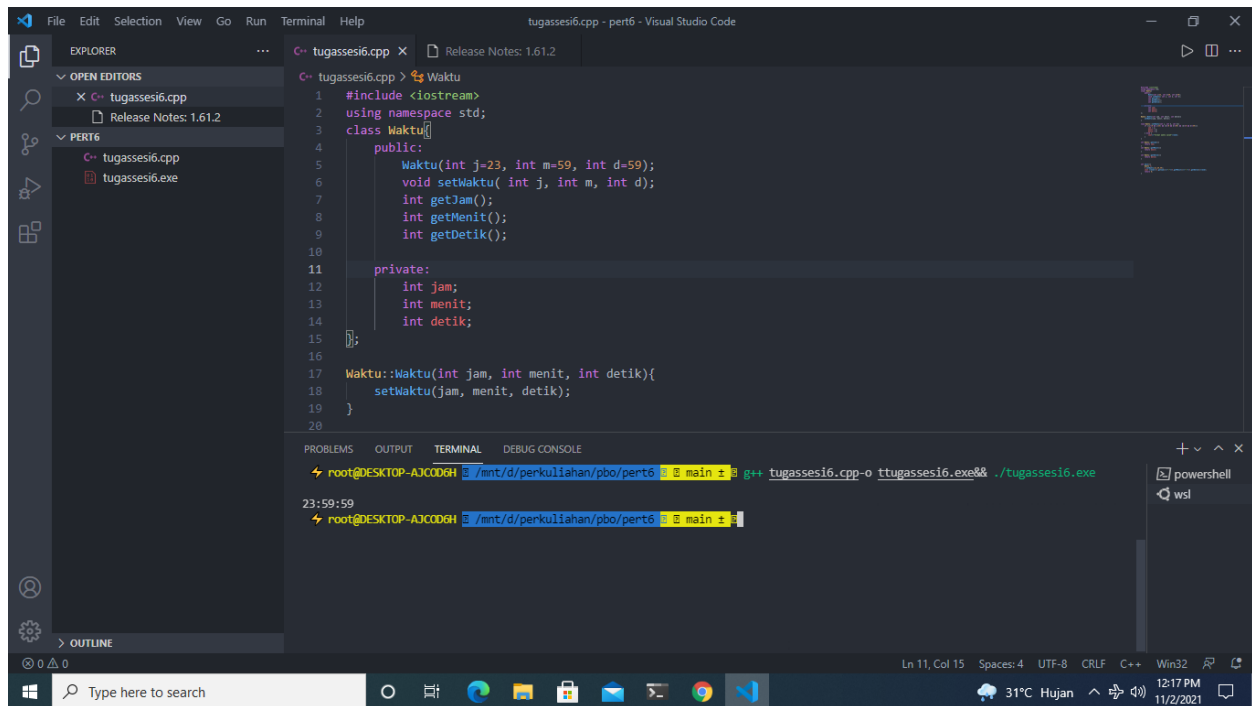
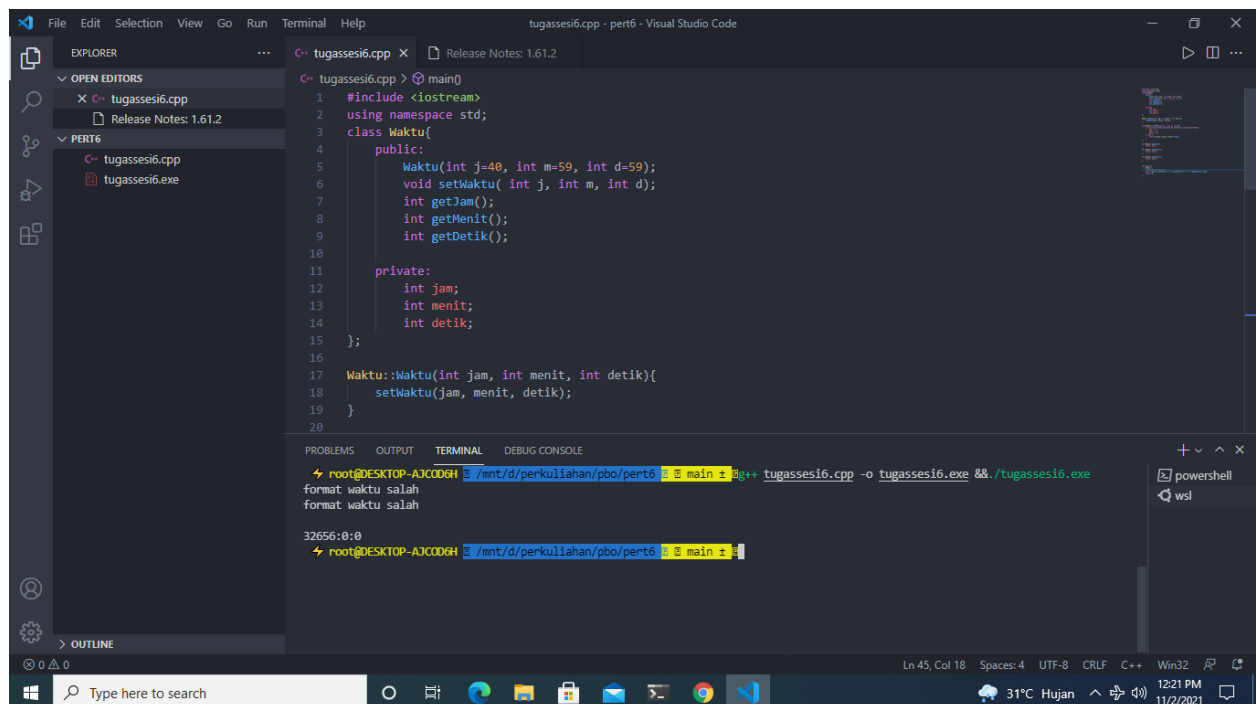


1. Pada hasil screenshoot dibawah ini adalah hasil sebelum perubahan menjadi 40 yang menunjukan bahwa hasil berjalan dengan semestinya



The screenshot shows the Visual Studio Code interface with a C++ file named `tugasassi6.cpp` open. The code defines a `Waktu` class with methods `setWaktu`, `getJam`, `getMenit`, and `getDetik`. The `main` function is not visible in the editor. The terminal at the bottom shows the command `g++ tugasassi6.cpp -o ttugasassi6.exe && ./tugasassi6.exe` being executed, resulting in the output `23:59:59`. The status bar at the bottom indicates the file is at line 11, column 15.

Pada Screenshoot kedua dibawah ini menunjukan bahwa angka 23 telah diganti dengan 40 dan Ketika dirunning menunjukan tidak berjalan dengan semestinya



The screenshot shows the same Visual Studio Code interface, but the `main` function in `tugasassi6.cpp` is now visible, showing `Waktu w(40, 59, 59);`. The terminal shows the same compilation and execution command as the first screenshot. However, the output is `format waktu salah` followed by `32656:0:0`, indicating a runtime error. The status bar at the bottom indicates the file is at line 45, column 18.

2. Abstraksi

Abstraksi sendiri berdasarkan penjelasan dari Modul adalah memisahkan antara ide dengan detail. Abstraksi adalah aksi menyajikan fitur penting tanpa menunjukkan detail dan penjelasan di belakangnya. Kelas menggunakan konsep abstraksi, dengan mendefinisikan serangkaian abstraksi dari atribut dan fungsi yang dimiliki kelas. Abstraksi pada class waktu ini sendiri terdapat pada public yang mana pada public ini memberitahu informasi seperti jam, menit, dan detik yang pada nantinya dijalankan sesuai dengan informasi yang telah diberikan public

Enkapsulasi

Berdasarkan modul yang telah diberikan. Enkapsulasi itu sendiri adalah membungkus data dan fungsi yang mengoperasikan data tersebut dalam sebuah kelas, dan membatasi akses langsung data tersebut. Enkapsulasi menyembunyikan detail implementasi dari klien. Enkapsulasi memaksa abstraksi, memisahkan antara hal yang nampak dari luar (perilaku) dengan yang ada di dalam (status). Enkapsulasi juga bertujuan untuk melindungi integritas dari data yang dimiliki oleh objek. Dan enkapsulasi pada program tersebut terdapat pada private yang mana pada private sendiri dapat diketahui informasi yang di sembunyikan dan hanya dapat diakses oleh variable yang memiliki akses kedalam private