

**Praktikum 0x1**  
**IF2130 - Organisasi dan Arsitektur Komputer**  
**II2130 - Arsitektur dan Sistem Komputer**

Revisi 04 10 2023 11:37

*"MaiBit"*

***Bit-level Manipulation***

Dipersiapkan oleh:  
Asisten Lab Sistem Terdistribusi

Didukung Oleh:



**Waktu Mulai:**  
28 September 2023, 18.30.00 WIB

**Waktu Akhir:**  
5 Oktober 2023, 23.59.59 WIB

## Daftar Isi

I. Latar Belakang	1
II. Deskripsi Tugas	3
III. Teknis dan Penilaian	3
IV. Langkah Umum Pengerjaan	4
V. Pengumpulan dan Deliverables	8
VI. Referensi	10

## Daftar Revisi

28/09/2023 18:46 - Penambahan informasi terkait file .zip

28/09/2023 19:51 - Penambahan informasi terkait apresiasi khusus

04/10/2023 11:36 - Penambahan informasi terkait ARM

## I. Latar Belakang

*Nainai* (ditulis sebagai *nainai*) merupakan sebuah permainan ritme arkade yang dikembangkan dan didistribusikan oleh **MEGA**, dimana pemain berinteraksi dengan objek-objek di layar sentuh dan mengeksekusi gerakan seperti tarian.



Logo *nainai* DX, iterasi terbaru *nainai*

Pertama kali kamu bertemu dengan *nainai*, kamu tidak melihatnya sebagai suatu hal yang menarik. Setiap kali kamu dan teman-temanmu bermain ke **GAMEZONE**, seringkali kamu lebih memilih untuk menghambur-hamburkan uang di mesin gacha demi figur **Sabar** (yang versi *alter*-nya ya, bukan yang biasa) ketika teman-temanmu sedang tertawa ria, *tryhard*, dan berdansa di mesin *nainai*.

Namun, kamu adalah seorang mahasiswa informatika (atau STI), seseorang yang ~~berjiwa ksatria dan tidak mengenal keluh kesah~~ seringkali stres akibat terjangan tsunami tugas besar, praktikum, dan kuis. Alhasil, di tengah kresendo beban kuliah, kamu memutuskan untuk mencoba permainan *nainai* ini yang konon se-seru itu sampai-sampai salah satu temanmu sudah menggesek kartunya sebanyak 1000 kali.

*Gesek*. Mungkin, jika dompet bisa berbicara, punyamu akan teriak saat itu juga, mengetahui seberapa kering ia akan menjadi di masa yang akan datang.



*Susu*, companion pilihanmu di *nainai*

Kamu disambut oleh **Susu**, sebuah *catgirl* putih yang comel dan lucu. Dengan suaranya yang amat imut itu, ia memandumu untuk mengatur opsi dan memilih lagu yang hendak dimainkan. Setelah meniru konfigurasi yang digunakan teman-temanmu, kamu

kembali ke menu pemilihan lagu, dan kamu pun terkagum akan seleksi yang tersedia. Lagu-lagu JPop dan opening anime? Ada. Lagu-lagu dari unit musik TUYUL? Ada juga. Lagu-lagu resmi dan *dōjin* serial permainan Tahu Project? Banyak. Diatas itu semua, terdapat banyak sekali lagu yang dinyanyikan **Matsune Hiku** (sang penyintesis suara dan virtual idol terkenal dari Jepang), antara lain beberapa favoritmu seperti "Aturan Hantu" (ゴーストルール), "Percikan" (ヒバナ), dan "Suara Matsune Hiku yang Amat Berisik" (松音日クの激唱).



**Matsune Hiku**

Pada lagu pertama, hasil yang kamu dapatkan tidak begitu bagus. Sebatas B, itupun di tingkat kesulitan *basic* (~~skill-issue sih kalo kata penulis~~). Namun, tidak butuh waktu yang lama bagimu untuk membiasakan diri, dan di akhir gesekan pertamamu tadi, kamu ternyata berhasil mendapatkan A di *Advanced* ber-level rendah.

Seiring berjalannya waktu, skill kamu semakin bertambah, dan akhirnya kamu juga dapat meng-clear tingkat kesulitan *Expert*. Kamu juga mulai mengenal lagu-lagu original *nainai*, seperti "Kalibakar ~Kisah Pedang Legendaris~" (Caliburne ~Story of the Legendary Sword~) dan "135 Detik Isinya Api Semua" (封焰の135秒). Teman-temanmu pun banyak yang bertanya, mengapa kamu dapat tiba-tiba jago ~~dan teradiksi~~ dalam waktu sesingkat itu.

Di perjalanan kamu untuk menjadi pe-*nainai* handal nomor satu, kamu mendengar dari sebuah forum Jepang bernama /2ch/ akan sebuah metode gelap yang konon sangat membantu dalam meningkatkan skill. Rupanya, *chart-chart nainai* dapat dianalisis dengan cara... melihat dan memanipulasi rangkaian bit yang menyusunnya, menggunakan sebuah sistem operasi spesial. Sembari menunggu antrian di **GAMEZONE**, kamu membuka laptop kamu dan langsung mengunduh *chart-chart*, kemudian langsung menjalankan **KotakVirtual** untuk membuka sistem operasi dan mulai menganalisis *chart-chart* tersebut dengan sungguh-sungguh. "Demi dipuja sama Susu, hehehe".

"それでは行きましょう!"

## II. Deskripsi Tugas

Pada tugas ini, kalian akan mengeksplorasi cara kerja komputer untuk melakukan operasi pada bilangan di tingkatan bit. Hal ini akan kalian lakukan dengan membuat fungsi operator-operator modern dengan operasi bit yang terbatas. Bahasa yang digunakan adalah bahasa C. Namun, kalian akan mengerjakannya pada sistem operasi yang sudah disediakan oleh asisten yang bisa kalian jalankan pada virtual machine.

Perhatikan fungsi contoh berikut

```
int nainai(int x) {  
    return x < 0;  
}
```

Operasi `x < 0` merupakan operasi tingkat tinggi. Pada praktikum ini, kalian harus mengubahnya menjadi operasi tingkat bit hingga menjadi seperti berikut.

```
int nainai(int x) {  
    return (x >> 31) & 0x1;  
}
```

Assignment variabel (operator `=`) tidak akan memakan jatah operator dan selalu diperbolehkan di seluruh soal. Ketentuan lain terkait jumlah operator dan jenis operator legal disebutkan pada deskripsi setiap soal.

## III. Teknis dan Penilaian

1. Pada praktikum ini, terdapat 12 soal yang wajib dikerjakan.
2. Pada setiap soal, tertulis rating atau poin yang akan didapatkan, beserta dengan constraint legal ops dan max ops.
3. Untuk setiap jawaban soal wajib yang tidak melanggar constraint legal ops, akan mendapatkan poin *correct* sebanyak rating soal. Constraint legal ops adalah operator-operator yang boleh digunakan oleh praktikan untuk menjawab soal tersebut.
4. Untuk setiap jawaban soal wajib yang tidak melanggar constraint max ops, akan mendapatkan 2 poin *perfect*. Constraint max ops adalah jumlah operator maksimum yang boleh digunakan.
5. Apabila seluruh soal wajib dijawab dengan benar, maka praktikan akan mendapatkan poin sebanyak 54 (30 *correct* + 24 *perfect*).

6. Selain itu, juga terdapat 5 soal *bonus* dan 1 soal *super bonus* yang dapat dikerjakan untuk penambahan nilai.
7. Untuk setiap jawaban soal *bonus* serta *super bonus* yang tidak melanggar constraint legal ops DAN maximum ops, akan mendapatkan poin *bonus* sebanyak rating soal.
8. Nilai maksimum yang dapat diperoleh dalam praktikum ini sebanyak 125, dengan perhitungan:

$$NA = ((\text{Correct} + \text{Perfect}) / 54) * 100 + \text{Bonus}$$

9. 3 NIM yang pertama kali berhasil menyelesaikan semua soal, **termasuk soal wajib, bonus, dan super bonus** setelah praktikum dimulai akan mendapatkan *apresiasi khusus* dari Sister '21. Bagi praktikan yang berhasil memenuhi ini, silahkan hubungi Duke (Line/Discord: pkmntrainerfrost) dan berikan tangkapan layar/foto yang seminimalnya berisi scoreboard serta jam.

#### IV. Langkah Umum Pengerjaan

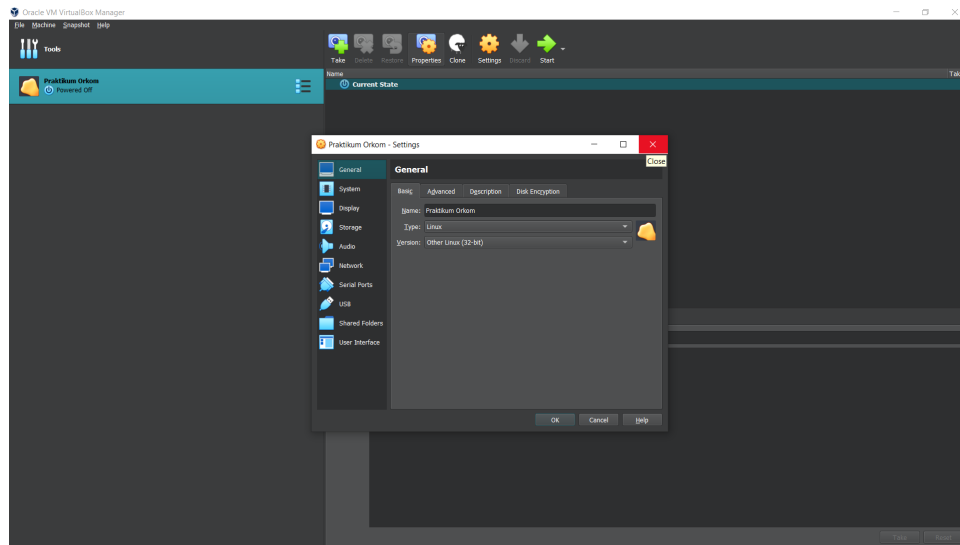
0. Bagi yang menggunakan ARM, silahkan ikuti [guide ini](#) bagian instalasi manual; jika tetap tidak bisa silahkan hubungi Duke (Line/Discord pkmntrainerfrost). Bagi yang sampai saat ini tidak dapat menggunakan VM (ARM atau tidak), berhubung waktu praktikum tinggal sedikit, disarankan untuk meminjam device temanya untuk melakukan pengecekan dan submisi. **Asisten memohon maaf apabila instruksi terkait ARM kurang jelas/datang terlambat, untuk praktikum selanjutnya akan dibuat guide yang baru yang akan diberikan di awal.**
1. Unduh ISO untuk praktikum pada [pranala ini](#). Terdapat file zip yang mengandung iso yang sama, bagi yang memiliki masalah dengan antivirus. Jika antivirus masih melunjak, matikan antivirus atau berikan pengecualian untuk file tersebut. (beneran guys nggak ada virusnya asli 100% ga bohong gak akan hilang data kalian plis percaya sama kita plisplisplis >///< 🙏🙏, jangan lupa dinyalain lg habis download)
2. Lakukan instalasi virtual machine apapun. Kami merekomendasikan VirtualBox.  
**Catatan:** Instruksi yang diberikan pada dokumen ini merupakan instruksi untuk VirtualBox. Penggunaan pada virtual machine lain dapat disesuaikan dengan instruksi ini.  
VirtualBox dapat diunduh pada [pranala ini](#).
3. Buatlah sebuah virtual machine. Jika menggunakan VirtualBox, gunakan spesifikasi seperti berikut  
**Type** : Linux

**Version** : Other Linux (32-bit)

**RAM size** : 256 MB

**Hard disk** : Do not add a virtual hard disk (abaikan jika ada peringatan)

Gambar 1 adalah contoh tampilan setelah virtual machine dibuat.



**Gambar 1.** Tampilan VirtualBox

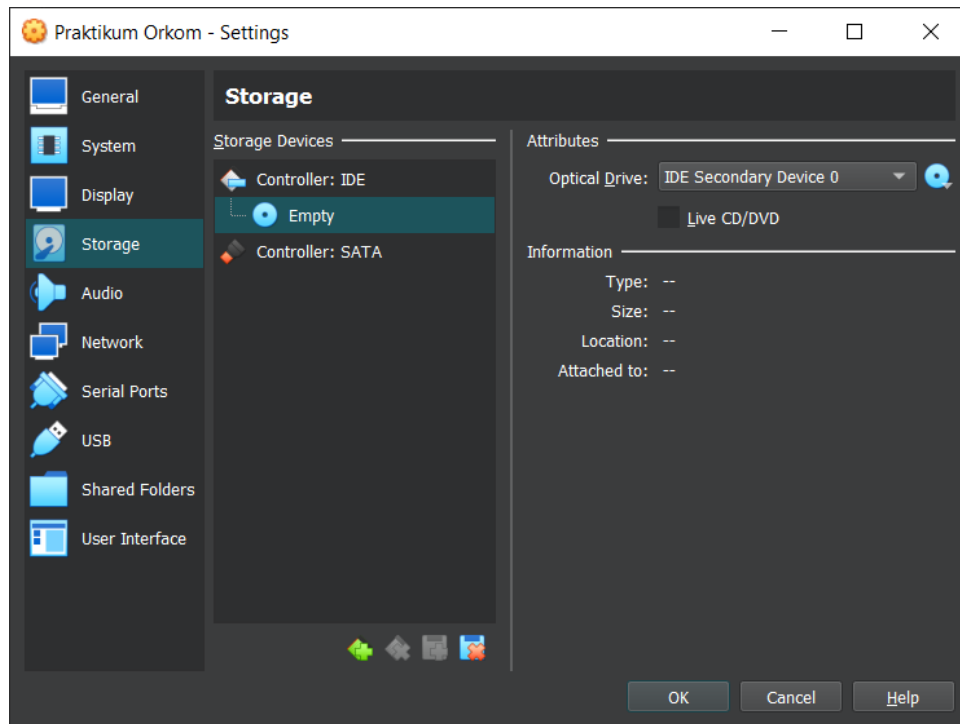
- Setelah itu, jalankan virtual machine yang telah kalian buat. Virtual machine akan bertanya mengenai boot medium. Pilih ISO pengerjaan praktikum sebagai boot medium dari virtual machine, kemudian jalankan virtual machine.



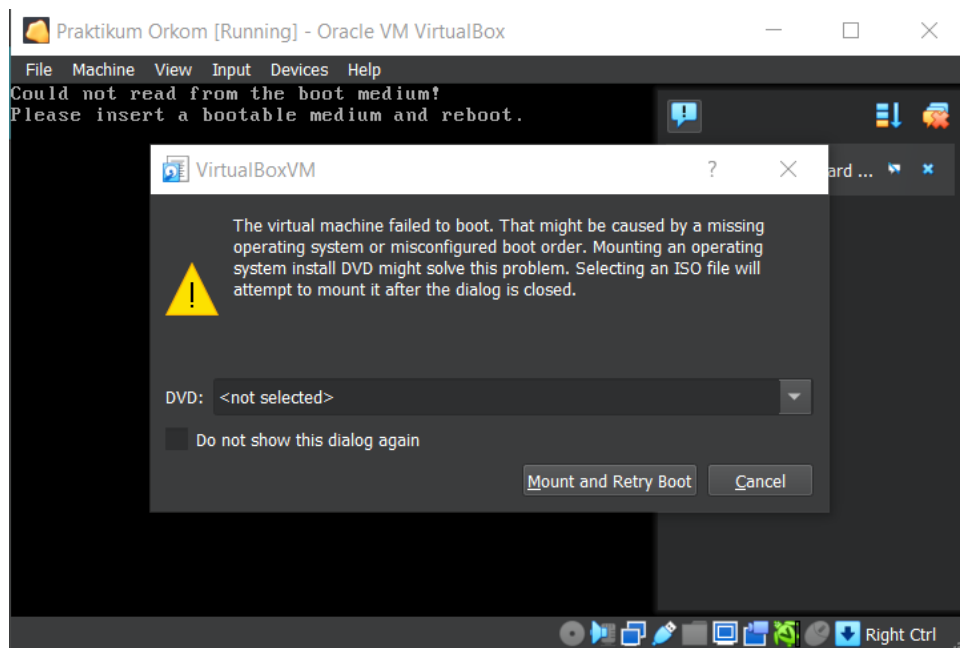
**Gambar 2.** Tampilan Virtual Machine



Jika menggunakan VirtualBox, pilih ISO melalui menu Settings->Storage->"Controller: IDE"->Empty->"Ikon CD", kemudian jalankan virtual machine. Pemilihan ISO juga dapat dilakukan pada saat pertama menjalankan virtual machine.

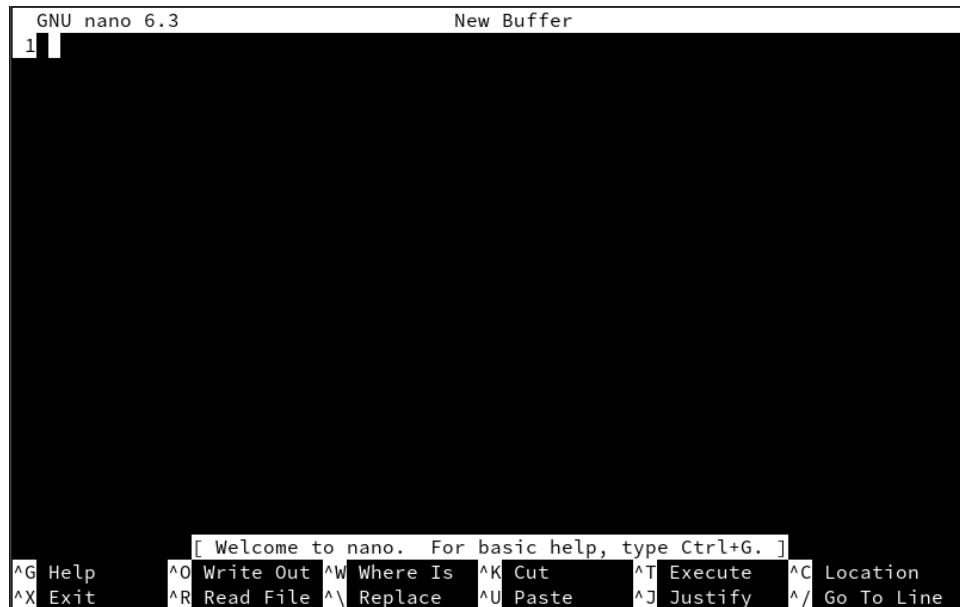


**Gambar 3.** Pemilihan ISO melalui Pengaturan Storage Device (VirtualBox)



**Gambar 4.** Pemilihan ISO melalui Boot Pertama VM (VirtualBox)

5. Masukkan NIM, lalu tekan ENTER. Jika kalian melakukan kesalahan dalam memasukkan NIM, tutup window virtual machine, matikan virtual machine (pada VirtualBox pilih "Power off the machine"). Lalu, jalankan ulang virtual machine.
6. Langkah-langkah persiapan telah selesai. Kalian dapat mengerjakan soal yang diberikan dengan menggunakan [teks editor nano](#) dengan perintah `nano <nama file>`



**Gambar 5.** Teks editor nano

7. Jika dilihat di bagian bawah, terdapat simbol `^` yang diikuti dengan huruf beserta nama perintah. Artinya untuk menjalankan suatu perintah, huruf yang bersangkutan harus ditekan bersamaan dengan CTRL. Contohnya :  
Tekan CTRL + O, lalu ENTER, untuk menyimpan file.  
Tekan CTRL + X untuk keluar dari [teks editor nano](#).  
**Perhatian! File tidak tersimpan di disk sehingga akan hilang saat VM dimatikan!** Semua file pada virtual machine ini disimpan dalam RAM, termasuk file pekerjaan kalian. Namun, isi RAM virtual machine kalian dapat disimpan dengan cara membuat [snapshot](#).
8. Soal dari praktikum ini berada pada file bernama **bits.c**
9. Pengumpulan jawaban praktikum membutuhkan password. Password ini tersedia pada executable file bernama **password**. Gunakan [gdb](#) untuk mencari password di dalam file tersebut. (**Note: setiap NIM memiliki password yang berbeda**)

10. Berikut ini adalah beberapa perintah yang dapat kalian masukkan pada terminal virtual machine

- Perintah `check` untuk memeriksa jawaban.
- Perintah `submit` untuk memeriksa dan mengirimkan jawaban kepada server.
- Perintah `setnickname` untuk mengubah nama yang akan tampil di scoreboard.
- Perintah `setpassword` untuk memasukkan password (selengkapnya terkait password di Bagian V - Pengumpulan dan Deliverables)

11. Jika kalian ingin melanjutkan pekerjaan kalian nanti, maka tutup window virtual machine, lalu pilih "Save the machine state" (VirtualBox). Setelah itu, jalankan virtual machine kembali untuk melanjutkan.

## V. Pengumpulan dan Deliverables

1. Untuk melakukan pengumpulan, masukkan password terlebih dahulu menggunakan perintah `setpassword`. Dapatkan password dengan cara membuka file `password` menggunakan gdb - cara penggunaan gdb terlampir di referensi. Berikut merupakan *clue* terkait cara mencari password:

- a. Cek (dan *disassemble*) fungsi-fungsi yang ada.
- b. Cek variabel-variabel yang ada.
- c. Password terdapat pada salah satu variabel. Variabel tersebut diproses menggunakan salah satu fungsi.
- d. File password dapat dijalankan untuk mengecek kebenaran password.

2. Pekerjaan kalian pada file `bits.c` akan otomatis dikumpulkan ketika kalian menjalankan perintah `submit`.

**Penting:** Nilai yang tercatat adalah nilai dari file yang terakhir kali di `submit`.

3. Kalian dapat melihat nilai kalian yang tercatat pada server melalui scoreboard di <http://sister21.tech/>

4. **Mulai** Kamis, 28 September 2023 pukul 18:00 WIB waktu server.

**Deadline** Kamis, 5 Oktober 2022 pukul 23.59 WIB waktu server. Server grading akan ditutup setelah waktu tersebut.

5. **DILARANG KERAS** melakukan serangan Denial of Service (DoS) ataupun serangan lain terhadap server.

6. Dilarang melakukan submisi dengan kode orang lain maupun men-submit dengan NIM orang lain. Kami memiliki rekap semua submisi yang anda lakukan sehingga segala bentuk kecurangan akan ditindak lanjuti. Kami akan melakukan pemeriksaan terhadap file **bits.c** yang kalian kirim.
7. Kami akan menindaklanjuti segala bentuk kecurangan yang terstruktur, masif, dan sistematis.
8. Diharapkan untuk mengerjakan sendiri terlebih dahulu sebelum mencari sumber inspirasi lain (Google, ~~mesin nainai DX~~, maupun teman Anda yang sudah bisa). Percayalah jika menemukan sendiri jawabannya anda akan mendapatkan *sense of pride and accomplishment*.
9. Jika ada pertanyaan atau masalah pengerjaan (atau ada yang tidak sengaja melakukan submisi diluar format yang diberikan) harap segera mengirimkan pertanyaan ke sheets QNA: <https://bit.ly/QNAOrkomArsikom2023>.

## VI. Referensi

1. **Nano Text Editor** -  
<https://www.howtogeek.com/42980/the-beginners-guide-to-nano-the-linux-command-line-text-editor/>
2. **VirtualBox Snapshots** -  
<https://howtogeek.com/150258/how-to-save-time-by-using-snapshots-in-virtualbox/>
3. **GDB** -  
<https://docs.google.com/document/d/1u-wryLpYS2CyWxOxGI5VQEIsYemFzNL12FRb8EB6yy4/edit?usp=sharing>



シスターは面白い ~ Rumei (Tahu Project)

~ satu soal bitwise 250 operator digantung excavator goreng utuh! ~

Johann

~ semoga bisa tidur nyenyak ~

Saddam

~ ajarin bitwise dong puuuh, sepuuuuh - aku mah masih pemula ~

黄

~ yang mau beli akun genshin, pc, tapi bukan pc gw >///< < ~

Awe

~ LOVE FOREVER ~

Duke

~ selamat datang di permulaan ~

Willy

~ dulu operator aritmetik bang tak ganas, sekarang dah tukar bitwise bang MANTAP ~

Toper

~ bitwise left me broken 🐱🐱🐱🐱🐱 ~

yujin

~ furina ayo pulang ~

barkod