Microservice GO Programming

M Octaviano Pratama, S.Kom., M.Kom
Chief Scientist BISA AI

Golang

Bisa.AI
AI FOR EVERYONE

Konsep Dasar golang

Apa itu Golang?

- Bahasa program open source yang dikembangkan oleh tim Google.
- Mengadaptasi keunggulan bahasa bahasa program lama seperti keluarga C dengan kombinasi bahasa modern.
- Biasanya digunakan sebagai bahasa backend.

Kelebihan Golang

- Mendukung konkurensi di level bahasa dengan pengaplikasian cukup mudah
- Mendukung pemrosesan data dengan banyak prosesor dalam waktu yang bersamaan *(pararel processing)*
- Memiliki *garbage collector*
- Proses kompilasi sangat cepat
- Bukan bahasa pemrograman yang hirarkial, menjadikan developer tidak perlu *ribet* memikirkan segmen OOP-nya
- Package/modul yang disediakan terbilang lengkap. Karena bahasa ini open source, banyak sekali developer yang juga mengembangkan modul-modul lain yang bisa dimanfaatkan

Instalasi Golang (Windows)

- Download *installer* di https://golang.org/dl/. Pilih *installer* untuk sistem operasi Windows sesuai jenis bit yang digunakan.
- Setelah ter-*download*, jalankan *installer*, klik *next* hingga proses instalasi selesai. *By default* jika anda tidak merubah path pada saat instalasi, Go akan ter-*install* di `C:\go`. *Path* tersebut secara otomatis akan didaftarkan dalam `PATH` *environment variable*.
- Buka *Command Prompt* / *CMD*, eksekusi perintah berikut untuk mengecek versi Go.

  ```
  go version
  ```
- Jika output adalah sama dengan versi Go yang ter-*install*, menandakan proses instalasi berhasil.

Instalasi Golang (MacOS)

- *Install* terlebih dahulu Homebrew (jika belum ada), caranya jalankan perintah berikut di **terminal**.

```
$ ruby -e "$(curl -fsSL http://git.io/pVOl)"
```

- *Install* Go menggunakan command `brew`.

```
$ brew install go
```

- Tambahkan path binary Go ke `PATH` *environment variable*.

```
$ echo "export PATH=$PATH:/usr/local/go/bin" >> ~/.bash_profile

$ source ~/.bash_profile
```

- Jalankan perintah berikut mengecek versi Go.

```
go version
```

- Jika output adalah sama dengan versi Go yang ter-*install*, menandakan proses instalasi berhasil.

Instalasi Golang (Linux)

- Unduh arsip *installer* dari https://golang.org/dl/, pilih installer untuk Linux yang sesuai dengan jenis bit komputer anda. Proses download bisa dilakukan lewat CLI, menggunakan `wget` atau `curl`.

```
$ wget https://storage.googleapis.com/golang/go1...
```

- Buka terminal, *extract* arsip tersebut ke `/usr/local`.

```
$ tar -C /usr/local -xzf go1...
```

- Tambahkan path binary Go ke `PATH` *environment variable*.

```
$ echo "export PATH=$PATH:/usr/local/go/bin" >> ~/.bashrc

$ source ~/.bashrc
```

- Selanjutnya, eksekusi perintah berikut untuk mengetes apakah Go sudah terinstal dengan benar.

```
go version
```

- Jika output adalah sama dengan versi Go yang ter-*install*, menandakan proses instalasi berhasil.

Instalasi Golang (Linux)

- Unduh arsip *installer* dari https://golang.org/dl/, pilih installer untuk Linux yang sesuai dengan jenis bit komputer anda. Proses download bisa dilakukan lewat CLI, menggunakan `wget` atau `curl`.

```
$ wget https://storage.googleapis.com/golang/go1...
```

- Buka terminal, *extract* arsip tersebut ke `/usr/local`.

```
$ tar -C /usr/local -xzf go1...
```

- Tambahkan path binary Go ke `PATH` *environment variable*.

```
$ echo "export PATH=$PATH:/usr/local/go/bin" >> ~/.bashrc

$ source ~/.bashrc
```

- Selanjutnya, eksekusi perintah berikut untuk mengetes apakah Go sudah terinstal dengan benar.

```
go version
```

- Jika output adalah sama dengan versi Go yang ter-*install*, menandakan proses instalasi berhasil.

Bisa.AI
AI FOR EVERYONE

Basic Programming

Basic Programming

- Setiap program Go terbuat dari paket-paket (package).
- Program mulai berjalan dalam paket main.

main.go

```go
package main

import (
  "fmt"
  "math/rand"
)

func main() {
  fmt.Println("Bilangan kesukaan saya adalah", rand.Intn(10))
}
```

Basic Programming

- Multiple Return Value

main.go

```go
package main

import "fmt"

func swap(x, y string) (string, string) {
  return y, x
}

func main() {
  a, b := swap("hello", "world")
  fmt.Println(a, b)
}
```

Basic Programming

- Go tidak memiliki class. Namun, anda bisa mendefinisikan method pada tipe.
- Sebuah method adalah sebuah fungsi dengan argumen khusus *receiver*.
- *receiver* muncul pada bagian antara kata kunci `func` and nama method.
- Pada contoh berikut, method `Abs` memiliki *receiver* dengan tipe `Vertex` bernama `v`.

main.go

```go
package main

import (
    "fmt"
    "math"
)

type Vertex struct {
    X, Y float64
}

func (v Vertex) Abs() float64 {
    return math.Sqrt(v.X*v.X + v.Y*v.Y)
}

func main() {
    v := Vertex{3, 4}
    fmt.Println(v.Abs())
}
```

Basic Programming

- Seleksi Kondisi

main.go

```go
package main

import (
  "fmt"
  "math"
)

func sqrt(x float64) string {
  if x < 0 {
    return sqrt(-x) + "i"
  }
  return fmt.Sprint(math.Sqrt(x))
}

func main() {
  fmt.Println(sqrt(2), sqrt(-4))
}
```

Basic Programming

- Seleksi Kondisi

main.go

```go
package main

import (
  "fmt"
  "math"
)

func pow(x, n, lim float64) float64 {
  if v := math.Pow(x, n); v < lim {
    return v
  } else {
    fmt.Printf("%g >= %g\n", v, lim)
  }
  // v tidak dapat digunakan disini
  return lim
}

func main() {
  fmt.Println(
    pow(3, 2, 10),
    pow(3, 3, 20),
  )
}
```

Basic Programming

- Perulangan For

main.go

```go
package main

import "fmt"

func main() {
    sum := 0
    for i := 0; i < 10; i++ {
        sum += i
    }
    fmt.Println(sum)
}
```

Basic Programming

- Perulangan While

main.go

```go
package main

import "fmt"

func main() {
    sum := 1
    for sum < 1000 {
        sum += sum
    }
    fmt.Println(sum)
}
```

Basic Programming

- Pointer

main.go

```go
package main

import "fmt"

func main() {
    i, j := 42, 2701

    p := &i          // menunjuk ke i
    fmt.Println(*p) // baca i lewat pointer
    *p = 21          // set i lewat pointer
    fmt.Println(i)  // lihat nilai terbaru dari i

    p = &j          // p menunjuk ke j
    *p = *p / 37    // bagi nilai j lewat pointer
    fmt.Println(j) // lihat nilai terbaru dari j
}
```

Basic Programming

- Struct

main.go

```go
package main

import "fmt"

type Vertex struct {
  X, Y int
}

var (
  v1 = Vertex{1, 2}  // memiliki tipe Vertex
  v2 = Vertex{X: 1}  // Y:0 adalah implisit
  v3 = Vertex{}      // X:0 dan Y:0
  p  = &Vertex{1, 2} // memiliki tipe *Vertex
)

func main() {
  fmt.Println(v1, p, v2, v3)
}
```

Basic Programming

- Array

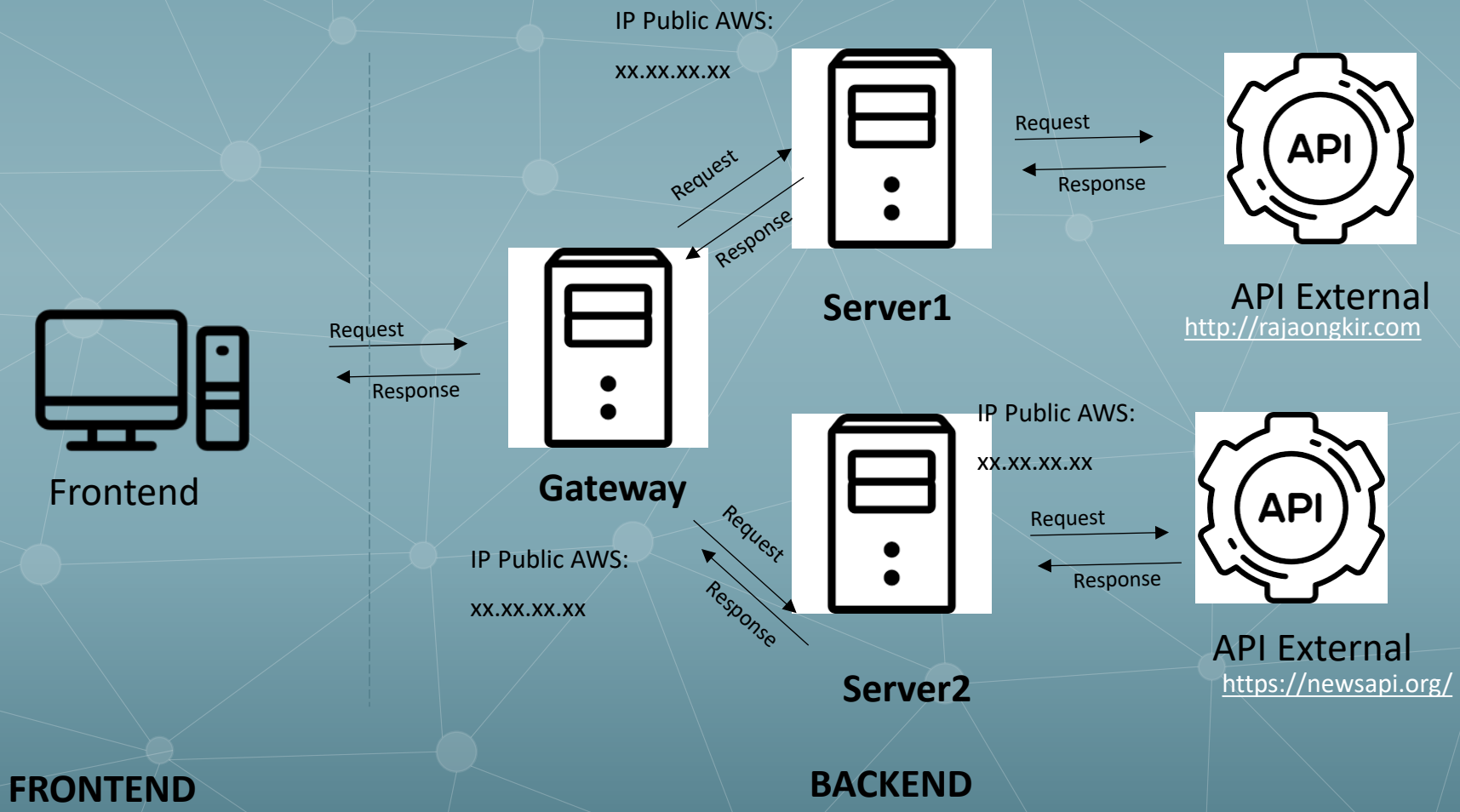
main.go

```go
package main

import "fmt"

func main() {
    var a [2]string
    a[0] = "Hello"
    a[1] = "World"
    fmt.Println(a[0], a[1])
    fmt.Println(a)

    primes := [6]int{2, 3, 5, 7, 11, 13}
    fmt.Println(primes)
}
```

Bisa.AI
AI FOR EVERYONE

Microservice vs Monolitik

Backend vs Frontend



Web Service

- Web Server

main.go

```go
package main

import "fmt"
import "net/http"

func index(w http.ResponseWriter, r *http.Request) {
    fmt.Fprintln(w, "apa kabar!")
}

func main() {
    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        fmt.Fprintln(w, "halo!")
    })

    http.HandleFunc("/index", index)

    fmt.Println("starting web server at http://localhost:8080/")
    http.ListenAndServe(":8080", nil)
}
```

Web Service

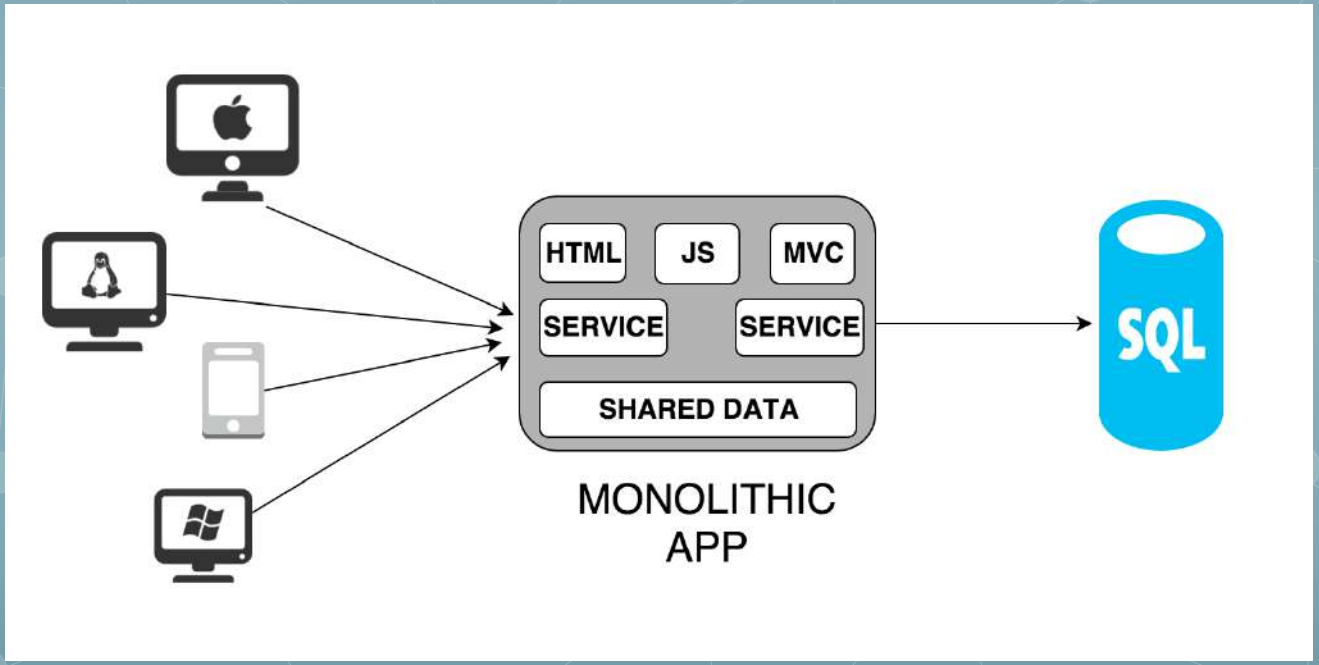
- Parsing URL

main.go

```go
package main

import "fmt"
import "net/url"

func main() {
    var urlString = "[redacted]"
    var u, e = url.Parse(urlString)
    if e != nil {
        fmt.Println(e.Error())
        return
    }

    fmt.Printf("url: %s\n", urlString)

    fmt.Printf("protocol: %s\n", u.Scheme) // http
    fmt.Printf("host: %s\n", u.Host)       // kalipare.com:80
    fmt.Printf("path: %s\n", u.Path)       // /hello

    var name = u.Query()["name"][0] // john wick
    var age = u.Query()["age"][0]   // 27
    fmt.Printf("name: %s, age: %s\n", name, age)
}
```

Monolitik Architecture

- Monolitik



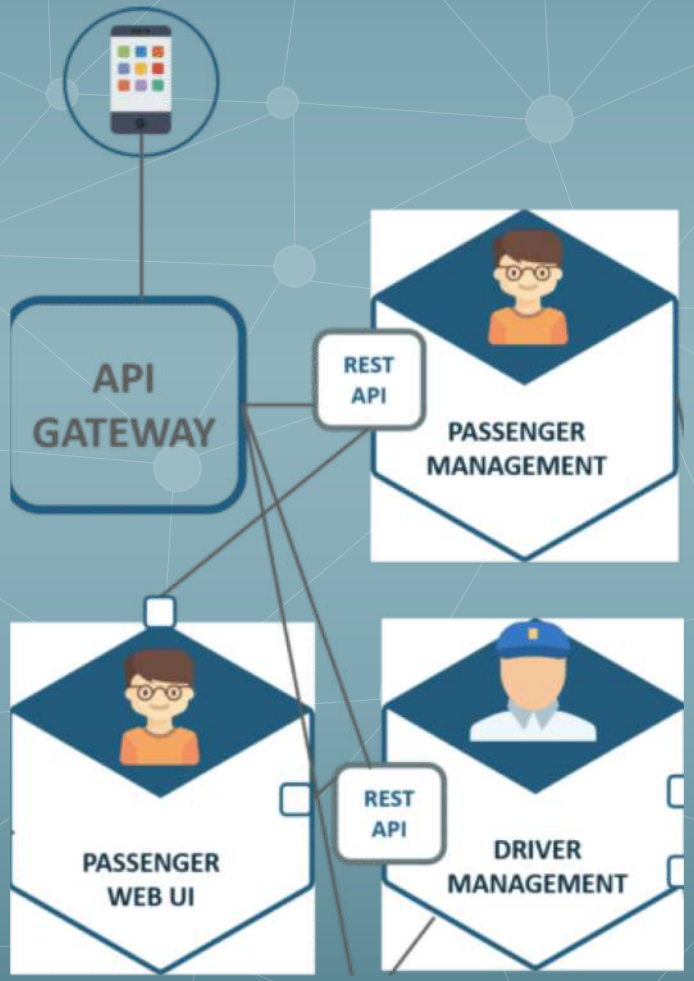
Monolitik

- Aplikasi Enterprise dibangun dalam tiga bagian: database, client-side, dan server-side.
- Dimana Server-side akan menangani request HTTP kemudian menjalankan beberapa logika sesuai dengan domain, kemudian mengambil dan memperbarui data dari database, dan mengurim data tersebut ke sisi client-side.

Monolitik

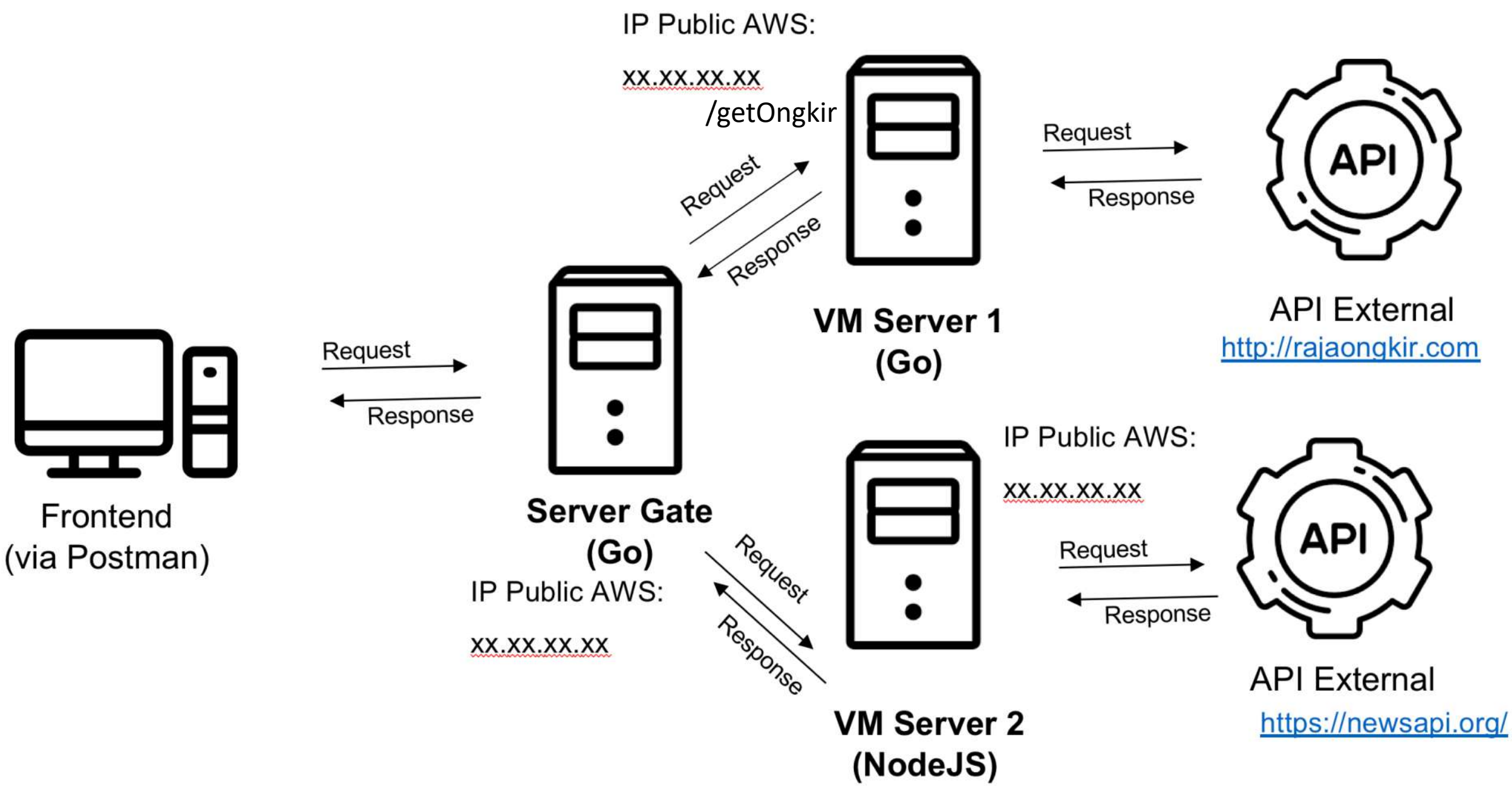
- Ketika aplikasi menjadi besar (banyak yang akses) peforma akan menurun (kecuali punya banyak uang buat bayar server yang lebih bagus)
- Ketika akan merubah teknologi pada aplikasi maka akan merubah secara keseluruhan aplikasi.
- Jika terjadi error pada salah satu fungsi maka akan mempengaruhi keseluruhan aplikasi.

Microservice



Microservice

- Microservices berarti membagi aplikasi menjadi layanan yang lebih kecil dan saling terhubung tidak seperti aplikasi monolitik.
- Setiap microservice merupakan aplikasi kecil yang memiliki arsitektur heksagonal sendiri yang terdiri dari logika beserta berbagai adapternya (bahasa pemrograman, dan lain-lain).
- Jadi intinya microservice yaitu membagi service ke bagian yang lebih kecil dimana service-service tersebut saling berhungan satu sama lain.
- Selain itu, dalam setiap services yang dibuat bisa menggunakan teknologi yang berbebeda.


IP Public AWS:
xx.xx.xx.xx
/getOngkir
Request
Response
Request
Response
API
API External
http://rajaongkir.com
VM Server 1
(Go)
Request
Response
Frontend
(via Postman)
Server Gate
(Go)
IP Public AWS:
xx.xx.xx.xx
Request
Response
IP Public AWS:
xx.xx.xx.xx
Request
Response
API
API External
https://newsapi.org/
VM Server 2
(NodeJS)

Web Service

- Connect to MySQL

```go
package main

import (
    "database/sql"
    "fmt"
    "log"
    _ "github.com/go-sql-driver/mysql"
)

type Products struct {
    Sku          string
    Product_name string
    Stocks       int
}
```

```go
func main(){
    var products Products
    var arr_products []Products
    db, err := sql.Open("mysql","root:@tcp(127.0.0.1:3306)/go_coba")
    defer db.Close()

    if(err != nil) {
        log.Fatal(err)
    }
    rows, err := db.Query("Select sku,product_name,stocks from products ORDER BY sku DESC")
    if err!= nil {
        log.Print(err)
    }
    count:=0
    for rows.Next(){
        if err := rows.Scan(&products.Sku, &products.Product_name, &products.Stocks); err != nil {
            log.Fatal(err.Error())

        }else{
            arr_products = append(arr_products, products)
            fmt.Println(arr_products[count])
        }
        count++
    }
}
```

Web Service

- Setup REST API

```go
package main

import (
    "database/sql"
    "log"
    "encoding/json"
    _ "github.com/go-sql-driver/mysql"
    "net/http"
    "github.com/gorilla/mux"
)

type Products struct {
    Sku          string
    Product_name string
    Stocks       int
}
```

```go
type ResponseProduct struct {
    Status int
    Message string
    Data   []Products

}
```

```go
func main(){
    router := mux.NewRouter()
    router.HandleFunc("/getproducts",returnAllProducts).Methods("GET")
    http.Handle("/",router)
    log.Fatal(http.ListenAndServe(":12345",router))

}
```

```go
func returnAllProducts(w http.ResponseWriter, r *http.Request){
    var products Products // variable untuk memetakan data product yang terbagi menjadi 3 field
    var arr_products []Products // menampung variable products ke dalam bentuk slice
    var responseProd ResponseProduct //variable untuk menampung data arr_products yang nantinya aka
    db, err := sql.Open("mysql","root:@tcp(127.0.0.1:3306)/go_coba")
    defer db.Close()

    if(err != nil) {
        log.Fatal(err)
    }

    rows, err := db.Query("Select sku,product_name,stocks from products ORDER BY sku DESC")
    if err!= nil {
        log.Print(err)
    }
    //bentuk perulangan untuk me render data dari mySQL ke struct dan slice data products
    for rows.Next(){
        if err := rows.Scan(&products.Sku, &products.Product_name, &products.Stocks); err != nil {
            log.Fatal(err.Error())

        }else{
            arr_products = append(arr_products, products)
        }
    }

    responseProd.Status = 1 //mengisi valus status = 1 , dengan asumsi pasti success
    responseProd.Message = "Success"
    responseProd.Data = arr_products // mengisi komponen Data dengan data slice arr_products

    //mengubah data sstruct menjadi JSON
    json.NewEncoder(w).Encode(responseProd)
```

Web Service

- Setup REST API


```
{"status":1,"message":"Success","Data":[{"sku":"SSI-D01466064-X3-BLA","product_name":"Salyara Plain Casual Big Blouse (XXXL,Black)","stocks":52},{"sku":"SSI-D01466013-XX-
BLA","product_name":"Salyara Plain Casual Big Blouse (XXL,Black)","stocks":77},{"sku":"SSI-D01401071-LL-RED","product_name":"Zeomila Zipper Casual Blouse
(L,Red)","stocks":76},{"sku":"SSI-D01401064-XL-RED","product_name":"Zeomila Zipper Casual Blouse (XL,Red)","stocks":44},{"sku":"SSI-D01401050-MM-
RED","product_name":"Zeomila Zipper Casual Blouse (M,Red)","stocks":73},{"sku":"SSI-D01326299-LL-NAV","product_name":"Siunfhi Ethnic Trump Blouse (L,Navy)","stocks":127},
{"sku":"SSI-D01326286-LL-KHA","product_name":"Siunfhi Ethnic Trump Blouse (L,Khaki)","stocks":210},{"sku":"SSI-D01326223-MM-KHA","product_name":"Siunfhi Ethnic Trump
Blouse (M,Khaki)","stocks":209},{"sku":"SSI-D01326205-MM-NAV","product_name":"Siunfhi Ethnic Trump Blouse (M,Navy)","stocks":143},{"sku":"SSI-D01326201-XL-
KHA","product_name":"Siunfhi Ethnic Trump Blouse (XL,Khaki)","stocks":186},{"sku":"SSI-D01322275-XL-WHI","product_name":"Thafqya Plain Raglan Blouse
(XL,White)","stocks":116},{"sku":"SSI-D01322234-LL-WHI","product_name":"Thafqya Plain Raglan Blouse (L,White)","stocks":105},{"sku":"SSI-D01220388-MM-
SAL","product_name":"Devibav Plain Trump Blouse (M,Salem)","stocks":216},{"sku":"SSI-D01220357-SS-YEL","product_name":"Devibav Plain Trump Blouse
(S,Yellow)","stocks":74},{"sku":"SSI-D01220355-XX-YEL","product_name":"Devibav Plain Trump Blouse (XXL,Yellow)","stocks":140},{"sku":"SSI-D01220349-LL-
YEL","product_name":"Devibav Plain Trump Blouse (L,Yellow)","stocks":101},{"sku":"SSI-D01220346-LL-SAL","product_name":"Devibav Plain Trump Blouse
(L,Salem)","stocks":151},{"sku":"SSI-D01220338-XX-SAL","product_name":"Devibav Plain Trump Blouse (XXL,Salem)","stocks":65},{"sku":"SSI-D01220334-XL-
YEL","product_name":"Devibav Plain Trump Blouse (XL,Yellow)","stocks":110},{"sku":"SSI-D01220322-MM-YEL","product_name":"Devibav Plain Trump Blouse
(M,Yellow)","stocks":121},{"sku":"SSI-D01220307-XL-SAL","product_name":"Devibav Plain Trump Blouse (XL,Salem)","stocks":182},{"sku":"SSI-D01037822-XX-
BLA","product_name":"Dellaya Plain Loose Big Blouse (XXL,Black)","stocks":8},{"sku":"SSI-D01037812-X3-BLA","product_name":"Dellaya Plain Loose Big Blouse
(XXXL,Black)","stocks":54},{"sku":"SSI-D01037807-X3-BWH","product_name":"Dellaya Plain Loose Big Blouse (XXXL,Broken White)","stocks":74},{"sku":"SSI-D00864661-MM-
NAV","product_name":"Deklia Plain Casual Blouse (M,Navy)","stocks":13},{"sku":"SSI-D00864652-SS-NAV","product_name":"Deklia Plain Casual Blouse (S,Navy)","stocks":2},
{"sku":"SSI-D00864614-XL-NAV","product_name":"Deklia Plain Casual Blouse (XL,Navy)","stocks":97},{"sku":"SSI-D00864612-LL-NAV","product_name":"Deklia Plain Casual Blouse
(L,Navy)","stocks":8},{"sku":"SSI-D00791091-XL-BWH","product_name":"Zalekia Plain Casual Blouse (XL,Broken White)","stocks":137},{"sku":"SSI-D00791077-MM-
BWH","product_name":"Zalekia Plain Casual Blouse (M,Broken White)","stocks":138},{"sku":"SSI-D00791015-LL-BWH","product_name":"Zalekia Plain Casual Blouse (L,Broken
White)","stocks":154},{"sku":"ffffff-ccc-ikik","product_name":"Zalekia Plain Casual Jeans (L,Broken White)","stocks":35}]}
```

Web Service

- Get Service

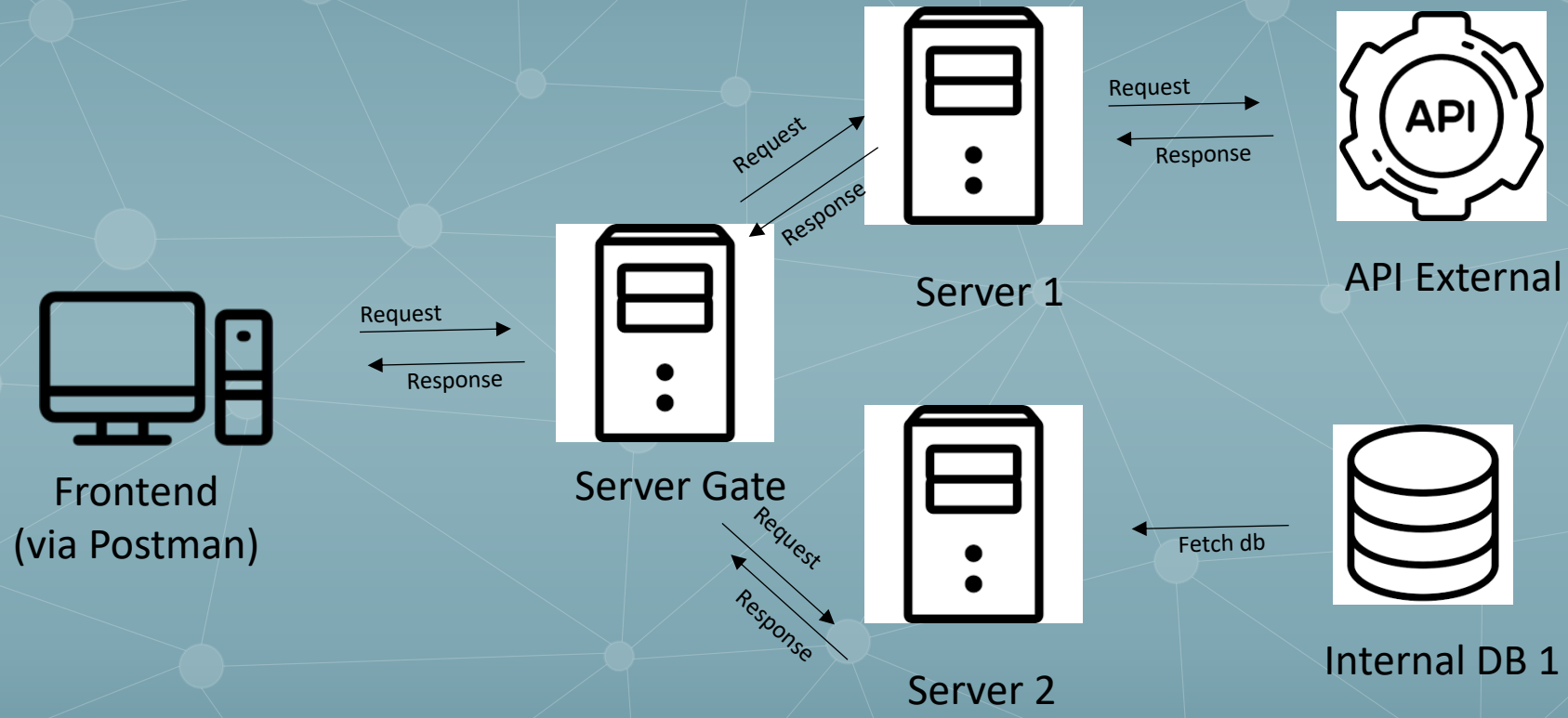
```go
package main

import (
    "encoding/json"
    "fmt"
    "io/ioutil"
    "log"
    "net/http"
    "time"
)

type Products struct {
    Status int
    Message string
    Data   []Products
}
```

```go
func main() {

    url := "http://localhost:12345/getproducts"

    spaceClient := http.Client{
        Timeout: time.Second * 2, // Timeout after 2 seconds
    }

    req, err := http.NewRequest(http.MethodGet, url, nil)
    if err != nil {
        log.Fatal(err)
    }

    req.Header.Set("User-Agent", "spacecount-tutorial")

    res, getErr := spaceClient.Do(req)
    if getErr != nil {
        log.Fatal(getErr)
    }

    if res.Body != nil {
        defer res.Body.Close()
    }
```

```go
    body, readErr := ioutil.ReadAll(res.Body)
    if readErr != nil {
        log.Fatal(readErr)
    }

    prod1 := Products{}
    jsonErr := json.Unmarshal(body, &prod1)
    if jsonErr != nil {
        log.Fatal(jsonErr)
    }

    fmt.Println(prod1.Status)
    fmt.Println(prod1.Message)
    fmt.Println(prod1.Data)
}
```

Web Service

- Get Service



REFERENSI

- Digital Talent Scholarsip Kementrian Kominfo