

14 Maret 2019

TUGAS 4
DESAIN & ANALISIS ALGORITMA
PENERAPAN ALGORITMA NON-REKURSIF



PARALEL : B

Sunu Ilham Pradika

17081010045

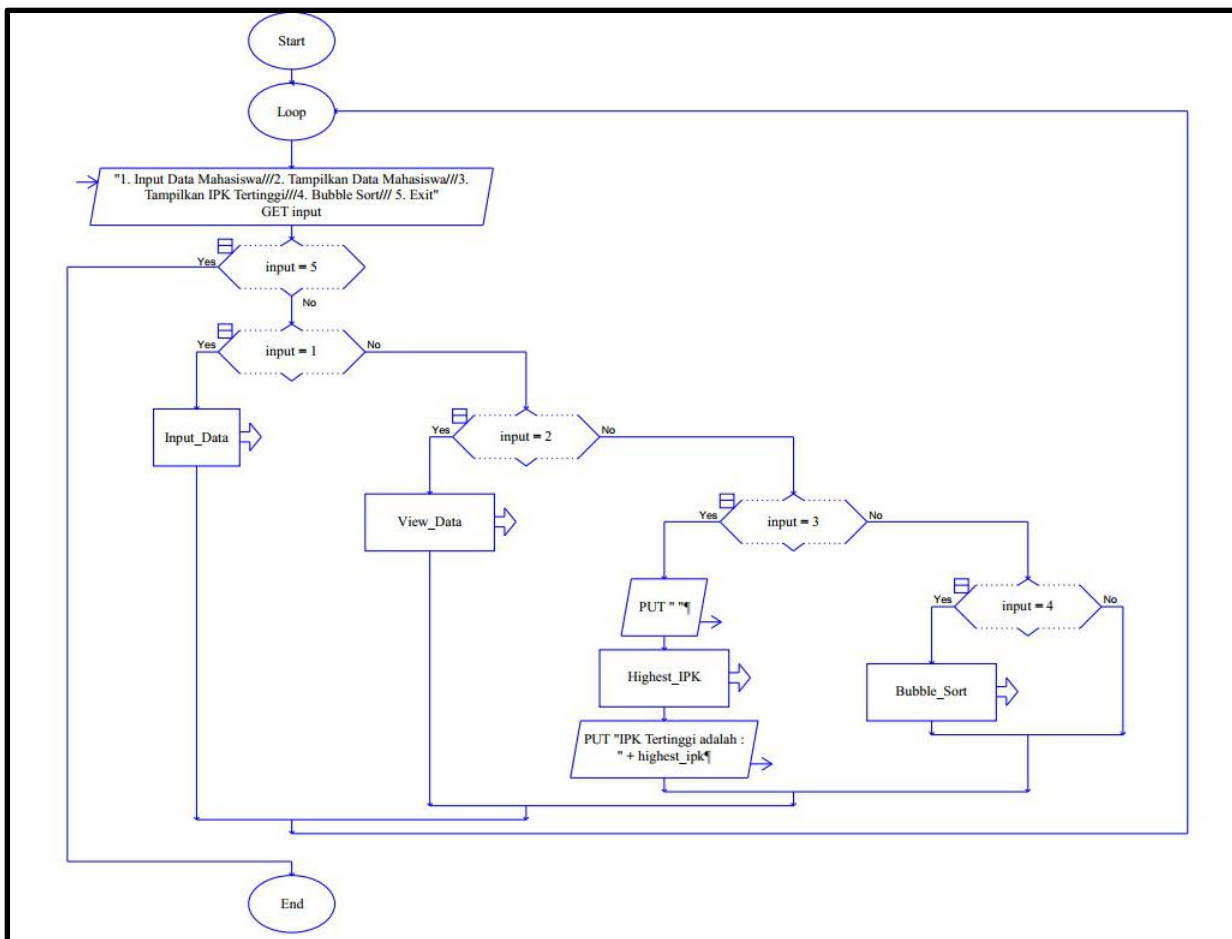
DOSEN PENGAMPU :

Pratama W A, S.Kom, M.Kom

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
JAWA TIMUR

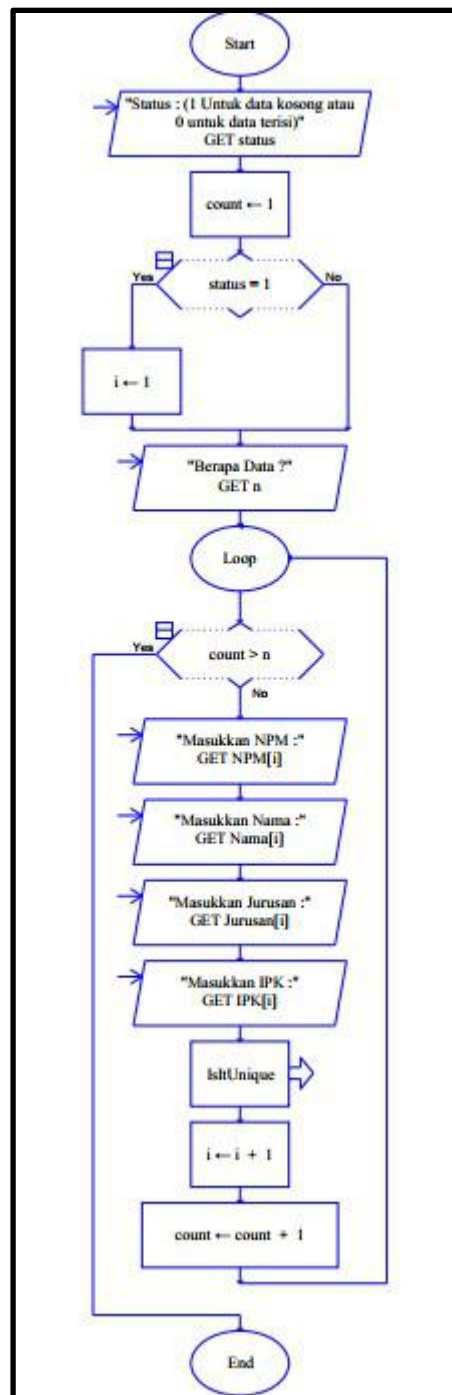
2019

PROGRAM UTAMA



Gambar 1 - Flowchart Program Utama.

FUNGSI TAMBAH DATA



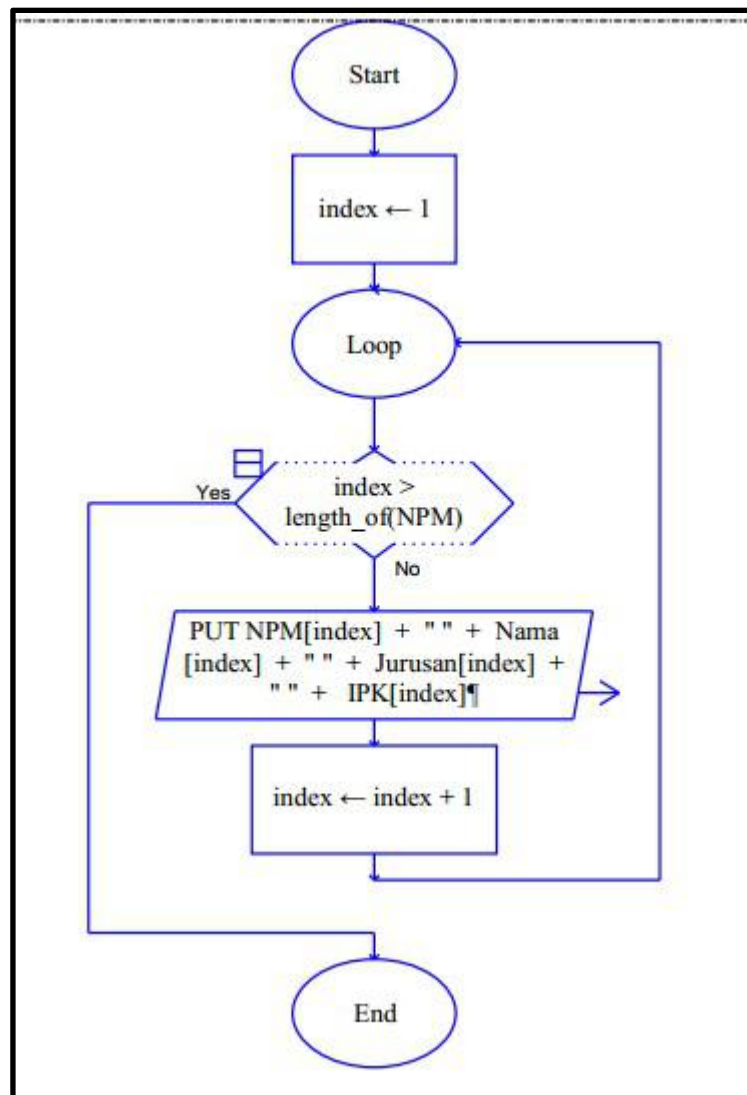
Gambar 2 - Flowchart Fungsi Tambah Data.

PENJELASAN/TRACING FUNGSI TAMBAH DATA

Algoritma :

1. Menginput status dari *array* yang akan diproses, yaitu agar saat dilakukan pemilihan menu tambah data sebanyak dua kali data *array* tidak kembali *reset* ke awal dan hanya pemilihan menu tambah data kedua yang nantinya terekam ke *array*.
2. Menginialisasi nilai dari count sebesar satu, nantinya akan menjadi *counter* agar *loop* berhenti.
3. Pengecekan status apakah *array* kosong atau sudah terisi. Jika kosong, maka akan mengeset nilai *i* sebesar satu. Jika tidak kosong maka akan langsung diminta menginput nilai *n*.
4. Menginput nilai *n* yaitu yang nantinya akan berapa data yang ditambah.
5. Melakukan pengecekan apakah count lebih besar dari *n*. Jika ya maka akan menuju end. Jika tidak maka akan diminta mengisi data NPM, Nama, Jurusan, dan IPK.
6. Setelah selesai menginput data maka akan dilakukan pemanggilan fungsi *IsItUnique* untuk cek data NPM apakah ada kesamaan data dengan yang sebelumnya.
7. Melakukan increment *i* dan count hingga syarat terpenuhi.

FUNGSI LIHAT DATA



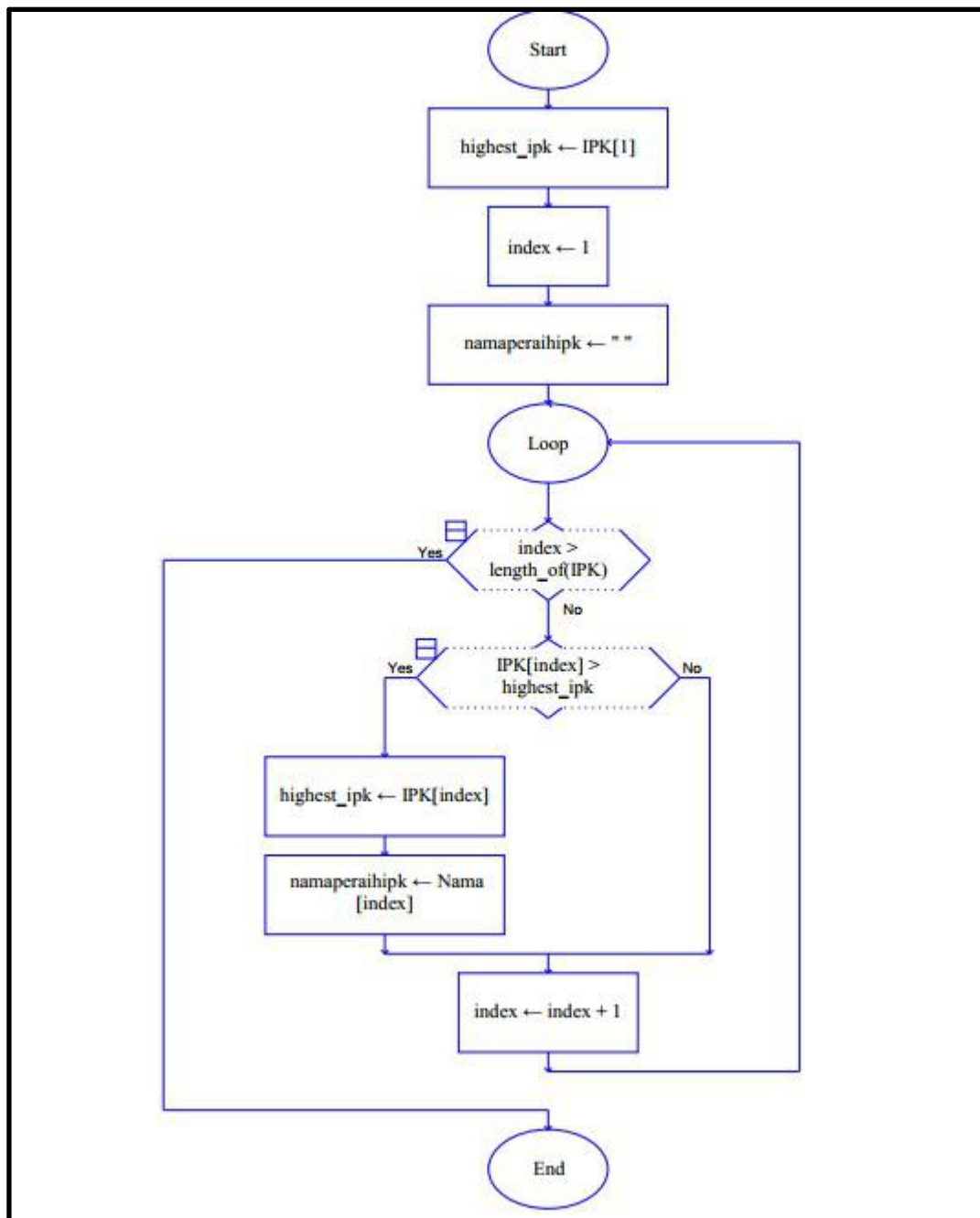
Gambar 3 - Flowchart Fungsi Lihat Data.

PENJELASAN/TRACING FUNGSI LIHAT DATA

Algoritma :

1. Menginialisasi nilai index sebesar satu.
2. Melakukan *looping* dimana di dalamnya dilakukan pengecekan apakah index lebih besar daripada data sepanjang NPM.
3. Mencetak *output* data mahasiswa mulai dari index pertama hingga sebanyak data sepanjang NPM.
4. *Increment* nilai index hingga syarat terpenuhi.

FUNGSI MENCARI IPK TERTINGGI



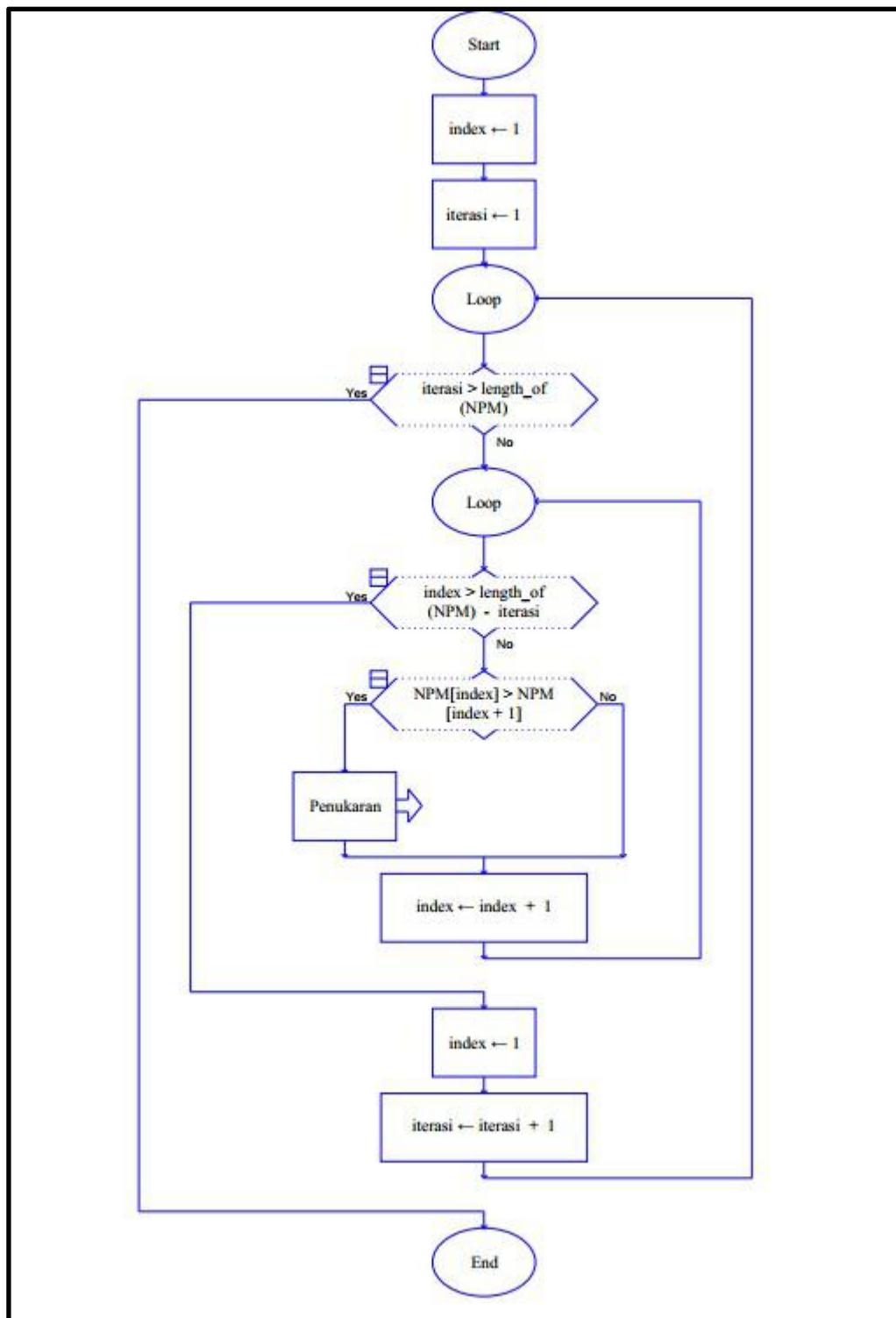
Gambar 4 - Flowchart Mencari IPK Tertinggi.

PENJELASAN/TRACING FUNGSI MENCARI IPK TERTINGGI

Algoritma :

1. Menginialisasi sebuah variabel (`highest_ipk`) untuk menyimpan nilai array NPM index pertama.
2. index diubah menjadi satu.
3. Membuat sebuah variabel untuk menyimpan nama peraih IPK tertinggi nantinya.
4. Melakukan *looping* yang di dalamnya mengecek apakah index sudah melebihi panjang data array IPK.
5. Membandingkan apakah IPK sekarang lebih besar daripada nilai IPK yang sudah disimpan pada variabel `highest_ipk`. Jika ya, maka `highest_ipk` mengambil nilai IPK index sekarang. Jika tidak maka akan dilanjutkan menuju index array selanjutnya.
6. Menyimpan nama peraih IPK tertinggi berdasarkan indexnya.
7. Melakukan *increment* hingga syarat terpenuhi.

FUNGSI BUBBLE SORT



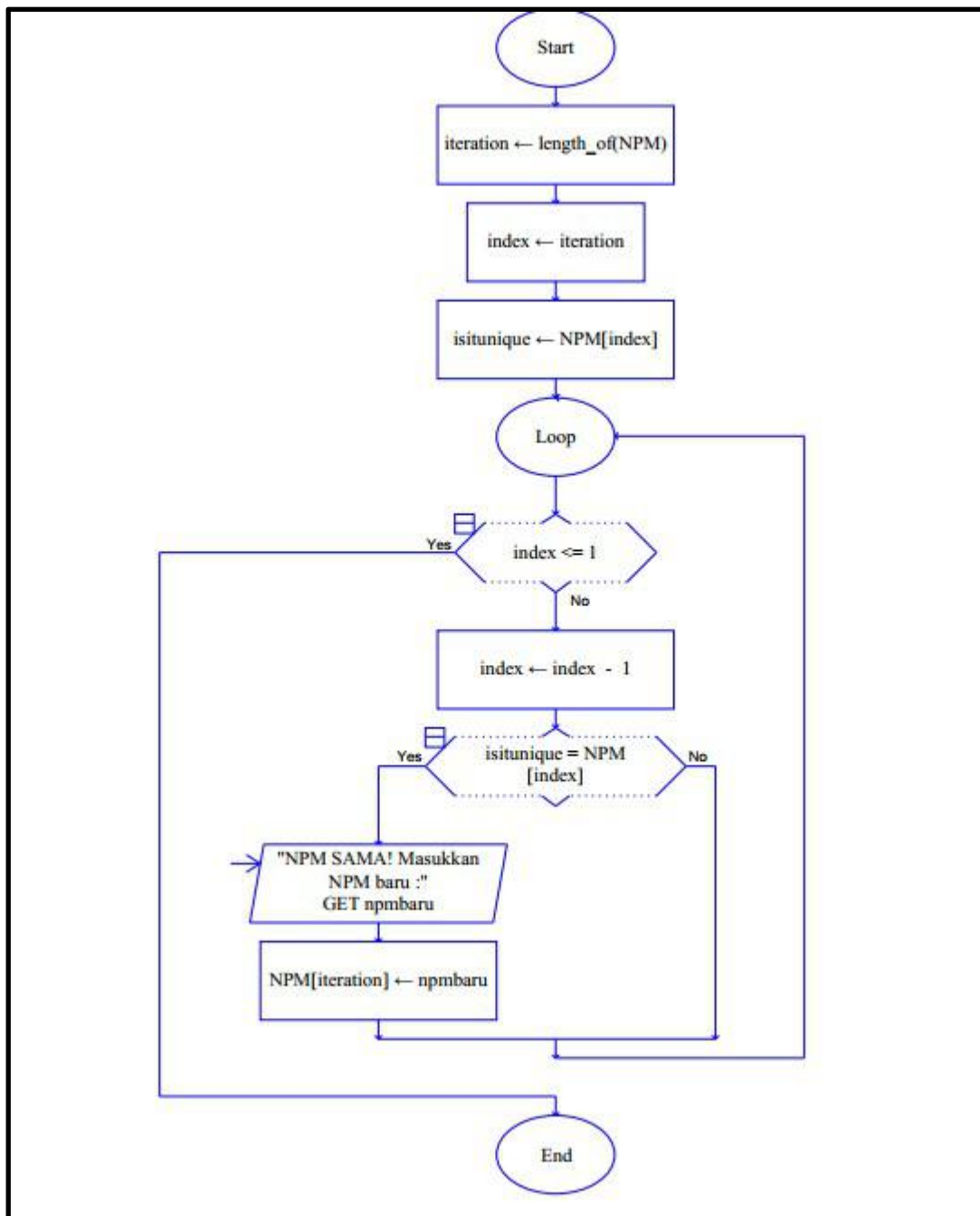
Gambar 5 - Flowchart Fungsi Bubble Sort.

PENJELASAN/TRACING FUNGSI BUBBLE SORT

Algoritma :

1. Menginialisasi variabel index dan iterasi dengan nilai sebesar satu.
2. Melakukan *looping* di dalamnya apakah iterasi lebih besar dibandingkan dengan panjang data NPM. Jika, tidak maka akan melanjutkan proses. Jika tidak maka akan menuju end.
3. Melakukan pengecekan apakah index lebih besar daripada panjang data NPM dikurangi iterasi. Jika ya maka melakukan increment pada iterasi dan memberikan nilai index menjadi 1.
4. Melakukan pengecekan kembali apakah nilai NPM pada index sekarang lebih besar daripada nilai NPM selanjutnya. Jika ya maka akan memanggil fungsi penukaran. Jika tidak maka lanjutkan proses.
5. Melakukan increment index hingga syarat terpenuhi.

FUNGSI MENGECEK KEUNIKAN DATA



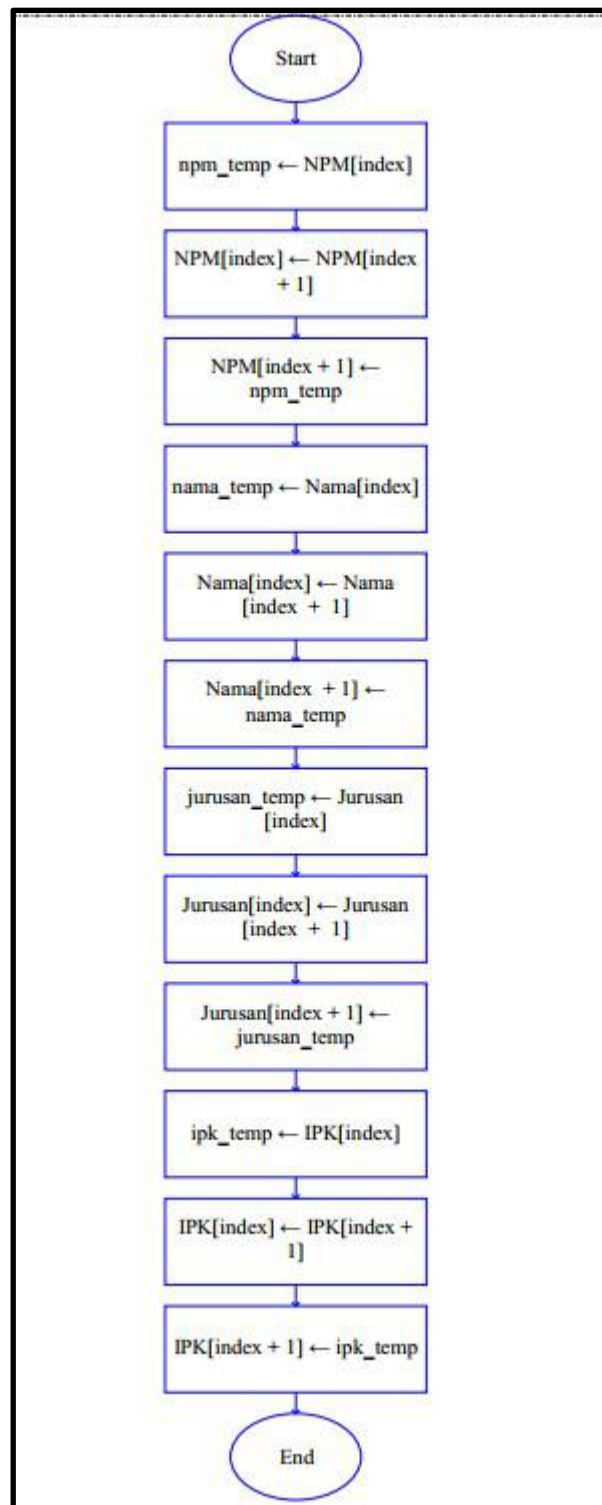
Gambar 6 - Flowchart Fungsi Mengecek Keunikan Data.

PENJELASAN/TRACING FUNGSI MENGECEK KEUNIKAN DATA

Algoritma :

1. Menginialisasi variabel iteration senilai panjang data NPM.
2. index mengambil nilai iteration.
3. Variabel isitunique mengambil nilai NPM pertama.
4. Melakukan looping hingga nilai index kurang dari sama dengan satu.
5. Membandingkan nilai isitunique dengan nilai index array yang sedang diproses apakah sama. Jika ya, maka akan mencetak warning bahwa IPK sama dan meminta user untuk mengganti nilainya.
6. Melakukan input pada variabel baru dan menyimpannya pada index array yang sedang diproses.
7. Proses dilakukan hingga data pertama, sehingga apabila ada data yang sama maka akan diminta langsung menggantinya.

FUNGSI PENUKARAN



Gambar 7 - Flowchart Fungsi Penukaran.

PENJELASAN/TRACING FUNGSI PENUKARAN

Pada fungsi penukaran sebenarnya adalah lanjutan dari fungsi bubble sort agar tidak membuat flowchart terlalu panjang maka dibuat sebuah fungsi baru. Karena hanya lanjutan maka hanya terjadi sebuah pemrosesan sesuai dengan simbolnya yaitu persegi panjang.

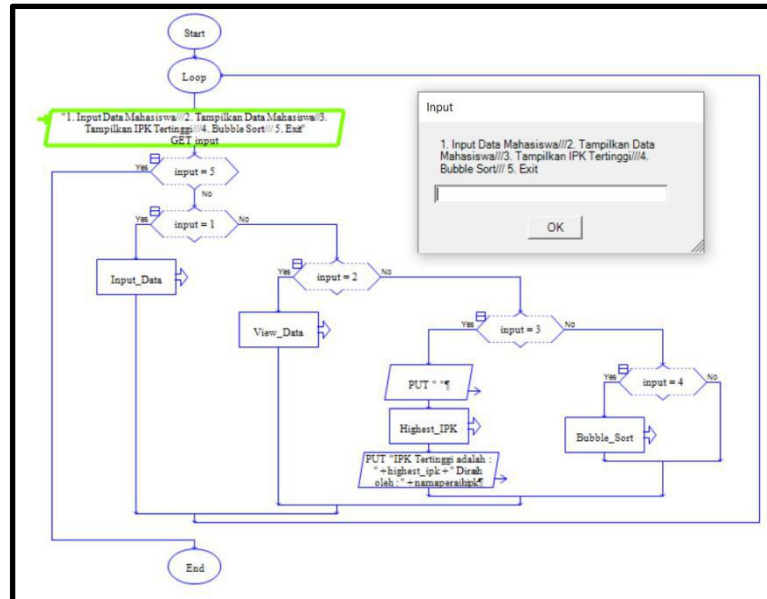
Algoritma :

1. Melakukan penyimpanan pada masing - masing (NPM, Nama, Jurusan, dan IPK) temp index array yang sedang diproses.
2. Melakukan penyimpanan nilai index array yang diproses pada masing - masing (NPM, Nama, Jurusan, dan IPK) menjadi nilai dari index array selanjutnya.
3. Melakukan penyimpanan nilai index array selanjutnya pada masing - masing (NPM, Nama, Jurusan, dan IPK) menjadi nilai temp sebelumnya.

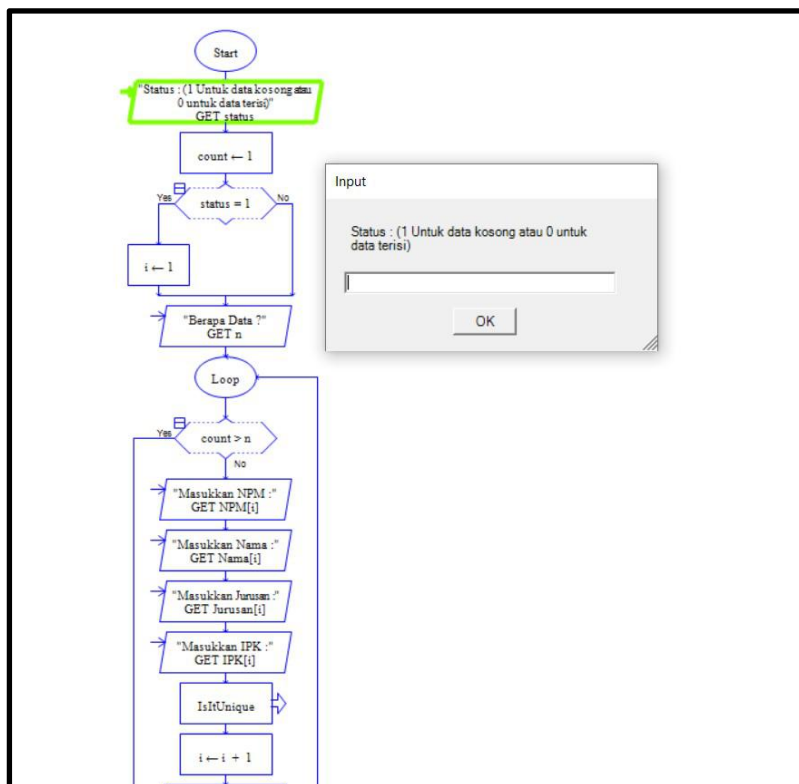
PENJELASAN/TRACING

KETERANGAN :

1. Gambar 8 - 12 menjawab soal mengenai **input data (menambahkan data)**.
2. Gambar 12 - 16 menjawab soal mengenai **unique element (data unik)**.
3. Gambar 17 - 18 menjawab soal mengenai **view data (melihat seluruh data)**.
4. Gambar 19 - 20 menjawab soal mengenai **sorting bubble sort (pengurutan data)**.
5. Gambar 21 - 22 menjawab soal mengenai **max element (nilai tertinggi)**.



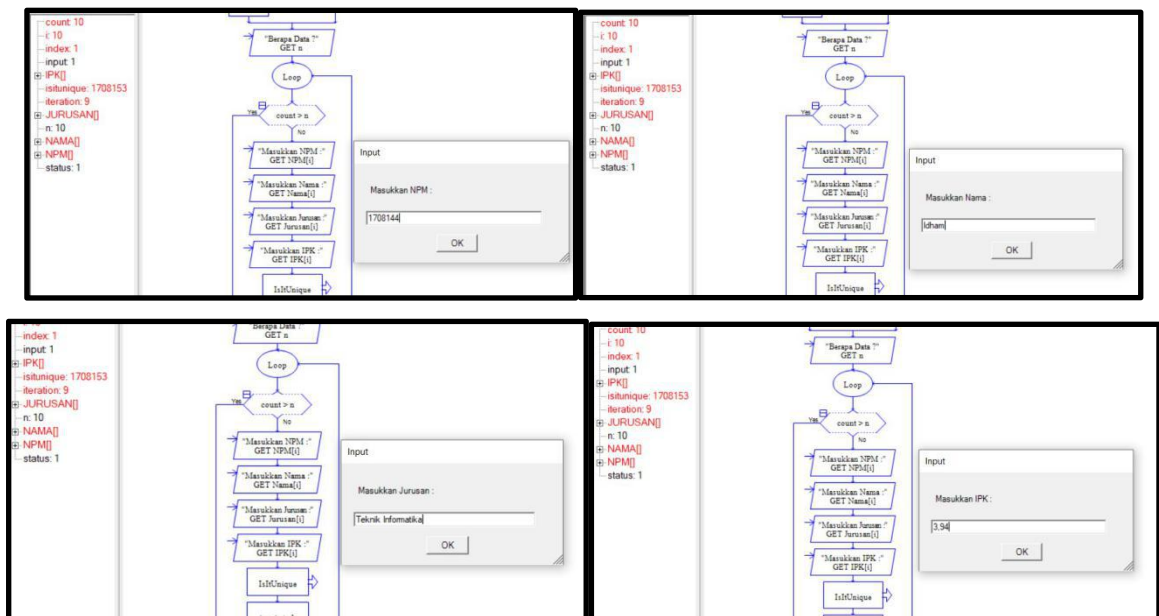
Gambar 8 - Pilihan Menu.



Gambar 9 - Menentukan array kosong atau tidak.



Gambar 10 - Tambah Data.

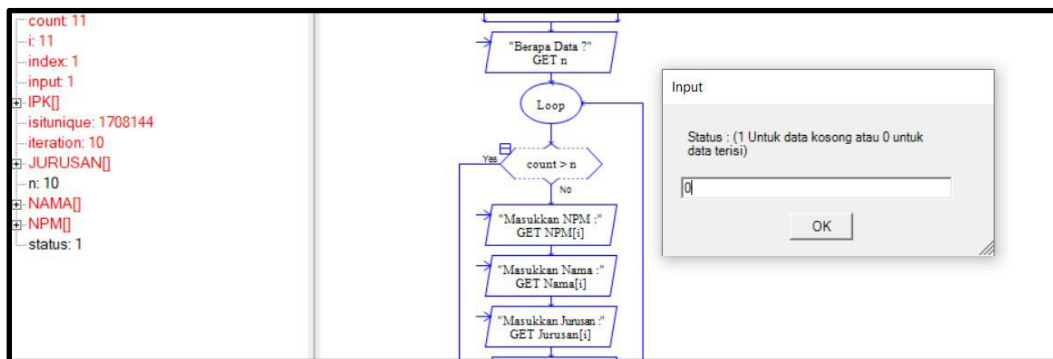


Gambar 11 - Membuat NPM dengan nilai tidak urut untuk bubble sort.

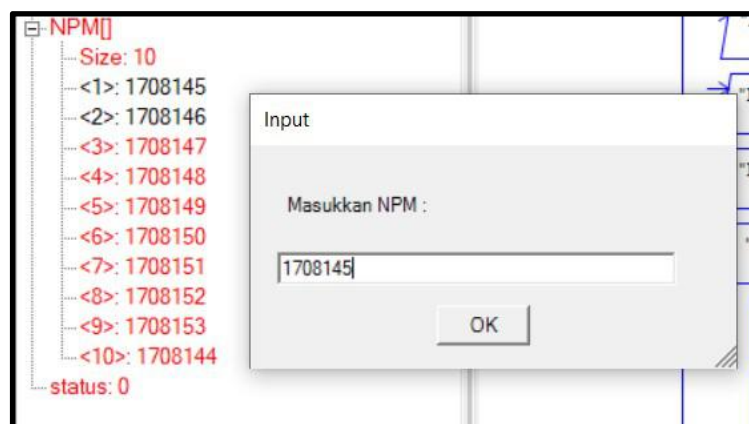
IPK[]	JURUSAN[]
Size: 10	Size: 10
<1>: "4,00"	<1>: "Teknik Informatika"
<2>: "3,96"	<2>: "Teknik Informatika"
<3>: "3,97"	<3>: "Teknik Informatika"
<4>: "3,98"	<4>: "Teknik Informatika"
<5>: "3,99"	<5>: "Teknik Informatika"
<6>: "3,90"	<6>: "Teknik Informatika"
<7>: "3,91"	<7>: "Teknik Informatika"
<8>: "3,92"	<8>: "Teknik Informatika"
<9>: "3,93"	<9>: "Teknik Informatika"
<10>: "3,94"	<10>: "Teknik Informatika"

NAMA[]	NPM[]
Size: 10	Size: 10
<1>: "Sunu"	<1>: 1708145
<2>: "Fitri"	<2>: 1708146
<3>: "Cherry"	<3>: 1708147
<4>: "Fathur"	<4>: 1708148
<5>: "Adi"	<5>: 1708149
<6>: "Eka"	<6>: 1708150
<7>: "Amir"	<7>: 1708151
<8>: "Yunus"	<8>: 1708152
<9>: "Rizky"	<9>: 1708153
<10>: "Idham"	<10>: 1708144

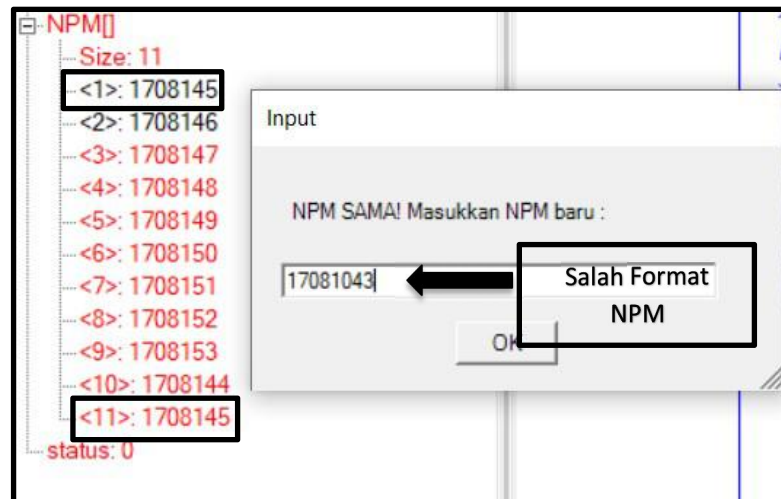
Gambar 12 - Bukti bahwa telah melakukan 10 tambah data.



Gambar 13 - Menambah data baru untuk nanti digunakan sebagai pengecekan keunikan data.



Gambar 14 - Memasukkan NPM yang sama seperti sebelumnya.

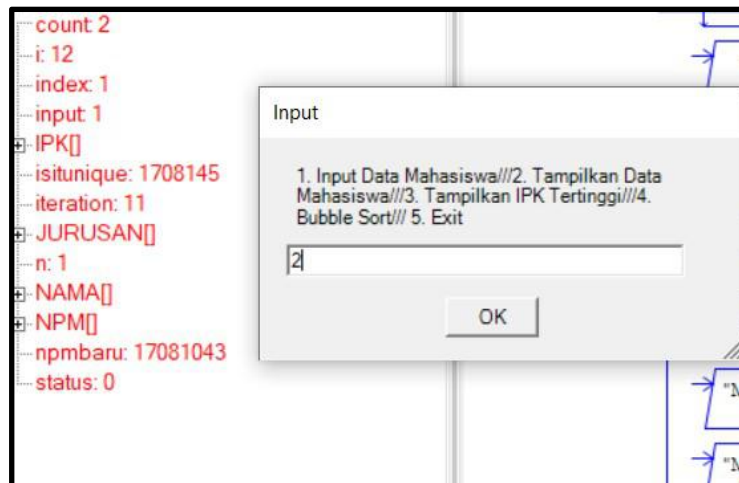


Gambar 15 - Muncul Error diminta memasukkan NPM baru. lihat tabel NPM ada dua NPM sama.

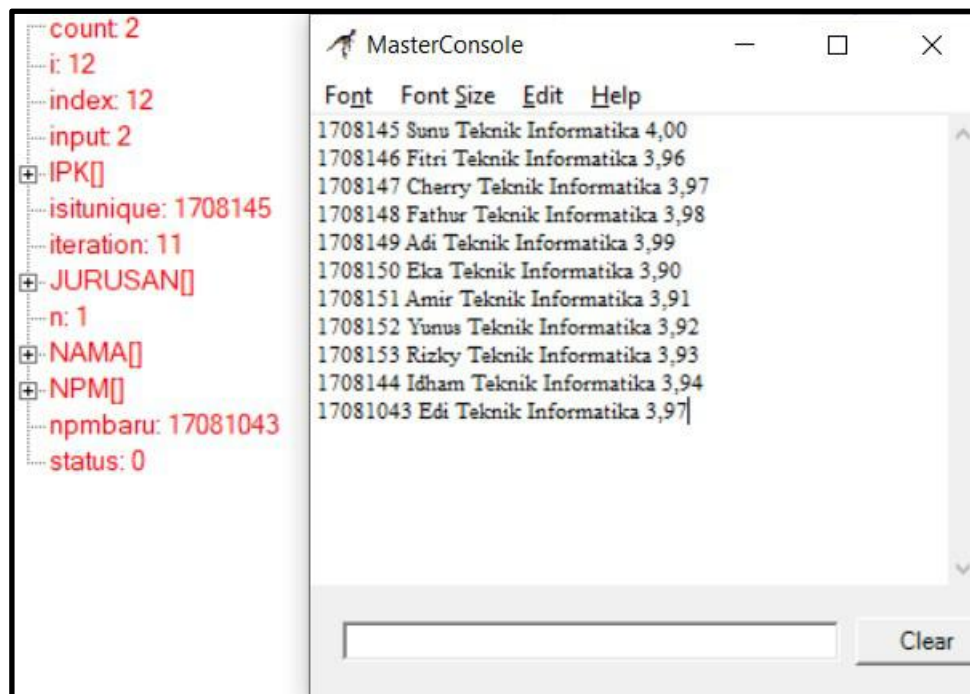
Pada data ini terdapat kesalahan penulisan format NPM sehingga formatnya berbeda dari yang lain.



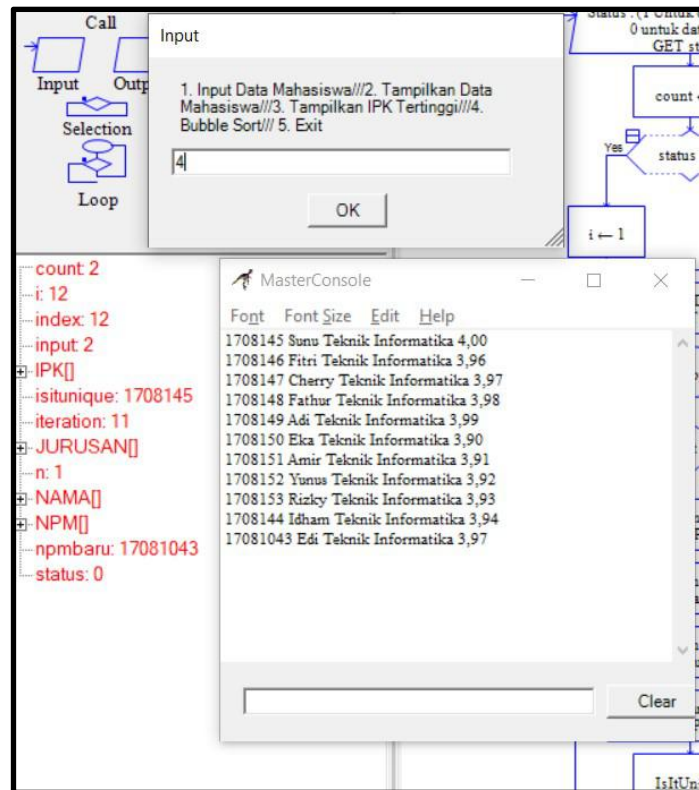
Gambar 16 - Terlihat ada NPM baru yang ditambahkan mengubah NPM yang sama.



Gambar 17 - Memilih menu tampilkan data.



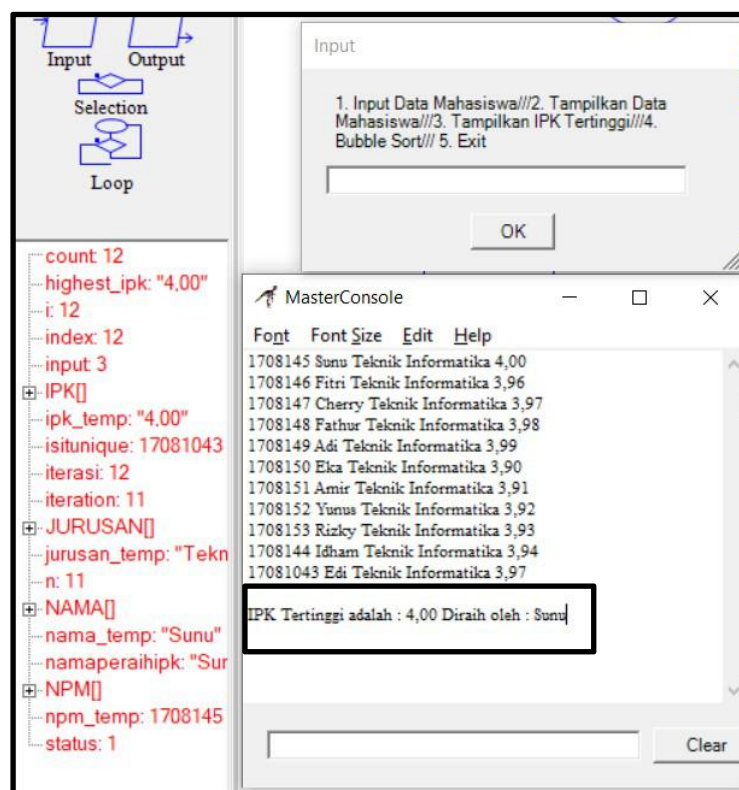
Gambar 18 - Terlihat bahwa data sesuai dengan inputan user.



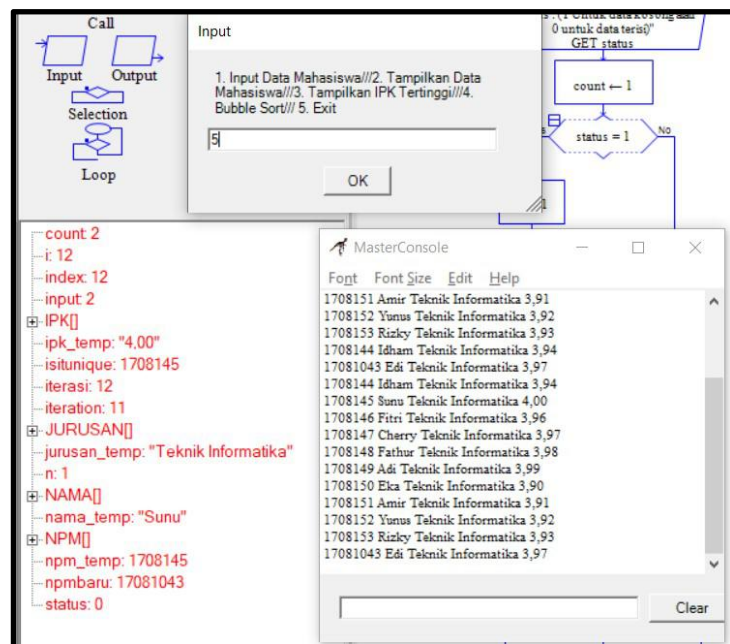
Gambar 19 - Memilih menu Bubble Sort.



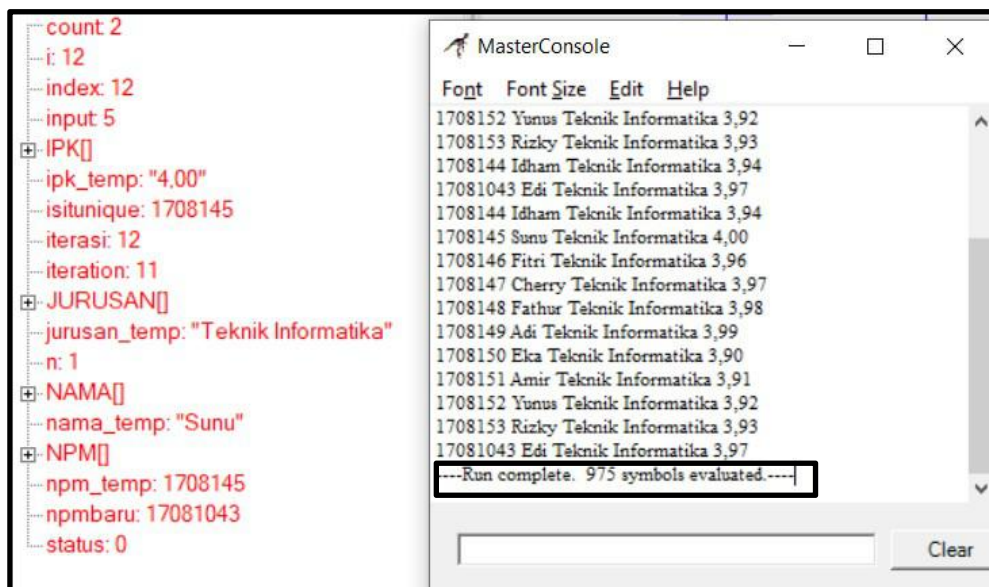
Gambar 20 - Terlihat bahwa data 1708144 Idham berpindah dari posisi bawah menuju atas.



Gambar 21 - Memilih menu tampilkan IPK tertinggi.



Gambar 22 - Memilih menu exit.



Gambar 23 - Berhasil Keluar.