

## DESAIN DAN IMPLEMENTASI SISTEM OPERASI WAKTU NYATA UNTUK SISTEM TERTANAM BIOMEDIS BERBASIS PENJADWALAN

### DESIGN AND IMPLEMENTATION OF SIMPLE REAL TIME OPERATING SYSTEM FOR EMBEDDED SYSTEM BASED SCHEDULING

Jafar Hifdzullisan<sup>1</sup>, Sony Sumaryo<sup>2</sup>, Achmad Rizal<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>hifdzullisan0@gmail.com, <sup>2</sup>sony.sumaryo@yahoo.co.id, <sup>3</sup>achmadrizal@telkomuniversity.ac.id

#### Abstrak

*Real time operating system* adalah sebuah sistem operasi yang dapat digunakan untuk *embedded system* agar dapat mengolah *task* yang diberikan kepada sistem dengan teratur sesuai dengan urutan *task* serta waktu yang dibutuhkan untuk melaksanakan *task* tersebut. *Embedded system* itu sendiri adalah sebuah sistem yang dirancang khusus untuk melaksanakan tujuan tertentu untuk meningkatkan fungsi sistem tersebut.

Pada tugas akhir ini akan dirancang sebuah *real time operating system* berbasis penjadwalan dengan tipe *soft real-time system* untuk mengoperasikan sebuah *embedded system* biomedis berupa elektrokardiogram. *Real time operating system* berfungsi untuk menerima *task* yang diberikan kepada sistem dan memperurutkan *task* tersebut sesuai dengan prioritas yang diberikan kepada *task* tersebut. *Task* yang diberikan kepada *real time operating system* adalah melakukan pengecekan terhadap elektroda yang digunakan sebagai sensor untuk pembacaan tegangan listrik pada jantung, membaca tegangan listrik pada jantung, dan menampilkan tegangan listrik tersebut melalui Graphical LCD 128x64.

**Kata Kunci :** *Real Time Operating System, embedded system, elektrokardiogram (EKG)*

#### Abstract

*The real time operating system is an operating system that can be used for embedded systems to be able to process tasks that are given to the system regularly and the tasks need to be executed in real time. The embedded system is a system specifically designed to carry out certain objectives to improve the function of the system.*

*In this final project, a scheduling-based real time operating system with a soft real-time system will be designed to operate an embedded biomedical system in the form of an electrocardiogram. The real time operating system functions to receive the task given to the system and sort the task according to the priority given to the task. The task given to the real time operating system is to check the electrodes used as sensors for reading the voltage on the heart, reading the voltage on the heart, and displaying the voltage through Graphical LCD 128x64.*

**Keywords:** *Real Time Operating System, embedded system, electrocardiogram (ECG)*

#### 1. Pendahuluan

Perkembangan ilmu pengetahuan dan teknologi dalam bidang teknik elektro saat ini telah mengalami pertumbuhan yang sangat pesat. Diantaranya penerapan sistem *microcontroller* pada berbagai bidang seperti otomotif, *robotic*, peralatan rumah tangga, serta peralatan medis yang membutuhkan kinerja yang baik. Sebuah sistem yang bertujuan untuk melakukan tugas tertentu disebut sebagai *embedded system*. *Embedded system* membutuhkan sebuah *controller* yang dapat mengeksekusi tugas yang diberikan dengan cepat dan tepat. Oleh karena itu, diperlukan sebuah *microcontroller* untuk mengeksekusi tugas dengan baik.

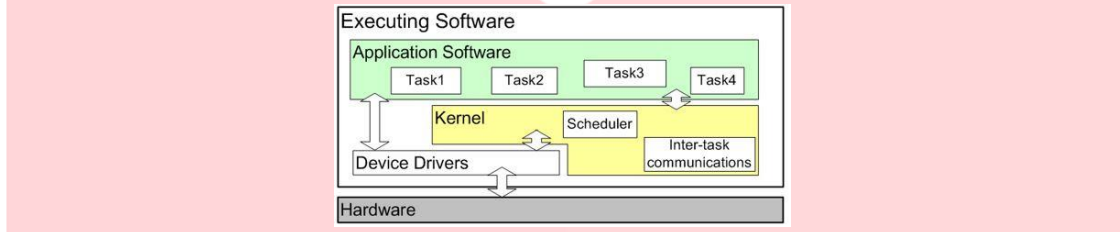
*Embedded system* membutuhkan sebuah sistem operasi untuk mengatur masukan serta keluaran dalam sistem tersebut. Selain untuk mengatur masukan serta keluaran sistem, sistem operasi juga diperlukan untuk menjamin waktu eksekusi dari masing-masing tugas. *Real-Time Operating System* (RTOS) adalah sebuah sistem operasi yang tepat untuk digunakan pada sebuah *embedded system*. RTOS adalah suatu program yang bertugas untuk menjadwalkan pekerjaan dengan sangat teratur, dan juga mengatur semua input dan output dari sistem dengan baik [1]. Hal inilah yang menjadi alasan bahwa RTOS adalah sistem operasi yang baik untuk digunakan pada sebuah *embedded system*.

Pada percobaan ini, dilakukan sebuah desain dan implementasi serta pengujian terhadap RTOS sederhana berbasis penjadwalan pada sistem biomedis berupa elektrokardiograf. *Embedded system* yang dirancang menggunakan sebuah sistem minimum dengan *microcontroller* ATMEGA32A, serta memiliki beberapa *device*. Yang pertama adalah *Single Lead Heart Rate Monitor* AD8232 sebagai modul yang berfungsi untuk mengukur aktifitas kelistrikan jantung. Kemudian sebuah *Graphical LCD* 128x64 ST7920 yang berfungsi untuk menampilkan keluaran dari sistem. RTOS yang digunakan pada sistem ini berfungsi untuk menjadwalkan semua *task* yang akan dieksekusi berdasarkan prioritas dari masing masing *task* tersebut.

## 2. Dasar Teori

### 2.1 Real Time Operating System

*Real-time operating system* (RTOS) adalah sebuah sistem operasi yang dibangun untuk dapat digunakan pada sebuah *embedded system* yang dapat mengerjakan beberapa *task* dalam waktu bersamaan serta memiliki ketepatan waktu yang sangat baik. RTOS mengeksekusi program-program dalam sebuah pola yang teratur. RTOS dijalankan oleh program otomatis yang disebut dengan *kernel*. Pada saat sistem operasi dinyalakan, maka *kernel* akan menyalakan terlebih dahulu kemudian *kernel* tersebut akan menyalakan *Real time operating system*.



Gambar II- 1 Cara Kerja RTOS

### 2.2 Elektrokardiograf

Elektrokardiograf adalah alat yang digunakan untuk membaca sinyal yang dihasilkan dari aktifitas kelistrikan otot jantung. Sewaktu impuls jantung melewati jantung, arus listrik akan menyebar kedalam jaringan disekeliling jantung dan sebagian kecil dari arus tersebut akan menyebar ke permukaan tubuh. Bila pada permukaan tubuh diletakkan elektroda-elektroda maka potensial listrik yang muncul bisa direkam. Rekaman tersebut disebut dengan elektrokardiogram atau lebih dikenal dengan sinyal EKG. EKG adalah grafik hasil catatan potensial listrik yang dihasilkan oleh denyut jantung.

### 2.3 Mikrokontroler

Mikrokontroler merupakan sebuah *chip* atau IC (*Integrated Circuit*) yang memiliki fungsi sebagai pengontrol sebuah rangkaian elektronik yang bisa menyimpan program di dalamnya. Mikrokontroler merupakan sistem mikroprosesor yang sudah lengkap di dalam satu chip. Sebuah mikrokontroler biasanya terdiri dari CPU (*Central Processing Unit*), memori, *input/output* dan unit pendukung seperti ADC (*Analog to Digital Converter*). Kelebihan dari sebuah mikrokontroler adalah memiliki kelengkapan dari semua sistem mikroprosesor yang dibuat seringkali mungkin di dalam satu *chip*.

### 2.4 Heart Rate Monitor AD8232

AD8232 adalah sebuah rangkaian elektronika yang sudah tersusun secara baik untuk dapat digunakan secara langsung dalam mengukur aktivitas kelistrikan jantung. Aktivitas kelistrikan jantung atau elektrokardiogram digunakan sebagai alat untuk membantu dalam hal mendiagnosis berbagai kondisi jantung. Rangkaian AD8232 berisi beberapa rangkaian penguat diferensial untuk menguatkan tegangan dari sekitar jantung. Untuk dapat membaca tegangan listrik yang dihasilkan oleh jantung maka diperlukan beberapa elektroda untuk menangkap sinyal listrik tersebut.

### 2.5 LCD Graphic 128x64

*Liquid Crystal Display* (LCD) graphic adalah suatu modul yang berfungsi sebagai alat untuk menampilkan gambar yang memiliki resolusi 128 kolom dan 64 baris. LCD ini memiliki warna dasar biru dan karakter berwarna putih dengan menggunakan *backlight*. LCD ini menggunakan IC berupa ST7920 dengan *supply* tegangan yang dibutuhkan sebesar 5 volt DC. Selain dapat menghasilkan karakter dan gambar, LCD ini juga dapat digunakan untuk menghasilkan gambar grafik.

## 3. Perancangan Sistem

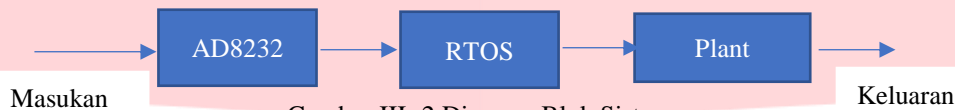
### 3.1. Desain Sistem

Pada tugas akhir ini akan dirancang sebuah *real-time operating system* sebagai sistem operasi untuk sebuah *embedded system* berupa elektrokardiograf. Rancangan umum sistem yang digunakan adalah *input*, *process* dan *output* seperti dapat dilihat



Gambar III- 1 Rancangan Umum Sistem

Dalam sistem ini *input* adalah nilai tegangan listrik yang ditangkap oleh elektroda di sekitar jantung. *Process* adalah tahap untuk menguatkan sinyal listrik yang sudah ditangkap oleh elektroda agar tegangannya dapat terbaca dengan baik oleh mikrokontroler. *Output* adalah proses untuk menampilkan sinyal listrik dari jantung yang sudah dikuatkan oleh penguat diferensial.

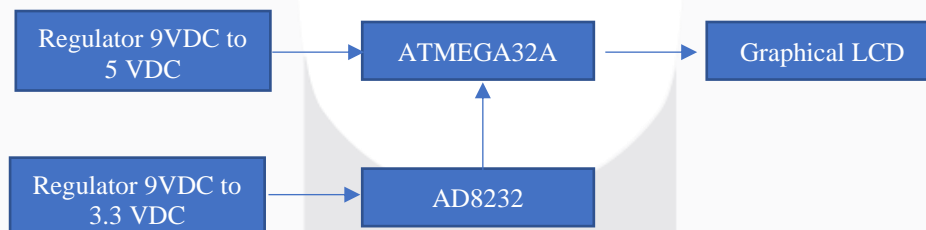


Gambar III- 2 Diagram Blok Sistem

Secara umum sistem dibagi menjadi beberapa bagian seperti pada gambar III.1, antara lain *input*, AD8232, *Real-time operating system*, dan *output*. Sistem ini merupakan sistem pembaca sinyal listrik yang berasal dari jantung. Sumber daya yang digunakan pada sistem ini adalah sebuah baterai 9 VDC yang akan digunakan untuk memenuhi kebutuhan daya dari ATmega 32A. ATmega 32A akan digunakan sebagai sistem minimum untuk mengolah data. Elektroda berfungsi sebagai alat pembaca sinyal listrik. Pada saat jantung berkontraksi, jantung akan menyebarkan arus listrik disekitar jantung. Sebagian dari arus listrik tersebut akan menyebar ke permukaan kulit disekitar jantung. Namun arus listrik yang tersebar di permukaan kulit tersebut sangat kecil. arus listrik. Oleh karena itu diperlukan sebuah penguat diferensial untuk menguatkan arus tersebut. Pada alat ini digunakan sebuah modul AD8232 sebagai penguat diferensial. Penguat diferensial diperlukan untuk dapat membedakan sinyal listrik yang asli dengan *noise* yang akan dibaca oleh elektroda dan juga berfungsi untuk menguatkan tegangan yang dibaca oleh elektroda. Setelah arus listrik dari elektroda dikuatkan, arus tersebut akan diteruskan ke *Real Time Operating System* untuk diolah.

### 3.2. Perancangan Perangkat Keras

Desain perangkat keras pada elektrokardiograf ini meliputi perancangan rangkaian sistem minimum dengan menggunakan ATMEGA 32A serta menghubungkannya dengan beberapa komponen tambahan seperti *power supply*, modul AD8232, serta *graphical LCD*. Berikut ini akan dijelaskan mengenai masing-masing perancangan tersebut.

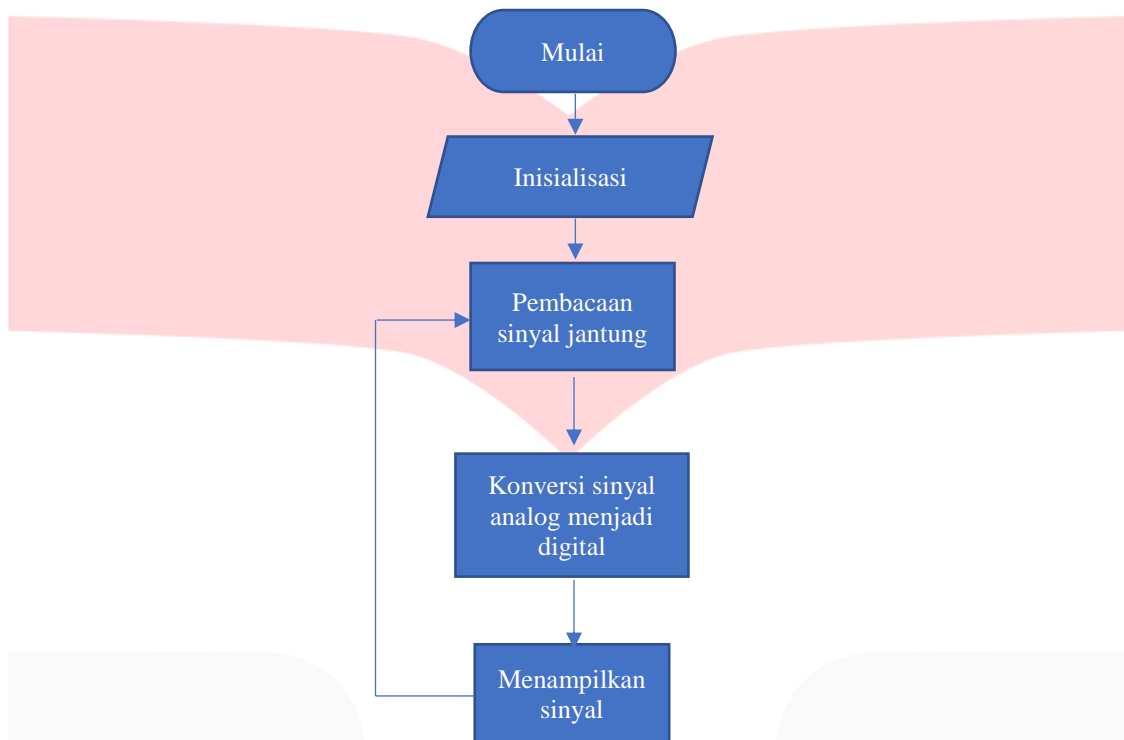


Gambar III- 3 Rancangan Desain Perangkat keras

Regulator 9VDC to 5VDC digunakan sebagai catu daya untuk mikrokontroler serta *graphical LCD* sedangkan regulator 9VDC to 3.3VDC digunakan untuk catu daya modul AD8232. Modul AD8232 berfungsi sebagai rangkaian penguat diferensial serta filter sinyal listrik yang berasal dari jantung. ATMEGA32A berfungsi sebagai mikrokontroler atau otak dari sistem ini. *Graphical LCD* berfungsi sebagai media untuk menampilkan keluaran dari sistem yang diinginkan.

### 3.3. Perancangan Perangkat Lunak

Berikut merupakan diagram alur pada sistem:



Gambar III- 9 Flowchart sistem elektrokardiogram

Pada sistem ini *real-time operating system* yang dibuat merupakan sebuah *real-time operating system* berbasis penjadwalan. Pada RTOS jenis ini *task* dijalankan berdasarkan prioritas dari *task* tersebut.



Gambar III- 10 Flowchart Real-time operating system

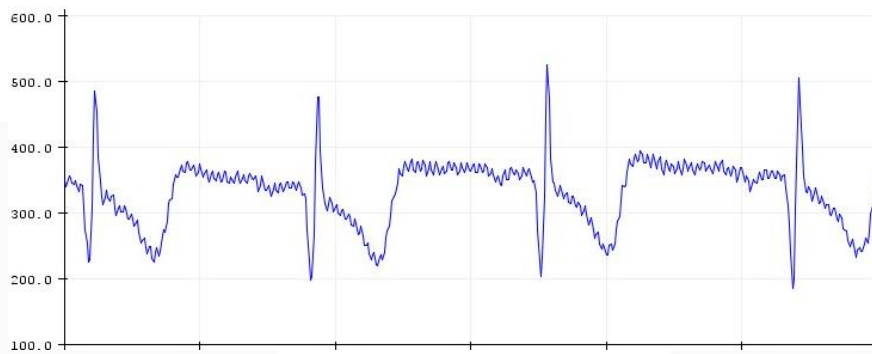
#### 4. Hasil Pengujian dan Analisis

##### 4.1 Hasil Pengujian Modul AD8232

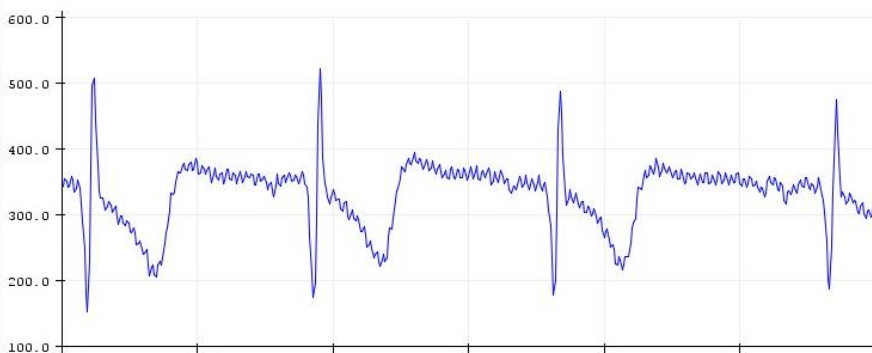
Modul AD8232 adalah sebuah rangkaian yang berfungsi untuk mendeteksi sinyal elektrokardiogram dan sinyal biopotensial lainnya. Modul AD8232 dirancang untuk dapat menangkap sinyal biopotensial, memperkuat sinyal biopotensial tersebut, serta menyaring sinyal biopotensial tersebut dari berbagai *noise*. Tujuan pengujian ini adalah untuk melihat seberapa baik sensor ini untuk diaplikasikan dengan RTOS untuk menjadi elektrokardiogram.



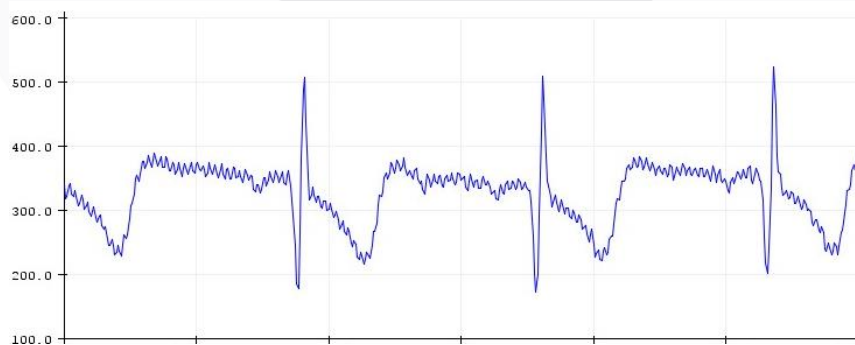
Gambar IV- 1 Pengujian 1 AD8232 Subjek 1



Gambar IV- 2 Pengujian 2 AD8232 Subjek 1



Gambar IV- 3 Pengujian 3 AD8232 Subjek 1



Gambar IV- 4 Pengujian 4 AD8232 Subjek 1



Gambar IV-5 Pengujian 5 AD8232 Subjek 1

Tabel IV-1 Data Hasil Sadapan AD8232 Subjek 1

Data ke-	Nilai	Data ke-	Nilai	Data ke-	Nilai	Data ke-	Nilai	Data ke-	Nilai	Data ke-	Nilai	Data ke-	Nilai
1	272	31	349	61	324	91	375	121	386	151	308	191	345
2	259	32	337	62	318	92	375	122	379	152	334	192	325
3	265	33	345	63	329	93	367	123	385	153	347	193	340
4	259	34	354	64	316	94	369	124	365	154	365	194	299
5	259	35	338	65	319	95	364	125	320	155	358	195	243
6	268	36	345	66	306	96	377	126	234	156	381	196	193
7	269	37	331	67	318	97	368	127	402	157	362	197	398
8	279	38	339	68	297	98	386	128	542	158	371	198	476
9	304	39	335	69	306	99	369	129	431	159	360	199	353
10	308	40	349	70	289	100	385	130	350	160	374	200	296
11	330	41	326	71	302	101	367	131	368	161	351		
12	338	42	328	72	273	102	390	132	347	162	366		
13	359	43	312	73	278	103	372	133	359	163	343		
14	348	44	331	74	254	104	376	134	333	164	354		
15	363	45	328	75	263	105	380	135	345	165	334		
16	346	46	355	76	253	106	393	136	327	166	346		
17	358	47	331	77	275	107	372	137	332	167	338		
18	337	48	348	78	270	108	389	138	327	168	350		
19	353	49	329	79	291	109	373	139	323	169	338		
20	337	50	349	80	296	110	392	140	312	170	345		
21	353	51	328	81	335	111	369	141	315	171	332		
22	332	52	357	82	335	112	379	142	294	172	344		
23	345	53	328	83	360	113	361	143	291	173	338		
24	330	54	283	84	362	114	366	144	278	174	341		
25	351	55	220	85	376	115	372	145	278	175	339		
26	332	56	362	86	379	116	377	146	267	176	333		
27	352	57	500	87	379	117	377	147	271	177	340		
28	340	58	393	88	370	118	381	148	275	178	336		
29	337	59	330	89	373	119	386	149	269	179	339		
30	346	60	337	90	372	120	380	150	297	180	327		

Tabel adalah data hasil pembacaan sinyal listrik jantung yang dilakukan oleh modul AD8232. Akan dianalisis untuk gelombang R dari elektrokardiogram atau grafik dengan amplitude tertinggi di setiap siklus grafik elektrokardiogram. Data diatas diambil berdasarkan nilai tegangan listrik yang dibaca oleh mikrokontroler setiap 10 *miliseconds* dan *sample* data diambil dari 200 data pertama. Terdapat 4 buah data dengan amplitudo tertinggi atau puncak gelombang R yaitu pada data ke-57 dengan amplitudo 500, data ke-128 dengan amplitudo 542, data ke-198



dengan amplitudo 198. dapat dilihat bahwa grafik gelombang R pada elektrokardiogram tidak menampilkan hasil yang akurat dan stabil serta terdapat banyak *ripple* pada seluruh gelombang PQRST. Terdapat beberapa faktor yang menyebabkan modul AD8232 tidak menampilkan hasil yang akurat dan stabil, diantaranya adalah ketepatan dalam menentukan titik dari ketiga *lead* yang akan dipasang di sekitar jantung, rangkaian penguat pada modul AD8232 belum akurat dan berbagai faktor lainnya. Hal ini menyebabkan modul AD8232 tidak dapat dijadikan menjadi acuan satu-satunya dalam mengambil data aktivitas kelistrikan jantung.

#### 4.2 Hasil Pengujian Pengujian Reliability dan Performance RTOS

*Real-time operating system* harus bisa memiliki ketahanan dan kestabilan yang baik serta dapat melakukan tugasnya sesuai dengan tujuan dari RTOS tersebut. Pada tugas akhir ini, RTOS yang akan dirancang adalah RTOS dengan tipe *soft real-time system*. RTOS dengan tipe ini harus dapat mengeksekusi *task* secara teratur berdasarkan prioritas dari masing masing *task* tersebut. Parameter pengujian *reliability* dari RTOS ini adalah kemampuan RTOS dapat konsisten dalam menjadwalkan *task* berdasarkan prioritas masing masing *task*. Parameter pengujian *performance* dari RTOS ini adalah kemampuan RTOS dalam menjadwalkan *task* sesuai dengan prioritas ketika prioritas dari masing masing *task* tersebut diganti. berikut adalah 5 *task* yang harus di jadwalkan oleh RTOS:

1. *Task* 1: PORTB = 0b00000000;
2. *Task* 2: PORTB = 0b00000011;
3. *Task* 3: PORTB = 0b00001111;
4. *Task* 4: PORTB = 0b00111111;
5. *Task* 5: PORTB = 0b11111111;

##### a. Pengujian 1

Berikut adalah prioritas dari masing masing *task*:

Tabel IV-3 Data Hasil Pengujian 1

- *Task*1 = 1
- *Task*2 = 2
- *Task*3 = 3
- *Task*4 = 4
- *Task*5 = 5

Jadwal ke-	Nama <i>task</i>	Jadwal ke-	Nama <i>task</i>	Jadwal ke-	Nama <i>task</i>
1	<i>task</i> 1	11	<i>task</i> 1	21	<i>task</i> 1
2	<i>task</i> 2	12	<i>task</i> 2	22	<i>task</i> 2
3	<i>task</i> 3	13	<i>task</i> 3	23	<i>task</i> 3
4	<i>task</i> 4	14	<i>task</i> 4	24	<i>task</i> 4
5	<i>task</i> 5	15	<i>task</i> 5	25	<i>task</i> 5
6	<i>task</i> 1	16	<i>task</i> 1	26	<i>task</i> 1
7	<i>task</i> 2	17	<i>task</i> 2	27	<i>task</i> 2
8	<i>task</i> 3	18	<i>task</i> 3	28	<i>task</i> 3
9	<i>task</i> 4	19	<i>task</i> 4	29	<i>task</i> 4
10	<i>task</i> 5	20	<i>task</i> 5	30	<i>task</i> 5

##### b. Pengujian 2

Berikut ini adalah prioritas dari masing masing *task*:

Tabel IV-4 Data Hasil Pengujian 2

- *Task*1 = 5
- *Task*2 = 4
- *Task*3 = 3
- *Task*4 = 2
- *Task*5 = 1

Jadwal ke-	Nama <i>task</i>	Jadwal ke-	Nama <i>task</i>	Jadwal ke-	Nama <i>task</i>
1	<i>task</i> 5	11	<i>task</i> 5	21	<i>task</i> 5
2	<i>task</i> 4	12	<i>task</i> 4	22	<i>task</i> 4
3	<i>task</i> 3	13	<i>task</i> 3	23	<i>task</i> 3
4	<i>task</i> 2	14	<i>task</i> 2	24	<i>task</i> 2
5	<i>task</i> 1	15	<i>task</i> 1	25	<i>task</i> 1
6	<i>task</i> 5	16	<i>task</i> 5	26	<i>task</i> 5
7	<i>task</i> 4	17	<i>task</i> 4	27	<i>task</i> 4
8	<i>task</i> 3	18	<i>task</i> 3	28	<i>task</i> 3
9	<i>task</i> 2	19	<i>task</i> 2	29	<i>task</i> 2
10	<i>task</i> 1	20	<i>task</i> 1	30	<i>task</i> 1

## c. Pengujian 3

Berikut ini adalah prioritas dari masing masing *task*:

Tabel IV-5 Data Hasil Pengujian 3

- *Task1* = 1
- *Task2* = 3
- *Task3* = 5
- *Task4* = 4
- *Task5* = 2

Jadwal ke-	Nama task	Jadwal ke-	Nama task	Jadwal ke-	Nama task
1	<i>task1</i>	11	<i>task1</i>	21	<i>task1</i>
2	<i>task5</i>	12	<i>task5</i>	22	<i>task5</i>
3	<i>task2</i>	13	<i>task2</i>	23	<i>task2</i>
4	<i>task4</i>	14	<i>task4</i>	24	<i>task4</i>
5	<i>task3</i>	15	<i>task3</i>	25	<i>task3</i>
6	<i>task1</i>	16	<i>task1</i>	26	<i>task1</i>
7	<i>task5</i>	17	<i>task5</i>	27	<i>task5</i>
8	<i>task2</i>	18	<i>task2</i>	28	<i>task2</i>
9	<i>task4</i>	19	<i>task4</i>	29	<i>task4</i>
10	<i>task3</i>	20	<i>task3</i>	30	<i>task3</i>

Dari hasil pengujian keseluruhan, hasil yang didapat adalah RTOS dapat menjadwalkan *task* sesuai dengan prioritas yang diberikan kepada masing-masing *task*. Sehingga tidak ada *task* yang terlewat untuk di eksekusi.

#### 4.3 Kecepatan RTOS

Pengujian *predictability* RTOS dilakukan untuk mengetahui seberapa tepat waktu pelaksanaan *task* yang bisa dilakukan pada RTOS berbasis penjadwalan dengan tipe *soft real-time system*.

Tabel IV-6 Data Hasil Pengujian *Predictability* RTOS

No	frame ke-	Nilai ADC	No	frame ke-	Nilai ADC	No	frame ke-	Nilai ADC
1	425	431	11	465	269	21	505	356
2	429	349	12	469	348	22	509	353
3	433	326	13	473	350	23	514	356
4	437	342	14	478	354	24	518	409
5	441	333	15	481	320	25	522	242
6	445	304	16	485	272	26	526	351
7	449	211	17	489	361	27	530	359
8	453	328	18	493	353	28	534	364
9	457	321	19	497	307	29	538	316
10	461	298	20	501	303	30	542	282

Data diatas merupakan hasil rekaman 30 data pertama yang ditampilkan oleh LCD pada elektrokardiograf. Dengan kemampuan kamera yang dapat merekam secepat 30 FPS. Maka akan didapatkan jarak waktu antar *frame* adalah 33.33 *miliseconds*. Dari hasil pengujian dapat diketahui jarak *frame* antar data adalah 4 . Jika dikonversi kedalam satuan waktu maka jarak antar data atau *time sampling* elektrokardiograf adalah 133.33 *miliseconds*. Di program RTOS yang dibuat, sudah ditentukan bahwa *time sampling* data adalah 100 *miliseconds*. Dari data tersebut dapat diambil kesimpulan bahwa RTOS mengalami keterlambatan selama 33.33 *miliseconds* dalam mengeksekusi *task*. Artinya pengujian *predictability* dari RTOS tidak terlalu baik karena terdapat keterlambatan dalam mengeksekusi *task*.

#### 4.4 Analisis Hasil Data Terhadap Keberhasilan RTOS

*Real time operating system* yang dirancang pada tugas akhir ini adalah sebuah RTOS berbasis penjadwalan dengan tipe *soft real-time system*. Parameter keberhasilan dari RTOS dengan tipe ini adalah kemampuan RTOS itu dalam menjadwalkan beberapa *task* sesuai dengan prioritasnya dan tidak ada *task* yang terlewat untuk dieksekusi. Keterlambatan pada saat mengeksekusi *task* pada RTOS dengan tipe ini bukan merupakan sebuah masalah yang mengakibatkan kegagalan sistem. Akan tetapi keterlambatan pada saat mengeksekusi *task* berpengaruh terhadap performa dari sistem.

Dari data keseluruhan yang sudah diambil dapat diambil kesimpulan bahwa RTOS sudah dapat menjadwalkan semua *task* yang dimasukkan dengan benar dan tidak ada *task* yang terlewat. Namun RTOS yang dirancang memiliki keterlambatan waktu sebesar 33.33 *miliseconds*. Akan tetapi keterlambatan waktu eksekusi *task* tersebut tidak



menyebabkan kegagalan pada sistem ini. Dan dapat diambil kesimpulan bahwa RTOS yang dirancang sudah memenuhi parameter keberhasilan dari *real-time operating system* berbasis penjadwalan dengan tipe *soft real-time system*.

## 5. Kesimpulan dan Saran

### 5.1. Kesimpulan

Setelah membuat perancangan *real-time operating system* berbasis penjadwalan untuk elektrokardiograf serta dan melakukan pengujian serta analisis, maka ditarik kesimpulan bahwa:

1. *Real-time operating system* berbasis penjadwalan dan bertipe *soft real-time system* sangat cocok digunakan sebagai sistem operasi untuk *embedded system* yang memiliki *task* dengan *delay* yang sangat kecil.
2. RTOS berbasis penjadwalan dapat mengeksekusi *task* dengan teratur dan sesuai dengan prioritas yang diberika kepada masing-masing *task* tanpa ada yang terlewat.
3. Waktu eksekusi *task* pada RTOS mengalami perlambatan pada setiap *task* selama kurang lebih 33.33 *milliseconds*. Perlambatan ini disebabkan oleh beberapa sebab seperti pengisian karakter pada LCD, pembacaan nilai ADC jantung dan lain-lain
4. Untuk mendapatkan grafik elektrokardiogram yang lebih baik dan bisa dijadikan acuan untuk mendeteksi kondisi jantung manusia harus menggunakan sebuah penguat diferensial yang lebih akurat serta pengambilan data dilakukan lebih dari 1 buah *lead*.

### 5.2 Saran

Dari hasil pengujian yang telah dilakukan, Tugas Akhir ini masih memiliki beberapa kekurangan. Tentunya dengan memperbaiki kekurangan tersebut dapat mengembangkan alat yang sudah dibuat. Berikut beberapa saran penulis, yang dapat dikembangkan di penelitian berikutnya:

1. RTOS ditambahkan fitur *interrupt* agar bisa membuat deadline pada masing masing *task* dan waktu pelaksanaan *task* lebih terprediksi.
2. Disarankan untuk menggunakan modul elektrokardiograf yang lebih akurat, agar hasil pembacaan bisa lebih akurat.

## Daftar Pustaka

- [1] Jatmiko,W, et.al, " Real Time Operating System (RTOS) Teori dan Aplikasi: Real time operating system," Depok: Universitas Indonesia, 2015
- [2] Williams, L., Wilkins., "ECG Interpretation made Incredible Easy!: ECG Fundamentals," Edisi ke- lima. China: Wolters kluwer, 2011.
- [3] Kautsar, Muhammad, "Pengertian RTOS: Definisi RTOS", 25 Juni 2013. <http://uchanetizen.blogspot.co.id/2013/06/pengertian-rtos.html> [Diakses 10 November 2017]
- [4] Pujo, Setiawan, "Penerapan Real Time Operating Systems (RTOS) Pada Mikrokontroler AVR (Studi Kasus Chibios/RT) ". Semarang: Universitas Diponegoro, 2010
- [5] Suryana, Yaya, "Sistem Pemonitor Detak Jantung Portable Menggunakan Tiga Sensor Elektroda". Jakarta: Universitas Al Azhar Indonesia, 2017
- [6] Wiedjaja A., et.al, "Kajian Dan Implementasi Real Time Operating System Pada Single Board Computer Berbasis ARM". Vol.5 No. 1,pp. 91-99, Juni 2014
- [7] George, Ligo, "Interfaving LCD with Atmega32 Microcontroller Using Atmel Studio", 20 Mei 2013. <https://electrosome.com/interfacing-lcd-atmega32-microcontroller-atmel-studio/> [Diakses 13 April 2018]
- [8] S. R. Ramanarayana, "Selection of RTOS for an Efficient Design if Embedded Systems". *International Journal of Computer Science and Network Security*, Vol.6 No. 6, Juni 2006.
- [9] V. Shinde, S. C. Biday, "Comparison of Real Time Task Scheduling Algorithm". *International Journal of Computer Applications (0975-8887)*, Vol. 158 No. 6, January 2017.
- [10] D. G. Harkut, A. M Agrawal, "Comparison of Different Task Scheduling Algorithm in RTOS: A Survey". *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 4 No. 7, July 2014.