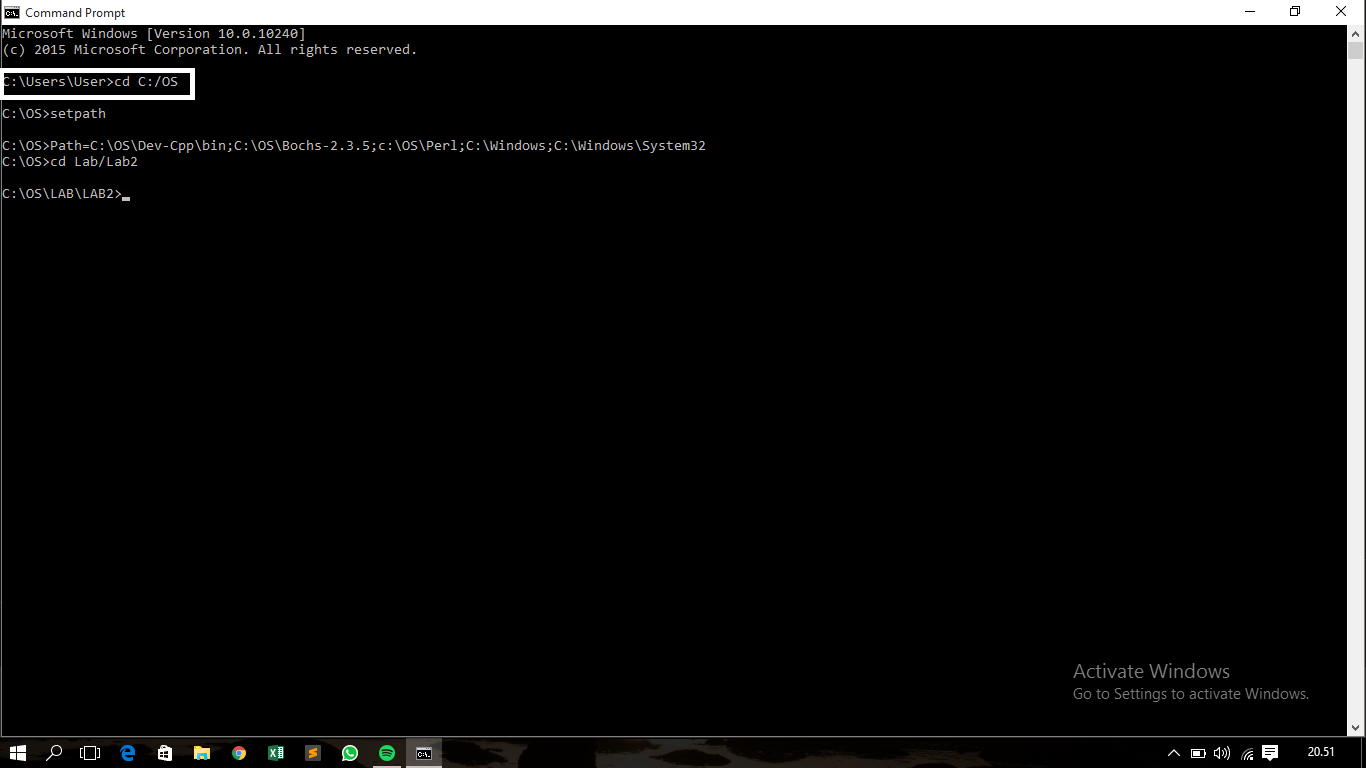
Nama : Rima Dwi Novika

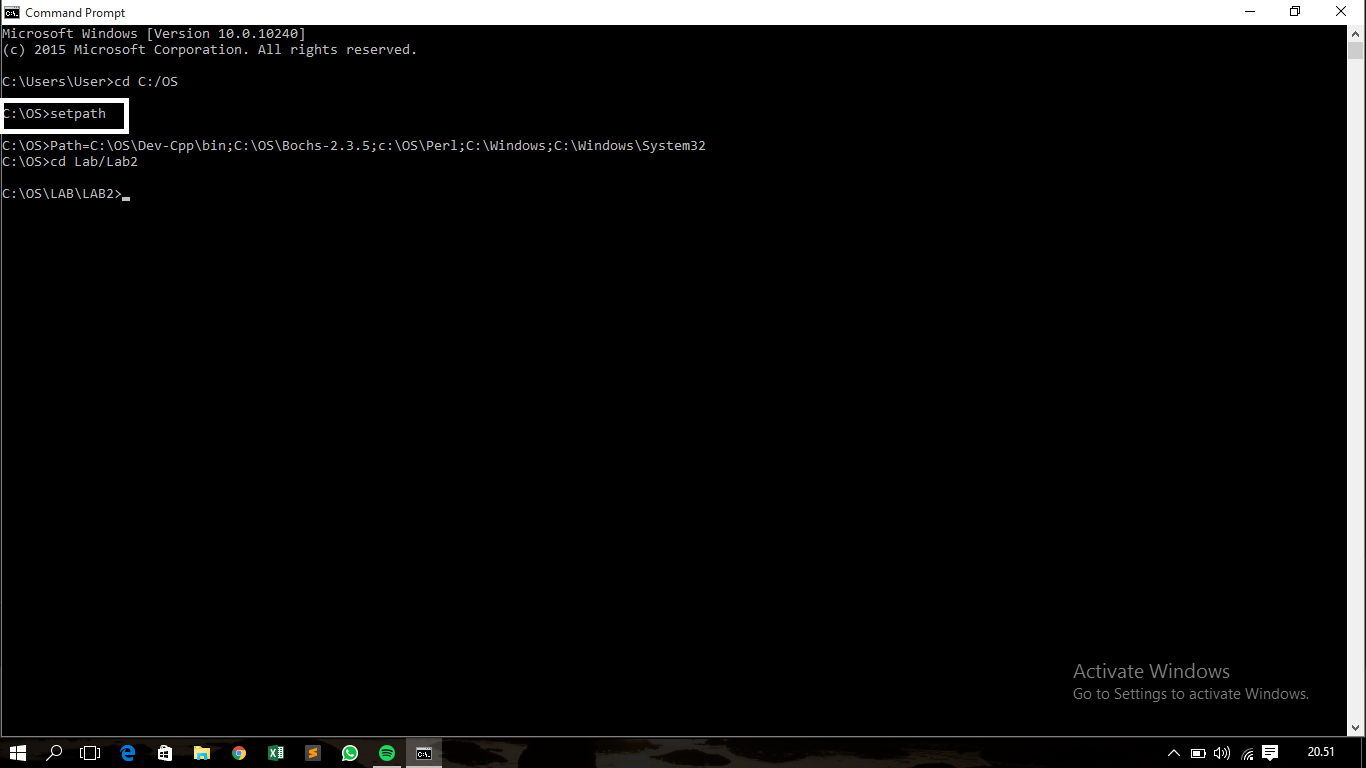
NIM : L200170044

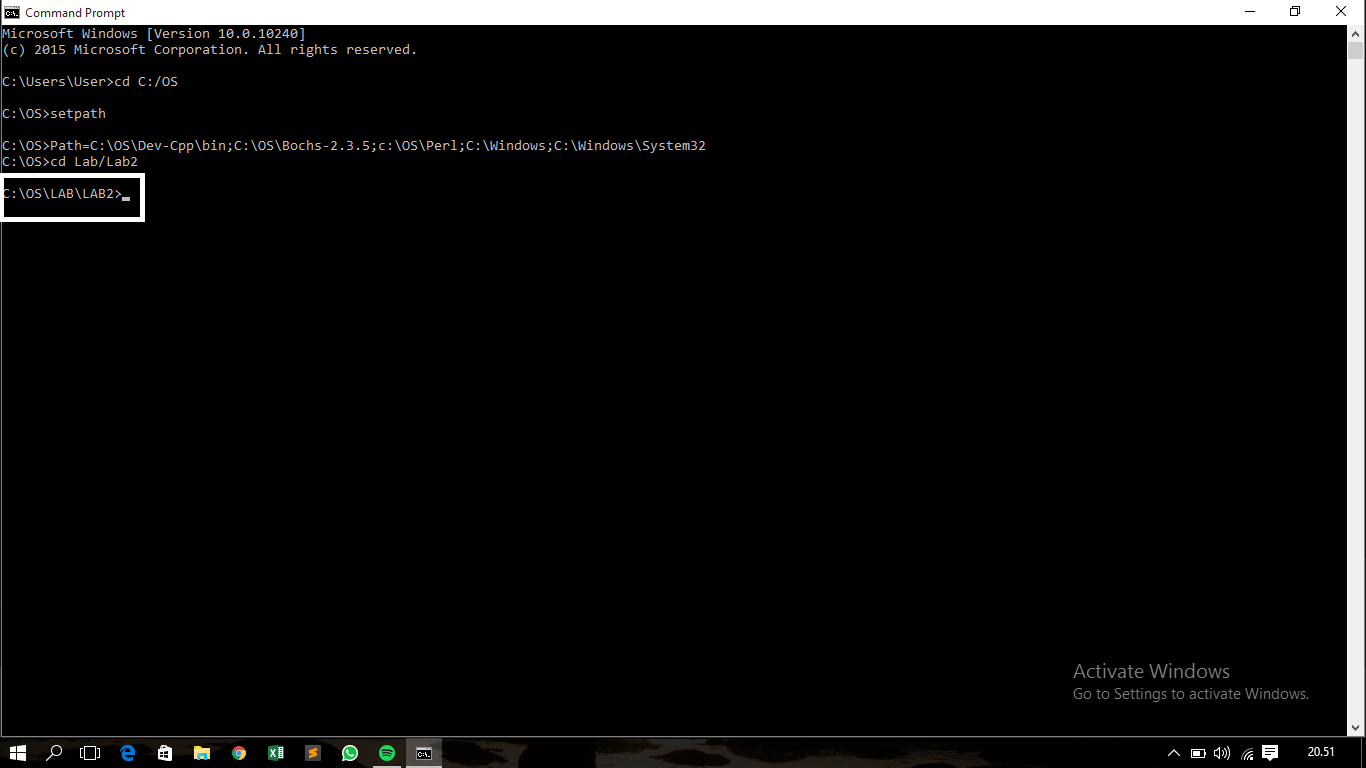
Kelas : C

1. Masuk ke direktori C:/OS dan melakukan setpath, kemudia masuk ke dorektori lab/lab2. Kemudia lakukan dir untuk mengecek isi dari direktori

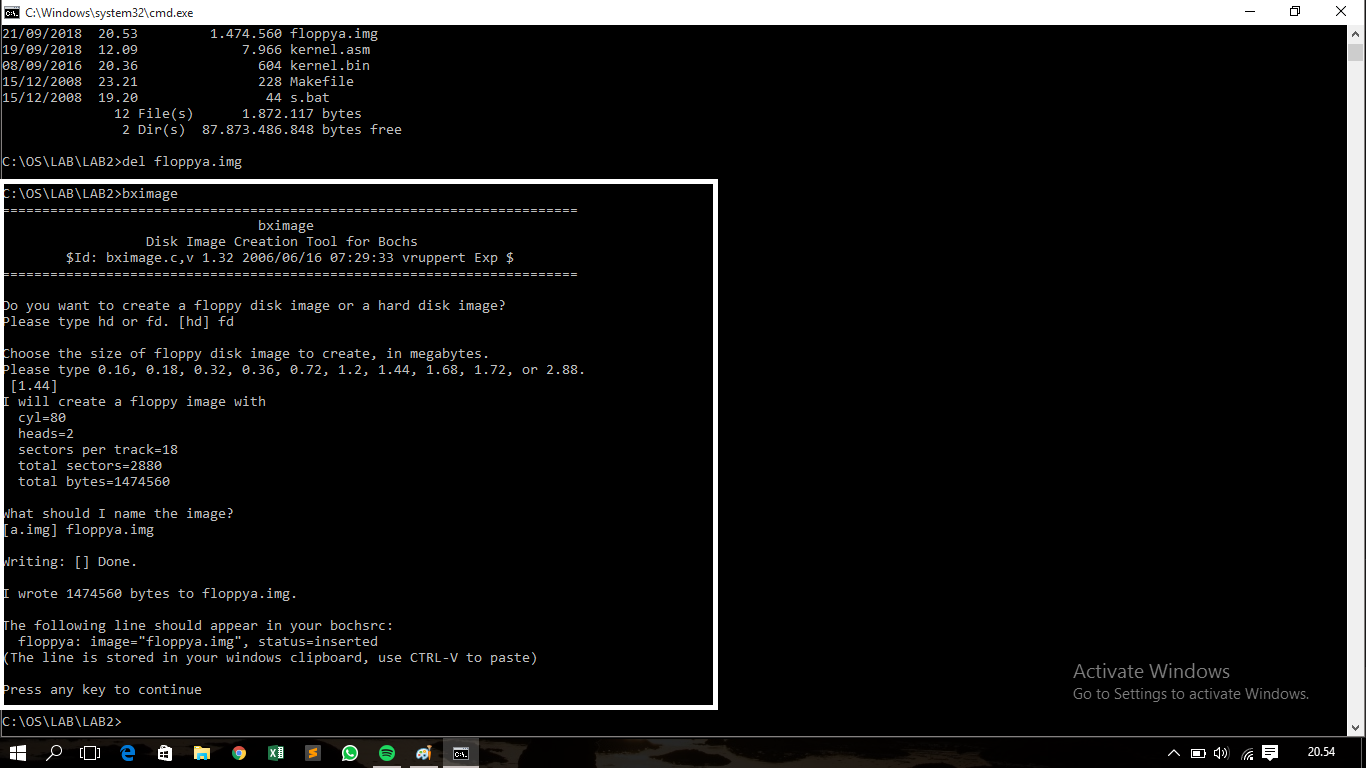
CC



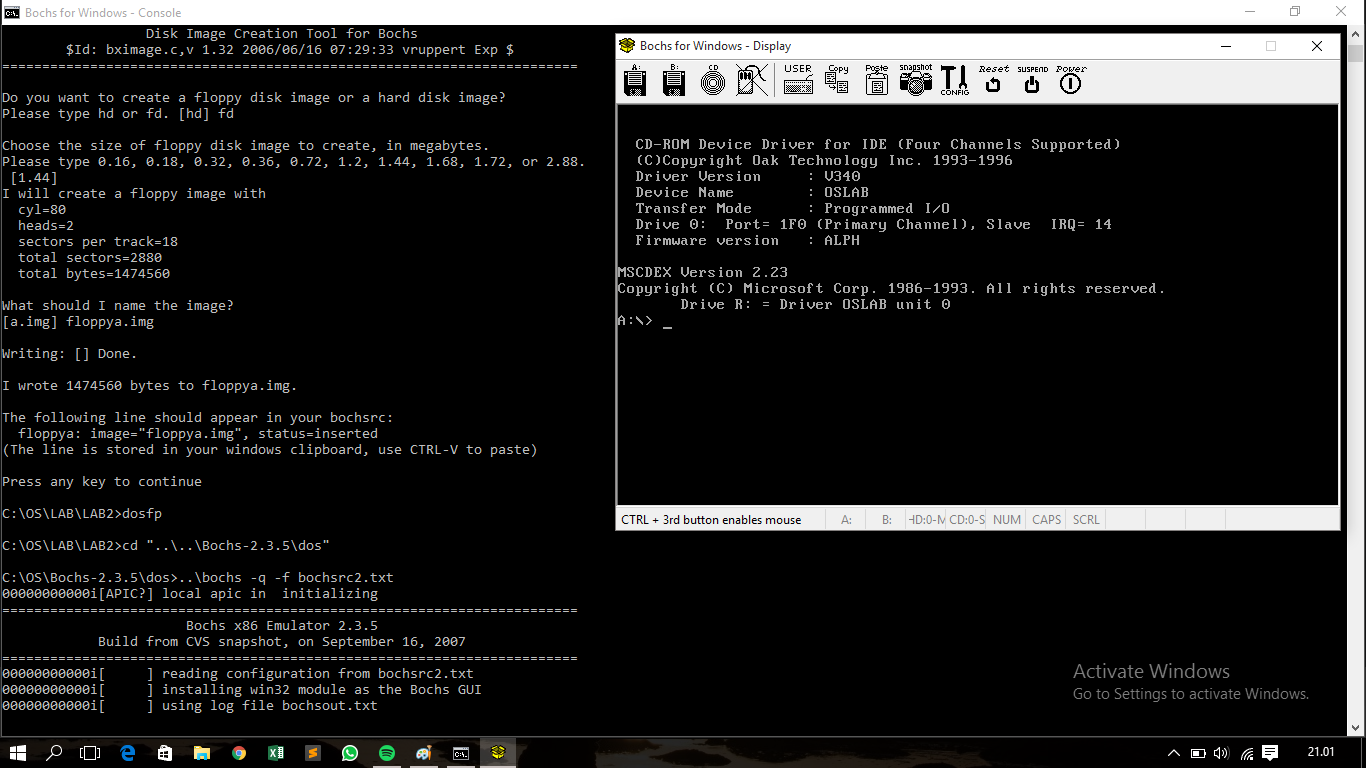




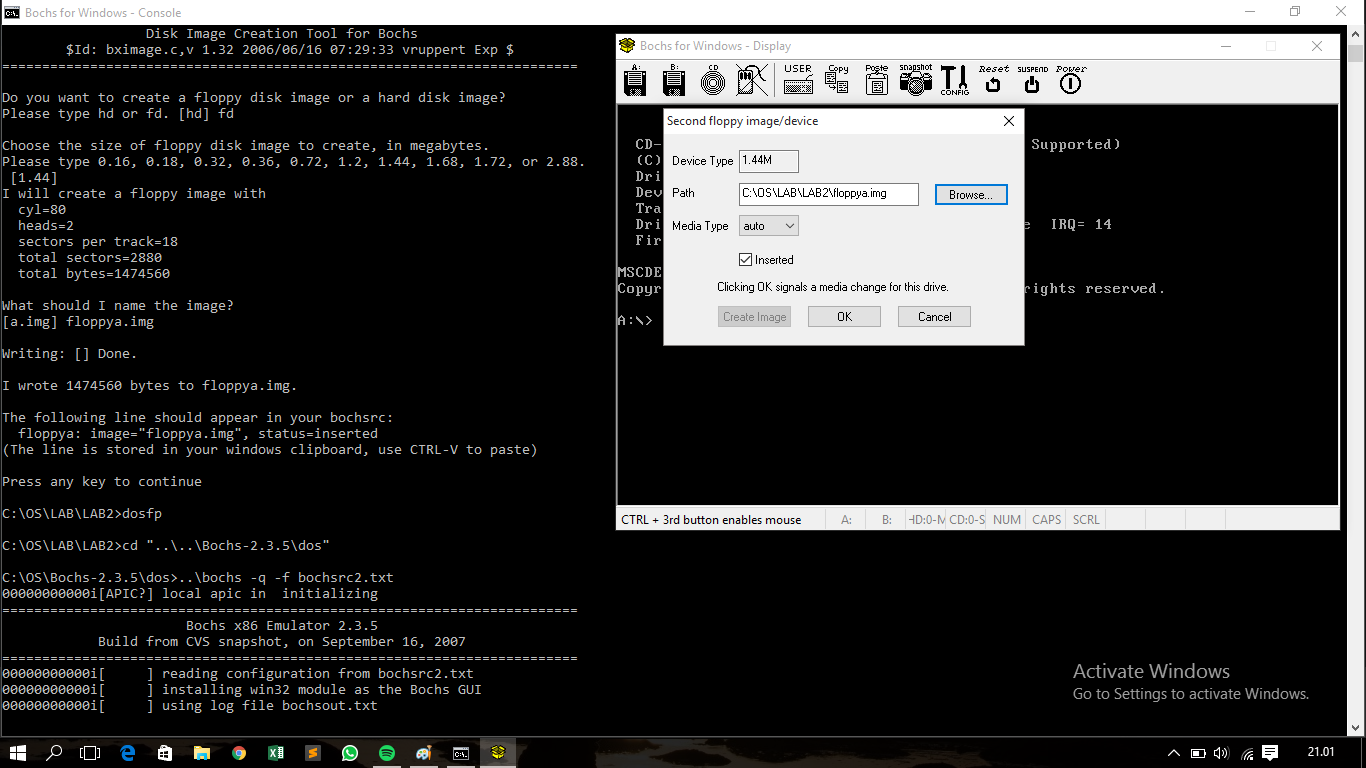
2. Kemudian buat suatu file bernama “floppya.img”, dengan bantuan aplikasi bximage

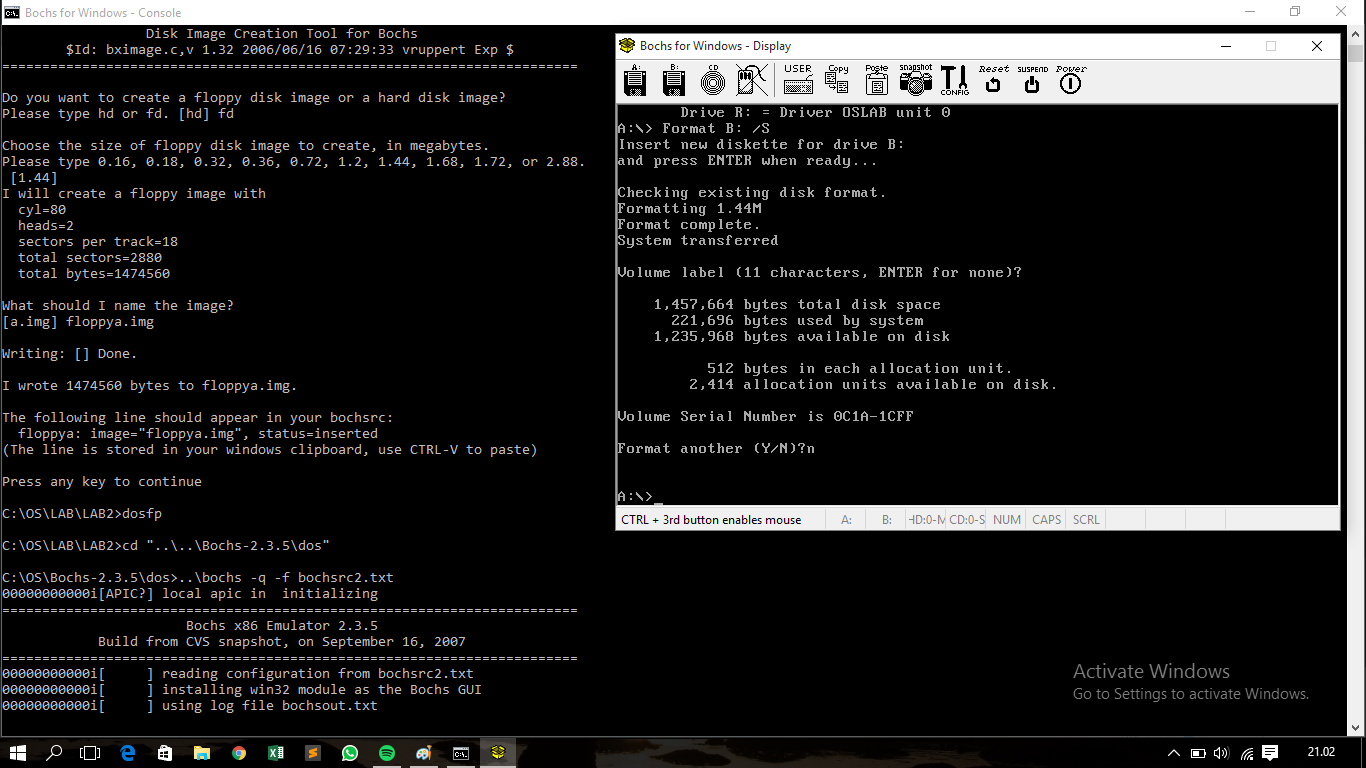


3. Kemudian format floppya.img dengan mengisi DOS ke dalamnya

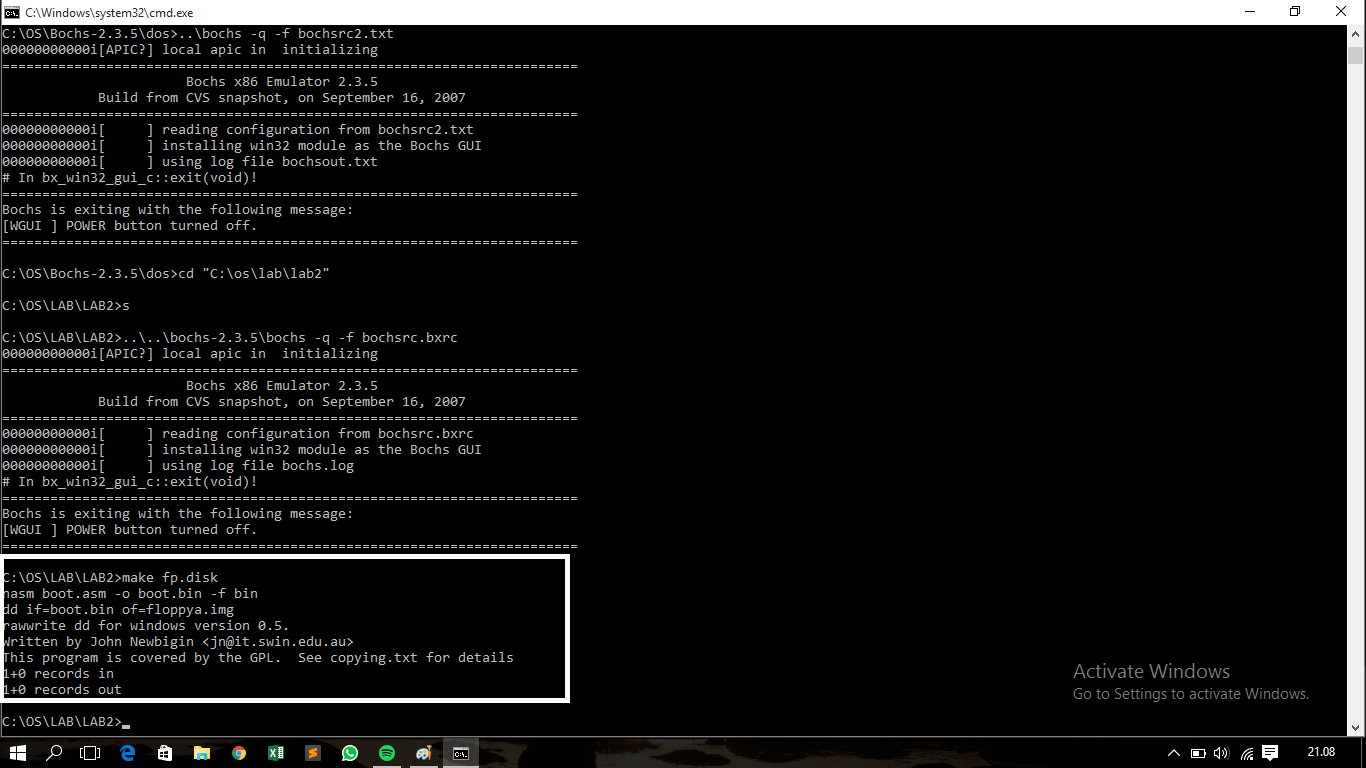


4. Aturl lokassi floppya.img yang ada dalam folder lab2. Kemudia lakukan pemformatan.

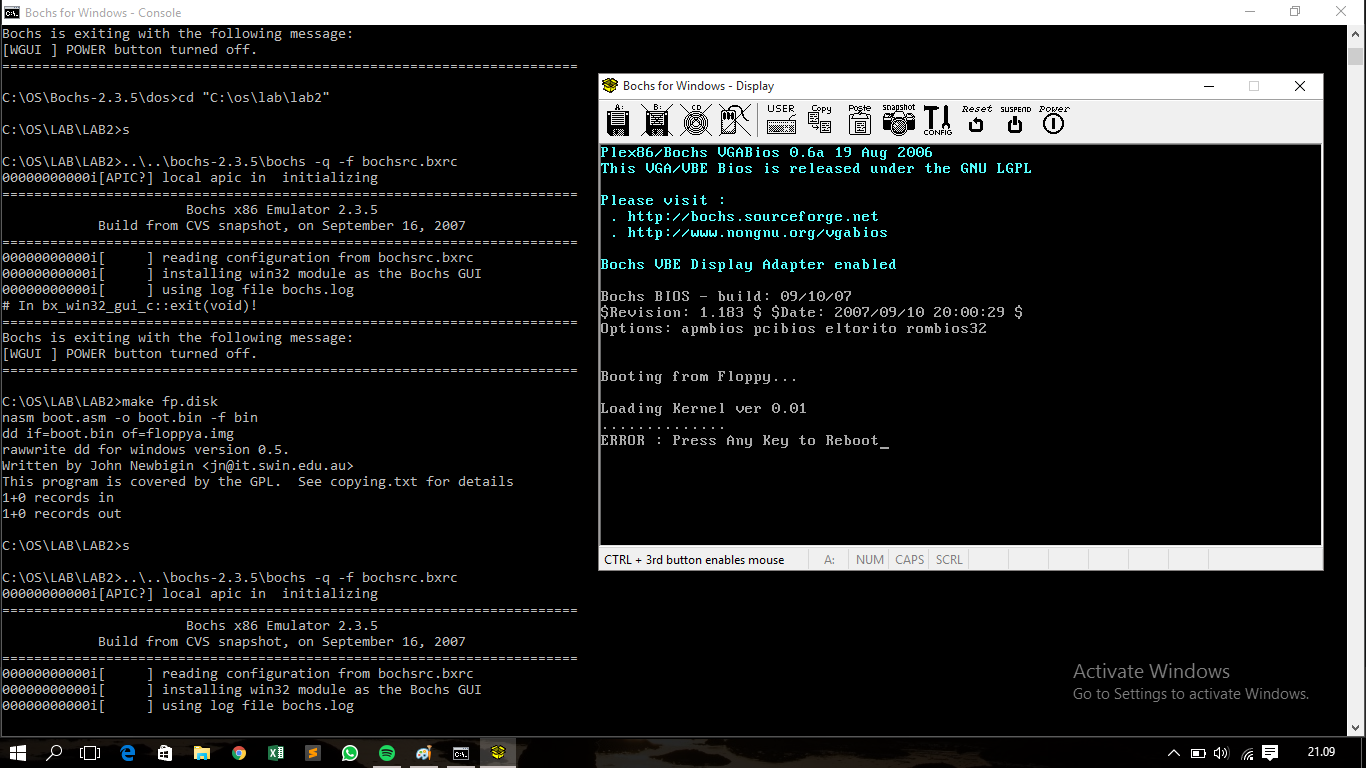




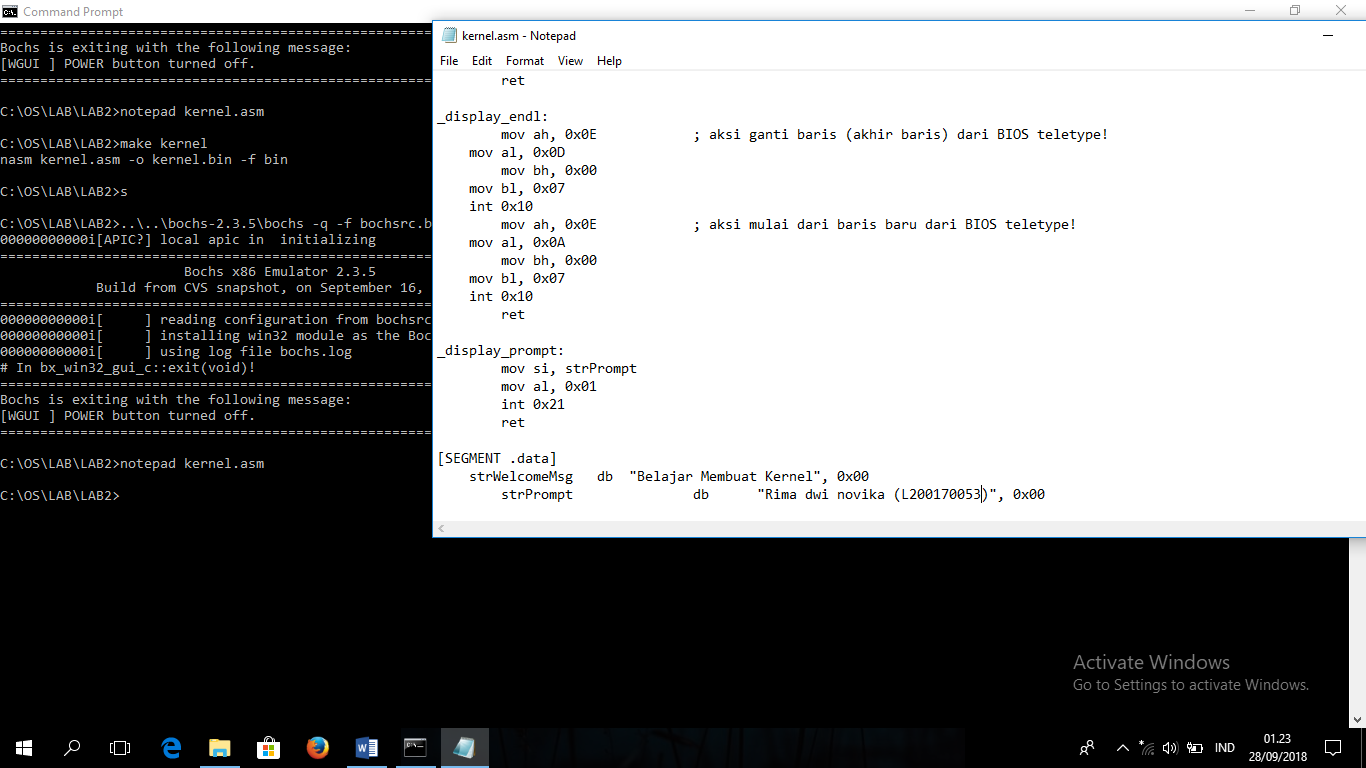
5. Memasukkan source code dari boot.asm ke bootsector “floppya.img”. Kemudian jalankan perintah make fp.disk



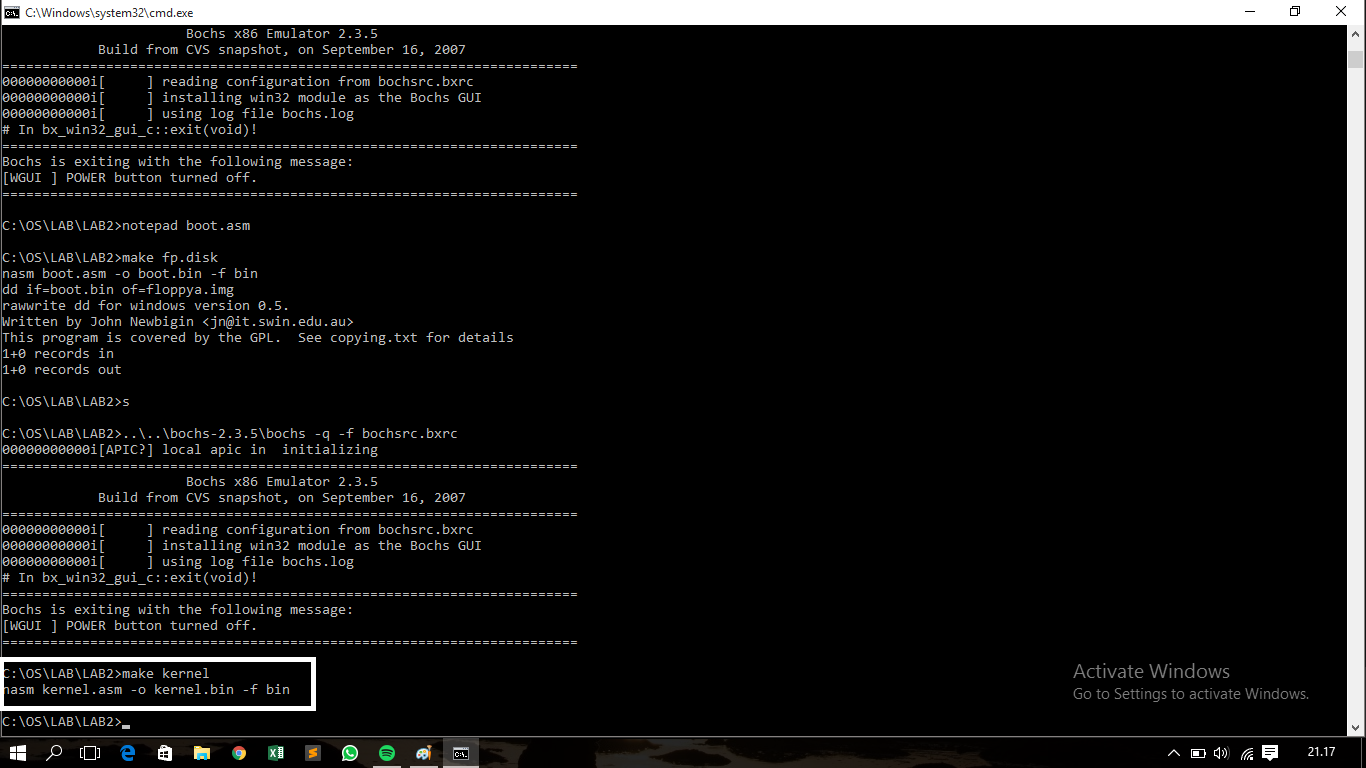
6. Lalu ketikkan ‘S’

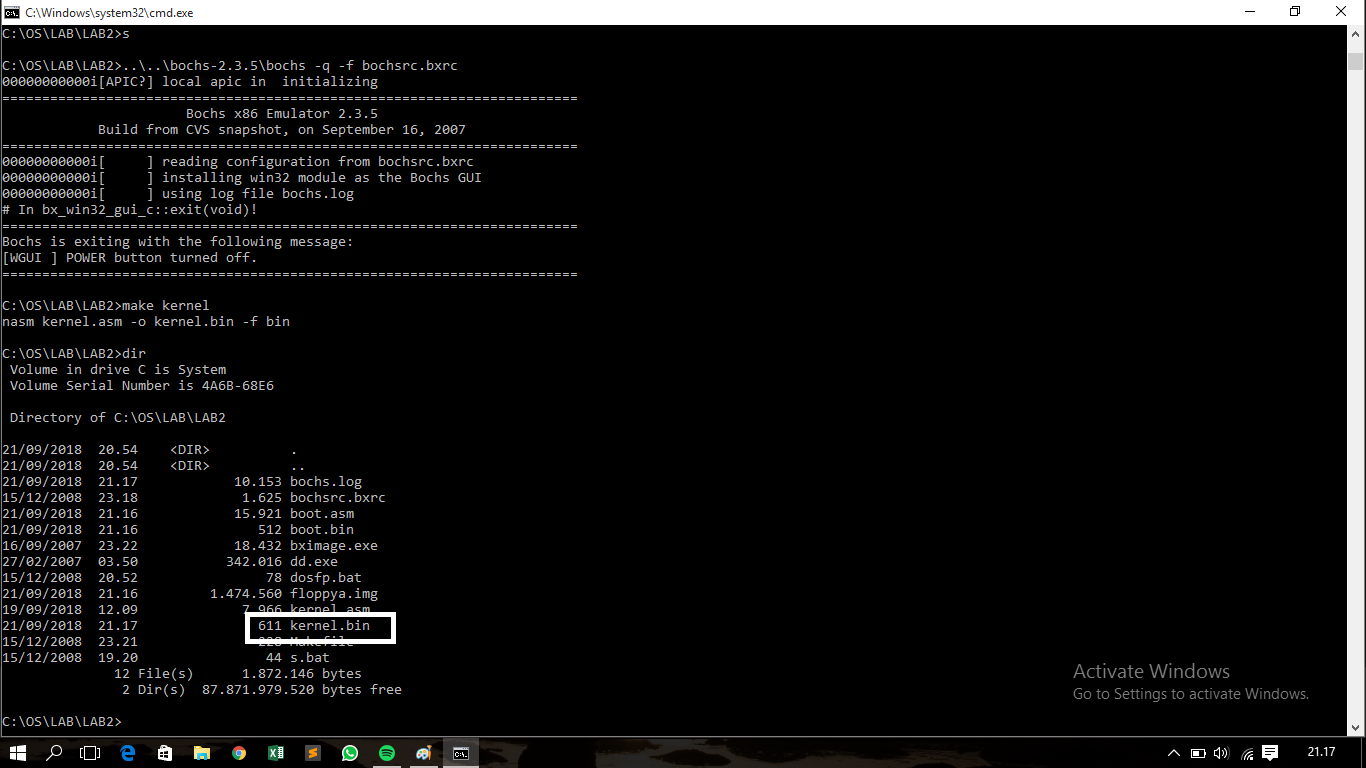


7. Selanjutnya suntukng boot.asm dengan mengetikkan ‘notepad boot.asm’ di mana membuka file boot.asm dengan menggunakan notepad. Kemudian ganti msgLoading sesuai perintah. Dan ganti error dengan nama dan nim.

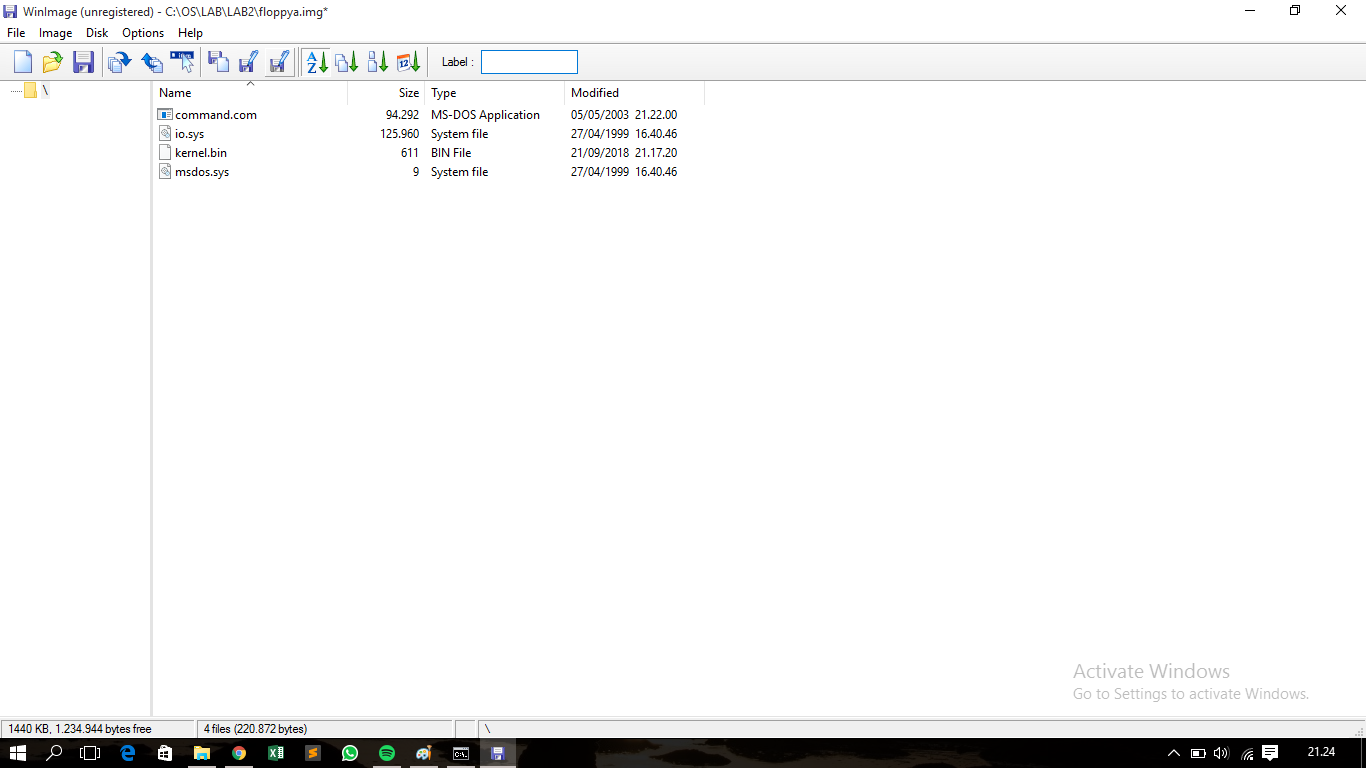


8. Ketikkan make kernel. Kemudian cek apakah sudah ada file baru bernama “kernel.bin”

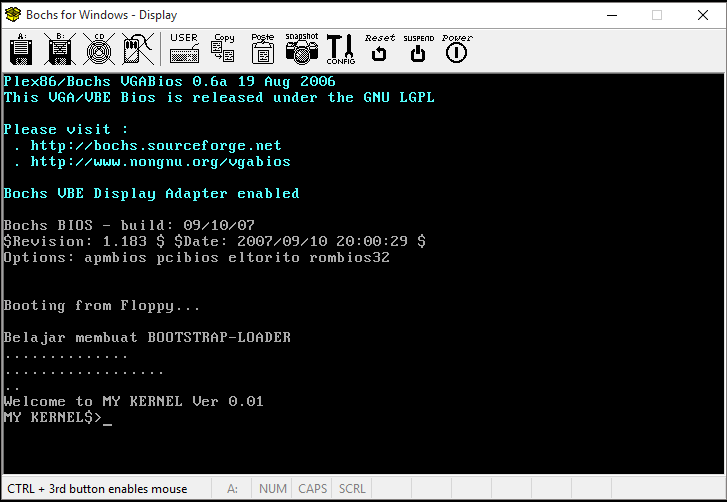




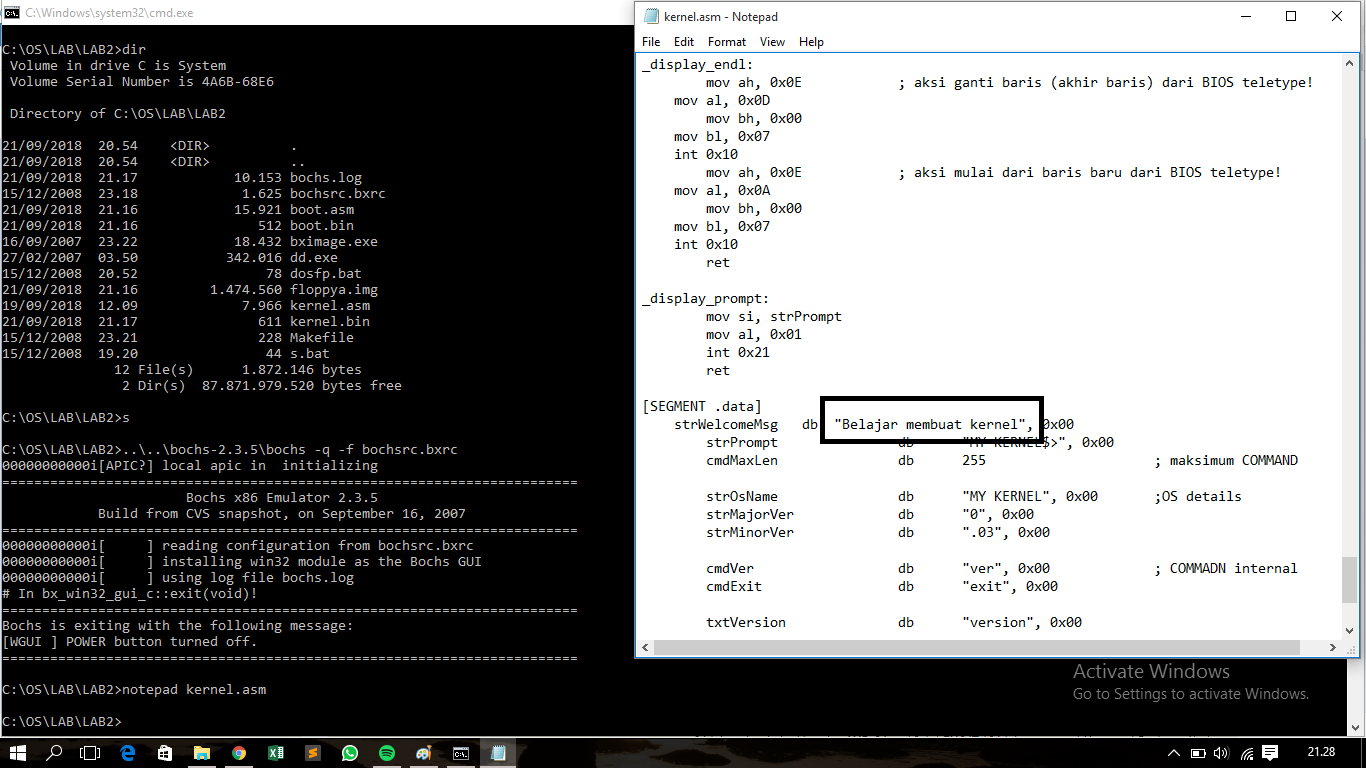
9. Memasukkan file kernel.bin ke dalam file floppya.img



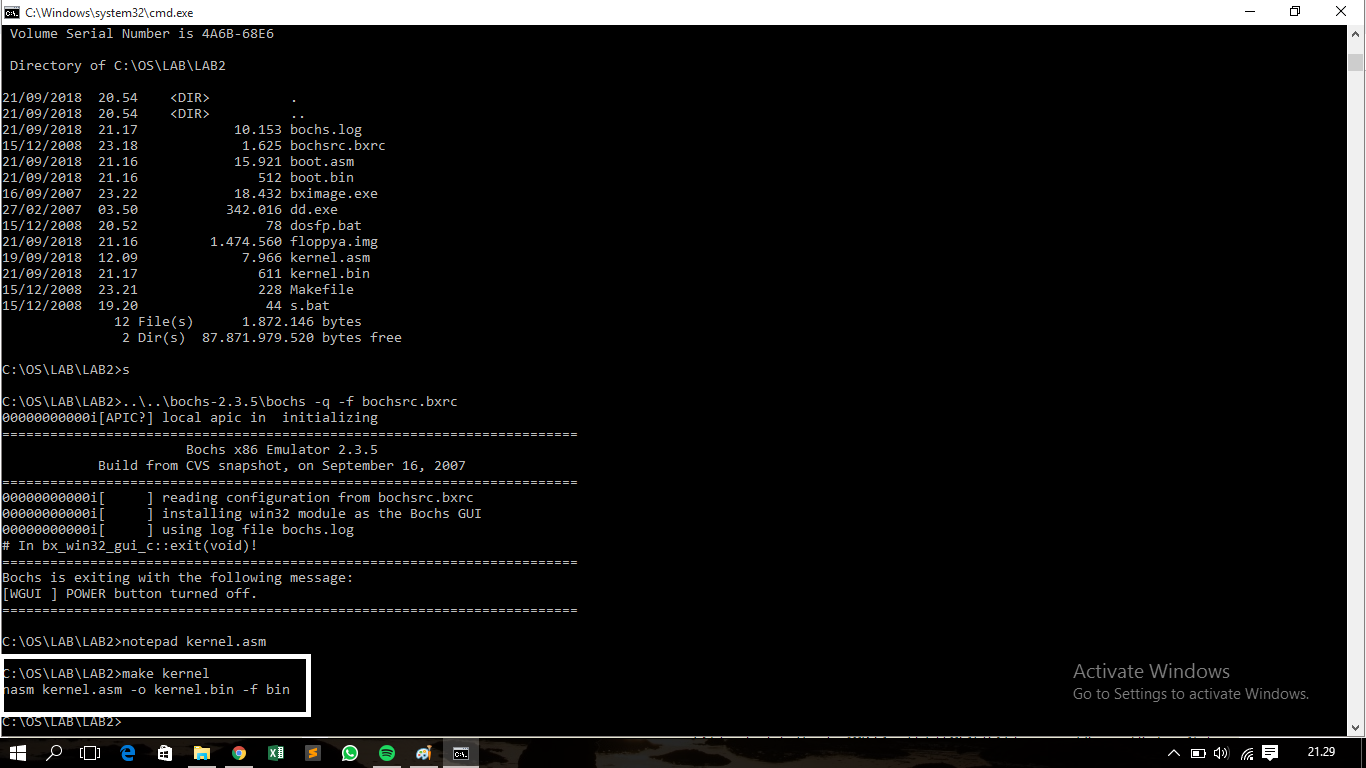
10. Cek dengan “S”



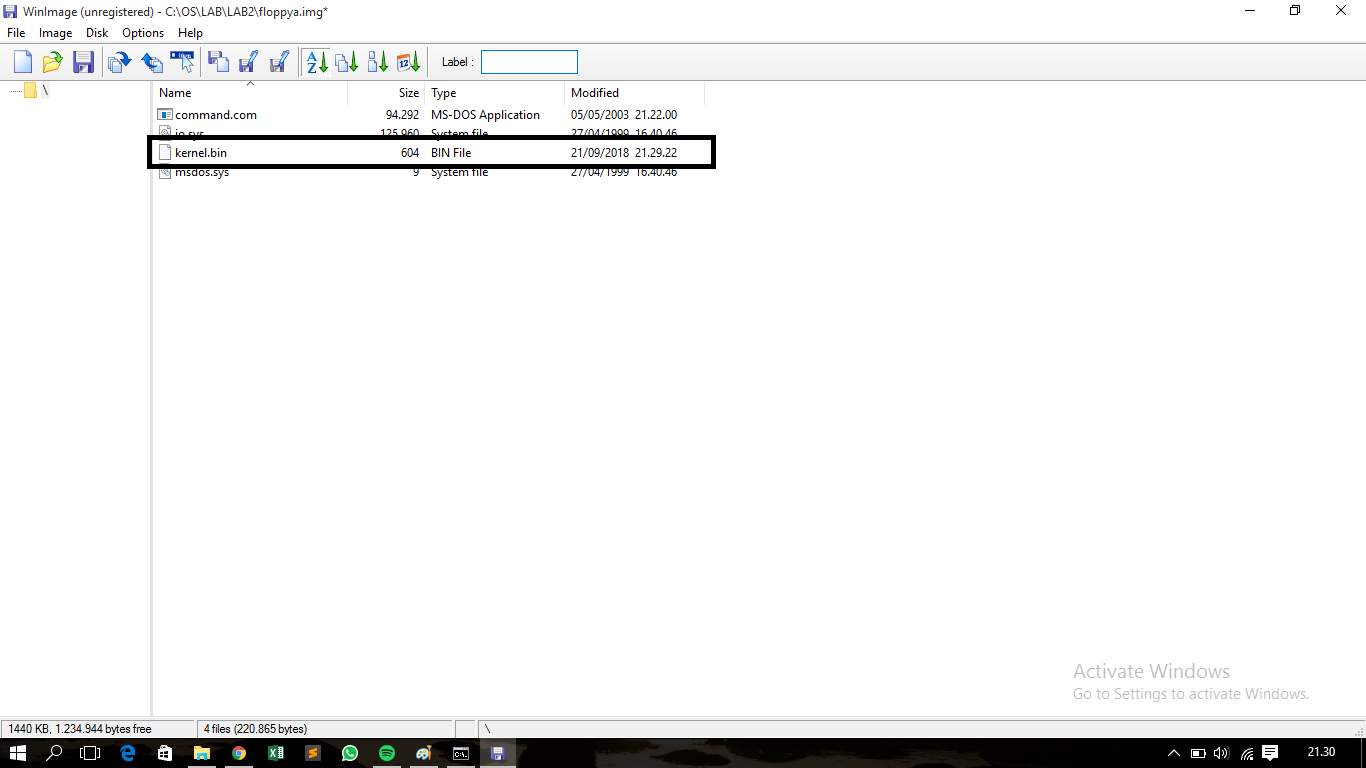
11. Modifikasi file kernel.asm. Ganti strWelcomeMsg sesuai perintah.



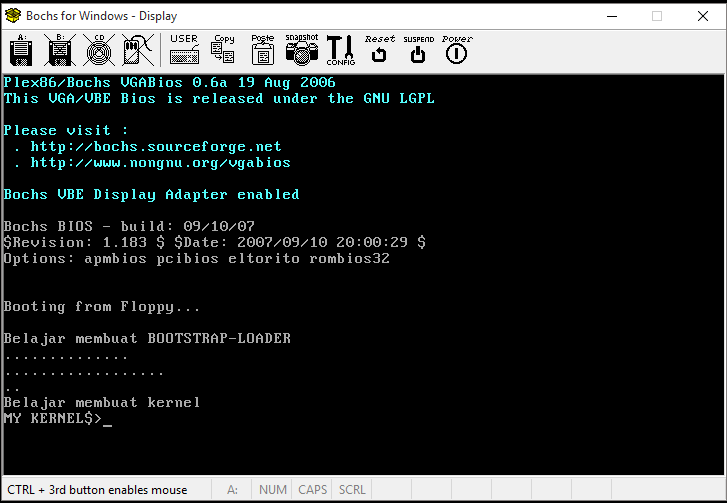
11. Kemudia make kernel kembali untuk merefresh.



12. Dan masukkan kembali kernel yang telah diperbaharui.



13. Kemudia cek dengan mengetikkan “S”



1. **boot.asm**

**Global**

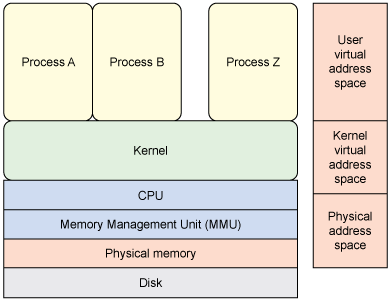
**Detail**

1. Mengatur mode kerja 16 bit (real-mode)
2. Menentukan lokasi awal dari program
3. BLOCK ID FAT12 Keterangan format FAT12 pada floppy disk
4. BLOCK BOOT CODE mengatur lokasi kode program,mengatur lokasi stack,menampilkan

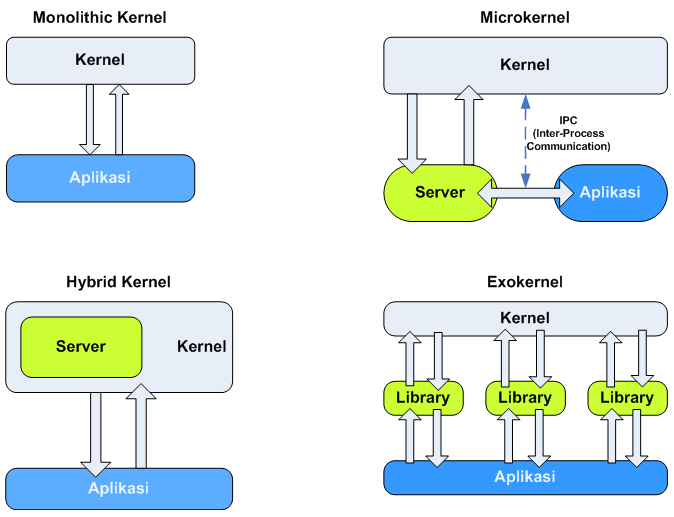
text dilayar,menghitung ukuran root directory dan menyimpanya dalam register,memindahkan isi root direktori ke memori ram,mencari nomer awal sektor lokasi FAT,baca isi FAT dipindahkan ke memory,baca isi file kernel.bin

1. **kernel.asm**

**Global**



**Deskriptif**



1. Kernel Monolitik, Kernel monolotik mengintegrasikan banyak fungsi di dalam kernel dan menyediakan lapisan [abstraksi Hardware](http://www.plimbi.com/article/3116/blackberry-hardware" \o "Rahasia Dibalik Sebuah BlackBerry Hardware) secara penuh terhadap perangkat keras yang berada di bawah sistem operasi.
2. Mikrokernel, Mikrokernel menyediakan sedikit saja dari abstraksi perngakat keras dan menggunakan aplikasi yang berjalan di atasnya untuk melakukan beberapa fungsionalitas lainnya.
3. Kernel Hybrida, kernel hybrida terdapat beberapa tambahan kode di dalam ruangan kernel untuk meningkatkan performanya.
4. Exokernel, Exokernel menyediakan hardware abstraction secara minimal, sehingga program dapat mengakses hardware secara langsung. Dalam pendeketana desain exokernel, library yang dimiliki oleh sistem operasi dapat melakukan abstraksi yang mirip dengan abstraksi yang dilakukan dalam desan monolithic kernel.