# LAPORAN AKHIR

Mata Praktikum : Organisasi Sistem Komputer

Kelas : 2IA16

Praktikum ke - : 3

Tanggal : Kamis, 28 Oktober 2021

Materi : Addresing

NPM : 50420093

Nama : Ajay Alfredo Almani

Ketua Asisten : Brigitta

Nama Asisten :

Paraf Asisten :

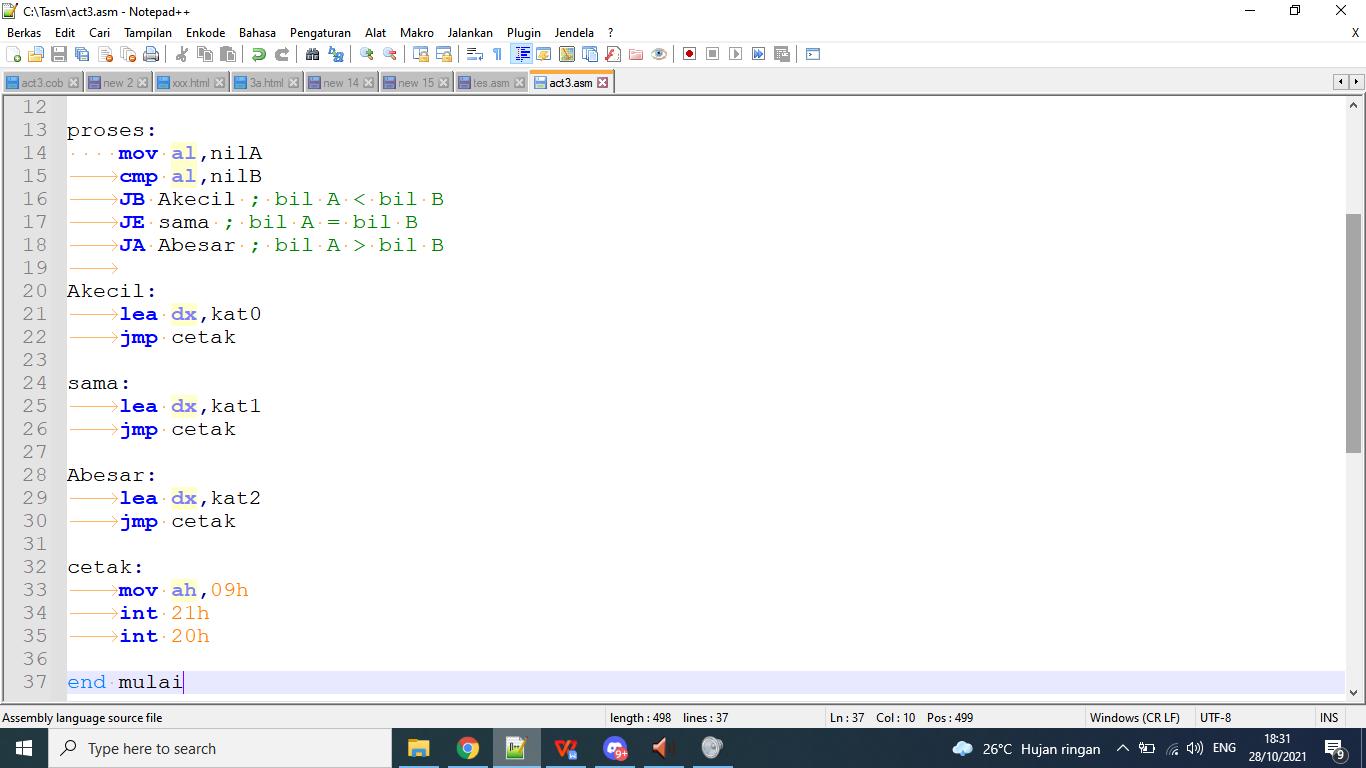
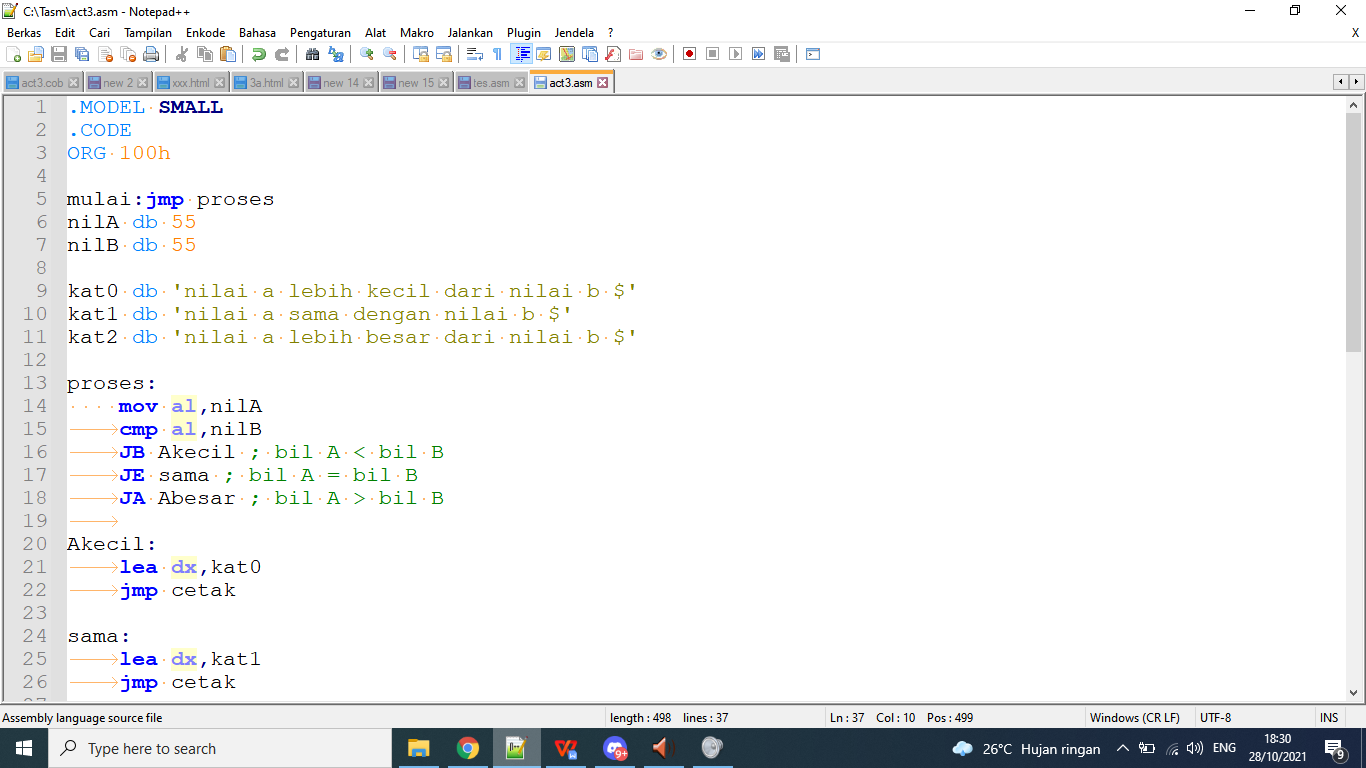
Jumlah Lembar : 5 Lembar



# LABORATORIUM TEKNIK INFORMATIKA UNIVERSITAS GUNADARMA

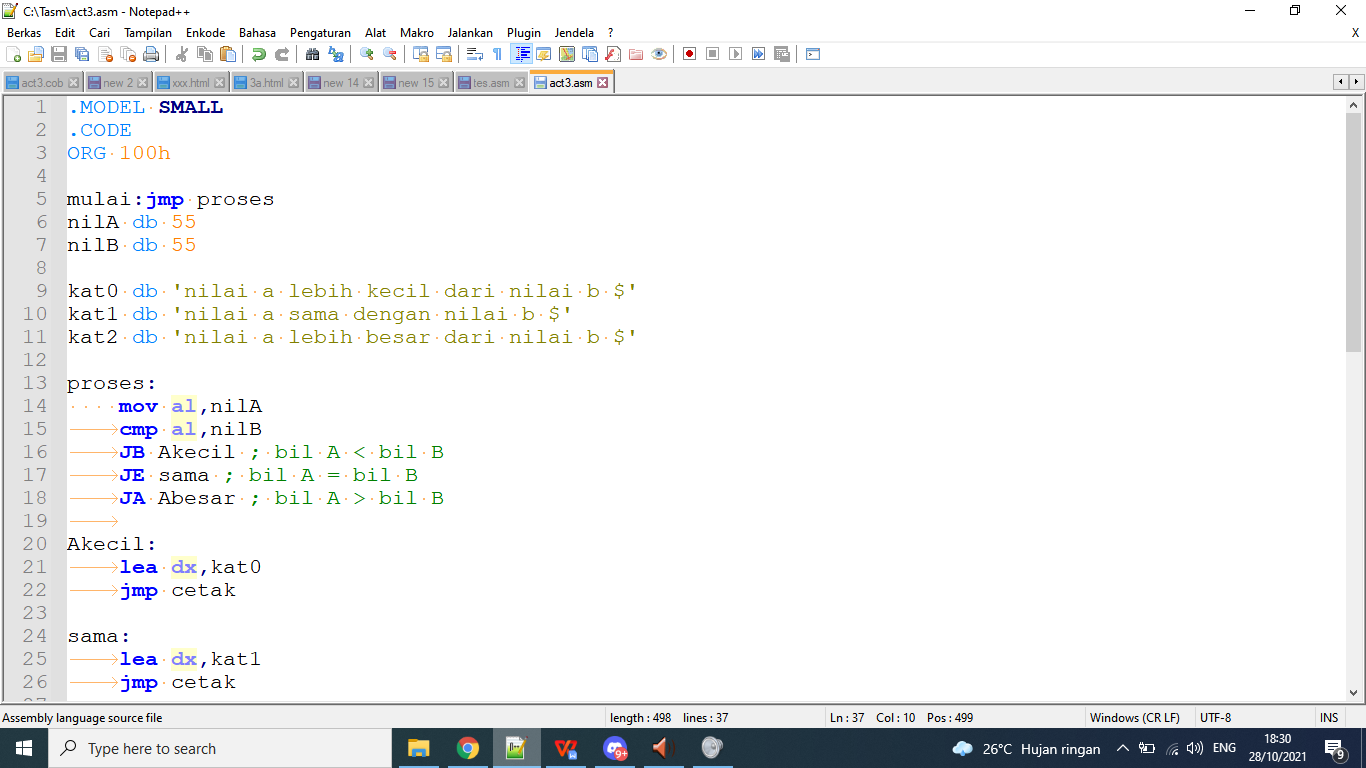
**2021**

**LISTING PROGRAM**

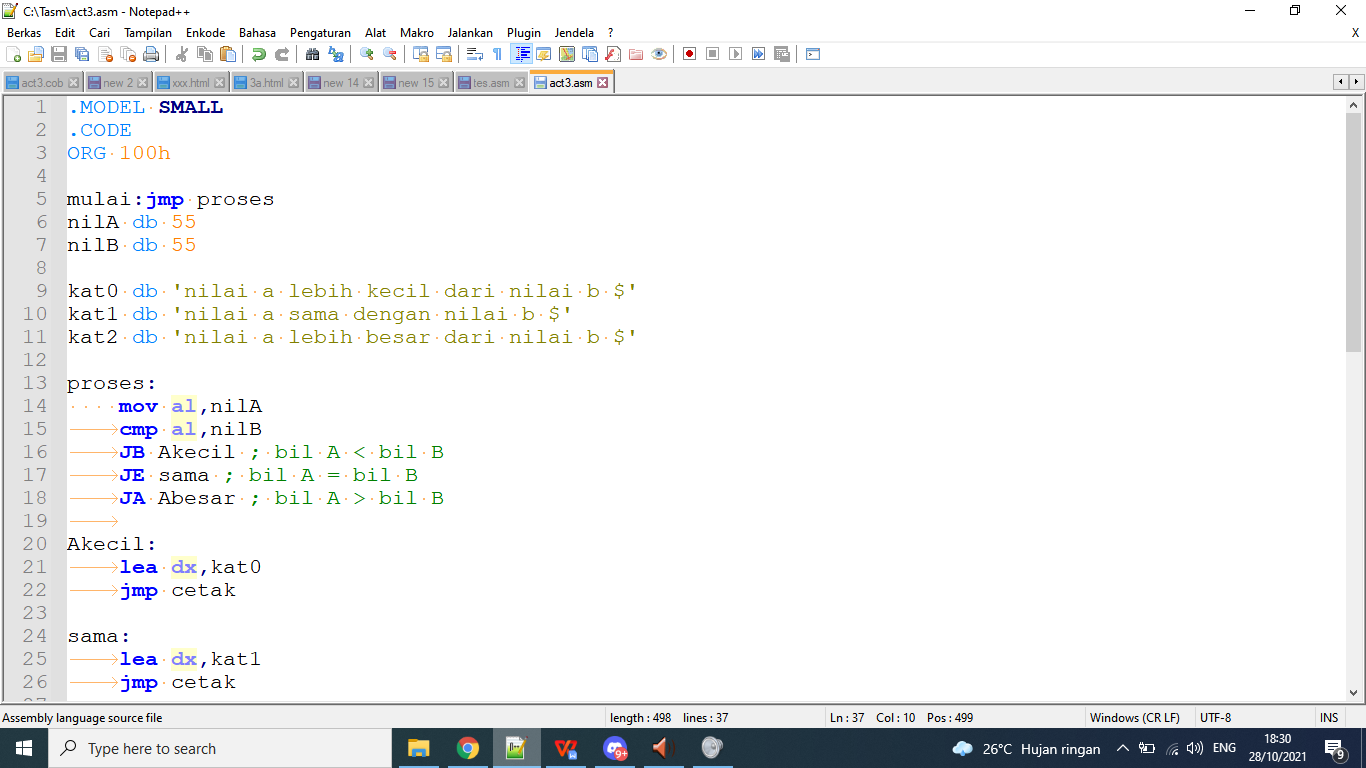


**LOGIKA PROGRAM**

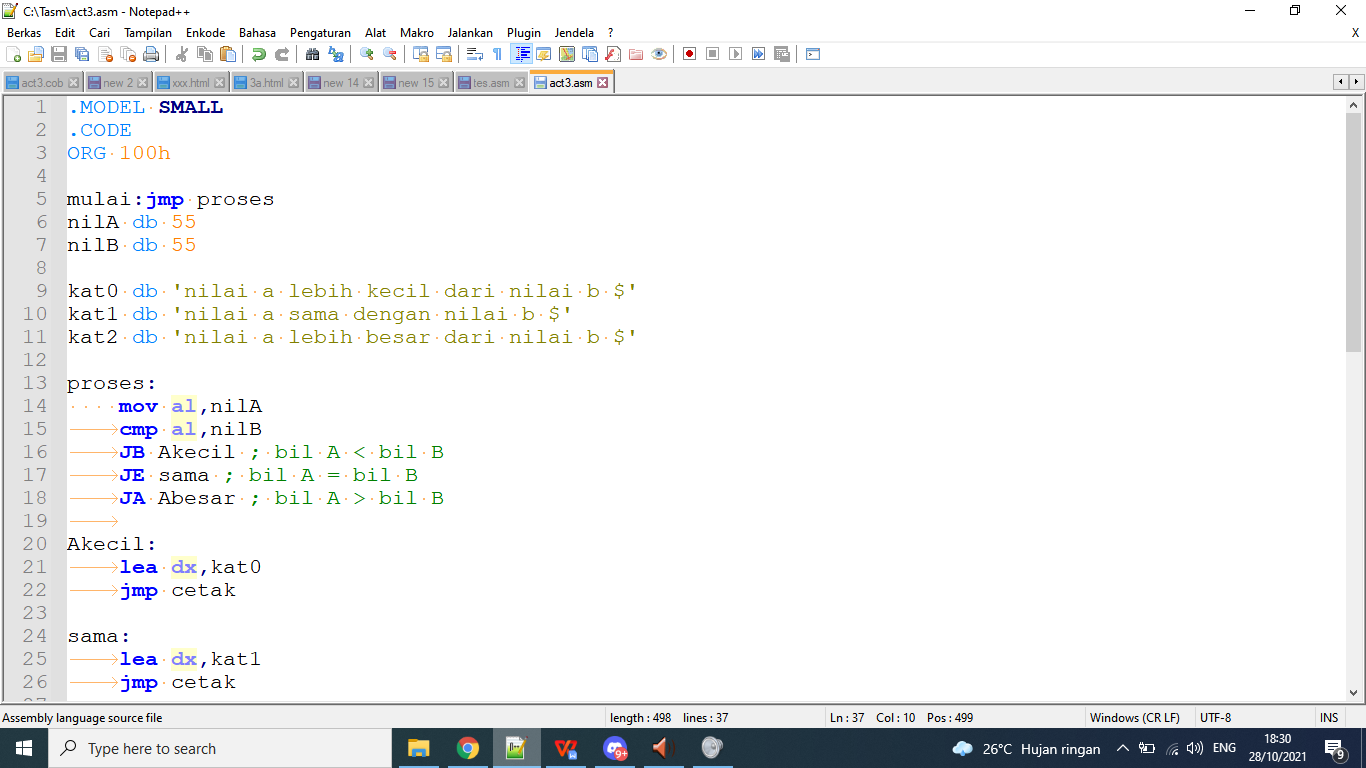
Pada Pertemuan Praktikum pertama ini yaitu Organisasi Sistem Komputer, Saya mendapat materi Addresing Turbo Assembler (TASM) merupakan perangkat lunak bahasa pemrograman Assembly yang dikembangkan Borland untuk komputer personal yang kompatibel dengan IBM, umumnya berbasis prosesor x86. Perangkat lunak ini diperdagangkan oleh Borland sebagai sebuah produk mandiri, atau dipaketkan bersama produk perangkat lunak bahasa pemrograman tingkat tinggi yang juga dikembangkan Borland, biasanya produk untuk tingkat mahir seperti (Borland Pascal, atau Borland C++), dan memiliki integrasi yang sangat baik dengan bahasa-bahasa pemrograman tingkat tinggi tersebut. Dengan berakhirnya era kejayaan keluarga Turbo, Turbo Assembler saat ini sudah tidak lagi dikembangkan oleh Borland. Dibawah ini kita akan menampilkan nilai a sama dengan nilai b.



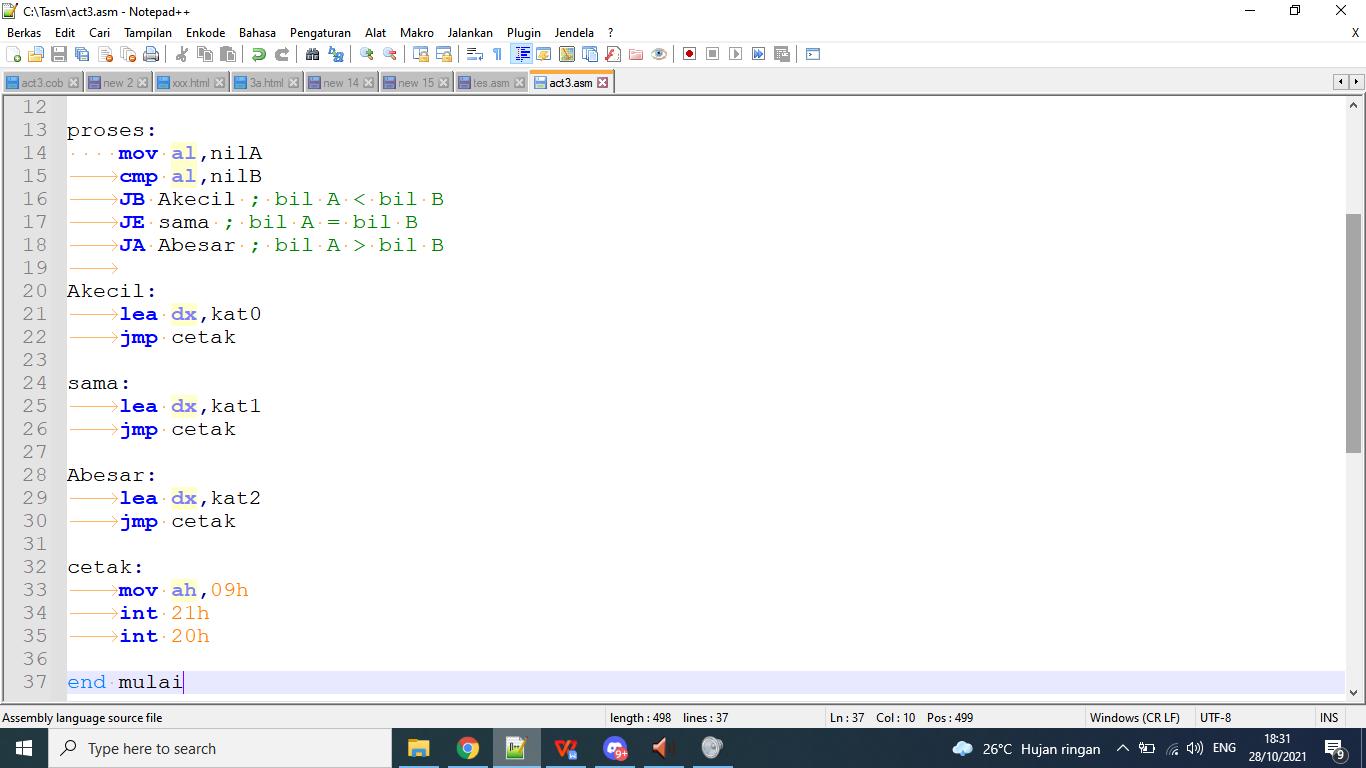
Langkah pertama pada gambar atas yaitu kita menuliskan **.MODEL SMALL** yang merupakan keterangan program yang berfungsi untuk mode dari program yang sudah ada. .**CODE ORG 100h** yang merupakan keterangan yang digunakan untuk menentukan alamat offset terhadap program yang akan digunakan. Perintah **JMP(JUMP)** yang berfungsi untuk melompat menuju tempat yang di tunjukkan oleh perintah JUMP. Code nya adalah **JUMP proses**. Selanjutnya kita membuat variabel **nilA dan nilB** sekaligus memasukkan nilainya. **nilA dan nilB** yang merupakan variabel nilai 1 dan nilai 2, lalu kita masukkan nilainya **55**. Disini saya membuat kedua nilai sama dengan tujuan untuk menampilkan output “**nilai a sama dengan nilai b”**.



Langkah Selanjutnya pada gambar atas yaitu berfungsi untuk membuat keterangan yang akan ditampilkan pada output program. Disini kita membuat 3 variabel, yaitu **kat0, kat1, dan kat2.** **Kat0** akan tampil jika **nilai a lebih kecil dari nilai b**, **Kat1** akan tampil jika **nilai a sama dengan nilai b,** dan **Kat2** akan tampil jika **nilai a lebih besar dari nilai b.** Selanjutnya pada bagian **proses** berfungsi untuk membuat proses agar program berjalan sesuai syarat yang telah ditentukan. **Mov** adalah perintah untuk mengisi nilai ke register, variabel, atau alamat memory tertentu. **mov al, nilA** adalah **nilai al akan berisi nilai dari variabel nilA**. **CMP** digunakan untuk membuat perbandingan antara pernyataan satu dan pernyataan lainnya. Disini cmp digunakan untuk membandingkan **nilA dengan nilB**. Kemudian terdapat kode **JB, JE, dan JA. JB (Jump If Below)**, lompat jika **Operand1 < Operand2**, **JE (Jump If Equal)**, lompat jika **Operand1 = Operand2**, **JA (Jump If Above)**, lompat jika **Operand1 > Operand2**. **Operand1** merupakan **bil A** dan **Operand2** merupakan **bil B.**



Langkah Selanjutnya pada gambar atas yaitu berfungsi untuk mencetak keterangan sesuai dengan hasil perbandingan kedua nilai. Bagian ini terbagi menjadi 3 label, yaitu **Akecil:** untuk **lebih kecil dari**, **sama:** untuk **sama dengan**.



Langkah Selanjutnya pada gambar terakhir di atas yaitu **Abesar**: untuk **lebih besar dari.** Pada program di atas yaitu hanya menggunakan **lea dan jmp. LEA** adalah perintah untuk mendapatkan alamat dari sebuah variabel. Programnya adalah **LEA** variabel. Berdasarkan contoh diatas yaitu **lea dx,kat0** akan mengambil variabel **kat0** yang telah dibuat sebelumnya. Kemudian **jmp cetak** untuk melakukan lompatan ke **cetak**. Begitu juga dengan bagian sama dan **Abesar**;, yang membedakan hanya variabel yang dipanggil untuk mencetak keterangan. Register **dx** harus berisi alamat dari string yang akan dicetak. Selanjutnya membuat **cetak** yang menggunakan sintaks **mov ah,09h** yaitu akan mencetak karakternya, **int 21h** dan **int 20h** yaitu sebagai penutup program.

**OUTPUT PROGRAM**

