

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Di dalam penelitian ini terdapat beberapa tinjauan pustaka untuk mendukung penelitian. Tinjauan pustaka pada penelitian ini berfungsi sebagai penelusuran dari hasil-hasil penelitian terdahulu yang sudah pernah diteliti oleh suatu individu untuk mengevaluasi dan mendukung penelitian-penelitian yang sedang kita lakukan saat ini sebagai acuan dan ulasan sehingga pada penulisan dapat memberikan penekanan dan penegasan terhadap suatu teori penelitian yang hendak dikerjakan. Penulis sudah menghimpunkan beberapa literatur dari tinjauan pustaka yang bisa dilihat dari tabel 2.1

**Tabel 2.1 Tinjauan Pustaka**

NO	Nama Penulis	Tahun Penelitian	Judul Penelitian
1	Atik Charisma & Enggar Prahargya	2020	“Implementasi purwarupa Sistem Monitoring Suhu dan Pengiriman Data untuk Muatan Roket Berbasis Web”
2	M. Mudaris & Satria Gunawan Zain	2020	“Implementasi Sensor <i>Inertial Measurement Unit</i> (IMU) untuk Monitoring Perilaku Roket”
3	Eka Nanda Sugianto, Wijaya Kurniawan dan Dahniel Syauqy	2018	“Implementasi Sistem Operasi Real-Time pada Arduino Nano dengan media Komunikasi NRF24L01 Untuk Pengukuran Suhu,

			Kelembaban, dan Intensitas Cahaya“
4	M. Sandy Anshori Sabriansyah Risqika Akbar Dan Rizal Maulana	2018	“Implementasi Sistem Sensor Dan Aktuator <i>Real Time</i> Pada Tanaman Jamur”
5	Hendriawan Dwi Saputro, Rizal Maulana, Mochammad Hannats Hanafi Ichsan	2018	“Implementasi <i>Real Time</i> Pada Penggerak Robot <i>Quandruped</i> Menggunakan Multisensor Dan RTOS”

### 2.1.1 Tinjauan Pustaka 1

Penelitian ini ditulis oleh (Charisma, Prahargyan and Iskandar, 2018) dengan judul “Implementasi purwarupa Sistem Monitoring Suhu dan Pengiriman Data untuk Muatan Roket Berbasis Web” pada penelitian ini sensor yang digunakan adalah sensor suhu, kelembaban, kecepatan angin, ketinggian dan GPS. Penelitian ini akan menggunakan mikrokontroler berbasis web, namun penelitian ini hanya berfokus pada sistem monitoring yang dapat diakses secara daring. Metode pengumpulan data yang akan dipakai untuk penelitian tersebut yaitu menggunakan metode observasi, studi pustaka serta perancangan dan pengujian. Metode observasi ini adalah suatu metode pengumpulan data dan melakukan pengamatan secara nyata atau langsung. Dari beberapa sensor yang telah terpasang akan memproses input, selanjutnya akan membaca nilai yang didapatkan dari anemometer sensor dan node mcu yang akan membaca nilai dari sensor BMP280 dan akan membaca Ublox GPS. Setelah arduino mendapat nilai sensor dari anemometer, nilai tersebut selanjutnya akan dikirim ke NodeMCU yang terdiri dari variabel suhu, kelembapan, ketinggian. Dan selanjutnya variabel yang telah diterima oleh web server akan

diproses untuk akan ditampilkan menggunakan pc, laptop dan smartpone. Langkah awal dari penelitian ini adalah dengan menginisialisasikan library serta konfigurasi nya dari sensor BMP280, kecepatan angin dan GPS. Setelah itu sensor membaca nilai sensor, apabila nilai sensor terbaca maka akan langsung keserial monitor dan apabila tidak maka akan melakukan pengecekan perangkat keras. Jika nilai sensor telah dapat dibaca pada serial monitor maka data dari hasil pembacaan akan disimpan kedalam database untuk diolah. Data yang telah diolah nantinya akan ditampilkan pada web browser yaitu pada suhu, kelembaban, kecepatan angin, ketinggian serta lokasi.

### **2.1.2 Tinjauan Pustaka 2**

Penelitian ini dilakukan oleh (Mudarris and Zain, 2020) dengan judul yaitu “Implementasi Sensor *Inertial Measurment Unit* (IMU) untuk Monitoring Perilaku Roket” Pada penelitian ini sensor yang akan dipakai untuk mengukur watak roket. Perilaku roket yang bisa diukur adalah kecepatan, ketinggian, posisi, sedangkan untuk mengukur sikap roket akan menggunakan sensor gyroscope. Sistem *monitoring* yang akan menggunakan *real time* kepada muatan untuk roket yang diuji menjadi suatu yang harus ada dalam pengetahuan performa dari roket nanti yang akan dikembangkan. Sistem dari perangkat ini berlaku dari perangkat sensor dan juga perangkat telemetri radio. Nantinya data dari sensor akan dikirimkan dari roket yang sedang melayang via telemetri radio. Dari pusat kendali, data sensor nantinya akan diamati dalam bentuk grafis dan array. Sensor yang telah dipasang kan pada muatan roket diharapkan nantinya bisa memberikan laporan terkait arah dan posisi roket. Untuk bisa mengukur kecepatan, percepatan

dan juga arah dari roket akan membutuhkan model sensor IMU yang nanti didalamnya sudah terpadu sensor *gyroscope & accelerometer*.

### 2.1.3 Tinjauan Pustaka 3

Penelitian ini ditulis oleh (Sugianto, Kurniawan and Syauqy, 2019) dengan judul “Implementasi Sistem Operasi *Real-Time* pada Arduino Nano dengan media Komunikasi NRF24L01 Untuk Pengukuran Suhu, Kelembaban, dan Intensitas Cahaya“. Penelitian ini akan dilakukan dengan mengimplementasi *metode preemptive priority based scheduling* pada modul arduino nano. Metode ini akan digunakan untuk metode penjadwalan pelaksanaan tugas berdasar prioritas paling tinggi. Metode ini juga akan digunakan untuk mengatasi problem terjadinya penundaan tugas pengiriman data yang akan dikirimkan. Pengimplementasian metode ini akan menggunakan *os real-time* dengan membuat perangkat *hardware* seperti yang dibutuhkan supaya nantinya dapat bekerja secara *multitasking*. Bahan yang akan digunakan dalam penelitian ini antara lain yaitu Modul Arduino Nano, Sensor LDR, Sensor DHT11, dan juga modul komunikasi *wireless*. Nantinya sensor dari DHT11 dan LDR pada *node client* dapat mengambil data secara bersamaan setelah itu data dari sensor akan dikirimkan kepada *node base* yang memiliki fungsi sebagai penerima data dari sensor.

### 2.1.4 Tinjauan Pustaka 4

Penelitian ini ditulis oleh (Anshori, Akbar and Maulana, 2018) dengan judul “Implementasi Sistem Sensor Dan Aktuator *Real Time* Pada Tanaman Jamur“. Penelitian dilaksanakan untuk memudahkan untuk suatu sektor budidaya tanaman jamur. Didalam penelitian ini adalah untuk menerapkan RTOS untuk memonitoring tanaman jamur serta dengan penyiraman otomatis dengan menggunakan sistem

operasi *FreeRTOS*. Nantinya *FreeRTOS* ini akan menggunakan tiga tugas ke semua *node* sensor dan aktuator yang akan berfungsi untuk memantau suhu dan kelembaban, memantau debit air, penyalan pompa air otomatis, dan mengirim data ke *node gateway*. Pada *FreeRTOS* yang akan menggunakan 3 tugas atau task yang terdiri dari *Taskdht*, *Taskwater*, *Taskend*. Semua task akan diber prioritas yang *balance* yaitu prioritas 1. Semua *task* akan eksekusi secara berurutan dari task *Taskdht*, *Taskwater*, lalu *Taskend*. Lalu *FreeRTOS* ini akan berjalan dengan terus-menerus sampai dengan tidak ada lagi source yang masuk ke Arduino Uno.

### 2.1.5 Tinjauan Pustaka 5

Penelitian ini ditulis oleh (Saputro *et al.*, 2018) dengan judul penelitian “Implementasi *Real Time* Pada Penggerak Robot *Quadruped* Menggunakan Multisensor Dan RTOS”. Penerapan RTOS didalam penelitian kali ini bermaksud untuk mengimplementasikan RTOS pada sebuah sistem tertanam yaitu pada penggerak robot *quadruped*. Os RTOS yang akan dipakai pada penelitian ini adalah *FreeRTOS* dikarenakan os pada sistem tersebut telah tersedia oleh mikrokontroler Arduino. Nantinya RTOS pada sistem akan implementasikan pada pengambilan data sensor dan juga proses olah data dalam pengambilan langkah keputusan. Robot didalam penelitian ini akan mempunyai sistem penggerak total 4 kaki yang disebut *quadruped* yang tertata pada bagian komponen HC-SR04, Arduino Mega 2560, OpenCM485, OpenCM9.04 , dan servo AX-12. Penelitian ini nantinya akan mempunyai beberapa pengujian yaitu, pengujian sensor HC-SR04, pengujian waktu eksekusi *task* dan prioritas, pengujian ketepatan robot bergerak, dan pengujian perbandingan waktu kerja sistem.

Berdasarkan dari penelitian-penelitian diatas, dapat disimpulkan oleh saya sebagai penulis yaitu pertama , sistem implementasi yang dilakukan dari penelitian sebelumnya relatif berbeda-beda dari setiap penulis dimana sistem, metode, dan alat sensor yang digunakan pun bervariasi ada yang hanya fokus kepada sistem monitoring ada juga dengan sistem monitoring serta pengerjaan task. Kedua, sektor fokus dari penelitian nya pun berbeda ada yang dibidang sektor budidaya jamur, sistem penggerak robot dan juga pada muatan roket. Ketiga, sensor yang digunakan dalam penelitian pun terdapat perbedaan ada yang hanya menggunakan 1 sensor ada juga yang menggunakan beberapa sensor sekaligus. Perbedaan dengan penelitian penulis yaitu terdapat pada fokus sensor yang akan diolah, setelah itu metode dari sistem yang digunakan serta pada sistem dibuat oleh penulis implementasi rtos untuk 2 tugas yang spesifik diantaranya dalam pembacaan banyak sensor serta pengiriman data sensor tersebut yang dilakukan secara *realtime*.

## **1.2. Landasan Teori**

### **2.2.1 Raket**

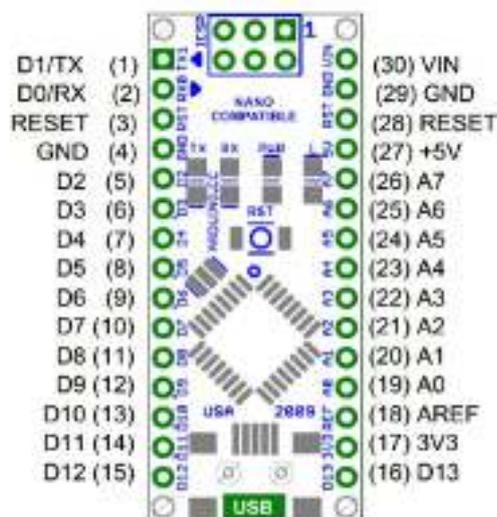
Raket adalah sebuah wahana peluncur yang dipergunakan untuk membawa sebuah beban atau muatan nya (*payload*) untuk sampai kepada tujuan yang ditentukan untuk sebuah misi ilmiah maupun sebuah senjata (Propelan, 2005). Muatan sendiri ialah sebuah substansi yang dibawa didalam roket, bisa itu berupa sebagai muatan roket pengindera dinamik roket itu sendiri ataupun sebagai misi tertentu, misalkan muatan sensor meteorologi (Panduan, 2021).

## 2.2.2 Arduino

Arduino merupakan sebuah *platform* fisik *computation open source* yang berbasis oleh rangkaian input dan output sederhana dalam suatu lingkungan pengembang yang mengimplementasi suatu bahasa pemrograman (Sokop *et al.*, 2016)

### 2.2.2.1 Arduino NANO

Arduino Nano adalah sebuah *board* yang pengembangan mikrokontroler yang dalam sistem nya menggunakan chip Atmeg328P, mikrokontroler ini berfungsi pada input tegangan 5 sampai 7 volt dan juga terdapat *flash* memori 32 KB dan juga dapat mampu berjalan pada *clock* 16 Mhz. Pada mikrikontroler ini memiliki 8 buah pin analog yaitu A1 sampai dengan pin A8, dan seluruh pin pada *analog* terhubung pada ADC pada *internal* mikrokontroler. Pada mikrokontroler ini juga terdapat 2 buah jalur pin SCL dan juga SDA masing-masing terletak pada pin A4 dan A5 (Kurniawan and Rivai, 2018)



Gambar 2.1 Pinout Arduino Nano

Sumber: (Kurniawan and Rivai, 2018)

### 2.2.3 Sensor

Sensor adalah komponen kelistrikan yang berfungsi sebagai *input* perangkat. Tidak semua *input* secara *eksplisit* merupakan sensor, tetapi hampir semua input menggunakan sensor. Secara lebih abstrak, Anda dapat merumuskan sensor sebagai komponen untuk mengukur stimulus yang berada di luar sistem (lingkungannya). Hasil data didasarkan pada pengukuran (Karvinen and Karvinen, 2015)

#### 2.2.3.1 Sensor IMU

Menurut (Suryanti, 2017) Sensor IMU (*Inertial Measurement Unit*) merupakan suatu komponen elektronik yang akan memanfaatkan *read* dari suatu sensor *accelerometer* dan sensor *gyroscope* untuk memperoleh suatu perkiraan nilai posisi relatif, akselerasi, dan juga kecepatan dari putaran motor. Sedangkan menurut (Asnada, 2020) Sensor IMU adalah sebuah sensor yang mempunyai *3axis accelerometer* dan *3axis gyroscope*. Sensor *accelerometer* berfungsi untuk mengukur sudut dengan baik saat diam dan sensor *gyro* dapat digunakan mengukur saat benda bergerak.

#### 2.2.4 NRF24L01

Menurut (Af'idah, Rochim and Widiyanto, 2014) Merupakan sebuah komponen komunikasi yang akan menggunakan pemanfaatan gelombang RF 2,4GHz ISM. Pada komponen ini akan menggunakan *interface Serial Peripheral Interface (SPI)* untuk komunikasi. NRF24L01 akan memadukan RF pengumpul, pengirim lengkap 2.4ghZ, dan akselerator berbentuk *Enhanced Shockburst* yang selaras dengan *interface SPI* yang memiliki kecepatan tinggi. Komponen

NRF24L01 memiliki sebuah pin yaitu: VCC, GND, CE, CSN, MOSI, MISO, SCK, IRQ, dan memiliki sebuah fitur diantaranya:

1. Memiliki kecepatan send data 250kbps sampa 2Mbps
2. Jangkauan transfer data 100m ditempat yang terbuka
3. Dapat beroperasi pada ISM 2.4GHz
4. Pengerjaan paket data yang otomatis
5. Pengerjaan transaksi paket data yang otomatis



**Gambar 2.2** NRF24L01

Sumber: (Af'idah, Rochim and Widiyanto, 2014)

### 2.2.5 PCB

PCB (*Printed Circuit Board*) merupakan sebuah komponen yang berbentuk papan yang difungsikan untuk menghubungkan suatu komponen elektronika dengan memanfaatkan suatu jalur daya hantar (*konduktif*) yang terukirkan ke lapisan bahan tembaga yang terlanimasi pada media bentuk *nonkonduktif*. Pada pembuatan papan PCB ini pada awal nya dilakukan secara manual, dengan cara membuat cetakan desain pada media kertas *glossy*, selanjutnya pemindahan serbuk tinta yang berada dikertas ke papan PCB dengan menggunakan strika, selanjutnya yaitu dengan melarutkan kedalam suatu larutan FeCl<sub>1</sub> (feri klorida), dan selanjutnya

akan dibor sesuai pada jalur pin yang akan digunakan untuk menghubungkan komponen (Pranata and Nuryadi, 2019).



**Gambar 2.3** PCB Board

(Sumber: (Brillianta Akbar, 2017))

### 2.2.6 RTOS

Pada biasanya jika task yang harus diselesaikan oleh suatu komponen elektronik dengan jumlah total yang banyak, penyelesaian yang efisien untuk menanggulangi kasus ini adalah dengan mengembangkan kinerja dari sebuah chip dari mikroontroler itu sendiri dengan memakai RTOS (*Real Time Operating System*). RTOS ini adalah sebuah *operating system* yang dipakai disistem tertanam. Dengan memakai RTOS suatu mikrokontroler dapat dioperasikan secara *multitasking* yang artinya dapat melakukan banyak suatu task dalam kurung satu waktu secara bersamaan (Gumelar, Syauqy and Akbar, 2018)

### 2.2.6.1 Karakteristik RTOS

Pada sistem operasi RTOS ini memiliki beberapa karakteristik yang ada pada sistem ini diantaranya yaitu *reliability*, *predictability*, *performance* (Wisnu Jatmiko, 2015)

#### a. *Reliability*

Sebuah sistem tertanam akan dibuat beroperasi secara terus-menerus dengan waktu yang relatif lama tanpa adanya intrusi *humanity*. Maka dari hal tersebut sebuah sistem tertanam diharuskan memiliki sistem yang *reliable*. RTOS harus bersifat *reliability* agar dapat menjaga suatu sistem tersebut dapat dioperasikan dengan waktu yang lama.

#### b. *Predictability*

Karakteristik ini akan menjamin RTOS setiap *real-time system* diharuskan untuk mengerjakan suatu tugas pada waktu yang telah dijadwalkan sebelumnya. Karakter dari RTOS itu pun dapat antisipasi atau dengan sebutan lain *call system* di RTOS akan tampak dalam waktu yang dapat diketahui. Pada sistem RTOS diharapkan mempunyai variasi waktu respon *call system* yang terbilang kecil.

#### c. *Performance*

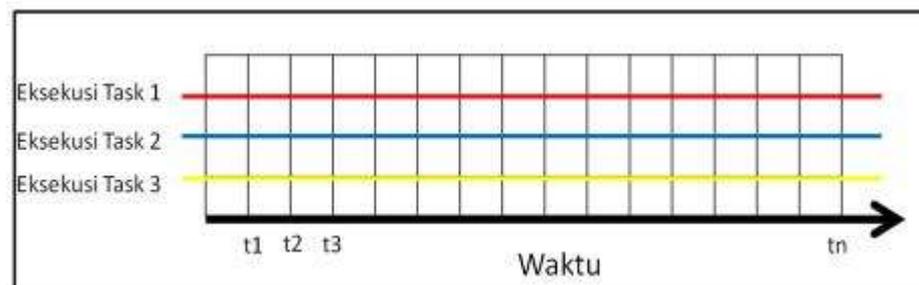
Karakteristik yang dipunya dari sistem RTOS ini iyalah dapat memproses dengan waktu yang relatif cepat. Dalam RTOS performa pada sistem pun dapat diukur memakai *size throuhput*, yang berarti sebuah *rate* pada sistem bisa menghasilkan suatu output dari input yang diterima.

### 2.2.6.2 Konsep Dasar

Konsep dasar pada RTOS mempunyai beberapa konsep yang diantaranya yaitu *multitasking*, *scheduling*, dan *real time application*. (Wisnu Jatmiko, 2015)

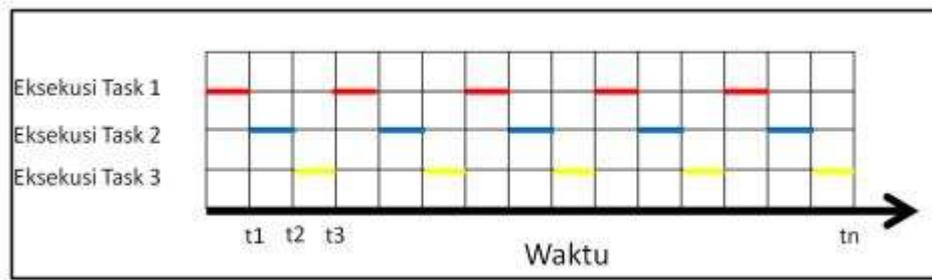
#### a. *Multitasking*

Pada sistem operasi yang mempunyai suatu komponen utama untuk mengelola adanya *multitasking*. *Multitasking* adalah suatu kemampuan suatu operasi sistem untuk memproses banyak suatu *task* atau pun *thread* dalam waktu tertentu. Konsepsi dari *multitasking* dari pada operasi sistem akan membantu kemudahan desain implementasi yang rumit.



**Gambar 2.4** Konsep Multitasking

Sumber: (Wisnu Jatmiko, 2015)



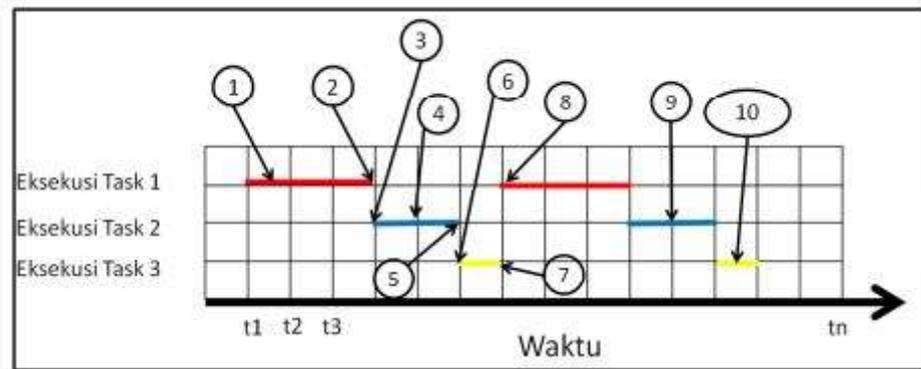
**Gambar 2.5** Konsep Konkurensi

Sumber: (Wisnu Jatmiko, 2015)

Pada gambar 2.4 mengartikan bagaimana suatu konsepsi *multitasking* ini bekerja. Tiga tugas yang ada pada konsep tersebut akan dilaksanakan dan juga berjalan secara serentak dan dalam waktu eksekusi yang sama. Sedangkan pada gambar 2.5 adalah suatu *task* yang dimana eksekusi nya dilakukan lebih dari satu tugas, tetapi task tersebut akan dieksekusi secara bergantian dengan cara membagi waktu eksekusi sesuai dengan penjadwalan dalam rentang waktu tertentu (Wisnu Jatmiko, 2015).

### **b. Scheduling**

Konsep adalah bagian dari pembagian waktu pemrosesan eksekusi sebuah task yang tersedia. Sebelum melakukan suatu *scheduling* akan dibutuhkan sebuah *scheduler* yang fungsinya adalah untuk menetapkan suatu waktu proses eksekusi sebuah *task*, durasi *task*, waktu tunda sebuah *task* atau *suspend*, dan waktu diproses kembali atau *resume*.



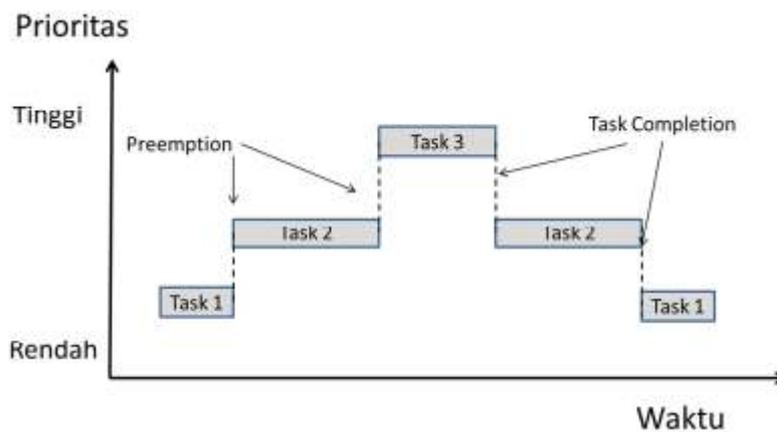
Gambar 2.6 Konsep Penjadwalan  
Sumber: (Wisnu Jatmiko, 2015)

Pada posisi 1 akan menentukan rentang waktu  $t_1$  sampai  $t_4$  *task 1* yang sedang dalam kondisi dieksekusi. Pada posisi akan menampilkan pada *time*  $t_4$ , dan juga *processor* akan membuat penundaan *task1*, dan selanjutnya akan memproses *task2*. Dan pada posisi 3, *task2* akan diproses. Pada posisi 4 waktu diwaktu  $t_1$  yang menggambarkan *task 2* sedang dalam pemrosesan. Pada posisi 5, *task2* sedang dalam keadaan ditunda akan dilanjutkan pada posisi 8 yang menggambarkan suatu *processor* akan menjalankan *task3*. Pada posisi 7, sebuah *task 3* akan mengalami penundaan dan akan dilanjutkan pada posisi 8 yang dimana *processor* akan kembali me-*resume task1* dan akan kembali lanjutan memproses *task1*. Pada posisi 9, sebuah *processor* akan me-*resume task2* dan akan kembali melanjutkan memproses *task2*. Dan pada posisi 10, *processor* pun akan me-*resume task3* dan kembali melanjutkan pemrosesan *task3* (Wisnu Jatmiko, 2015).

### 2.2.7 Algoritma Priority Scheduling

Algoritma ini adalah sebuah algoritma yang *basic* pada penjadwalan dan mendahulukan proses yang mempunyai prioritas paling tinggi. Pada setiap proses mempunyai prioritas yang berbeda-beda dan prioritas dari suatu proses tugas tersebut dapat ditetapkan melalui beberapa karakteristik diantaranya:

- a. Akses file
- b. Kebutuhan memori
- c. Batas waktu
- d. Tingkat kepentingan proses



**Gambar 2.7** Algoritma Priority-Based Scheduling  
Sumber: (Wisnu Jatmiko, 2015)

Pada *scheduling priority* ini bisa dijalankan dengan cara *preemptive* atau dengan cara *non-preemptive*. Jika pada preemptive suatu proses *task* yang baru tiba dan mempunyai prioritas yang lebih besar dari pada sebuah proses *task* yang sebelumnya sedang diproses, maka dari itu suatu proses yang sedang berjalan itu akan tertunda, dan processor akan dialihkan pada pemrosesan tugas yang baru tiba

tersebut. Sedangkan pada *non-preemptive*, suatu proses yang baru saja tiba tidak akan mempengaruhi proses *task* yang sedang dikerjakan atau dijalankan, tapi *task* tersebut hanya akan ditempatkan didepan baris antri. Adapun dalam *priority scheduling* ini jika suatu proses *task* mempunyai prioritas lebih rendah dapat kemungkinan untuk tidak dieksekusi apabila terdapat suatu *task* yang mempunyai prioritas lebih tinggi yang biasanya disebut dengan *blocking* (Mutasar, 2021).