

LAPORAN TUGAS BESAR V

IF3038 Pemrograman Internet

Disusun oleh:
Kelompok Adwadwadwadw

Irfan Kamil / 13510001
Johaner Ridho T.P / 13510103
Ahmad Faiz N / 18210023



Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung
Jl. Ganesha 10, Bandung 40132

Alamat appfog : core.hp.af.cm/api.php

Pada pengerjaan Tugas Besar V dibuat sebuah *protocol* yang digunakan untuk komunikasi antara *client* dan *server*. Berikut ini merupakan daftar *method* yang terdapat pada *protocol* yang kami buat :

- *getpublickey* : Method ini digunakan untuk mendapatkan public key dari server dengan menggunakan enkripsi RSA.
- *login* : Method ini digunakan client untuk login ke server
- *gettask* : Method ini digunakan untuk mendapatkan daftar *task* yang dimiliki oleh user tertentu
- *negatetask* : Method ini digunakan untuk menegasikan status *done* yang dimiliki oleh sebuah *task*.

PUBLIC KEY

Format *message client*

```
{ "method": "getpublickey" }  
###
```

Format response server

```
{ "publickey": "hasilpublickey" }  
###
```

Hasil public key adalah sebuah objek *PublicKey* di Java yang dikonversikan menjadi string.

LOGIN

Format message client

```
{ "method": "login", "username": <suatuusername>, "password": <suatupassword> }  
###
```

Format response server

```
{ "success": true/false, "user_id" = <useridlogin> }  
###
```

suatuusername adalah username dari user yang telah dienkrpsi dengan menggunakan public key RSA dan dikonversi menjadi string. *suatupassword* adalah password dari user yang telah dienkrpsi dengan menggunakan public key RSA dan dikonversi menjadi string. *useridlogin* adalah user id dari user yang telah login melalui client. User ini muncul jika status success adalah true.

GETTASK

Format message client

```
{ "method": "gettask", "timestamp": <waktusaatini>, "token": <token>, "userid": <useridlogin> }  
###
```

Format response server

Jika success true

```
{ "success": true, "data": <semuatask> }  
###
```

Jika success false

```
{ "success": false, "error": "bad authorization" }  
###
```

waktusaatini adalah waktu ketika request dibuat oleh client. Waktu dibuat dalam format epoch time ms. token adalah string hasil hash MD5 dari userid yang digabungkan dengan timestamp. Useridlogin adalah user id dari user yang dienkripsi dengan menggunakan public key RSA.

bad authorization terjadi jika hasil MD5 token tidak sesuai dengan MD5 yang seharusnya atau timestamp terlambat lebih dari 5 detik . semuatask adalah daftar task yang dimiliki user yang sedang login dengan format setiap task adalah sebagai berikut :

```
"data": [  
  {  
    "tags": [  
      { "name": "<nametag1>" },  
      { "name": "<nametag2>" },  
      { "... " }  
    ],  
    "last_update": "<waktu terakhir done yang dirubah>",  
    "category": "<nama kategori >",  
    "name": "<nama task>",  
    "done": "<status done ( 0 jika belum / 1 jika sudah )>",  
    "task_id": "<id dari task>",  
    "category_id": "<id dari kategori>",  
    "assignee": [  
      { "name": "<nama dari assignee task>" },  
      { "name": "<nama dari assignee task>" },  
      { "... " }  
    ],  
    "user_id": "<id dari user>",  
    "attachment": [  
      {  
        "attachment_id": "<id dari attachment>",  
        "task_id": "<id dari task yang memiliki attachment>",  
        "data": "<data dari attachment>"  
      }  
    ]  
  }  
]
```

NEGATETASK

Format message client

```
{ "method": "negatetask", "timestamp": <waktusaatini>, "token": <token>, "userid": <useridlogin>, "task_id": <idtask> }
###
```

Format response server

Jika success true

```
{ "success": true }
###
```

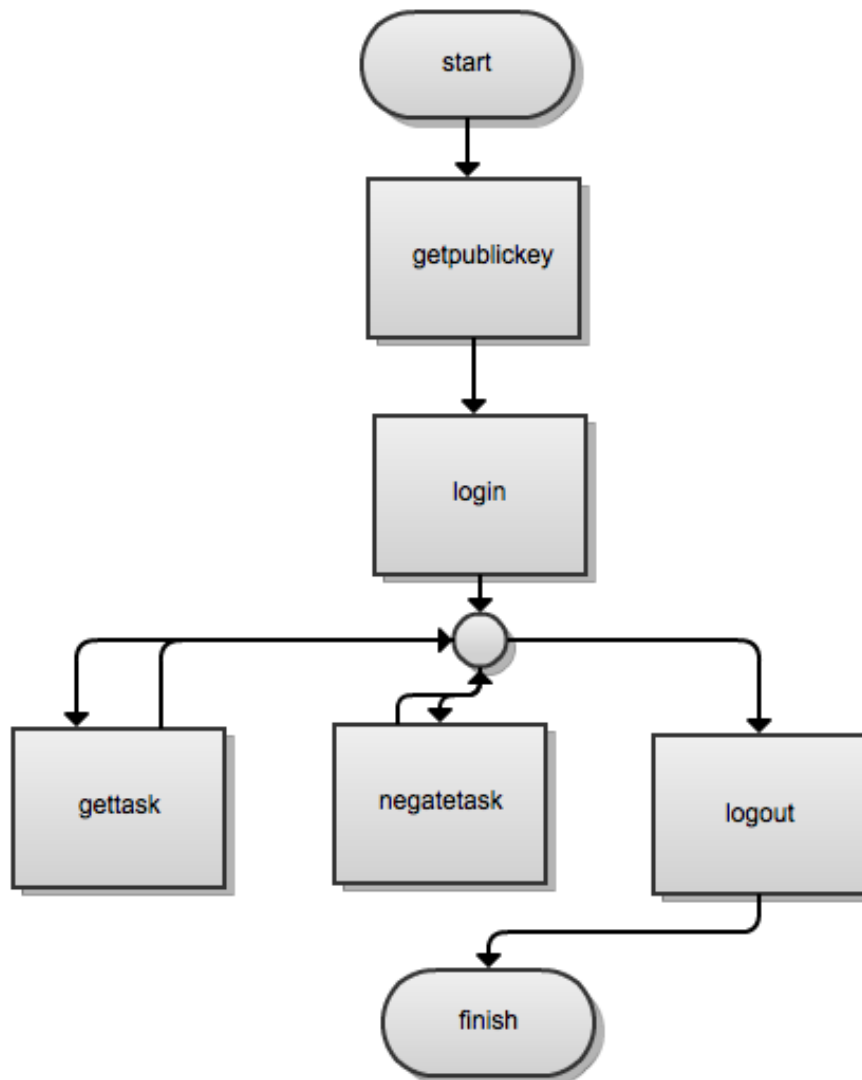
Jika success false

```
{ "success": false, "error": "bad authorization" }
###
```

waktusaatini adalah waktu ketika request dibuat oleh client. Waktu dibuat dalam format epoch time ms. token adalah string hasil hash MD5 dari userid yang digabungkan dengan timestamp. Useridlogin adalah user id dari user yang dienkrpsi dengan menggunakan public key RSA. Idtask adalah id task yang hendak dirubah status *done* yang dimiliki.

bad authorization terjadi jika hasil MD5 token tidak sesuai dengan MD5 yang seharusnya atau timestamp terlambat lebih dari 5 detik.

Berikut ini merupakan *flowchart* dari protocol yang digunakan



Gambar 1 Flowchart dari protocol yang digunakan