# BAB II

# PEMBAHASAN

## 2.1. Class

Class adalah cetak biru untuk membuat objek. Kelas menentukan apa yang dimiliki sebuah objek (atribut) dan apa yang dapat dilakukan objek (metode).



Contoh pada class diatas adalah terdapat class bernama MyCars, yang mempunyai attribute namaMobil dan warnaMobil. Terdapat instance atau variable sebagai objek nya. Lalu kita tampilkan hasil dari attribute dari instance nya.

## 2.2. Inheritance

Inheritance adalah konsep OOP dimana sebuah class dapat menurunkan property dan method yang dimilikinya kepada class lain. Konsep inheritance dipakai untuk memanfaatkan fitur code reuse, yakni menghindari terjadinya duplikasi kode program. Dalam bahasa Indonesia, inheritance ini disebut juga sebagai pewarisan atau penurunan.

Konsep inheritance membuat sebuah struktur atau hierarchy class dalam kode program. Class yang akan diturunkan bisa disebut sebagai class induk (parent class), super class, atau base class. Sedangkan class yang menerima penurunan bisa disebut sebagai class anak (child class), sub class, derived class atau heir class.

Tidak semua property dan method class induk akan diturunkan. Property dan method dengan hak akses private tidak akan diturunkan kepada class anak. Hanya property dan method dengan hak akses protected dan public saja yang bisa diakses dari class anak.

Cara Penggunaan Inheritance dalam PHP

Di dalam PHP, inheritance atau penurunan sebuah class kepada class lain menggunakan keyword extends, dengan penulisan dasar sebagai berikut:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **class** induk {     //...isi class induk  }    **class** anak **extends** induk  {     //... class anak bisa mengakses     //... property dan method class induk  } |

Agar lebih mudah dipahami, kita akan langsung masuk ke dalam contoh program penggunaan inheritance PHP:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35 | <?php  // buat class induk: komputer  **class** komputer {    **public** $merk;  **public** $processor;  **public** $memory;    **public** **function** beli\_komputer() {  **return** "Beli komputer baru";     }  }    // turunkan class komputer ke laptop  **class** laptop **extends** komputer {    **public** **function** lihat\_spec() {  **return** "merk: $this->merk, processor: $this->processor,       memory: $this->memory";     }  }    // buat objek dari class laptop (instansiasi)  $laptop\_baru = **new** laptop();    // isi property objek  $laptop\_baru->merk = "acer";  $laptop\_baru->processor ="intel core i5";  $laptop\_baru->memory = "2 GB";    //panggil method objek  echo $laptop\_baru->beli\_komputer();  echo "<br />";  echo $laptop\_baru->lihat\_spec();  ?> |

Dalam contoh kode ini terdapat class komputer dengan beberapa property dan sebuah method. Semua property milik class komputer masih belum berisi nilai apa-apa.

## 2.3. Constructor

Constructor adalah method khusus yang dijalankan secara otomatis pada saat sebuah object dibuat, yakni saat proses instansiasi. Constructor biasa dipakai untuk membuat proses awal dalam persiapan object, seperti memberi nilai kepada property, memanggil method internal serta beberapa proses lain yang dirasa perlu.

Dalam bahasa Java, constructor dibuat dengan cara menulis sebuah method yang namanya sama dengan nama class. Sebagai contoh jika kita memiliki class Laptop, maka function constructor juga harus bernama Laptop.

Sebuah constructor tidak mengembalikan nilai sehingga tidak perlu menulis tipe data sebelum nama function. Constructor juga harus memiliki hak akses public, yang apabila tidak ditulis dianggap public secara default.

Karakteristik Construktor :

1. Nama method konsutruktor = nama kelas Nama method konsutruktor = nama kelas
2. Tidak mengembalikan suatu nilai
3. Satu kelas bisa memiliki lebih dari satu konstruktor (Overloading Method Konstruktor)
4. Dapat di bubuhi modifier public, private, protected
5. Suatu konstruktor bisa dipanggil oleh konstruktor lain dalam satu kelas.

Berikut contoh kode program cara membuat constructor dalam bahasa Java:



Dalam kode program di atas saya kembali membuat class Laptop. Class Laptop memiliki 1 constructor Laptop() di baris 2-4 yang hanya berisi satu perintah System.out.println. Kemudian di dalam class BelajarJava terdapat 3 buah object dari class Laptop, yakni object laptopAndi, laptopLisa dan laptopRudi. Pada saat ketiga object di instansiasi, secara otomatis constructor akan berjalan. Hasilnya tampil 3 baris teks berikut di jendela cmd:

Satu object Laptop sudah di buat

Satu object Laptop sudah di buat

Satu object Laptop sudah di buat

Semua teks ini berasal dari perintah System.out.println di dalam method Laptop.

## 2.4. Menciptakan Objek

Object adalah elemen dasar dari konsep pemrograman, merupakan sesuatu yang memiliki identitas (nama), pada umumnya juga memiliki data tentang dirinya maupun object lain dan mempunyai kemampuan untuk melakukan sesuatu dan bisa bekerja sama dengan objek lainnya.

Pada dasarnya semua benda yang ada di dunia dapat dianggap sebagai sebuah object. Rumah, mobil, sepeda motor, meja, dan komputer merupakan contoh-contoh object yang ada di dunia nyata. Setiap object memiliki dua karakteristik yang utama yaitu, atribut dan behavior. Atribut merupakan status object dan behavior merupakan tingkah laku dari object tersebut.

Sebelumnya, kita perlu tahu dulu bahwa sesuatu tidaklah dikatakan objek kecuali jika memiliki atribut atau perilaku.

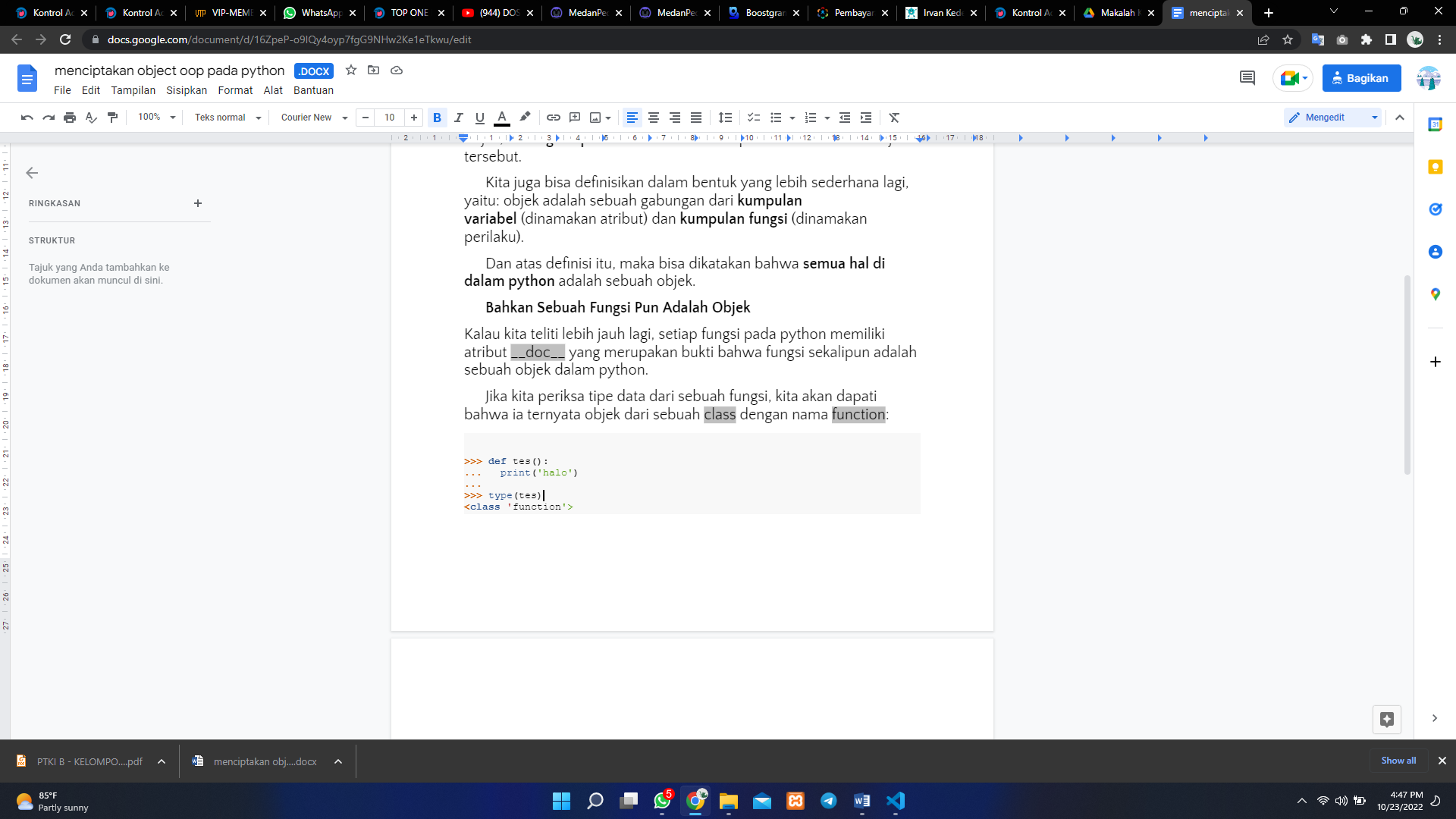
Atribut adalah semacam identitas atau variabel dari suatu objek, sedangkan perilaku adalah “kemampuan” atau fitur dari objek tersebut.

Kita juga bisa definisikan dalam bentuk yang lebih sederhana lagi, yaitu: objek adalah sebuah gabungan dari kumpulan variabel (dinamakan atribut) dan kumpulan fungsi (dinamakan perilaku).

Dan atas definisi itu, maka bisa dikatakan bahwa semua hal di dalam python adalah sebuah objek.

Bahkan Sebuah Fungsi Pun Adalah Objek. Kalau kita teliti lebih jauh lagi, setiap fungsi pada python memiliki atribut \_\_doc\_\_ yang merupakan bukti bahwa fungsi sekalipun adalah sebuah objek dalam python.

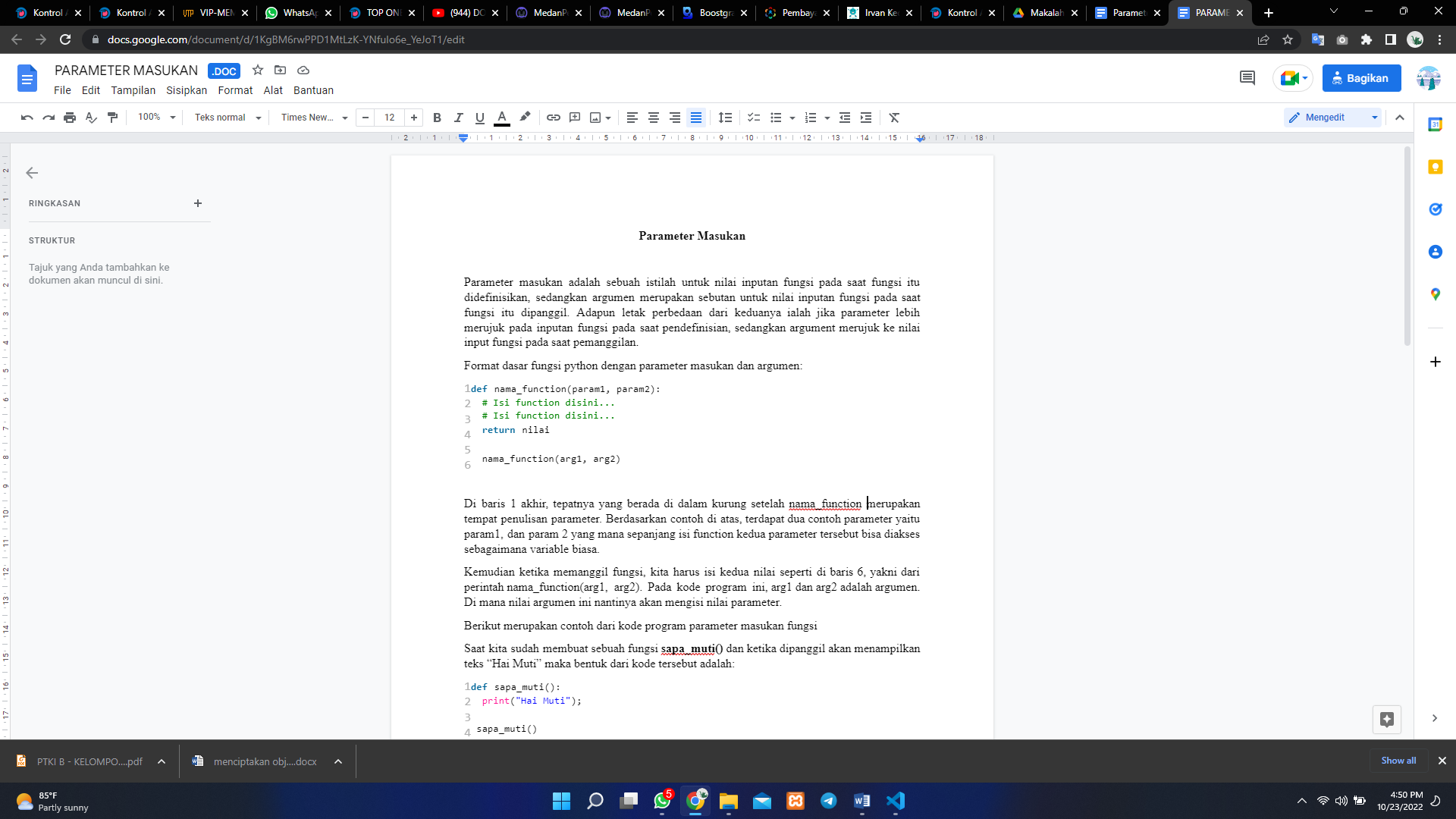
Jika kita periksa tipe data dari sebuah fungsi, kita akan dapati bahwa ia ternyata objek dari sebuah class dengan nama function:



## 2.5. Parameter Masukan

Parameter masukan adalah sebuah istilah untuk nilai inputan fungsi pada saat fungsi itu didefinisikan, sedangkan argumen merupakan sebutan untuk nilai inputan fungsi pada saat fungsi itu dipanggil. Adapun letak perbedaan dari keduanya ialah jika parameter lebih merujuk pada inputan fungsi pada saat pendefinisian, sedangkan argument merujuk ke nilai input fungsi pada saat pemanggilan.

Format dasar fungsi python dengan parameter masukan dan argumen:

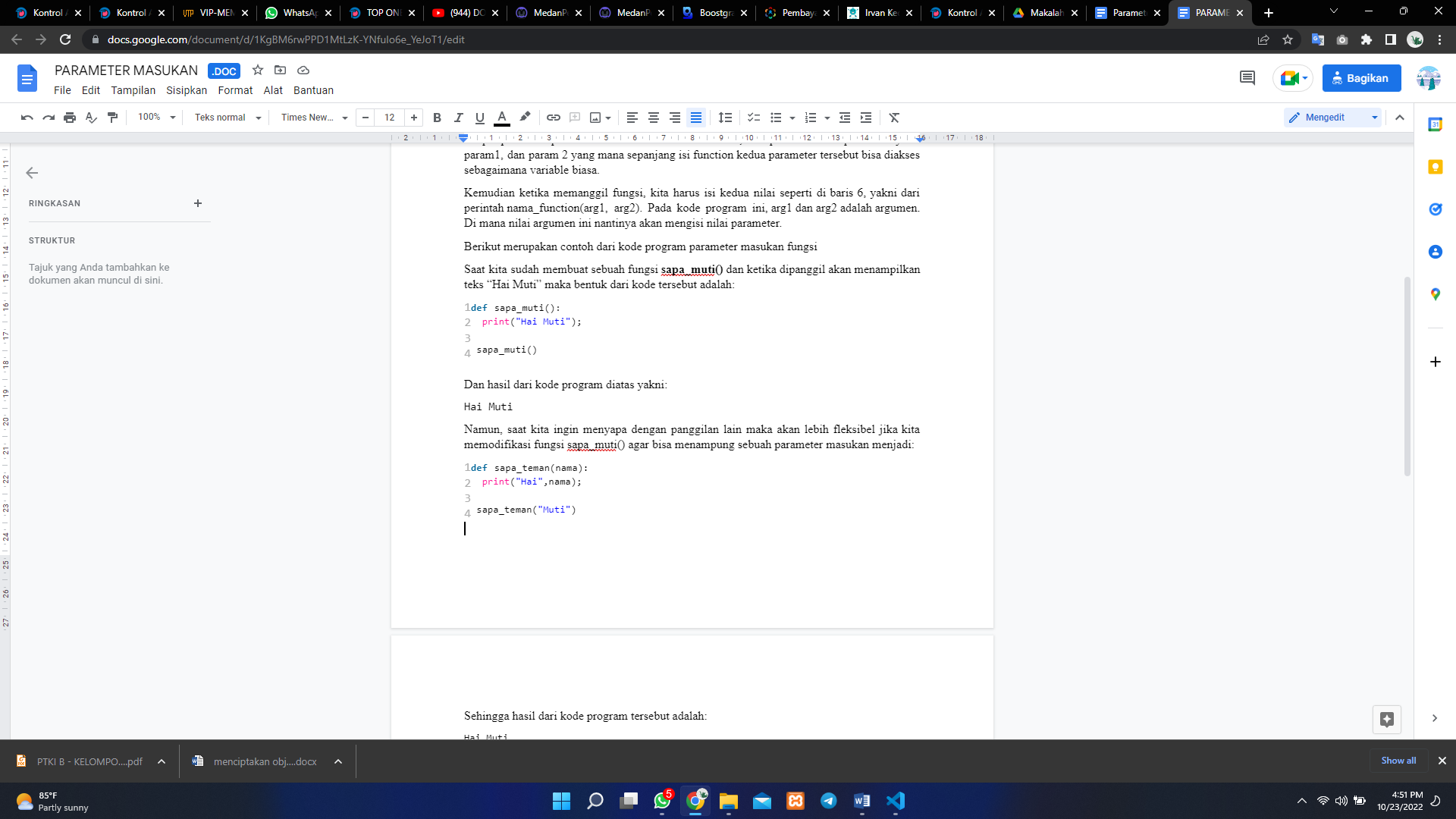


Di baris 1 akhir, tepatnya yang berada di dalam kurung setelah nama\_function merupakan tempat penulisan parameter. Berdasarkan contoh di atas, terdapat dua contoh parameter yaitu param1, dan param 2 yang mana sepanjang isi function kedua parameter tersebut bisa diakses sebagaimana variable biasa.

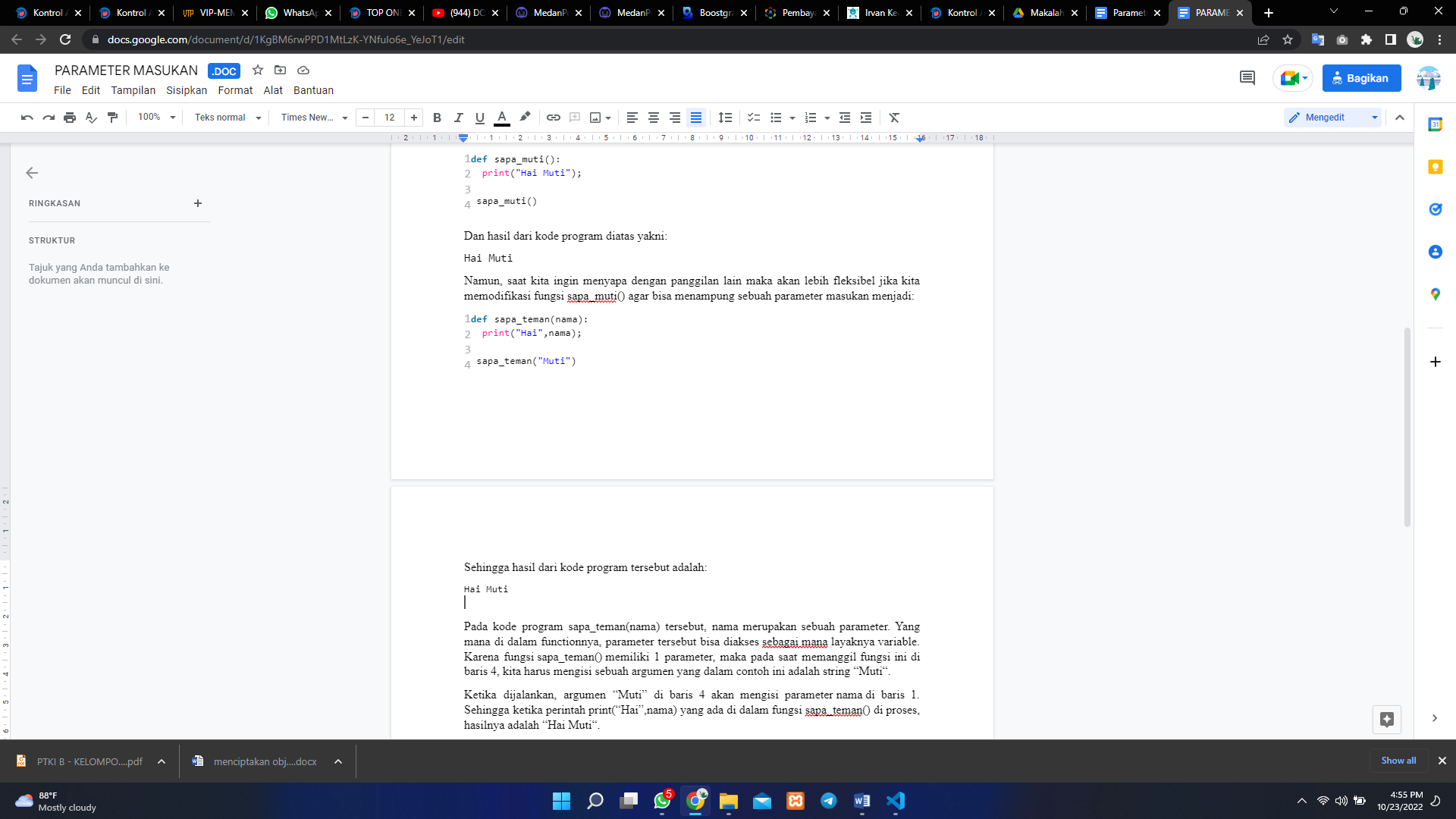
Kemudian ketika memanggil fungsi, kita harus isi kedua nilai seperti di baris 6, yakni dari perintah nama\_function(arg1, arg2). Pada kode program ini, arg1 dan arg2 adalah argumen. Di mana nilai argumen ini nantinya akan mengisi nilai parameter.

Berikut merupakan contoh dari kode program parameter masukan fungsi

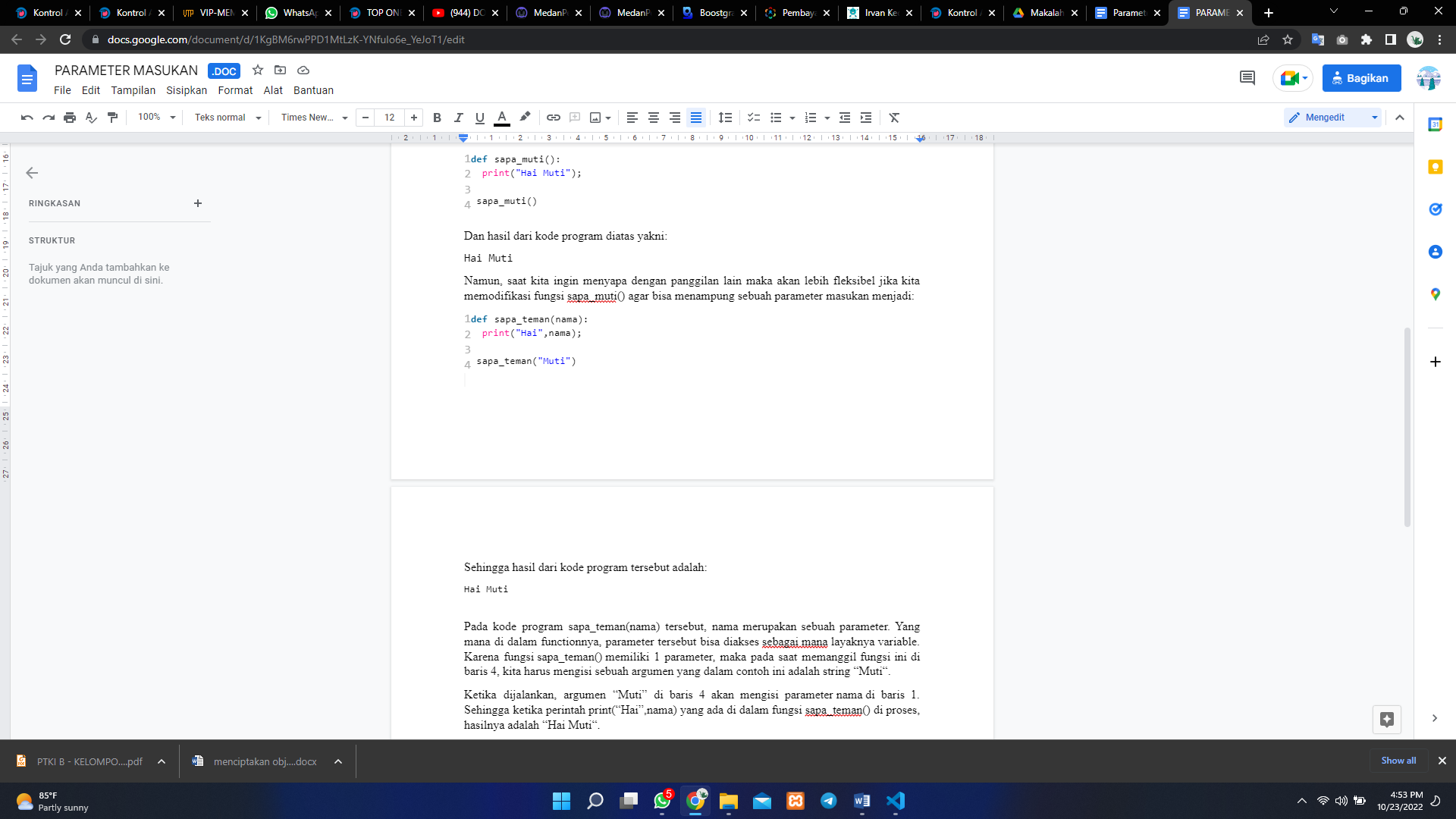
Saat kita sudah membuat sebuah fungsi sapa\_muti() dan ketika dipanggil akan menampilkan teks “Hai Muti” maka bentuk dari kode tersebut adalah:



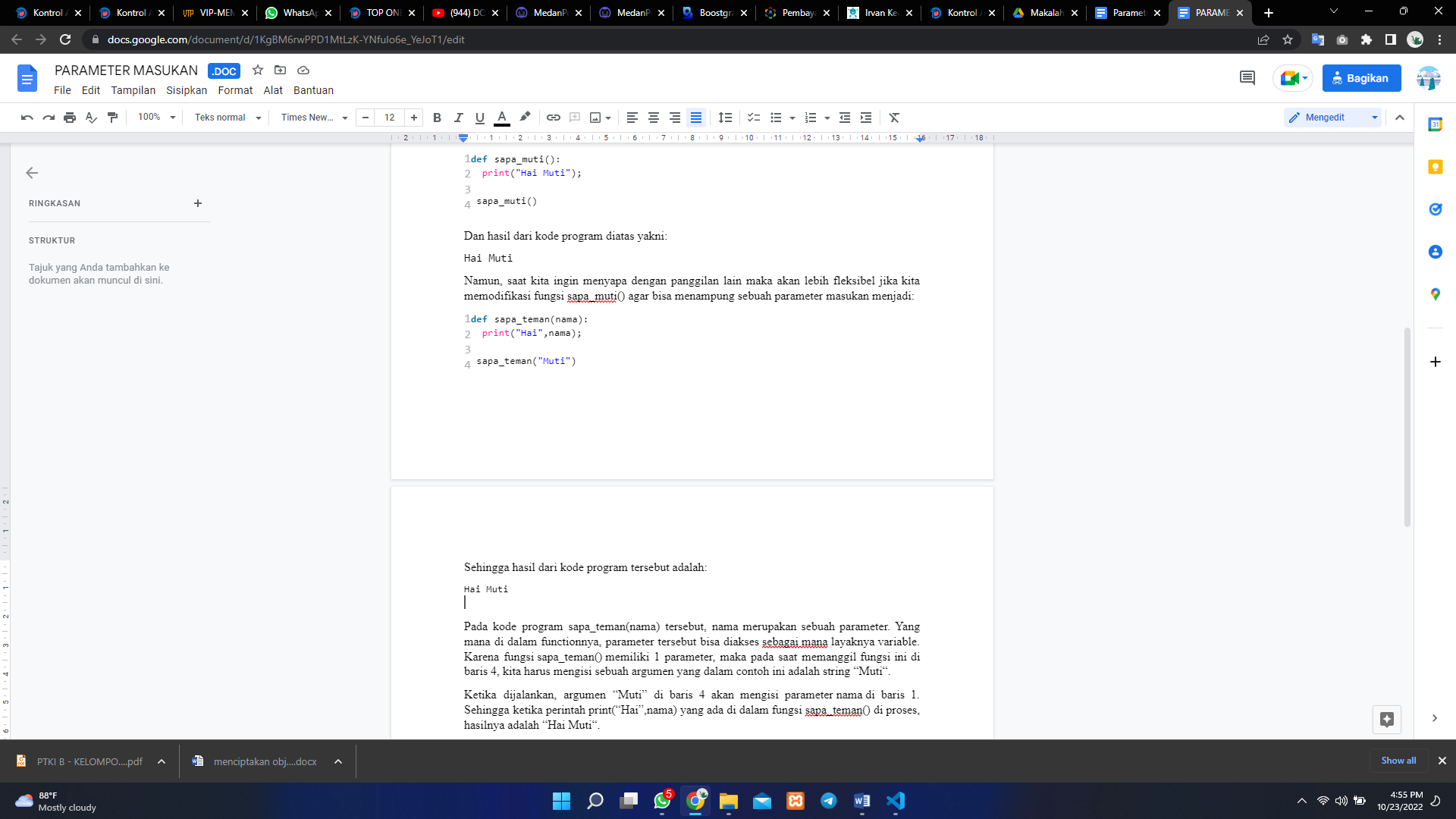
Dan hasil dari kode program diatas yakni:



Namun, saat kita ingin menyapa dengan panggilan lain maka akan lebih fleksibel jika kita memodifikasi fungsi sapa\_muti() agar bisa menampung sebuah parameter masukan menjadi:



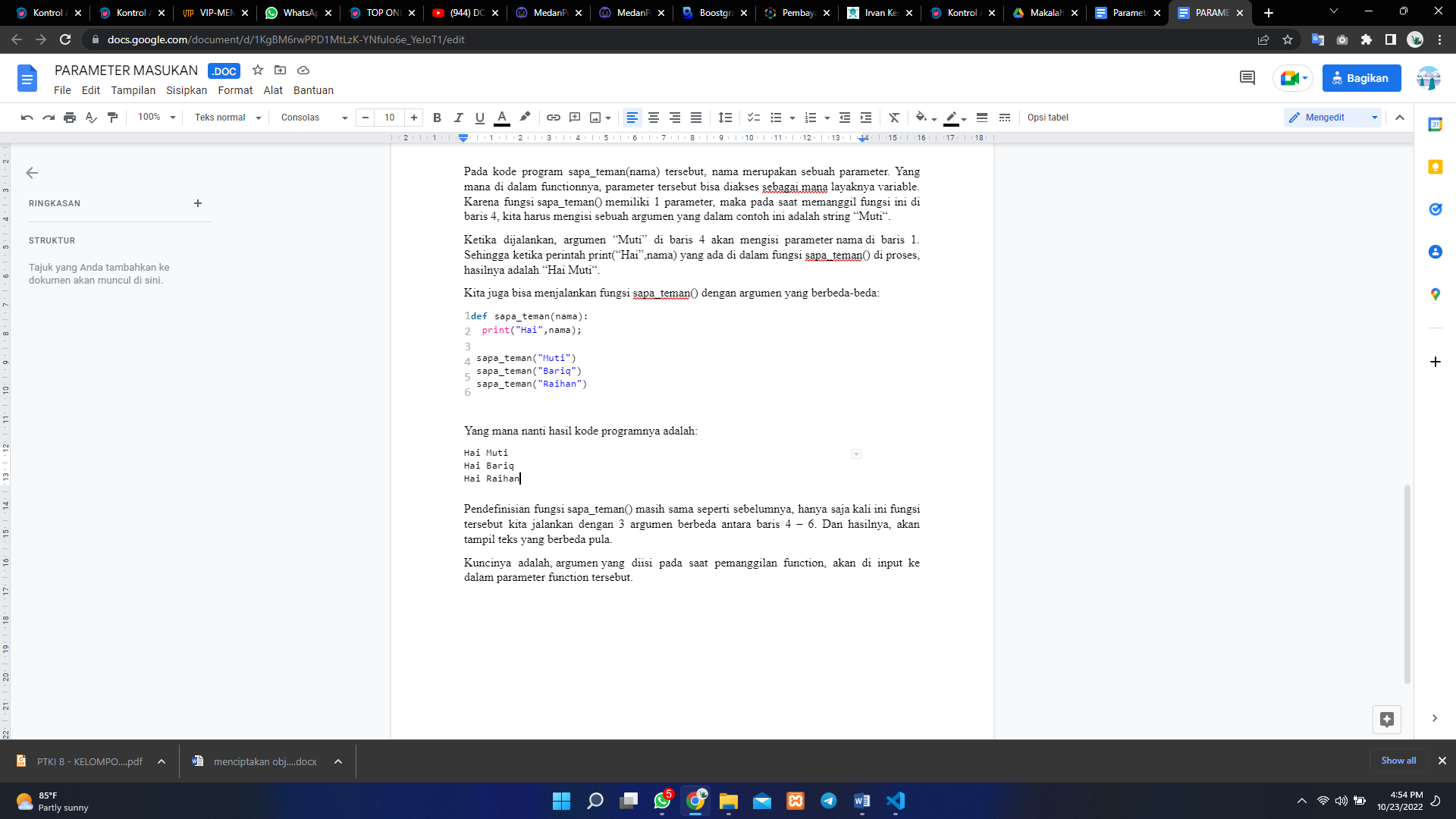
Sehingga hasil dari kode program tersebut adalah:



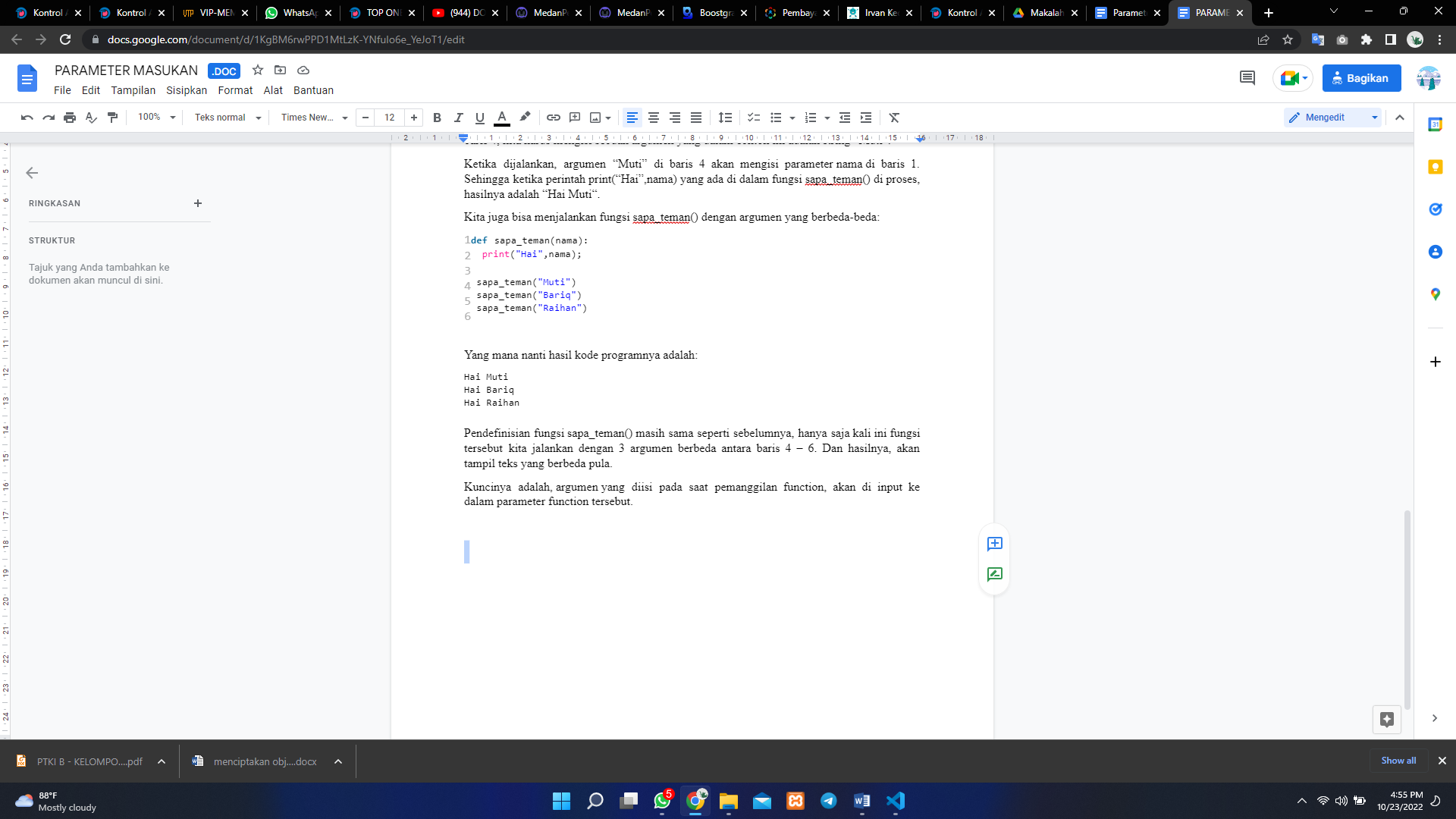
Pada kode program sapa\_teman(nama) tersebut, nama merupakan sebuah parameter. Yang mana di dalam functionnya, parameter tersebut bisa diakses sebagai mana layaknya variable. Karena fungsi sapa\_teman() memiliki 1 parameter, maka pada saat memanggil fungsi ini di baris 4, kita harus mengisi sebuah argumen yang dalam contoh ini adalah string “Muti“.

Ketika dijalankan, argumen “Muti” di baris 4 akan mengisi parameter nama di baris 1. Sehingga ketika perintah print(“Hai”,nama) yang ada di dalam fungsi sapa\_teman() di proses, hasilnya adalah “Hai Muti“.

Kita juga bisa menjalankan fungsi sapa\_teman() dengan argumen yang berbeda-beda:



Yang mana nanti hasil kode programnya adalah:



Pendefinisian fungsi sapa\_teman() masih sama seperti sebelumnya, hanya saja kali ini fungsi tersebut kita jalankan dengan 3 argumen berbeda antara baris 4 – 6. Dan hasilnya, akan tampil teks yang berbeda pula.

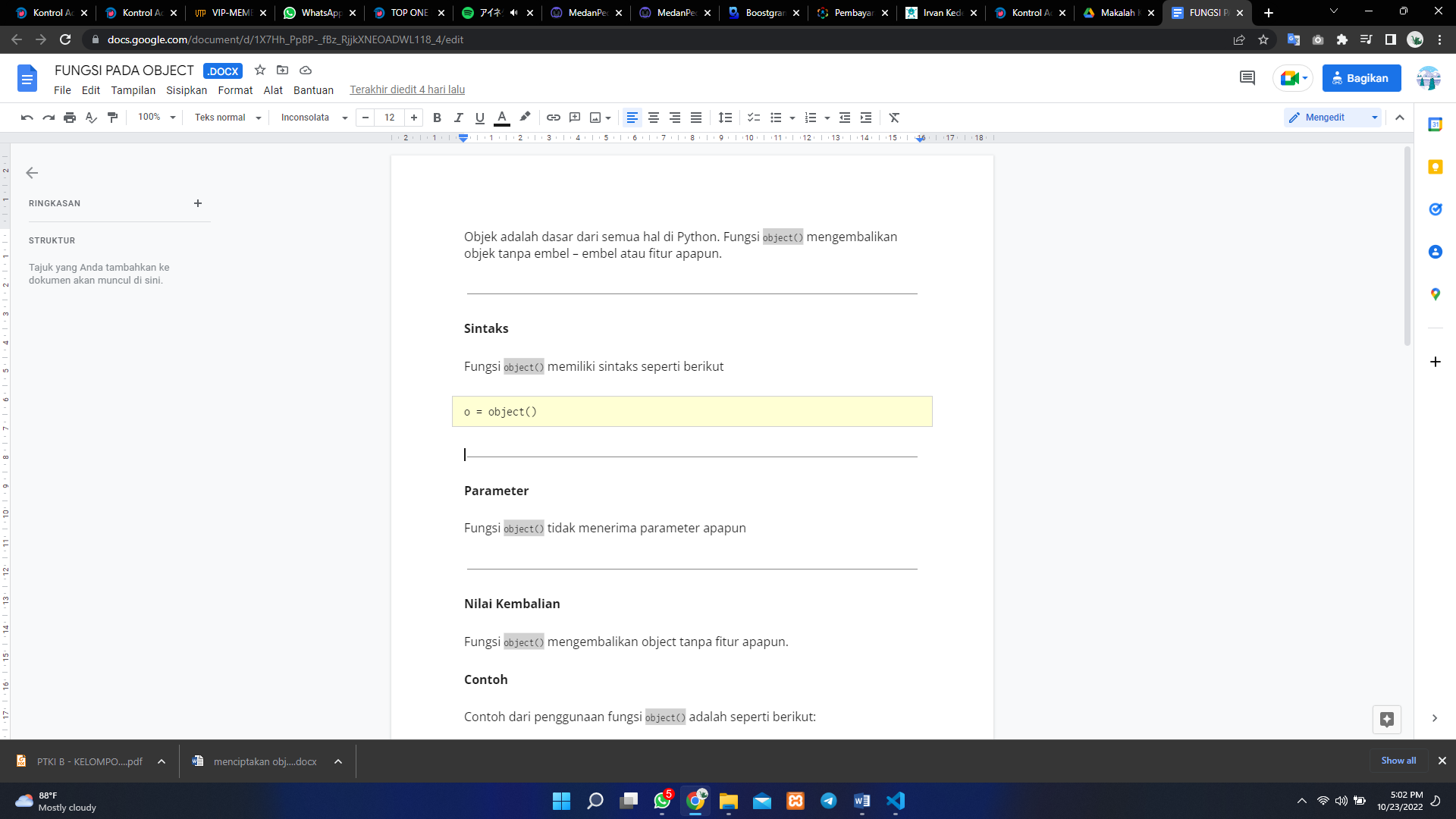
Kuncinya adalah, argumen yang diisi pada saat pemanggilan function, akan di input ke dalam parameter function tersebut.

## 2.6 Fungsi Pada Object

Objek adalah dasar dari semua hal di Python. Fungsi object() mengembalikan objek tanpa embel – embel atau fitur apapun.

1. Sintaks

Fungsi object() memiliki sintaks seperti berikut



1. Parameter

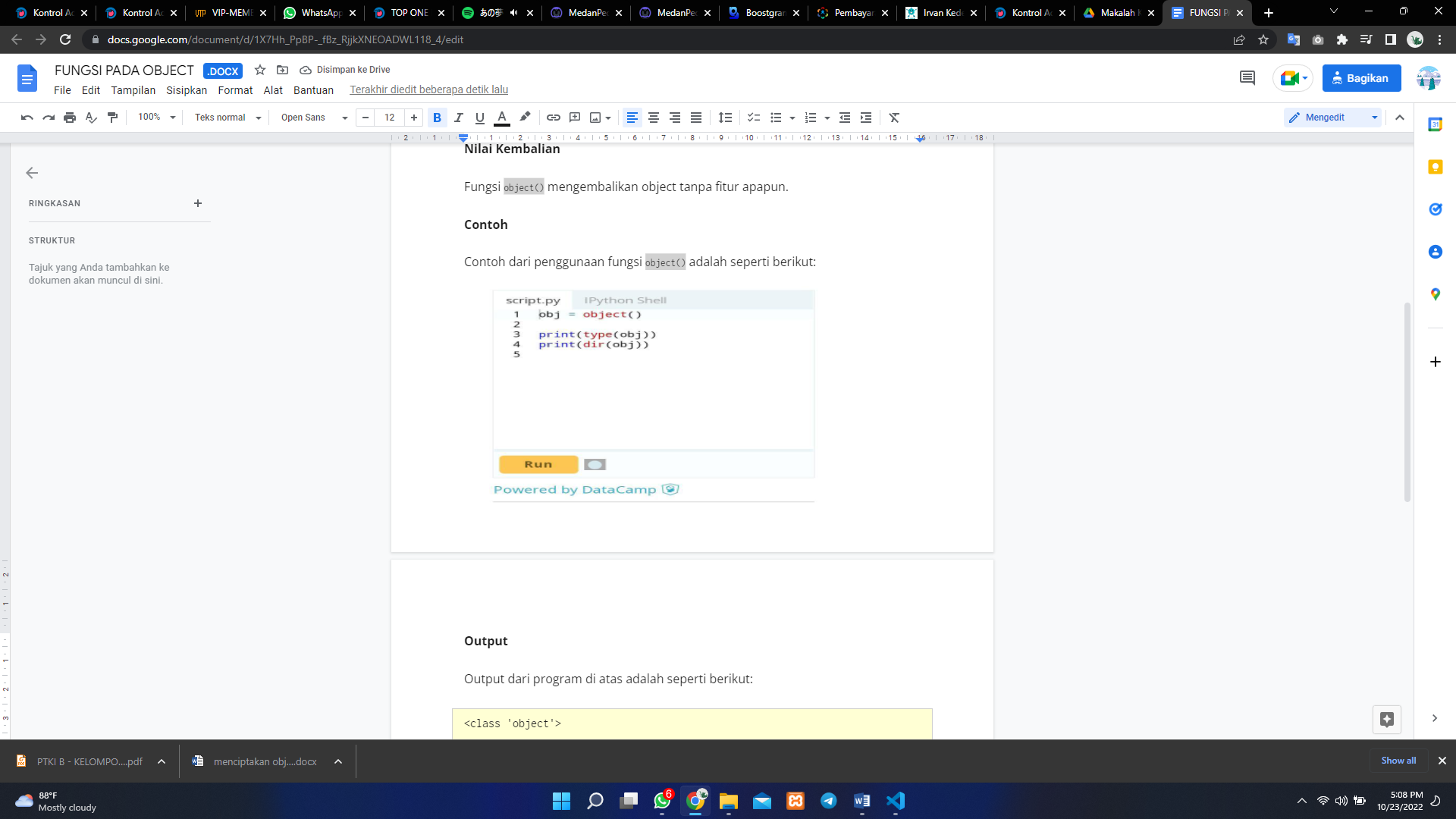
Fungsi object() tidak menerima parameter apapun

1. Nilai Kembalian

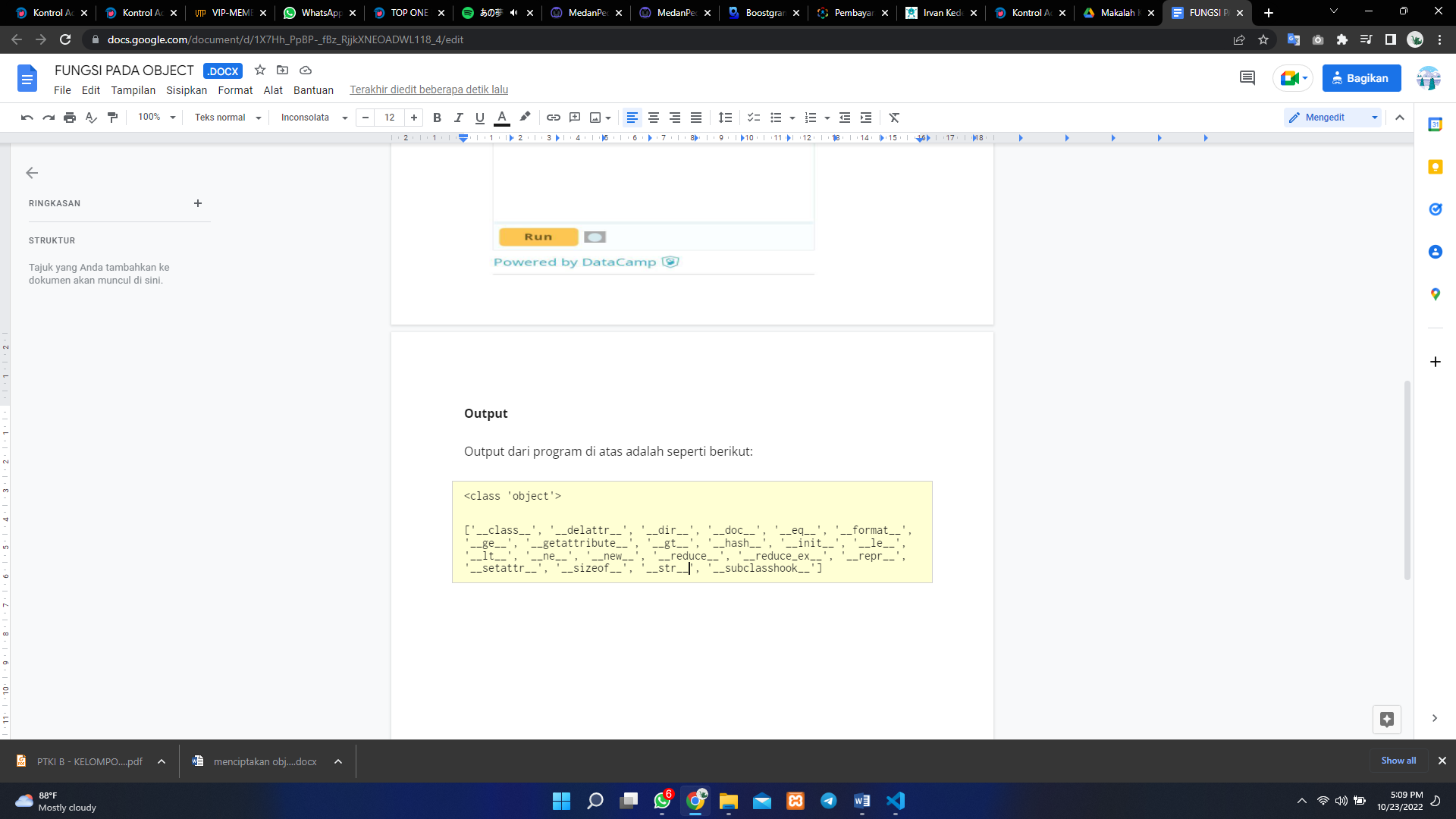
Fungsi object() mengembalikan object tanpa fitur apapun.

Contoh:

Contoh dari penggunaan fungsi object() adalah seperti berikut:



Output dari program di atas adalah seperti berikut:



## 2.7. Buatlah Program OOP Pada Python