

Nama : Faricha Aulia
Kelas : 2I / Informatics Engineering
NIM : 2141720155

Jobsheet 7 Inheritance Part 2

1. KOMPETENSI

- Memahami konsep dasar inheritance atau pewarisan.
- Mampu memahami jenis-jenis inheritance.
- Mampu mengimplementasikan konsep inheritance pada Bahasa pemrograman Java

2. PENDAHULUAN

Inheritance atau **pewarisan sifat** merupakan suatu cara untuk menurunkan suatu class yang lebih umum menjadi suatu class yang lebih spesifik. Inheritance adalah salah satu ciri utama suatu bahasa program yang berorientasi pada objek. Inti dari pewarisan adalah sifat reusable dari konsep object oriented. Setiap **subclass** akan “mewarisi” sifat dari **superclass** selama bersifat protected ataupun public.

Dalam inheritance terdapat dua istilah yang sering digunakan. Kelas yang menurunkan disebut kelas dasar (**base class/super class**), sedangkan kelas yang diturunkan disebut kelas turunan (**derived class/sub class/child class**). Di dalam Java untuk mendeklarasikan suatu class sebagai subclass dilakukan dengan cara menambahkan kata kunci **extends** setelah deklarasi nama class, kemudian diikuti dengan nama parent class---nya. Kata kunci extends tersebut memberitahu kompiler Java bahwa kita ingin melakukan perluasan class. Berikut adalah contoh deklarasi inheritance.

Contoh
kompiler
meng-

```
public class B extends A {  
    ...  
}
```

diatas memberitahukan
Java bahwa kita ingin
--extend class A ke

class B. Dengan kata lain, class B adalah subclass (class turunan) dari class A, sedangkan class A adalah parent class dari class B.

Karakteristik pada super class akan dimiliki juga oleh subclassnya. Terdapat 5 bentuk pewarisan: single inheritance, multilevel inheritance, hierarchical inheritance, hybrid inheritance, dan multiple inheritance.

- Single Inheritance : Suatu class yang hanya mempunyai satu parent class. Contoh:



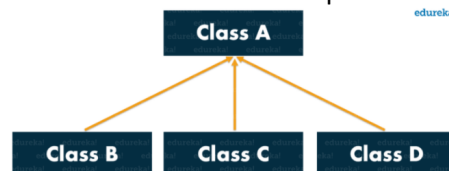
Berdasarkan Gambar 1 dapat diketahui bahwa class B merupakan subclass yang mempunyai satu parent yaitu class A sehingga disebut single inheritance.

- Multilevel Inheritance : Suatu subclass bisa menjadi superclass bagi class yang lain.



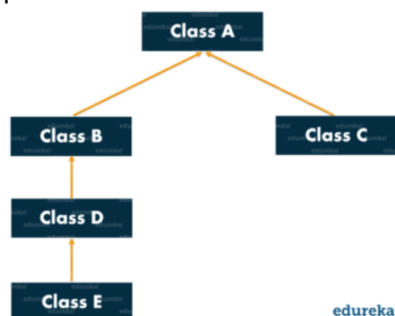
Berdasarkan Gambar 2 diatas dapat dilihat bahwa class B merupakan subclass dari class A, sehingga dalam hal ini class A adalah superclass dan class B adalah subclass. Kemudian class B yang awalnya merupakan subclass mempunyai subclass lagi yaitu class C sehingga class B menjadi superclass dari class C, begitu juga seterusnya jika class C memiliki subclass lagi.

3. Hierarchical Inheritance: Contoh pewarisan yang terjadi ketika sebuah kelas memiliki lebih dari satu kelas turunan (*subclass*) atau dengan kata lain, lebih dari satu kelas turunan memiliki kelas induk yang sama. Konsep ini sesuai dengan contoh di dunia nyata, dimana satu orangtua bisa memiliki lebih dari satu anak. Contoh bentuk hierarchical inheritance dapat dilihat pada Gambar 3.



Pada Gambar 3 dapat dilihat bahwa class A memiliki 3 subclass, yaitu class B, class C, dan class D. Dengan demikian, class B, class C, dan class D memiliki superclass yang sama, dan ada kemungkinan ketiga class subclass tersebut memiliki atribut dan behaviour yang sama dari class A.

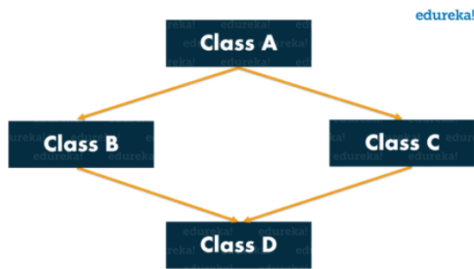
4. Hybrid Inheritance: Jenis pewarisan yang merupakan kombinasi dari dua atau lebih jenis pewarisan.



Pada Gambar 4 dapat dilihat bahwa bentuk pewarisan tersebut merupakan kombinasi dari hierarchical dan multilevel inheritance. Hierarchical inheritance ditunjukkan oleh Class A, class B, dan class C. Sedangkan multilevel inheritance ditunjukkan oleh class B, class D, dan class E.

Terdapat beberapa aturan pada pewarisan menggunakan Bahasa pemrograman Java, antara lain:

1. Multiple inheritance tidak dapat digunakan pada Bahasa pemrograman Java. Multiple inheritance mengacu pada proses dimana satu kelas turunan mencoba untuk memperluas (extend) lebih dari satu kelas induk.



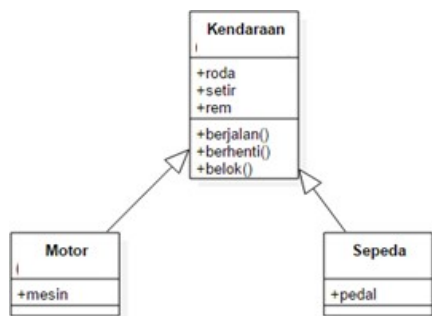
Misal terdapat method `show()` pada kelas B dan C dengan fungsi yang berbeda. Kemudian kelas D meng-extend kelas B dan C. Ketika objek dari kelas D mencoba memanggil method `show()`, kompilator akan bingung method di kelas mana yang akan dieksekusi (dari kelas B atau C). Sehingga hal ini mengarah pada ambiguitas.

2. Cyclic inheritance tidak dapat digunakan di Java. Cyclic inheritance merupakan jenis inheritance dimana sebuah kelas memperluas (extend) dirinya sendiri. Jenis ini tidak diizinkan oleh Java karena tidak ada peluang untuk memperluas kelas Object.

Atribut dan method dengan access modifier Private tidak diwariskan.

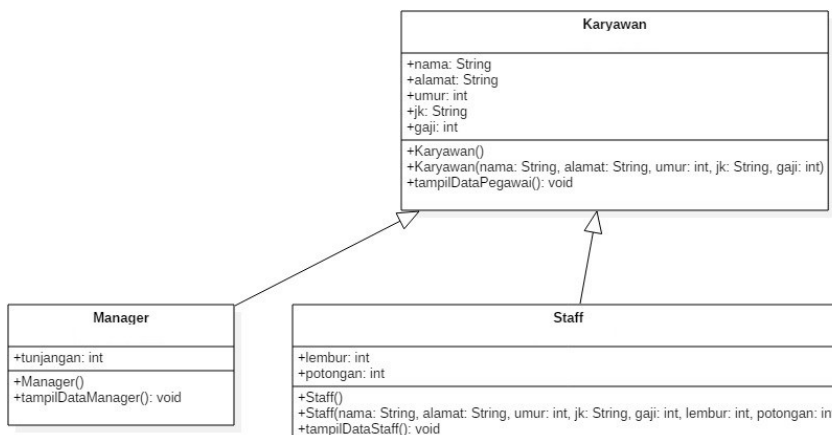
Constructor tidak dapat diwariskan. Jika ingin menggunakan constructor kelas induk, harus menggunakan perintah `super()` pada constructor kelas anak.

Pada class diagram, pewarisan digambarkan dengan sebuah garis tegas, dengan segitiga di ujungnya. Class yang dekat pada segitiga merupakan superclass, sedangkan class yang jauh dari segitiga merupakan subclass. Untuk membentuk sebuah subclass, keyword "extends" digunakan (lihat contoh pada sesi "Implementasi Pewarisan"). Berikut ini adalah contoh class diagram dari pewarisan:



3. PERCOBAAN

3.1. Percobaan 1



Program code:

```

public class Karyawan {
    public String nama, alamat, jk;
    public int umur, gaji;
    public Karyawan(){
    }
    public Karyawan(String nama, String alamat, String jk, int umur, int gaji){
        this.nama = nama;
        this.alamat = alamat;
        this.jk = jk;
        this.umur = umur;
        this.gaji = gaji;
    }
    public void tampilDataKaryawan(){
        System.out.println("Nama      : " +nama);
        System.out.println("Alamat    : " +alamat);
        System.out.println("Jenis Kelamin : " +jk);
        System.out.println("Umur      : " +umur);
        System.out.println("Gaji      : " +gaji);
    }
}

```

```

public class Manager extends Karyawan{
    public int tunjangan;
    public Manager(){
    }
    public void tampilDataManager(){
        super.tampilDataKaryawan();
        System.out.println("Tunjangan    : " +tunjangan);
        System.out.println("Total Gaji    : " +(super.gaji + tunjangan));
    }
}

```

```

public class Staff extends Karyawan{
    public int lembur, potongan;
    public Staff(){
    }
    public Staff(String nama, String alamat, String jk, int umur, int gaji, int lembur, int potongan){
        super(nama, alamat, jk, umur, gaji);
        this.lembur = lembur;
        this.potongan = potongan;
    }
    public void tampilDataStaff(){
        super.tampilDataKaryawan();
        System.out.println("Lembur      : " +lembur);
        System.out.println("Potongan    : " +potongan);
        System.out.println("Total Gaji    : " + (gaji + lembur - potongan));
    }
}

```

```

public class Inheritance1 {
    public static void main(String[] args) {
        Manager M = new Manager();
        M.nama = "Vivin";
        M.alamat = "Jl. Vinolia";
        M.umur = 25;
        M.jk = "Perempuan";
        M.gaji = 3000000;
        M.tunjangan = 1000000;
        M.tampilDataManager();
        Staff S = new Staff();
        S.nama = "Lestari";
        S.alamat = "Malang";
        S.umur = 25;
        S.jk = "Perempuan";
        S.gaji = 2000000;
        S.lembur = 500000;
        S.potongan = 250000;
        S.tampilDataStaff();
    }
}

```

Output:

```

Nama      : Vivin
Alamat    : Jl.Vinolia
Jenis Kelamin : Perempuan
Umur      : 25
Gaji      : 3000000
Tunjangan : 1000000
Total Gaji : 4000000
Nama      : Lestari
Alamat    : Malang
Jenis Kelamin : Perempuan
Umur      : 25
Gaji      : 2000000
Lembur    : 500000
Potongan  : 250000
Total Gaji : 2250000

```

3.1.1 Pertanyaan Percobaan 1

1. Sebutkan class mana yang termasuk super class dan sub class dari percobaan 1 diatas!

Jawab: Super class terdapat pada class karyawan sedangkan subclass terdapat pada class manajer dan class staff

2. Kata kunci apakah yang digunakan untuk menurunkan suatu class ke class yang lain?

Jawab: Menggunakan kata extend

3. Perhatikan kode program pada class Manager, atribut apa saja yang dimiliki oleh class tersebut? Sebutkan atribut mana saja yang diwarisi dari class Karyawan!

Jawab: Pada class Manager hanya memiliki atribut tunjangan(int). yang diwariskan dari class Karyawan adalah nama(String),alamat(String),jk(String),umur(int) dan gaji(int).

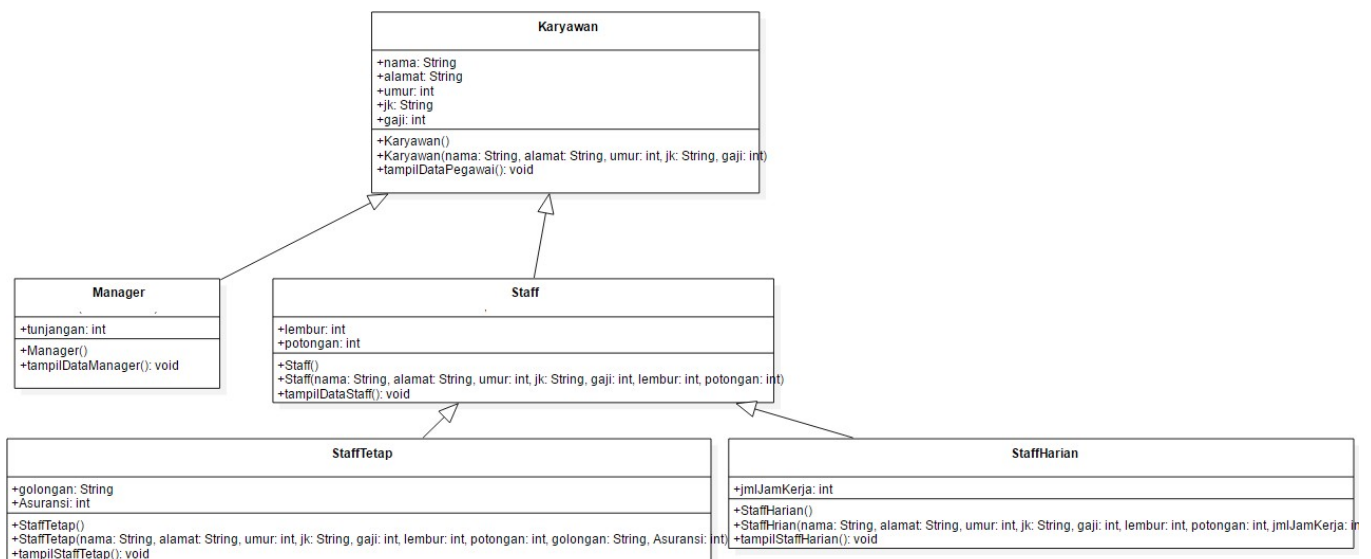
4. Jelaskan kata kunci super pada potongan program dibawah ini yang terdapat pada class Manager!

Jawab: Pada kata kunci super.gaji maksudnya menurunkan atribut gaji pada class karyawan atau
`System.out.println("Total Gaji =" + (super.gaji+tunjangan));`
 menunjukkan bahwa gaji berasal dari class parent.

5. Program pada percobaan 1 diatas termasuk dalam jenis inheritance apa? Jelaskan alasannya!

Jawab: Termasuk dalam single inheritance dikarenakan hanya satu pewarisan.

3.2. Percobaan 2



Program Code:

```

public class StaffHarian extends Staff{
    public int jmlJamKerja;

    public StaffHarian(){
    }

    public StaffHarian(String nama, String alamat, String jk, int umur, int gaji,int lembur, int potongan, int jmlJamKerja){
        super(nama, alamat, jk, umur, gaji, potongan, lembur);
    }
}

```

```

        this.jmlJamKerja = jmlJamKerja;
    }

    public void tampilStaffHarian(){
        System.out.println("=====Data Staff Harian=====");
        super.tampilDataStaff();
        System.out.println("Jumlah jam kerja: " +jmlJamKerja);
        System.out.println("Gaji bersih   : " +(gaji*jmlJamKerja+lembur-potongan));
    }
}

```

```

public class StaffTetap extends Staff{
    public String golongan;
    public int asuransi;

    public StaffTetap(){
    }

    public StaffTetap(String nama, String alamat, String jk, int umur, int gaji,int lembur, int potongan, String golongan, int asuransi){
        super(nama, alamat, jk, umur, gaji, potongan, lembur);
        this.golongan = golongan;
        this.asuransi = asuransi;
    }

    public void tampilStaffTetap(){
        System.out.println("=====Data Staff Tetap=====");
        super.tampilDataStaff();
        System.out.println("Golongan      : " +golongan);
        System.out.println("Jumlah asuransi : " +asuransi);
        System.out.println("Gaji bersih   : " +(gaji+lembur-potongan-asuransi));
    }
}

```

```

public class Inheritance1 {
    public static void main(String[] args) {
        StaffTetap ST = new StaffTetap("Budi", "Malang", "Laki-laki", 20, 2000000, 250000, 200000, "2A", 100000);
        ST.tampilStaffTetap();

        StaffHarian SH = new StaffHarian("Indah", "Malang", "Perempuan", 27, 10000, 100000, 50000, 100);
        SH.tampilStaffHarian();
    }
}

```

Output:

```

=====Data Staff Tetap=====
Nama       : Budi
Umur       : 20
Gaji       : 2000000
Lembur     : 200000
Potongan   : 250000
Total Gaji : 1950000
Golongan   : 2A
Jumlah asuransi : 100000
Gaji bersih : 1850000
=====Data Staff Harian=====
Nama       : Indah
Alamat     : Malang
Jenis Kelamin : Perempuan
Umur       : 27
Gaji       : 10000
Lembur     : 50000
Potongan   : 100000
Total Gaji : -40000
Jumlah jam kerja: 100
Gaji bersih : 950000

```

3.2. Pertanyaan Percobaan 2

1. Berdasarkan class diatas manakah yang termasuk single inheritance dan mana yang termasuk multilevel inheritance?

Jawab: Single inheritance terdapat pada class manager sedangkan multilevel inheritance terletak pada class staff

2. Perhatikan kode program class StaffTetap dan StaffHarian, atribut apa saja yang dimiliki oleh class

tersebut? Sebutkan atribut mana saja yang diwarisi dari class Staff!

Jawab: Pada class staffTetap terdapat atribut asuransi(int) dan golongan(String). Pada class staffHarian terdapat class jmlJamKerja(int). Kedua class tersebut mewarisi sifat dari class staff yaitu nama(String), alamat(String), jk(String), umur(int), gaji(int), lembur(int), dan potongan(int)

3. Apakah fungsi potongan program berikut pada class StaffHarian

Jawab: Memanggil atau menggunakan atribut nama, alamat, jk, umur, gaji, lembur, potongan melalui konstruktor yang berasal dari class parent.

4. Apakah fungsi potongan program berikut pada class StaffHarian

Jawab: Digunakan untuk memanggil atau menggunakan method tampilDataStaff yang terdapat pada class pewaris atau class parents yaitu class staff

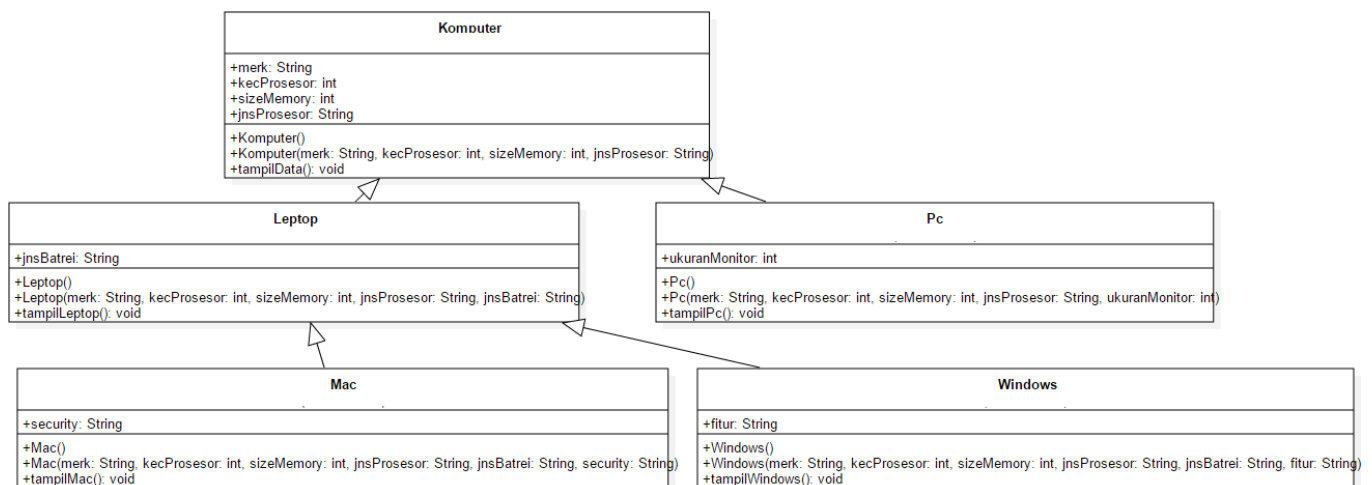
5. Perhatikan kode program dibawah ini yang terdapat pada class StaffTetap. Terlihat dipotong program diatas atribut gaji, lembur dan potongan dapat diakses langsung. Kenapa hal ini bisa terjadi dan bagaimana class StaffTetap memiliki atribut gaji, lembur, dan potongan padahal dalam class tersebut tidak dideklarasikan atribut gaji, lembur, dan potongan?

```
System.out.println("Gaji Bersih      "+(gaji+lembur-potongan-asuransi));
```

Jawab: Pada class StaffTetap, atribut dapat dideklarasikan karena pada atribut-atribut tersebut menggunakan modifier public yang sudah dideklarasikan pada class karyawan atau parents maka dari itu class lain dapat menggunakan atribut tersebut. Class StaffTetap dapat menggunakan atribut tersebut karena mewarisi dari class parents yaitu karyawan

4. TUGAS

Buatlah sebuah program dengan konsep pewarisan seperti pada class diagram berikut ini. Kemudian buatlah instansiasi objek untuk menampilkan data pada class Mac, Windows dan Pc!



Program Code:

```
public class Komputer {
    public String merk, jnsProsesor;
    public int kecProsesor, sizeMemory;

    public Komputer(){
    }

    public Komputer(String merk, int kecProsesor, int sizeMemory, String jnsProsesor){
        this.merk = merk;
        this.kecProsesor = kecProsesor;
        this.sizeMemory = sizeMemory;
        this.jnsProsesor = jnsProsesor;
    }

    public void tampilData(){
```

```

        System.out.println("Merk          : " + merk);
        System.out.println("Kecepatan Prosesor : " + kecProsesor + " GHz");
        System.out.println("Size Memory      : " + sizeMemory + " GB");
        System.out.println("Jenis prosesor   : " + jnsProsesor);
    }
}

```

```

public class PC extends Komputer{
    public int ukuranMonitor;

    public PC(){

    }

    public PC(String merk, int kecProsesor, int sizeMemory, String jnsProsesor, int ukuranMonitor){
        super(merk, kecProsesor, sizeMemory, jnsProsesor);
        this.ukuranMonitor = ukuranMonitor;
    }

    public void tampilPC(){
        System.out.println("===== PC =====");
        super.tampilData();
        System.out.println("Ukuran monitor   : " + ukuranMonitor);
    }
}

```

```

public class Laptop extends Komputer{
    public String jnsBaterai;

    public Laptop(){

    }

    public Laptop(String merk, int kecProsesor, int sizeMemory, String jnsProsesor, String jnsBaterai){
        super(merk, kecProsesor, sizeMemory, jnsProsesor);
        this.jnsBaterai = jnsBaterai;
    }

    public void tampilLaptop(){
        super.tampilData();
        System.out.println("Jenis Baterai    : " + jnsBaterai);
    }
}

```

```

public class Mac extends Laptop{
    public String security;

    public Mac(){

    }

    public Mac(String merk, int kecProsesor, int sizeMemory, String jnsProsesor, String jnsBaterai, String security){
        super(merk, kecProsesor, sizeMemory, jnsProsesor, jnsBaterai);
        this.security = security;
    }

    public void tampilMac(){
        System.out.println("===== Laptop Mac =====");
        super.tampilData();
        System.out.println("Security         : " + security);
    }
}

```



```

public class Windows extends Laptop{
    public String fitur;

    public Windows(){

    }

    public Windows(String merk, int kecProsesor, int sizeMemory, String jnsProsesor, String jnsBaterai, String fitur){
        super(merk, kecProsesor, sizeMemory, jnsProsesor, jnsBaterai);
        this.fitur = fitur;
    }

    public void tampilWindows(){
        System.out.println("===== Laptop Windows =====");
        super.tampilData();
        System.out.println("Fitur          : " + fitur);
    }
}

```

```

public class Main {
    public static void main(String[] args) {
        PC pc = new PC("ASUS", 4200, 1000, "Intel Core i9", 24);
        pc.tampilPC();

        Mac mac = new Mac("Apple", 2500, 500, "Chip Apple M1", "Lithium Ion", "Secure Enclave");
        mac.tampilMac();

        Windows wndw = new Windows("ASUS", 3200, 512, "Intel Core i7", "Nickel Cadmium", "Fingerprint");
        wndw.tampilWindows();
    }
}

```

Output:

```

===== PC =====
Merk          : ASUS
Kecepatan Prosesor : 4200 GHz
Size Memory    : 1000 GB
Jenis prosesor  : Intel Core i9
Ukuran monitor : 24
===== Laptop Mac =====
Merk          : Apple
Kecepatan Prosesor : 2500 GHz
Size Memory    : 500 GB
Jenis prosesor  : Chip Apple M1
Security       : Secure Enclave
===== Laptop Windows =====
Merk          : ASUS
Kecepatan Prosesor : 3200 GHz
Size Memory    : 512 GB
Jenis prosesor  : Intel Core i7
Fitur          : Fingerprint
=====
PC: 6.11.1 (ASUS) 6081.155705173

```