

## Intisari

Dalam kurun waktu beberapa tahun terakhir muncul banyak piranti elektronik yang memerlukan pengendalian menggunakan *single board computer*. Pengendalian tersebut dilakukan di dalam *operating system* (OS). Dari kumpulan piranti-piranti tersebut terdapat beberapa piranti yang memerlukan *response time* yang cepat, misal *fingerprint scanner* memerlukan *response time* cepat supaya data gambar yang diterima tidak *corrupted*. *Response time* ini bisa dimanipulasi dalam pemrograman *device driver*.

Umumnya *device driver* mengakses perangkat keras pada level *user space*. Konfigurasi ini menimbulkan masalah, yakni durasi *response time* akan sangat bergantung pada ada atau tidaknya antrian proses. Oleh karena itu dalam Tugas Akhir ini dibuatkan sebuah solusi, yakni dengan memindahkan sebagian fungsi akses perangkat keras dari *user space* ke dalam *kernel space*.

Eksperimen dilakukan dengan cara membuat sebuah *Loadable Kernel Module* (LKM) yang berisi fungsi-fungsi akses perangkat keras dalam *kernel space* pada OS Raspbian dan sebuah program di *user space* yang menggunakan fungsi-fungsi dalam LKM tersebut melalui metode *character device*. Perangkat keras yang dikendalikan dalam pengujian ini adalah sebuah piranti termodifikasi yang belum memiliki *device driver*. *Response time* untuk metode yang diajukan kemudian diukur dengan menggunakan osiloskop dan hasil yang didapat dibandingkan dengan *response time* dari metode *user space* konvensional.

Dari eksperimen didapati bahwa metode *user space* konvensional menghasilkan rerata *response time* 99,97 ms dengan simpangan baku 50,67 ms, sedangkan metode yang diajukan menghasilkan rerata *response time* 31,17 ms dengan simpangan baku 18,19 ms. Metode yang diajukan terhitung 68.82% lebih cepat daripada metode *user space* konvensional. Menggunakan data hasil eksperimen tersebut terbukti bahwa metode yang diajukan lebih cepat dan lebih *reliable* dibandingkan dengan metode *user space* konvensional.

**Kata kunci :** *kernel, kernel module, device driver, response time, piranti termodifikasi*

## Abstract

*Nowadays, there are many emerging electronic devices which need control using single board computers. The control is conducted in the operating system (OS). Out of those many emerging devices there are some which need fast response time, for example a fingerprint scanner would need a fast response time so that the received image data would not be corrupted. Response time can be manipulated with device driver programming.*

*Generally, device driver would gain access to hardware at user space. This configuration will raise an issue, in which the response time will be heavily dependent on whether there are any process queue or not. Therefore a solution method is proposed in this final project, which is transferring some parts of hardware access functions from the user space into kernel space.*

*An experiment was conducted by creating a Loadable Kernel Module (LKM) which contains hardware access functions in kernel space on Raspbian OS and a user space program that utilizes functions in the LKM through character device method. The controlled hardware in this experiment is a customized device that does not have a device driver yet. The response time from the proposed method was then measured using oscilloscope and compared to the ones from conventional user space method.*

*From the experiment it was found that the conventional user space method resulted in average response time of 99.97 ms with standard deviation of 50.67 ms, while the proposed method resulted in average response time of 31.17 ms with standard deviation of 18.19 ms. The proposed method is calculated as 68.82% faster than conventional user space method. By using these results, it can be concluded that the proposed method is proven to be faster and more reliable than conventional user space method.*

**Keywords :** *kernel, kernel module, device driver, response time, customized device*