

# TANTANGAN PENGEMBANGAN DAN PENGELOLAAN SOFTWARE

---

Dr. Noprianto

2020-Feb-17

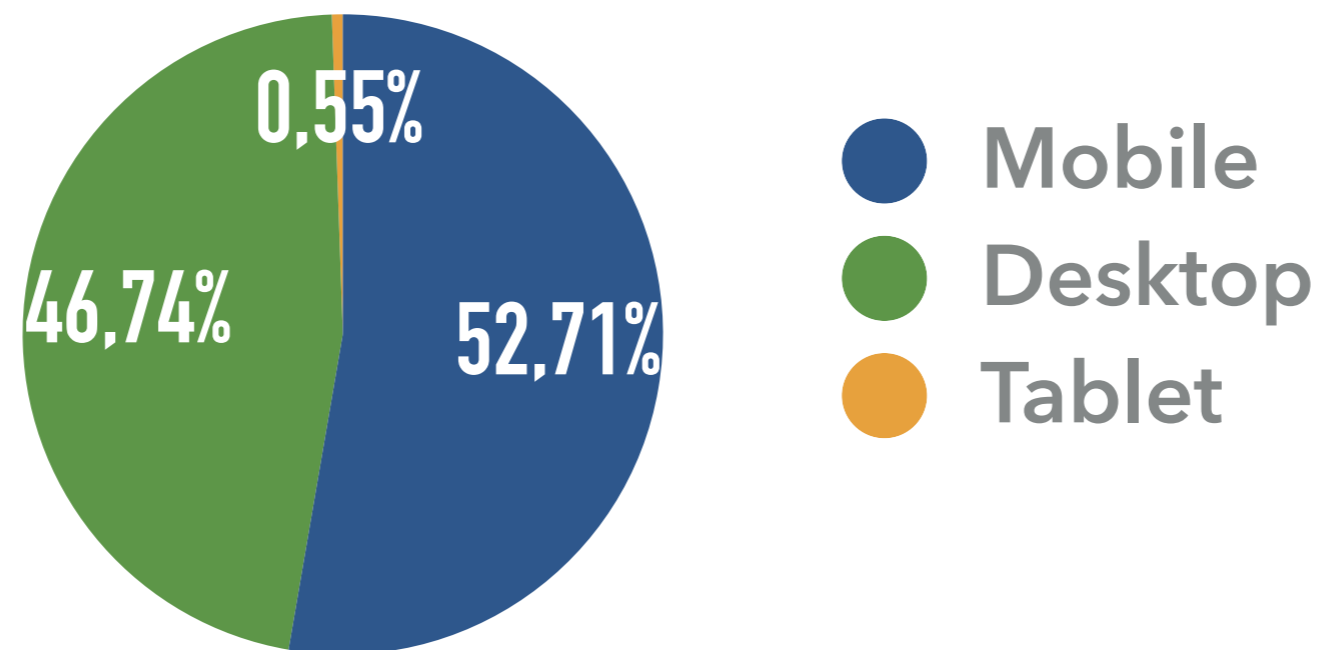
[nopri.anto@icloud.com](mailto:nopri.anto@icloud.com)

[noprianto.com](http://noprianto.com)

## A. PERANGKAT, AKSES INTERNET, PENGGUNA

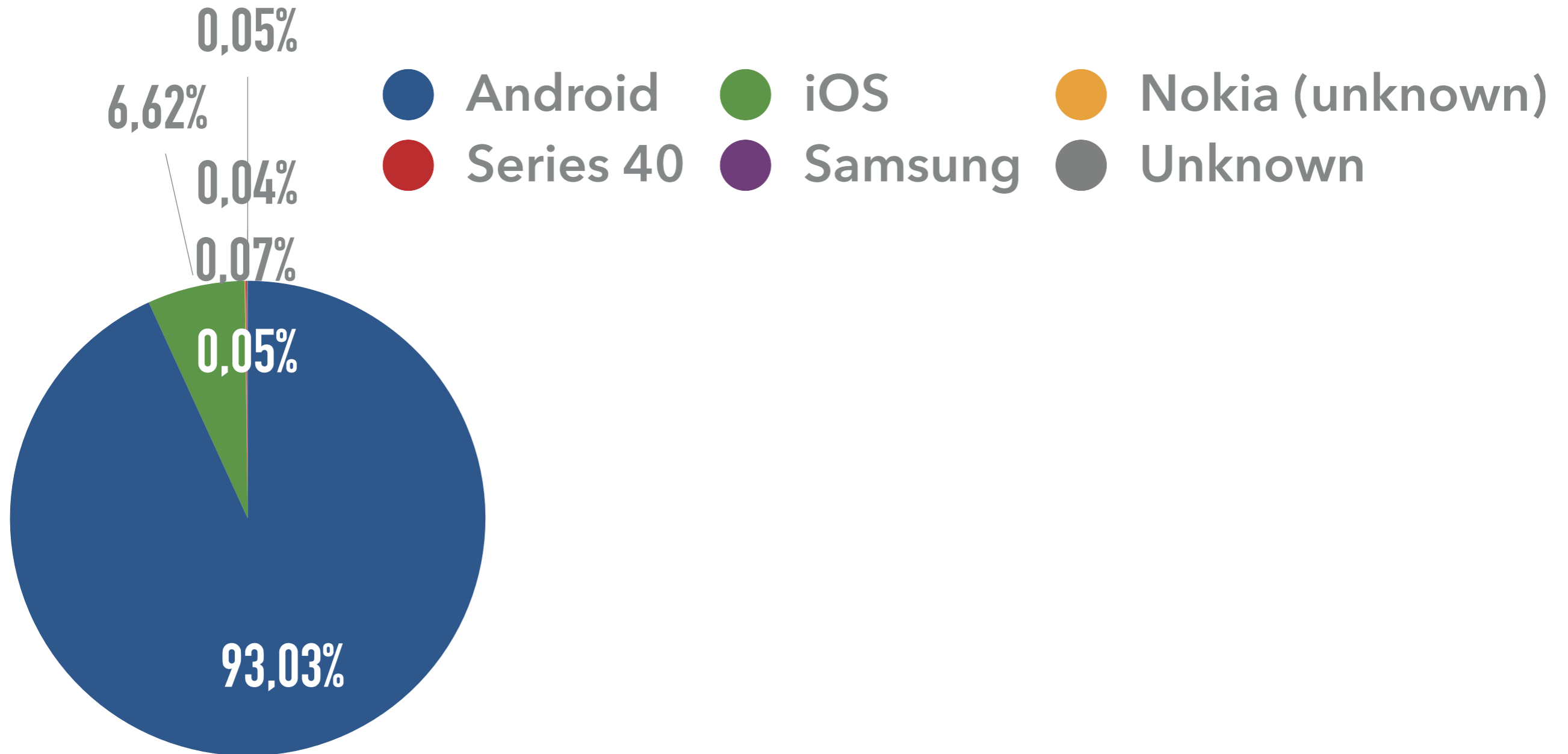
- ▶ Pengguna berpotensi mengakses software dari berbagai perangkat: ponsel pintar, tablet, laptop, desktop, alat rumah tangga, perangkat dalam kendaraan, dan lainnya.
  - ▶ Masing-masing perangkat memiliki kapabilitas yang bisa sangat berbeda.
- ▶ Perkembangan infrastruktur memungkinkan Internet semakin terjangkau: semakin banyak pengguna.
  - ▶ Skalabilitas: seberapa software yang dikembangkan dapat menyesuaikan dengan jumlah pengguna?

## MARKET SHARE: DESKTOP – MOBILE – TABLET (INDONESIA, 2020–JAN)

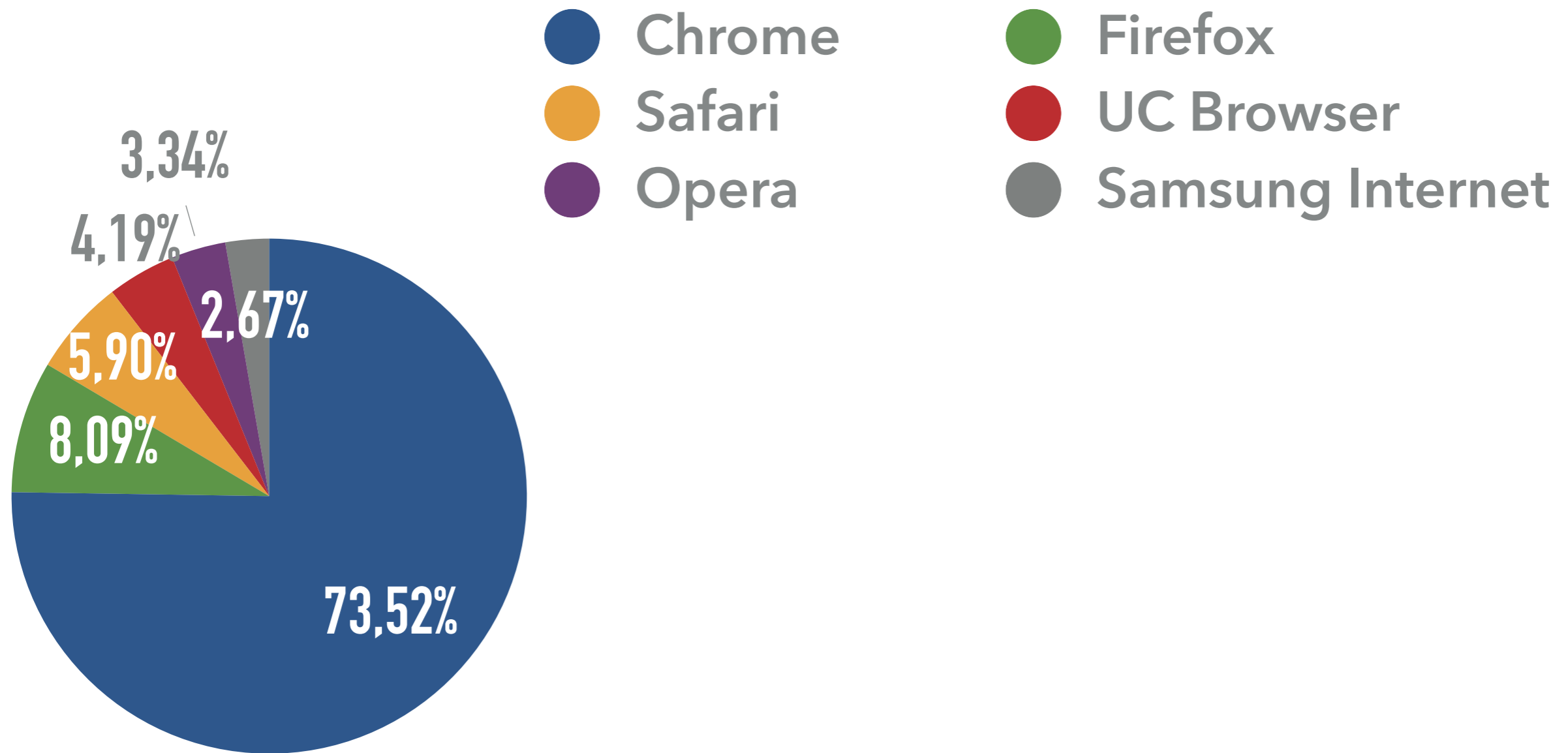


Data: <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/indonesia>

## MARKET SHARE: SISTEM OPERASI PERANGKAT MOBILE (INDONESIA, 2020-JAN)



## MARKET SHARE: BROWSER (INDONESIA, 2020-JAN)



## JUMLAH PENGGUNA INTERNET DI INDONESIA (2018)

Populasi (juta)	Pengguna Internet (juta)
264,16	171,17

## B. TEKNOLOGI PENGEMBANGAN SOFTWARE

- ▶ Terdapat sejumlah bahasa pemrograman, virtual machine, pustaka, framework: lahir, berkembang, tidak lagi dikembangkan
- ▶ Kita mungkin sedang menggunakan teknologi tertentu yang pengembangannya dapat dihentikan dalam beberapa waktu ke depan
- ▶ Perkembangan teknologi tidak selalu menawarkan kompatibilitas

## CERITA: PYTHON2 VS PYTHON3

- ▶ Python 3 dirilis pertama kali pada 2008 dan merupakan versi yang tidak kompatibel dengan Python 2.
- ▶ EOL untuk Python 2.7 diundur sampai 1 Januari 2020.
- ▶ Apa pelajaran yang bisa kita dapatkan?



## C. VENDOR LOCK-IN

- ▶ Untuk mengurangi biaya pengembangan dan pemeliharaan, serta mempercepat waktu ke pasar, perusahaan dapat membeli layanan dari pihak lain
- ▶ Vendor lock-in dapat terjadi
- ▶ Apa yang dapat kita lakukan ketika vendor tidak lagi beroperasi atau layanan dihentikan oleh vendor?

## D. KETIDAKSIAPAN SUMBER DAYA MANUSIA

- ▶ Tidak mudah mengelola tim pengembang software dengan berbagai karakteristik
- ▶ Kombinasikan ini dengan jadwal pengembangan yang ketat, kurangnya pengujian, kurangnya disiplin, pemilihan teknologi yang kurang cocok, dan lainnya

## CONTOH: DYNAMIC PROGRAMMING LANGUAGE

Isi file test.py

```
def test(x):  
    return x * x  
  
print(test(100))  
print(test('100'))
```

```
python test.py
```

```
10000
```

```
Traceback (most recent call last):
```

```
  File "test.py", line 6, in <module>
```

```
    print(test('100'))
```

```
  File "test.py", line 3, in test
```

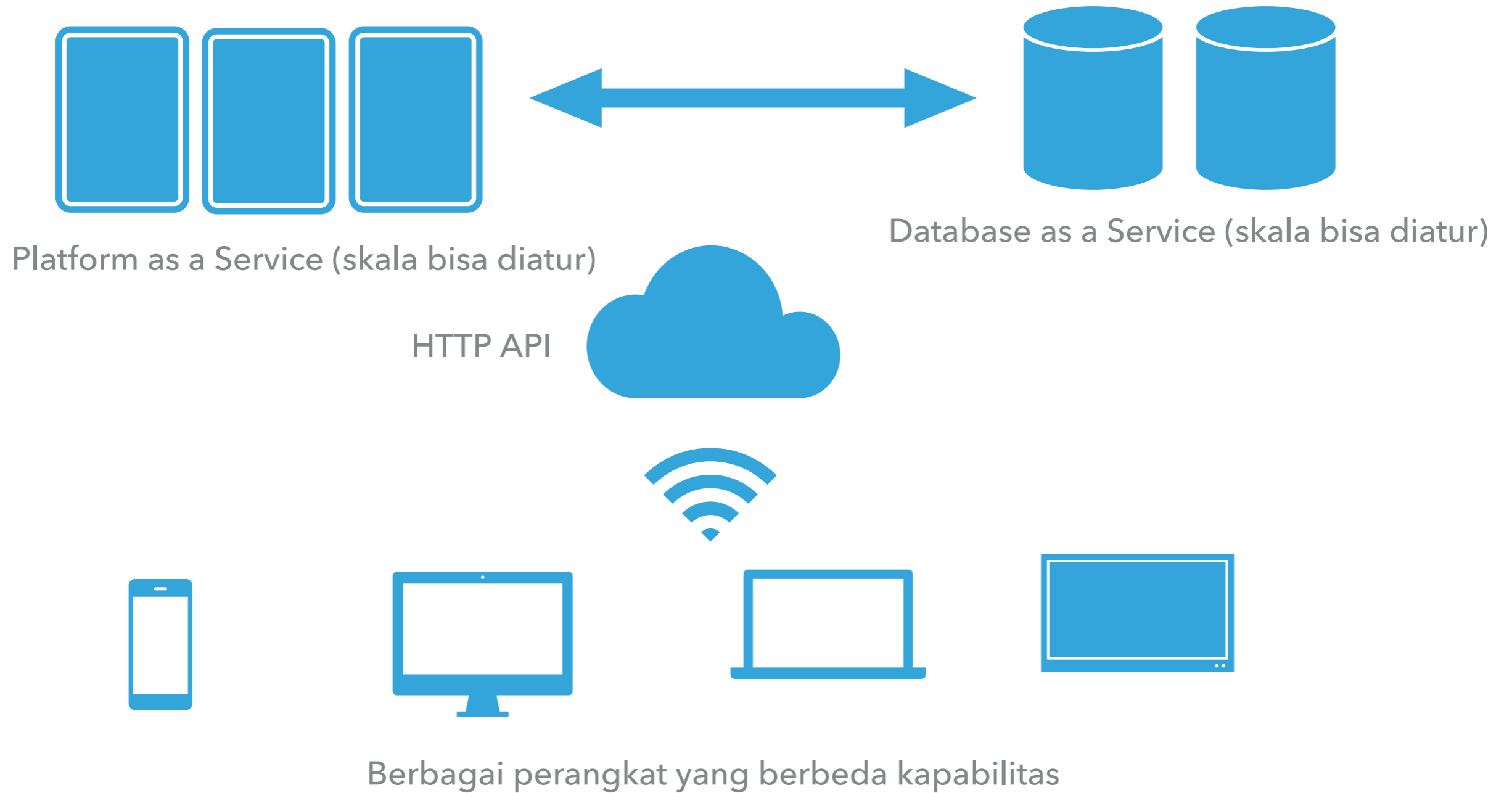
```
    return x * x
```

```
TypeError: can't multiply sequence by non-int of type 'str'
```

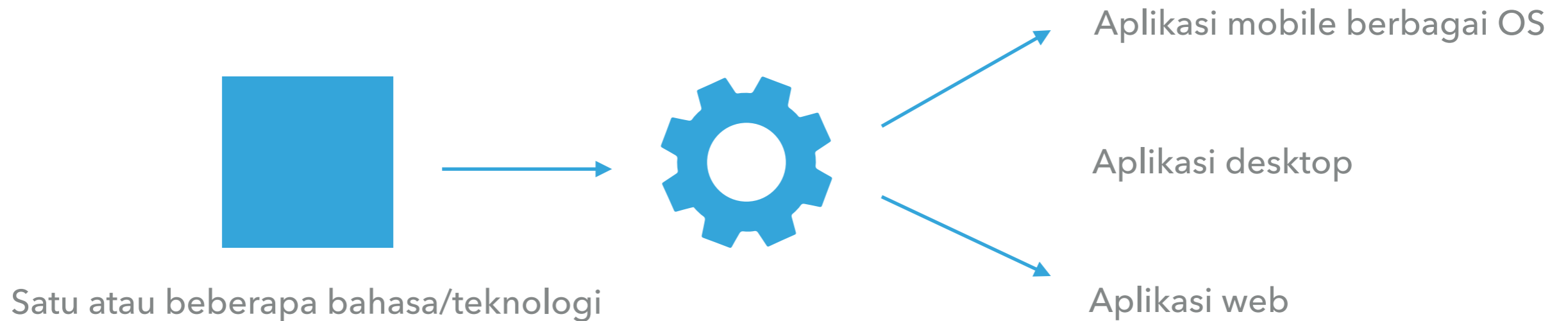
## A. PERANGKAT, AKSES INTERNET, PENGGUNA

- ▶ Berbagai perangkat: software dirancang dalam arsitektur *multitier*, dimana pada tier tertentu, API disediakan. API tersebut kemudian dapat digunakan oleh setiap perangkat, yang dapat memiliki kapabilitas yang berbeda-beda.
  - ▶ HTTP digunakan sebagai protokol.
  - ▶ Untuk mengantisipasi perangkat dengan kapabilitas yang terlalu terfragmentasi, fungsionalitas disediakan di sisi server, apabila dimungkinkan.
- ▶ Skalabilitas: software dirancang supaya application server tidak menyimpan data, sehingga dapat diatur skalanya secara horizontal, vertikal, atau kombinasinya.
- ▶ Fragmentasi perangkat: tool pengembangan cross-platform?

# GAMBARAN DASAR ARSITEKTUR



# CROSS-PLATFORM MOBILE/DESKTOP/WEB APPLICATION DEVELOPMENT



Diskusi: Bisakah kita memanfaatkan secara penuh kapabilitas yang disediakan oleh perangkat?

## B. TEKNOLOGI PENGEMBANGAN SOFTWARE

- ▶ Perubahan dan inkompatibilitas:
  - ▶ Microservices:
    - ▶ Dampak perubahan/inkompatibilitas bisa diminimalisir
  - ▶ Menggunakan teknologi yang sudah teruji, matang, kompatibel
  - ▶ Menggunakan bahasa pemrograman tersendiri - yang pengembangannya dapat dikontrol - untuk kebutuhan tertentu
    - ▶ Studi kasus: bahasa pemrograman Singkong

## C. VENDOR LOCK-IN

- ▶ Sebisa mungkin, gunakanlah layanan cloud computing dari penyedia yang telah teruji.
- ▶ Usahakan untuk memilih teknologi mendasar yang dapat di 'install di server sendiri', andaikata layanan tidak lagi disediakan.
- ▶ Atau, gunakanlah teknologi yang juga disediakan oleh vendor-vendor lain.



## D. KETIDAKSIAPAN SUMBER DAYA MANUSIA

- ▶ Dari sisi teknis:
  - ▶ Pilih teknologi yang dapat mengurangi dampak akibat: pemrograman yang ceroboh, pengujian yang kurang, dan lainnya
  - ▶ Manfaatkan alat bantu seperti source code management dan code review
  - ▶ Blue-Green deployment

# STATIC VS DYNAMIC PROGRAMMING LANGUAGE

```
def test(x):
    return x * x

print(test(100))
print(test('100'))
```

```
python test.py
10000
Traceback (most recent call last):
  File "test.py", line 6, in <module>
    print(test('100'))
  File "test.py", line 3, in test
    return x * x
TypeError: can't multiply sequence by non-int of type 'str'
```

```
public class Test {
    public static int test(int x) {
        return x * x;
    }

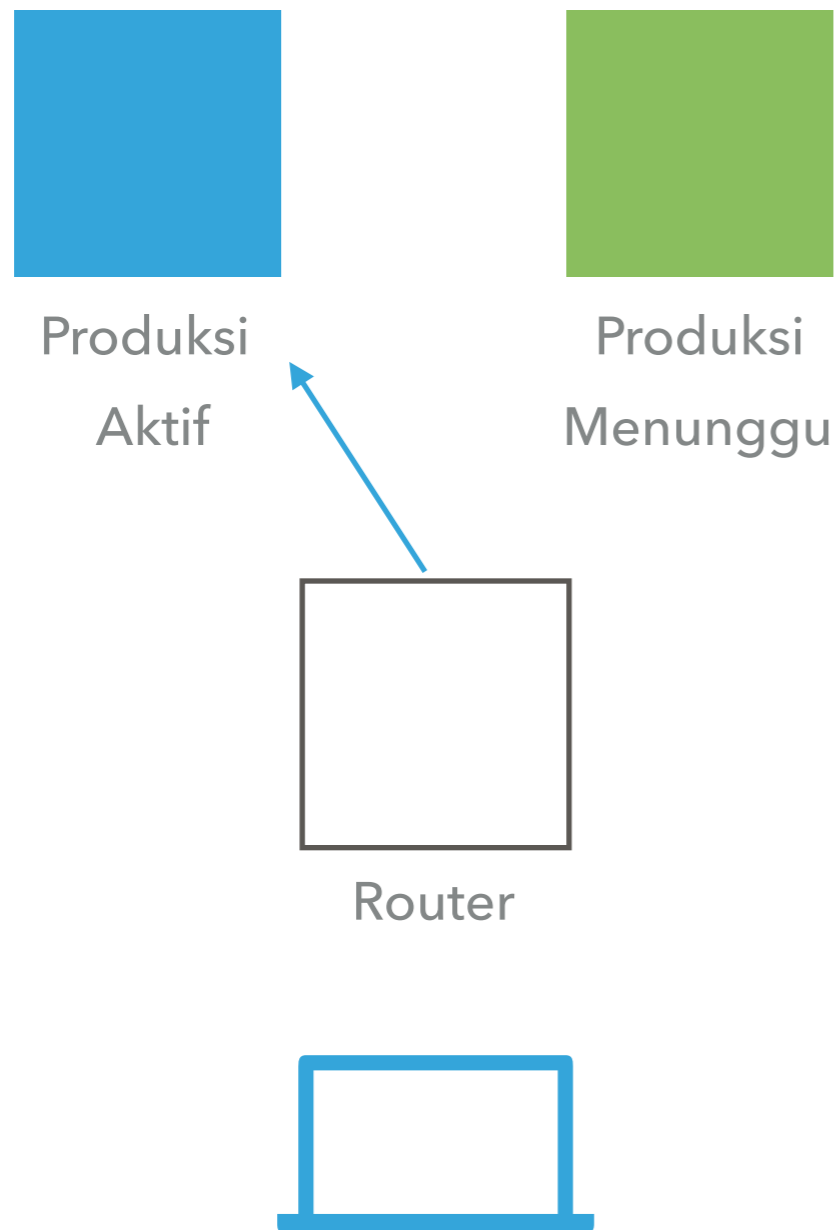
    public static void main(String[] args) {
        System.out.println(test(100));
        System.out.println(test("100"));
    }
}
```

Kompilasi gagal (kesalahan dicegah pada saat kompilasi)

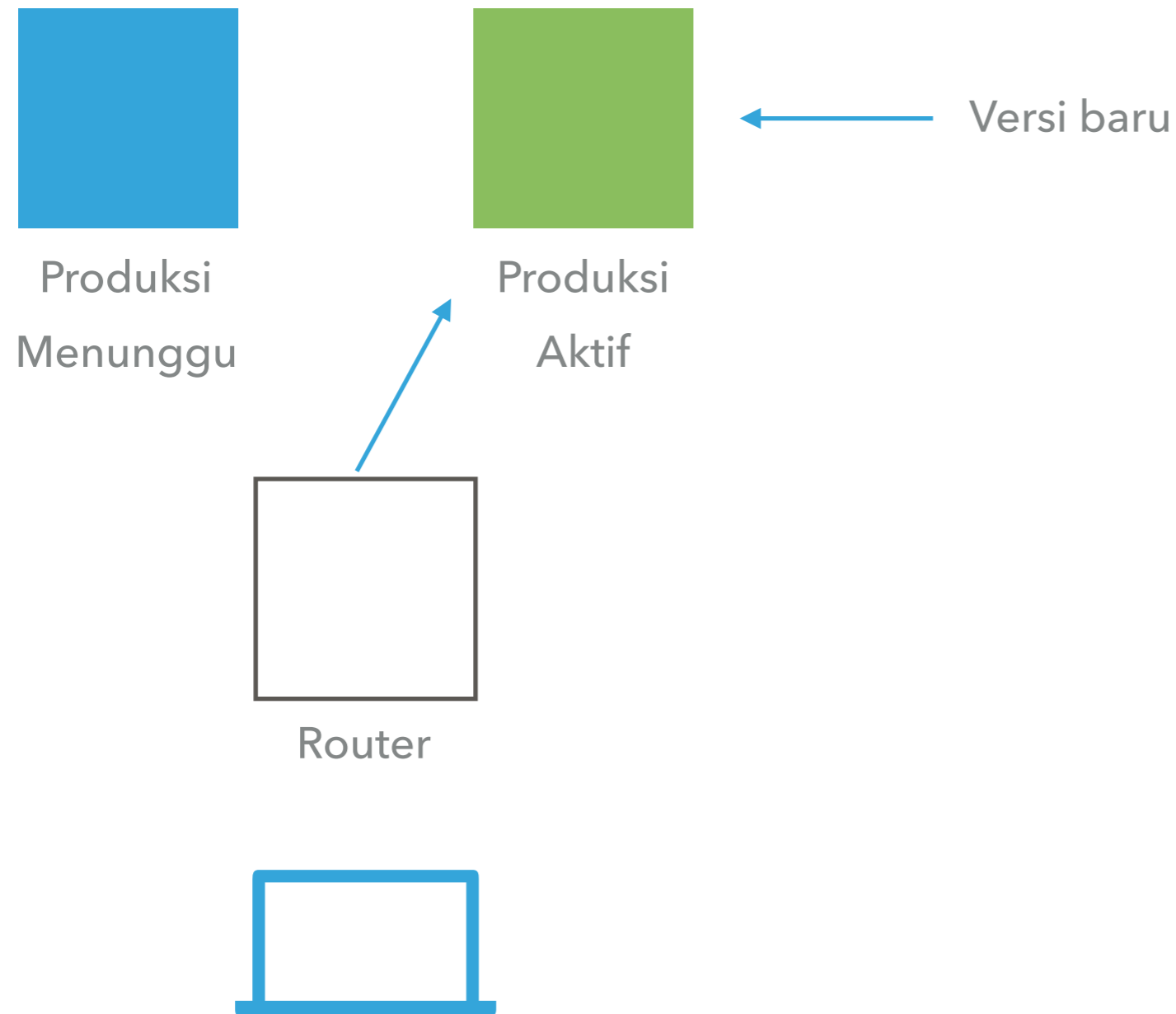
```
javac Test.java
Test.java:9: error: incompatible types: String cannot be converted to int
        System.out.println(test("100"));
                               ^
```

Note: Some messages have been simplified; recompile with `-Xdiags:verbose` to get full output  
1 error

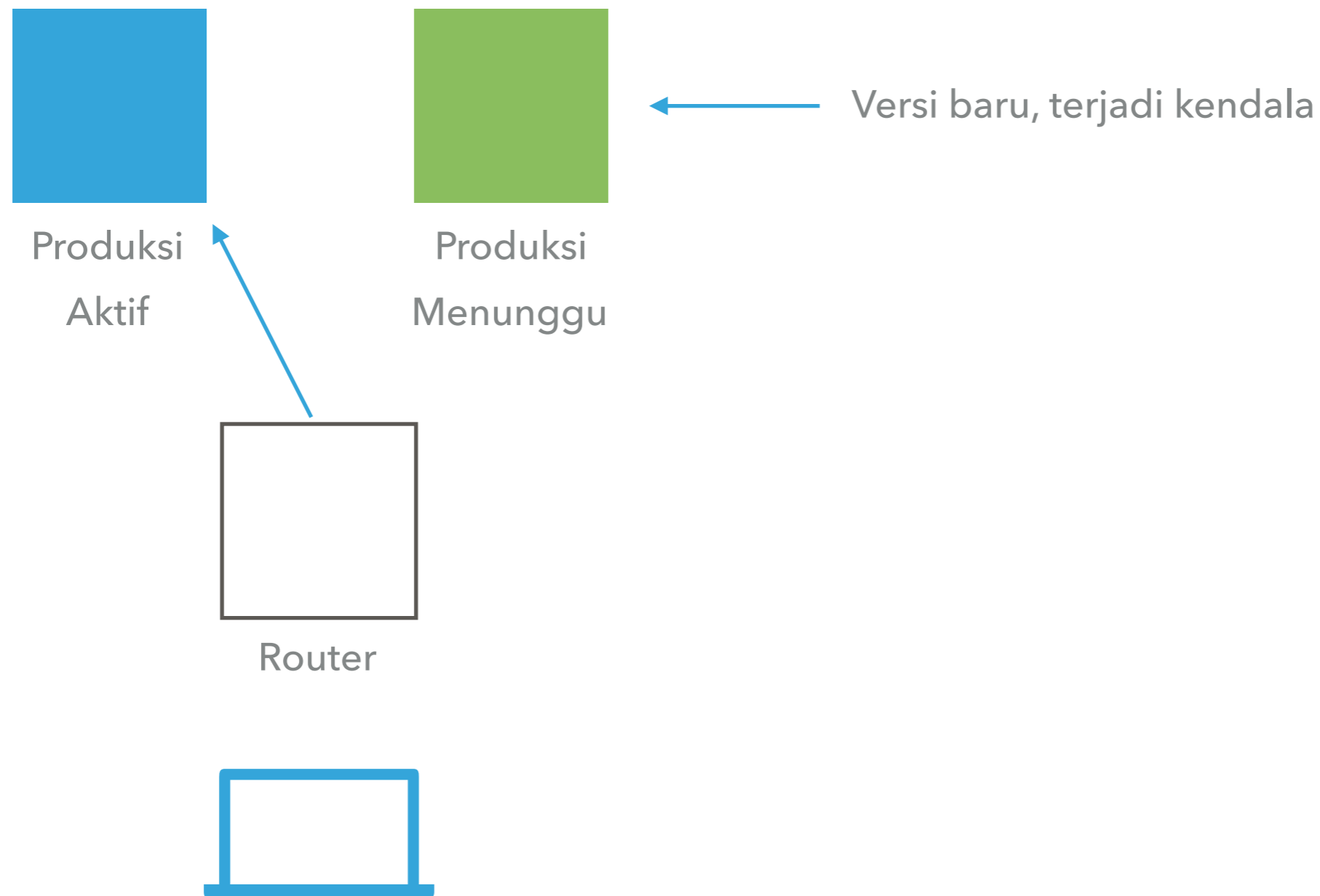
# BLUE-GREEN DEPLOYMENT



# BLUE-GREEN DEPLOYMENT (KETIKA ADA VERSI BARU)



# BLUE-GREEN DEPLOYMENT (ADA KENDALA PADA VERSI BARU)



**TERIMA KASIH**