

Laporan Tugas EL3014

Rekayasa Jam Digital

Nama

Andhika Rahadian

NIM

13218030

Contents

1	Analisis Kebutuhan.....	2
2	Spesifikasi	3
3	Perancangan	3
3.1	Perancangan Perangkat Lunak	3
3.2	Perancangan Perangkat Keras.....	11
4	Implementasi	14
4.1	Implementasi Perangkat Lunak	14
4.2	Implementasi Perangkat Keras.....	17
5	Pengujian	22

Daftar Gambar

Gambar 1	State Chart Sistem Jam Digital Keseluruhan	5
Gambar 2	Bagian dari State Chart : Blok State Display	6
Gambar 3	Bagian dari State Chart : Blok State Alarm	6
Gambar 4	Bagian dari State Chart : Blok State Stopwatch	7
Gambar 5	Bagian dari State Chart : Block State Setting	7
Gambar 6	Flowchart Utama (Top Level) dari Perancangan Perangkat Lunak	8
Gambar 7	Flowchat Interrupt Service Routine Timer Overflow untuk Definisi Clock.....	9
Gambar 8	Flowchart Blok State Setting Kalibrasi Menit/Detik.....	9
Gambar 9	Flowchart Blok State Setting Kalibrasi Jam	10
Gambar 10	Flowchart Block State Alarm, dan Setting Jam, Menit, Detik Alarm	10
Gambar 11	Flowchart Block State Stopwatch. Bergantung pada Last State. Apabila ditekan button 'mode', akan berpindah state yang menjadikan reset/start/stop stopwatch.	11
Gambar 12	Skematik Perancangan Hardware Elektrikal untuk Jam Digital	14
Gambar 13	Tampak Atas Arduino Nano.....	17
Gambar 14	Tampak atas dari perangkat 7 Segments 4 Digits Common Carhode	18
Gambar 15	Tampak bawah dari perangkat 7 Segments 4 Digits Common Cathode	18

Gambar 16 Foto Produk Jam Digital Ketika Berada dalam Fitur Alarm (nyala LED biru 10)	21
Gambar 17 Foto Produk Jam Digital Ketika Berada dalam Fitur Clock Yang Menunjukkan Menit Detik (Nyala LED Biru 00)	21

Daftar Tabel

Table 1 Pin Mapping Arduino Nano dan Perangkat 7 Seg 4 Dig Common Cathode	19
Table 2 Pin Mapping Arduino Nano dan Perangkat Pushbutton	19
Table 3 Pin Mapping Arduino Nano dengan Perangkat LED	20

1 Analisis Kebutuhan

- Menampilkan jam, menit dan detik

Pada Tugas 2 EL3014 Sistem Mikroprosesor kali ini, akan dirancang suatu sistem yang membangun jam digital. Jam digital yang dimaksud setidaknya dapat menampilkan jam yang sesuai dengan keadaan saat ini (real-time clock) yang sinkron dengan jam yang ada pada internet. Jam digital juga diperbolehkan untuk menyala sepanjang waktu, tidak harus menyimpan data jam ketika power pada jam digital dimatikan. Waktu yang di tunjukkan jam digital adalah sesuai dengan Waktu Indonesia Barat (WIB)

Sebagai kebutuhan demonstrasi jam digital untuk tugas 2, dan penghematan biaya pada jam digital, maka interface yang digunakan adalah 4 buah seven segment alih alih menggunakan 6 buah seven segment (jam:menit:detik). Kebutuhan demonstrasi jam digital, memiliki arti bahwa ketika melakukan perekaman, bagian jam akan sulit untuk berubah karena akan memakan waktu yang lama. Sehingga akan terdapat 2 mode tampilan waktu jam digital, yaitu mode jam dan mode menit:detik.

- Membantu aktivitas manusia (alarm dan stopwatch)

Jam digital yang akan dibangun, diharapkan dapat menghasilkan suatu fitur yang bermanfaat untuk menunjang aktivitas manusia, yaitu fitur alarm. Yang mana ketika pada jam tertentu yang sudah diatur jam menit detiknya, alarm akan memberikan sinyal tertentu. Sinyal alarm tersebut akan membantu kita untuk mengurangi kemungkinan terjadinya lupa atas aktivitas yang ingin dilakukan pada jam tertentu.

Tidak kalah penting, jam digital juga diharapkan dapat menampilkan selang waktu seseorang dalam mengerjakan sesuatu, seperti aktivitas *workout*/olahraga, aktivitas belajar,dsb. Dengan kata lain, terdapat fitur stopwatch pada jam digital ini.

Tentu penggunaan fitur alarm dan stopwatch tidak mengganggu keberjalanan fungsi utama jam digital, yaitu menampilkan jam, menit dan detik saat ini.

- Portabel (mudah dipindahkan dan di bawa)

Jam digital yang akan dibuat diharapkan juga dapat dengan mudah dibawa ke mana mana namun tidak harus selalu dipasang di tangan. Karena jam digital ini akan dipakai lebih dikhususkan untuk

mengetahui jam saat ini, dan fitur alarm dan stopwatch ketika melakukan aktivitas tertentu di suatu tempat yang berpindah pindah.

2 Spesifikasi

- Menampilkan jam, menit dan detik
- Tampilan mode 1: menit & detik
- Tampilan mode 2: jam
- Optional: menampilkan tanggal
- Prosesor: Atmega328, bisa pakai Arduino Nano atau Arduino UNO
- Display LED 7 segment, 4 digit.
- Ada fitur alarm
- Tidak wajib pakai buzzer, cukup dengan LED saja. Optional: LED berkedip ketika sedang alarm
- Ada fitur stopwatch
- Stopwatch tetap berjalan ketika di mode jam
- Jam tetap berjalan ketika di mode stopwatch
- Sumber waktu: kristal yang menghasilkan interupsi timer
- Tombol 2 atau 3 buah. Optional : pakai 1 tombol saja.

3 Perancangan

Dari bab 1 dan bab 2 sudah ditentukan analisis kebutuhan dan spesifikasi. Spesifikasi desain alat jam digital diturunkan dari analisis kebutuhan dari jam digital tersebut. Pada bab 3 ini, akan dipaparkan rencana perancangan/desain dari jam digital mulai dari segi software hingga segi hardware. Yang perlu ditekankan adalah pada perancangan hardware/perangkat keras, pada tugas kali ini tidak difokuskan pada desain mekanikalnya, alih alih lebih terfokus ke bagian elektrikal dari perangkat keras.

3.1 Perancangan Perangkat Lunak

Perancangan perangkat lunak yang akan di paparkan di bawah ini dibagi menjadi 2. Yang pertama adalah state chart yang merepresentasikan kondisi/state tertentu dari sistem jam digital ketika button mode/set di tekan. Yang kedua adalah tentang flowchart software yang merepresentasikan alur kerja sistematis program yang akan diflash ke microcontroller.

Yang pertama adalah state chart. Gambar 1 adalah diagram desain state chart yang akan diimplementasikan pada jam digital secara keseluruhan. Sementara gambar 2-4 merupakan sub bagian state chart agar lebih terlihat dalam laporan ini.

Perancangan state chart yang saya buat terbagi menjadi 4 blok state dibawah top level:

1. Blok Display

Pada blok state display, terdapat 4 substate yang tersedia. Pada substate 1, jam digital menunjukkan menit dan detik saat ini. Ketika button mode ditekan, maka akan berpindah ke substate 2 yang menunjukkan jam saat ini. Kemudian setelah menekan mode, akan berpindah ke substate 3 yaitu menunjukkan menit dan detik dan jam yang bergantian dari alarm (alarm diasumsikan selalu ON). Lalu setelah menekan mode setelah mode alarm, akan berpindah ke mode stopwatch (substate 4). Masing masing state apabila di tekan set, akan berpindah ke blok state lain seperti setting, alarm, stopwatch

2. Blok Setting

Blok state setting akan ditrigger ketika menekan tombol set pada saat substate menit:detik atau substate jam pada blok display.

Ketika menekan tombol set pada saat substate menit:detik, akan berpindah ke blok state setting untuk mengubah nilai menit dan detik. Substate diberikan tanda berupa substate 1-1, yang bertujuan untuk mengubah nilai menit. Ketika tombol set di tekan pada substate 1-1, nilai menit akan diincrement sebesar 1.

Pada saat substate 1-1, ketika ditekan tombol mode, akan berpindah ke substate 1-2 yang bertujuan untuk mengubah nilai detik. Sama seperti mengubah nilai menit, ketika ditekan tombol set, akan mengincrement nilai detik sebesar 1. Ketika substate 1-2, dan tombol mode ditekan, maka akan berpindah ke state 1, yaitu berpindah ke blok display dan menampilkan nilai menit:detik.

Yang terakhir dalam blok setting, secara umum cara kerja untuk mengubah nilai jam adalah sama, namun berangkat dari blok state display yang menampilkan jam, tekan tombol set untuk berpindah ke blok state setting jam, tekan tombol set untuk mengincrement jam, tekan tombol mode untuk kembali ke blok state display.

3. Blok Alarm

Block state alarm akan ditrigger ketika pada blok display substate alarm, ditekan tombol set. Pada saat ini, sama seperti blok setting, yaitu dengan mengatur jam, menit dan detik yang mana alarm akan mengeluarkan sinyal. Substate 3-1 3-2 dan 3-3 adalah substate yang mengatur masing masing Jam, menit, detik. Masing2 substate ketika ditekan tombol set, maka akan diincrement sebesar 1. Blok alarm akan kembali ke blok display setelah menekan tombol mode ketika substate 3-3 (edit detik)

4. Blok Stopwatch

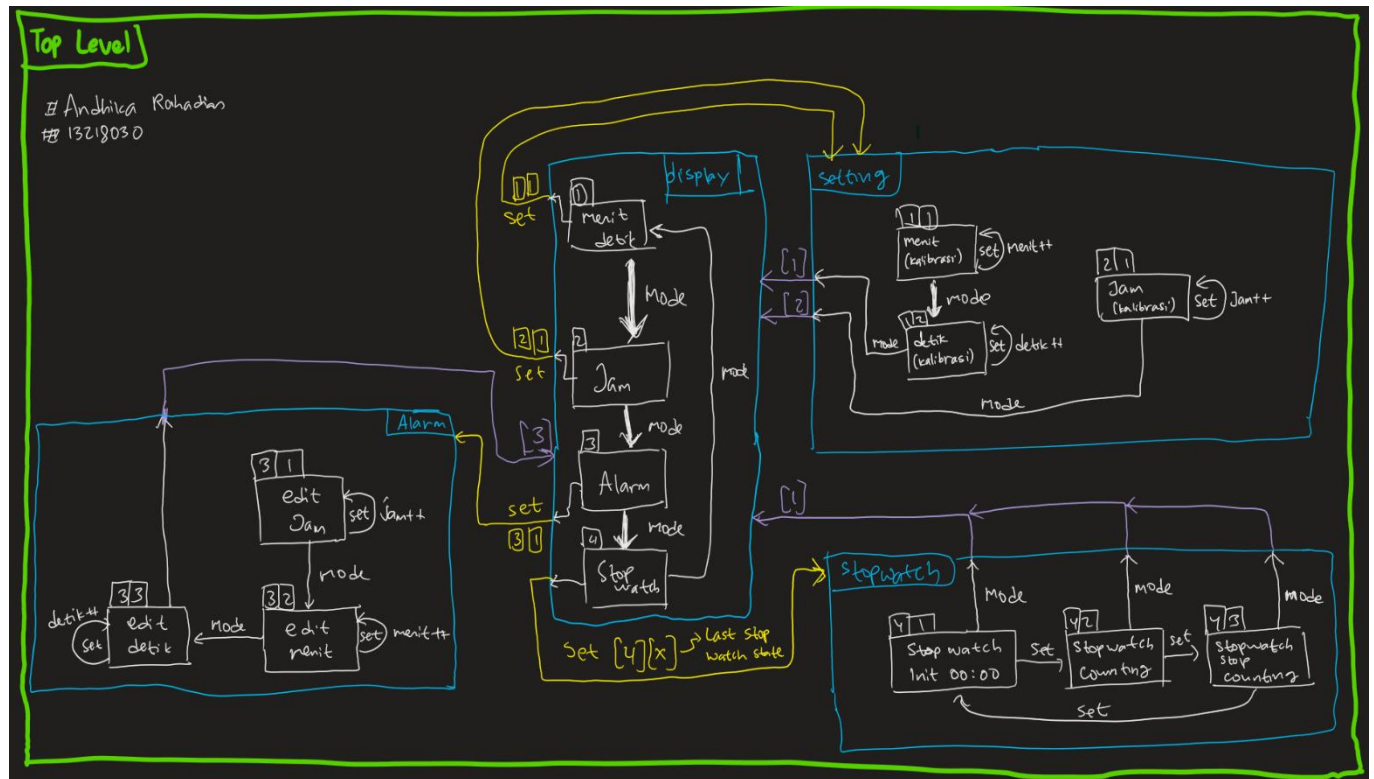
Block state stopwatch akan ditrigger ketika pada blok display substate stopwatch, ditekan tombol set.

Sedikit berbeda dengan blok yang lain, karena display stopwatch harus tetap berjalan walaupun mode display berubah ke selain stopwatch. Sehingga pada blok stopwatch akan dirancang tiga substate dengan kode 4-1,4-2 dan 4-3. Ketika menekan mode pada salah satu sub state ini, maka substate akan disimpan lalu berpindah ke blok display dan menampilkan menit-detik. Substate ini harus disimpan untuk dipakai kembali saat mentrigger blok stopwatch dan dapat menampilkan stopwatch yang masih counting/paused.

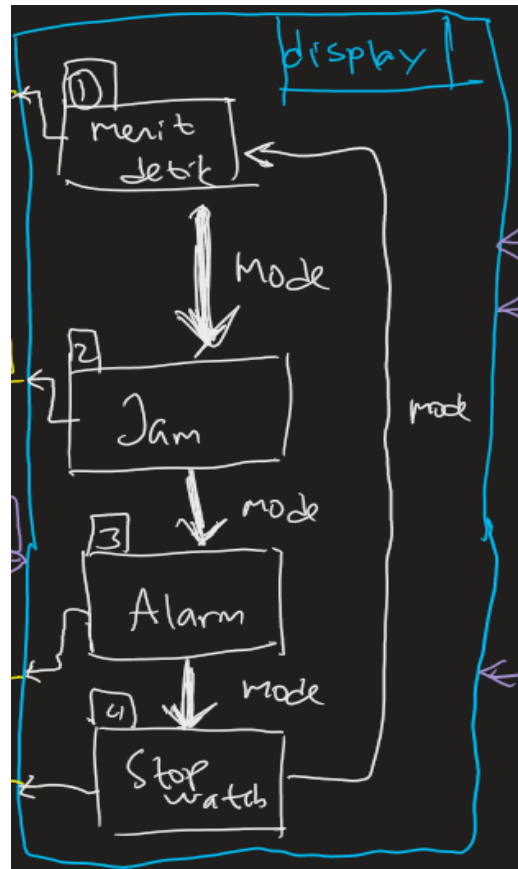
Substate 4-1 menunjukkan inisialisasi stopwatch, sehingga akan bernilai 00 00 untuk menit dan detiknya. apabila menekan tombol set akan berpindah ke substate 4-2, sementara apabila menekan tombol mode, akan kembali ke blok display, substate menit:detik.

Substate 4-2 Stopwatch yang sedang berjalan (counting. apabila menekan tombol set akan berpindah ke substate 4-3, sementara apabila menekan tombol mode, akan kembali ke blok display, substate menit:detik.

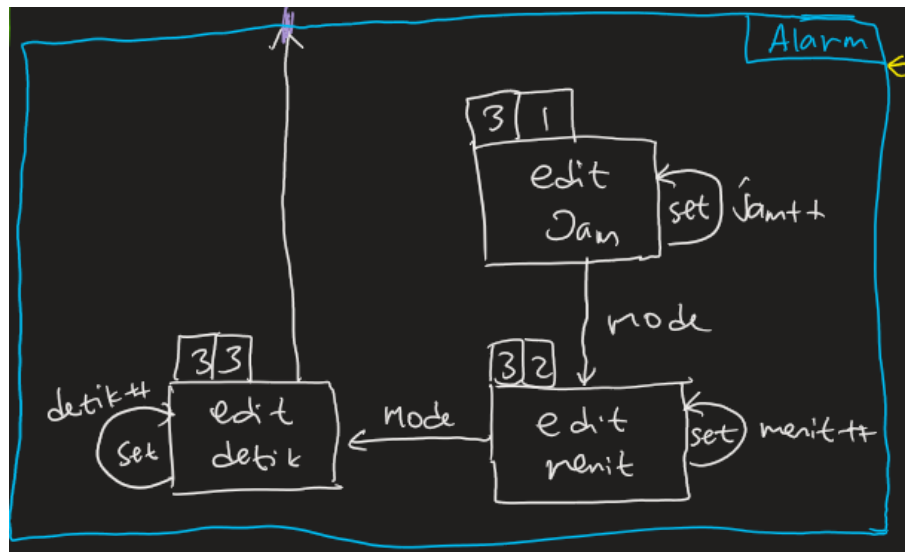
Substate 4-2 Stopwatch yang sedang dipause . Apabila menekan tombol set akan berpindah ke substate 4-1 (reset stopwatch), sementara apabila menekan tombol mode, akan kembali ke blok display, substate menit:detik.



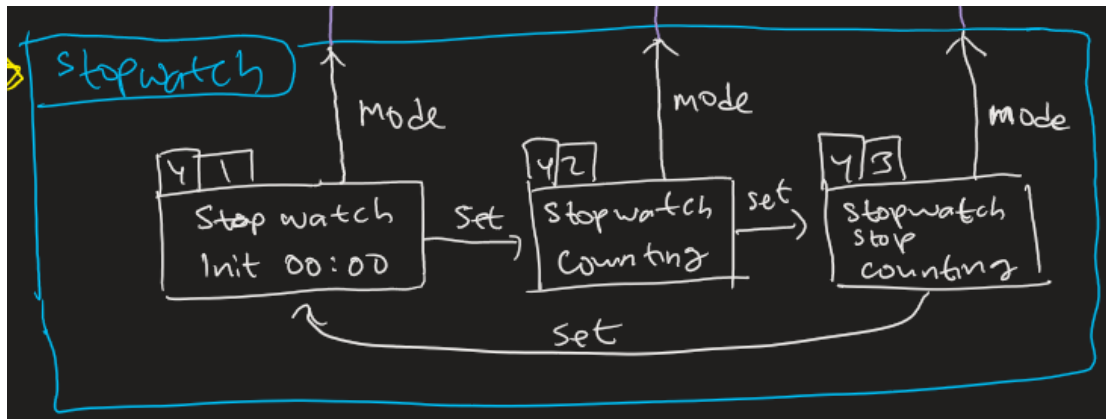
Gambar 1 State Chart Sistem Jam Digital Keseluruhan



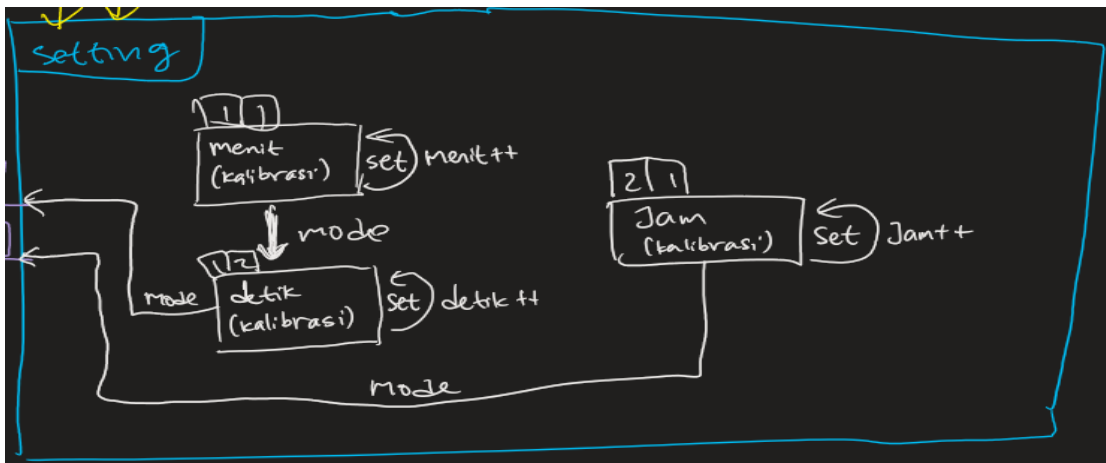
Gambar 2 Bagian dari State Chart : Blok State Display



Gambar 3 Bagian dari State Chart : Blok State Alarm

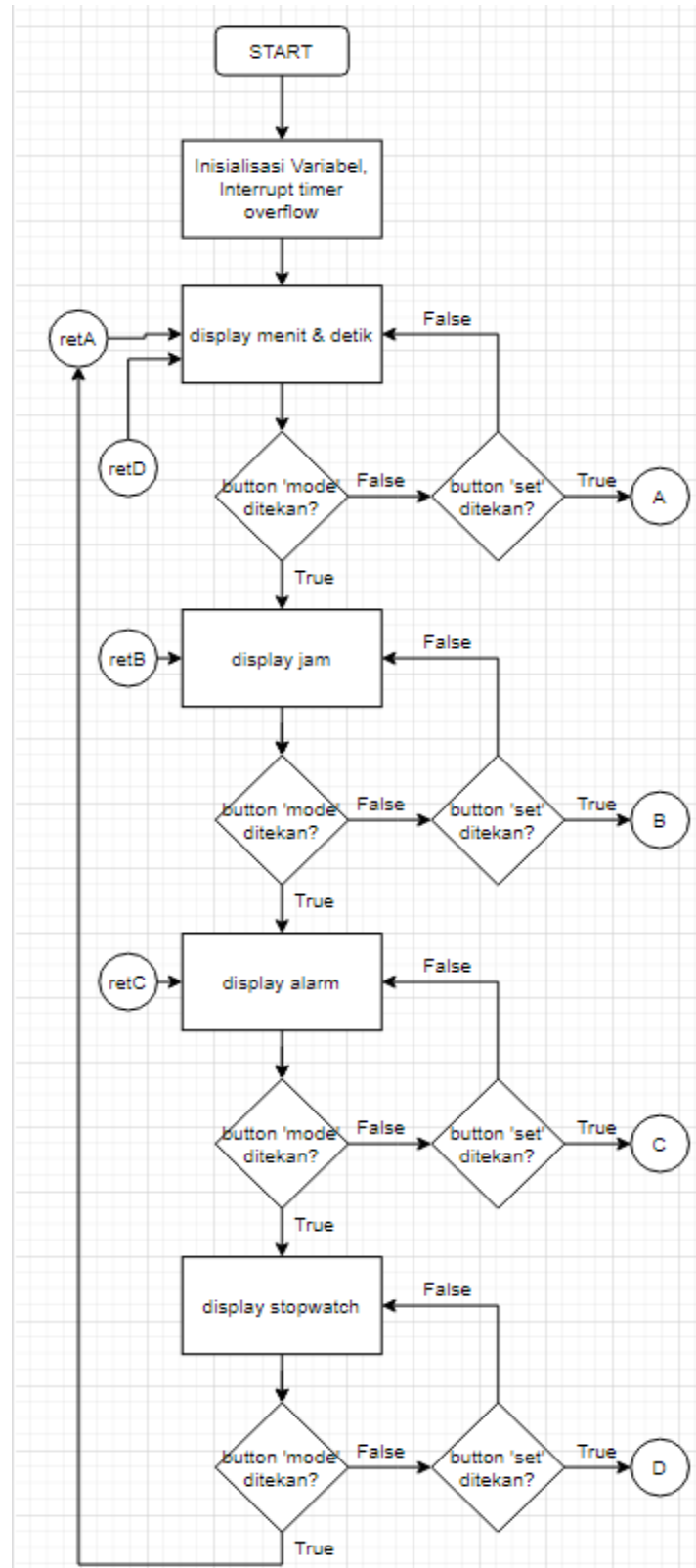


Gambar 4 Bagian dari State Chart : Blok State Stopwatch

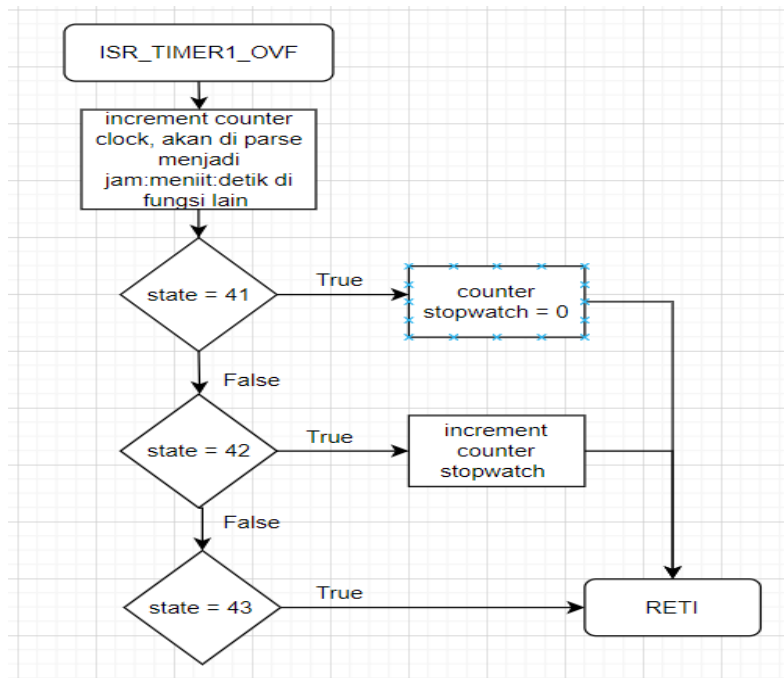


Gambar 5 Bagian dari State Chart : Block State Setting

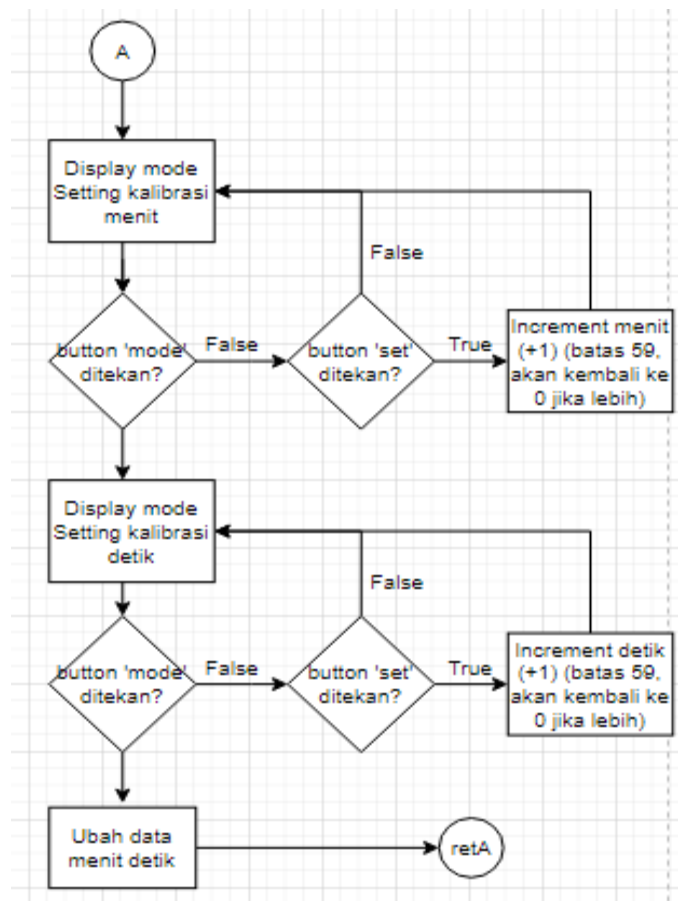
Perancangan flowchart software adalah sebagai berikut:



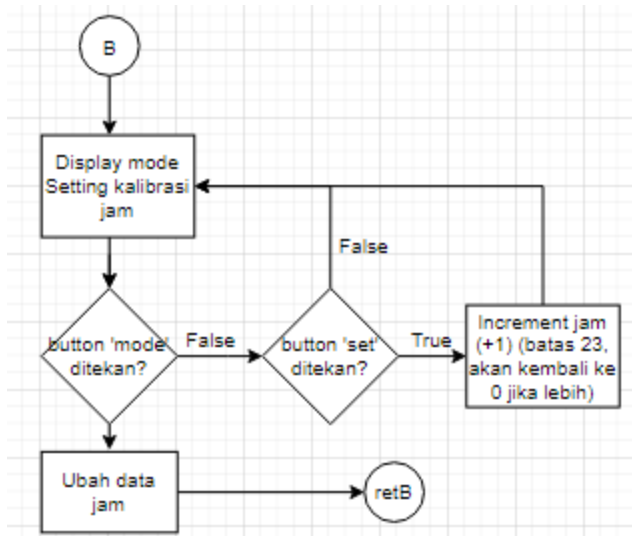
Gambar 6 Flowchart Utama (Top Level) dari Perancangan Perangkat Lunak



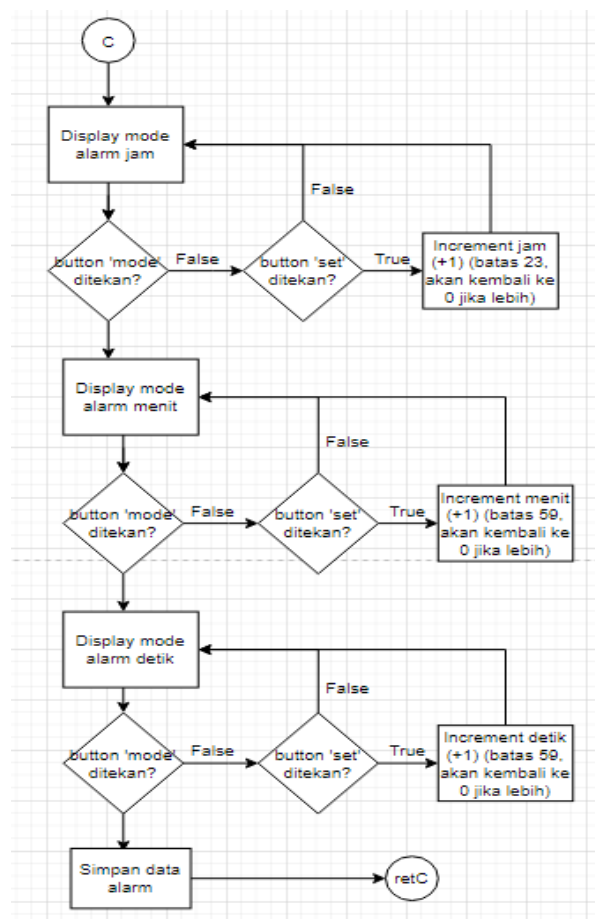
Gambar 7 Flowchart Interrupt Service Routine Timer Overflow untuk Definisi Clock



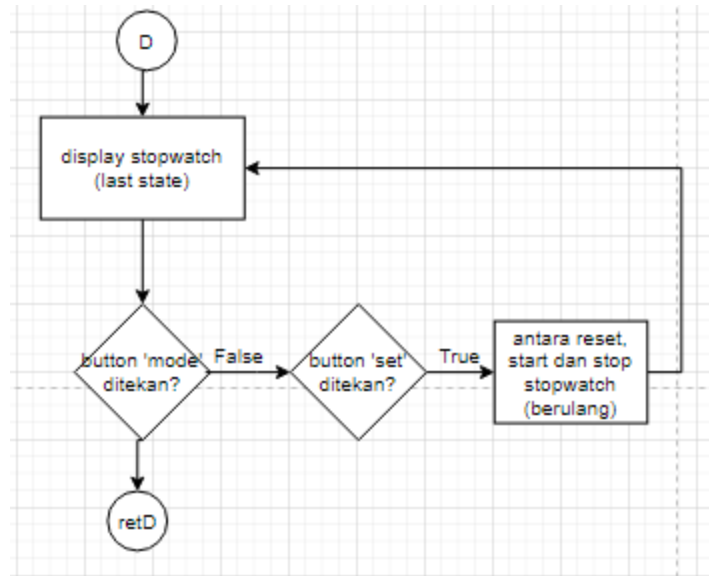
Gambar 8 Flowchart Blok State Setting Kalibrasi Menit/Detik



Gambar 9 Flowchart Blok State Setting Kalibrasi Jam



Gambar 10 Flowchart Block State Alarm, dan Setting Jam, Menit, Detik Alarm



Gambar 11 Flowchart Block State Stopwatch. Bergantung pada Last State. Apabila ditekan button 'mode', akan berpindah state yang menjadikan reset/start/stop stopwatch.

3.2 Perancangan Perangkat Keras

Setelah dilakukan perancangan perangkat lunak (software) yang akan diimplementasikan pada jam digital, akan dilakukan proses perancangan perangkat keras yang membangun jam digital. Perancangan perangkat keras lebih difokuskan kepada bagian elektrik dari jam digital, sementara bagian mekanikal adalah opsional. Perancangan perangkat keras sama pentingnya dengan perancangan perangkat lunak, karena perangkat keras merupakan salah satu benda yang dapat dilihat secara langsung oleh user.

- **Analisis Komponen 7 segments-4 digit untuk display**

Secara umum, akan digunakan 7 segments-4 digit common anoda untuk display utama dari jam digital. Karena 7 segments pada dasarnya adalah diode, maka dibutuhkan resistor pembatas arus agar LED dapat menyala dengan aman. Saya memilih arus bernilai **12 mA** untuk setiap segmen pada setiap digit pada 7 segments. Saya membuat nyala 7 segment tidak terlalu terang, dan karena untuk kebutuhan jam yang dapat dibawa bawa sehingga nyaman di lihat di setiap saat untuk pengelihatan jarak dekat. Untuk mendapatkan arus 12 mA yang ditarik dari PNP dan akan melakukan sink pada pin arduino, maka dibutuhkan resistor pembatas arus dengan nilai:

$$R = \frac{V_{cc} - V_{LED}}{I} = \frac{5 - 1.9}{12mA} = 258,33\Omega$$

Saya menggunakan batas bawah resistor yaitu **220 Ohm** agar perbedaan nilai arus akibat toleransi resistor, ditaksir ke atas (arus akan naik).

- **Analisis Komponen LED untuk alarm, dan untuk menunjukkan mode (display/alarm/stopwatch).**

Untuk mengetahui mode jam digital yang sedang ditampilkannya, dibutuhkan suatu interface. Saya menggunakan dua LED berwarna biru yang dapat merepresentasikan state display pada saat itu. Secara konsep bit, dua LED dapat merepresentasikan 4 state yang berbeda (2 bit). Saya mendefinisikannya sebagai berikut:

00 : State yang menampilkan menit:detik

01 : State yang menampilkan jam

10 : State yang menampilkan Alarm

11 : State yang menampilkan Stopwatch

Sehingga saya menggunakan 2 LED (berwarna biru) agar user mengetahui, jam digital tersebut sedang menampilkan mode yang mana.

Terdapat LED lain yang akan digunakan pada jam digital ini, yaitu untuk menampilkan sinyal alarm ketika jam alarm sesuai dengan jam saat ini. LED yang digunakan adalah berwarna merah sehingga menambah *awareness* bagi user.

Batas maksimum arus yang melewati LED adalah 10-30mA dengan rentang arus yang umum dipakai adalah 12-20mA[1]. Tegangan forward dioda yang digunakan adalah 1,9V[5]. Karena LED merah akan dibuat lebih terang, maka arus yang diberikan harus lebih besar dibandingkan LED biru. Sehingga saya menggunakan arus 12mA untuk masing masing LED biru, dan 18 mA untuk LED merah dan total arus yang ditarik untuk kebutuhan interfacing ke LED adalah **42mA**.

Untuk LED merah (alarm) dengan arus 18mA, dibutuhkan resistor pembatas arus yang dapat dikalkulasi menggunakan hukum ohm :

$$R = \frac{V_{GPIO} - V_{LED}}{I} = \frac{5 - 1.9}{18mA} = 172,22\Omega$$

Sehingga digunakan **resistor sebesar 172,22 ohm** untuk membatasi arus pada LED merah. Dan resistor yang terdekat sesuai dengan standard (electronics industry association) E12, dipilih resistor **220 ohm**.

Untuk LED biru (interface mode) dengan arus 12mA, dibutuhkan resistor pembatas arus yang dapat dikalkulasi menggunakan hukum ohm :

$$R = \frac{V_{GPIO} - V_{LED}}{I} = \frac{5 - 1.9}{12mA} = 258.33\Omega$$

Sehingga digunakan **resistor sebesar 258.33 ohm** untuk membatasi arus pada LED merah. Dan resistor yang terdekat sesuai dengan standard (electronics industry association) E12, dipilih resistor **330 ohm**.

- **Analisis Komponen Bipolar Junction Transistor PNP**

Pada jam digital ini, digunakan empat transistor PNP untuk men-*drive* 7 segments yang berjumlah 4 digit. Saya memilih PNP ketimbang NPN karena emitter PNP dihubungkan ke power supply 5V

arduino yang dapat menjadi source arus yang lebih besar untuk mentenagai komponen 7 segments. Saat arduino diberikan power dari USB, power supply 5V dapat memberikan kurang lebih 400mA[6].

Selain itu, tipe PNP yang akan digunakan adalah **BC557A**. dengan hFE sekitar 120 untuk arus kolektor dalam range -100mAdc[7]. Pada analisis sebelumnya sudah dijelaskan bahwa saya memilih arus bernilai 12 mA setiap segmen. Dari data ini, akan dipilih resistor base PNP untuk membatasi arus base yang ditarik pin mikrokontroler dengan formula (asumsi bekerja pada mode saturasi dengan Veb dekat dengan 0,7, dan active low):

$$R_{base} = \frac{(V_B - 0)}{I_{base}} = \frac{V_E - V_{EB}}{I_{base}} = \frac{5 - 0,7}{\frac{I_C}{(\beta)}} = \frac{4,3}{\frac{12mA}{120}} = 43k\Omega$$

Karena keterbatasan nilai resistor pada skematik, digunakan resistor 47k ohm untuk kebutuhan skematik.

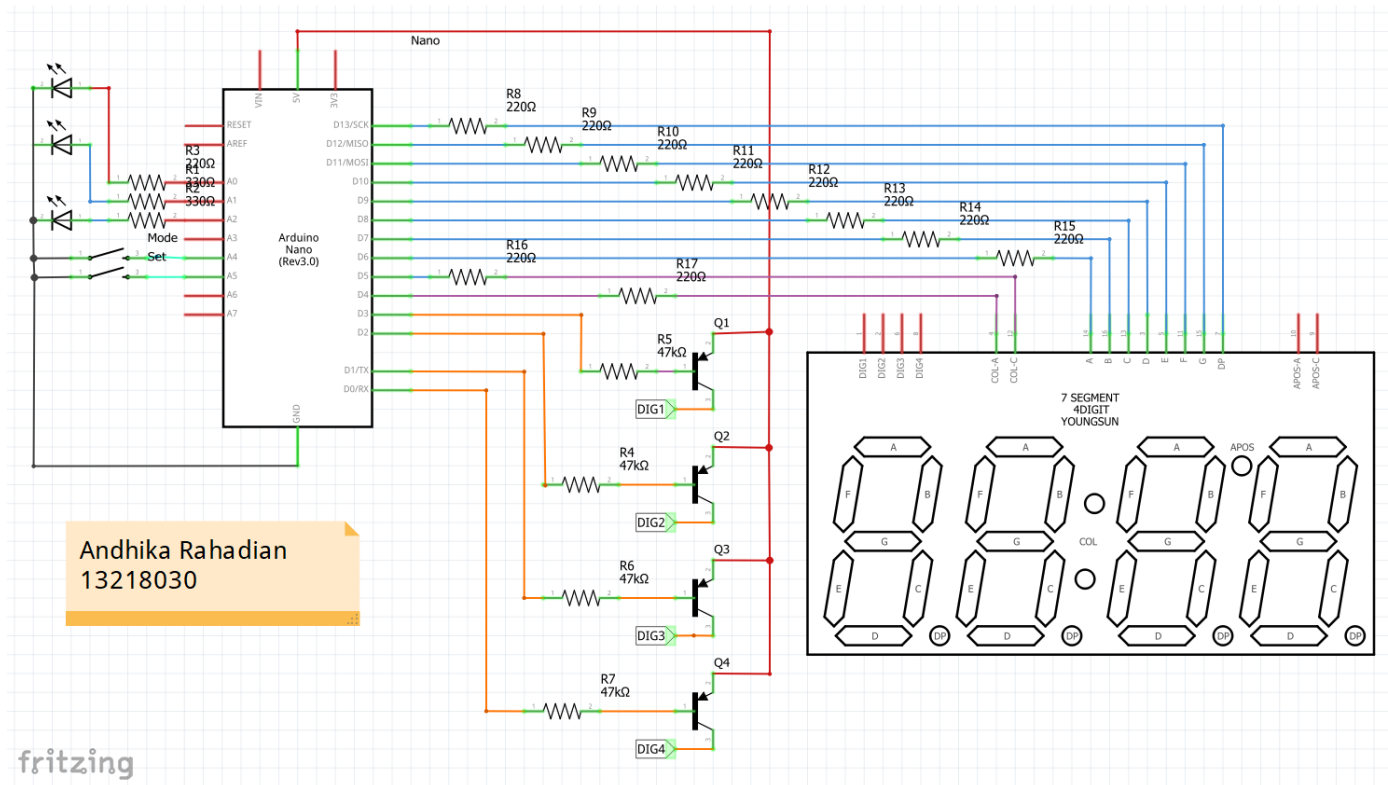
- **Analisis Cara Kerja Perangkat Keras Sistem**

Pada dasarnya, sistem hardware jam digital hardware yang dibuat terdiri dari 1 LED merah untuk visualisasi trigger alarm, 2 LED biru untuk menentukan mode pada saat ini(jam/menit/detik/alarm/stopwatch), 1 unit seven segments 4 digit display untuk menampilkan angka angka yang berisi jam/menit/detik, 4 buah transistor PNP untuk men-drive arus yang masuk pada seven segments 4 digit display. Arduino Nano sebagai microcontroller sebagai pengendali seluruh komponen sesuai dengan software yang sudah di rancang.

Arduino sebagai microcontroller, akan menerima input sinyal dari button sebagai penanda untuk berpindah state pada software. Ketika berada di state tertentu, arduino mengeluarkan sinyal sinyal pada pin pin digital (dan analog yang dijadikan digital), untuk mengaktifkan transistor dan transistor men-drive digit seven segments agar dapat menyala secara normal. Pin pin digital yang lain juga digunakan untuk mengendalikan nyala seven segments agar bentuknya dapat menyerupai jam/menit/detik pada saat ini. Nyala 2 LED biru untuk merepresentasikan mode (display/alarm/stopwatch), juga dikendalikan oleh microcontroller arduino, begitu juga dengan LED merah akan menyala apabila sinyal alarm ditrigger.

Salah satu yang menarik adalah cara kerja dari komponen 7 segments 4 digit. Komponen ini menggunakan konsep multiplexing seven segments untuk menghemat pin untuk mengendalikan 7 segment tersebut. Pin A-G pada komponen ini dipasang secara parallel dan nyala dari masing masing digit 7 segments ini **tidak simultan atau tidak sinkron**, melainkan nyala **secara bergantian**. Meskipun nyala bergantian setiap digitnya, karena selang waktu menyala bergantian dari setiap digit pada komponen ini sangat cepat, maka mata kita seolah olah melihat nyala yang sinkron (sekaligus 4 digit menyala). Dengan konsep multiplexing ini, memungkinkan untuk menghemat pin dan, arus nominal yang digunakan untuk menyalakan komponen ini, sama saja dengan arus nominal untuk menyalakan hanya satu seven segment saja.

Berikut adalah skematik dari sistem hardware dari jam digital yang akan dibuat:



Gambar 12 Skematik Perancangan Hardware Elektrikal untuk Jam Digital

4 Implementasi – Tugas 3

4.1 Implementasi Perangkat Lunak

- Finite State Machine Implementation

Menurut spesifikasi sistem, dijelaskan secara rinci bahwa sistem jam digital menggunakan konsep state. Sehingga dalam implementasi perangkat lunak, akan berbasis finite state machine. Secara umum, pada perancangan perangkat lunak dibuat beberapa fungsi pada source code Arduino (.ino) yang konsepnya berbasis finite state machine yang mana terdapat **next state logic** dan **output logic**. Perlu diketahui bahwa state ini menandakan fitur yang sedang ditampilkan pada interface jam digital. State-state tersebut dilambangkan dengan suatu integer berupa kode (2 digit dan 1 digit) sesuai dengan perancangan state yang sudah dibuat.

Next state logic ditrigger oleh button. Sehingga, dibuat suatu prosedur untuk mendeteksi adanya tekanan pada button yang dapat mentrigger perubahan state, sesuai state chart yang sudah dirancang. Prosedur ini menentukan arah perubahan state secara sekuensial sehingga membuat state menjadi dinamis dan user akan dapat menggunakan banyak fitur dari produk jam digital ini. Fitur-fitur yang ada pada produk ini didasari pada state-state yang ada, dan user dapat menekan dua buah button yang dapat mendinamisasi state dan menggunakan keseluruhan fitur dari produk jam digital seperti alarm, stopwatch, clock, dan settingsnya.

Output Logic didasari oleh state pada saat itu. Pada dasarnya, ketika berada dalam suatu state, akan dihasilkan keluaran berupa digit seven segment yang menunjukkan angka clock/alarm/stopwatch, dan

keluaran berupa nyala 2 buah LED biru yang memberitahu user terkait fitur apa yang ditunjukkan jam digital pada saat itu. Selain itu, apabila alarm memenuhi kondisi (waktu alarm = waktu clock), akan dikeluarkan berupa output digital untuk menyalakan LED secara kedap kedip selama 30 detik (dapat diatur durasinya pada program) dengan jeda 1 detik setiap kedipan. Karena Output Logic bergantung pada State pada saat itu, maka implementasi dapat dilakukan dengan menggunakan switch case untuk semua code state yang mungkin. Masing masing case merepresentasikan apa saja output yang harus di berikan pada 7 segments dan LED. Semua output logic ini diatur oleh suatu prosedur yang akan dijelaskan selanjutnya.

- Timer Interrupt Implementation

Selain finite state machine, pada perangkat lunak juga diimplementasikan timer interrupt dengan mode timer overflow. Timer overflow memungkinkan untuk mentrigger interrupt setelah timer mencapai batas maksimalnya. Timer yang dipakai pada Arduino Nano adalah Timer1 dengan resolusi 16 bit. Pemilihan timer 16 bit dimaksudkan untuk mendapatkan selang waktu triggering interrupt setiap 1 sekon.

Untuk merealisasikan interruptnya, dilakukan insialisasi timer dengan mengatur register TCCR1A dan TCCR1B menjadi 0. Lalu, untuk mendapatkan triggering interrupt setiap satu detik, dengan menggunakan metode timer overflow, dapat digunakan formula untuk mengisi register TCNT1:

$$TCNT = 2^{bittimer} - \frac{frekuensi\ clock}{prescaler} \cdot t$$

Sumber: Referensi[8] daftar pustaka

Yang mana register TCNT1 atau timer counter 1 ini menunjukkan nilai awal register untuk diincrement agar sampai ke batas over flow yaitu $2^{16} = 65536$. Frekuensi clock Arduino Nano adalah 16MHz dan digunakan prescaler sebesar 1024 (sebenarnya memungkinkan juga untuk memakai prescaler 256). Dengan adanya pengaturan prescaler dan mode overflow, maka sesuai dengan data sheet ATmega, digunakan assignment nilai register:

TCNT1 = 65536-15625 = 49911 -> sebagai nilai awal counter timer

TCCR1B = 0b00001101 -> sebagai timer counter control register (untuk menentukan mode prescaler 1024, dan mode kerja overflow)

TIMSK1 = 0b00000001 -> masking untuk mengaktifkan mode timer overflow.

ISR yang digunakan adalah ISR(TIMER1_OVF_vect) yang merupakan salah satu interrupt vector table yang dimiliki ATmega328 yaitu timer1 overflow. ISR yang dibuat, digunakan untuk melakukan increment terhadap clock dan increment terhadap stopwatch (masing masing memiliki variable yang independent satu sama lain). Untuk clock, counter akan selalu menambah setiap ISR dijalankan, untuk stopwatch, terdapat beberapa kondisi sesuai state yang dirancang. Yaitu mode berhenti, mode berjalan dan mode init/reset.

- Display to Seven Segments, display to LED Implementation

Untuk mengeluarkan output logic berupa display ke LED, cukup sederhana yaitu dengan menggunakan output digital dengan fungsi digitalWrite dalam kondisi tertentu seperti nyala alarm, dan nyala 2 buah LED

biru untuk indicator fitur yang sedang ditunjukkan user. Karena kehabisan pin digital (dipakai untuk input digital (button) dan seven segments, digunakan pin analog A0,A1,A2 untuk menyalakan ketiga LED tersebut.

Sementara untuk implementasi daripada nyala seven segments, akan dirancang suatu BCD/Binary Coded Decimal-To-7Seg secara software berupa prosedur, yang mana prosedur ini dapat mencetak ke seven segments dengan bentuk digit sesuai dengan parameter input prosedur yang sesuai. Isi dari prosedur ini adalah digitalWrite untuk setiap segments agar membentuk digit tertentu, dan dibungkus dalam suatu switch case (dengan kasus ada 9 digit, 0-9). Prosedur ini memungkinkan untuk mencetak bentuk digit yang sesuai dengan angka-angka yang ingin dicetak. Penentuan mana saja segments yang harus diberikan tegangan untuk menghasilkan bentuk digit tertentu, saya melakukan testing terhadap rangkaiannya terlebih dahulu menggunakan multimeter, lalu setelahnya akan diimplementasikan pada software.

Permasalahan selanjutnya muncul, yaitu prosedur tersebut hanya dapat menyalakan satu digit saja, sementara ada 4 digit yang harus dipakai secara bersamaan. Sehingga dibutuhkan suatu prosedur yang lebih top level, untuk mencetak keseluruhan digit (atau mungkin sebagian digit). Dengan menggunakan konsep multiplexing 7 segment 4 digit, akan ditambahkan delay yang sangat kecil untuk mengurangi flicker, dan seolah olah 7 segmen 4 digit menyala bersamaan. Konsep multiplexing ini juga mengantisipasi kelebihan arus yang dapat dikeluarkan Arduino pada satu waktu untuk menyalakan banyak segments. Pada prosedur ini juga di tentukan kasus mana saja yang dapat menjadikan titik dua di 7 segments akan menyala berkedip (seperti clock, stopwatch).

- List Fungsi dan Prosedur yang digunakan pada Perangkat Lunak

Inisialisasi variable global

Void setup() : untuk menginisiasi register timer, menginisiasi pin output dan input, menyalakan interrupt dan menginisiasi state (yaitu pada state = 1 yang menunjukkan menit:detik)

ISR(TIMER1_OVF_vect) : merupakan interrupt service routine sebagai fungsi yang akan dijalankan terus menerus ketika interrupt ditrigger dengan ketentuan timer overflow

Void loop() : proses utama dari semua program, berisi pemanggilan detectAlarm(int AlarmTime), detectButton(), dan parseHMS(count,SWcount), dan mengembalikan nilai count menjadi 0 ketika counter = 86399 atau 23:59:59

Void detectAlarm(AlarmTime) : meminta parameter AlarmTime untuk mengatur durasi Alarm, ketika terdapat alarm, maka LED merah akan berkedap kedip selama AlarmTime

Void detectButton : Implementasi dari Next State Logic FSM, Berisi banyak if else untuk mendinamisasi state ketika adanya sinyal button ditekan

Void stateHandler() : Implementasi dari Output Logic FSM, berisi switch case yang digunakan untuk mengeluarkan output ke seven segments dan ke 2 buah LED biru untuk indicator state/fitur. Dilakukan pemanggilan printToSevSeg(d1,d2,d3,d4,s1,s2,s3,s4)

Void printToSevSeg(d1,d2,d3,d4,s1,s2,s3,s4) : parameter d1 d2 d3 d4 adalah digit yang ingin ditampilkan pada seven segments, dan s1 s2 s3 s4 merupakan Boolean yang menentukan apakah digit tersebut

dinyalakan atau tidak. Disini juga terdapat beberapa kondisi if else untuk menentukan di state mana titik dua pada jam digital dibuat berkedip.

Void parseHMS(count,SWcount) : hasil counter yang menunjukkan nilai detik pada clock dan stopwatch akan diparsing menjadi hours, minutes, dan seconds untuk diketahui masing masing digitnya agar ditampilkan pada seven segments. Void-stateHandler selalu dipanggil karena nilai H,M,S berubah setiap satu detik dan perlu dilakukan pencetakan ulang ke seven segments.

- Perbedaan Implementasi Perangkat Lunak dan Perancangan

Secara umum, untuk implementasi perangkat lunak, secara umum sudah sesuai dengan perancangan, terutama di bagian Finite State Machine yang terdiri Output Logic dan Next State Logiknya sesuai dengan state chart pada perancangan. Perbedaan yang terlihat adalah pada isi ISR dari timer interrupt. Awalnya pada perancangan, fungsi parseHMS() untuk mengubah counter clock ke jam:menit:detik berada di ISR. Namun pada implementasinya, fungsi tersebut terpisah dan fungsi ISR hanya digunakan untuk mengincrement counter clock dan counter stopwatch.

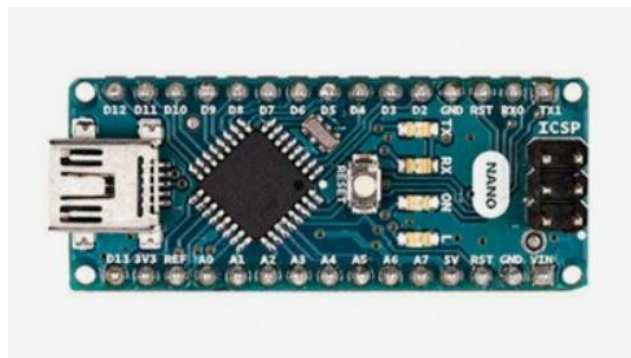
4.2 Implementasi Perangkat Keras

Pada subbab ini akan dijelaskan terkait implementasi perangkat keras yang digunakan.

- Microcontroller

Microcontroller yang digunakan adalah **Arduino Nano**. Hal ini didasari dengan jumlah pin yang masih cukup memadai untuk mengendalikan semua pin segments dan digit 7segment-4digit beserta LED alarm dan LED indicator state. Total pemakaian digital pin adalah 17. Meskipun hanya terdapat D0-D13 pada Arduino nano, pin A0,A1,A2 dapat digunakan untuk output digital yang digunakan pada LED alarm dan LED indicator state.

Microcontroller Arduino Nano relative berukuran kecil dibandingkan microcontroller Arduino yang lain. Sehingga masih cocok untuk spesifikasi yang mana jam digital dapat di bawa secara mudah (portable).



Gambar 13 Tampak Atas Arduino Nano

- Perangkat 7 Segments-4 Digits

Menurut spesifikasi yang sudah dirancang pada tugas sebelumnya, akan diimplementasikan 7 segments dengan 4 digits yang menjadi satu. Didalamnya sudah terdapat transistor yang mendrive arus dari Arduino nano agar setiap digit pada perangkat ini tidak mengalami overcurrent. Transistor ini akan digunakan sebagai switch yang dikendalikan oleh sinyal digital 0 atau 5V dari pin digital Arduino untuk menyalakan atau mematikan digit.

Pada perancangan, akan digunakan 7 segments 4 digit dengan konfigurasi common anode. Namun pada implementasinya, akan lebih mudah menggunakan common cathode untuk membuat implementasi BCD pada software. Karena secara intuitif, active high lebih disukai untuk menyalakan segments yang berjumlah 8 per digit dengan alasan tegangan HIGH identic dengan ON atau menyala. Namun, untuk menyalakan satu digit tetap digunakan tegangan LOW untuk switching transistor dikarenakan arus mengalir dari segment diode ke tegangan LOW/GND yang di set pada base transistor built in pada perangkat. Berikut adalah perangkat 7 segments 4 digit common cathode yang saya beli pada salah satu toko online (sumber gambar pada referensi [9]).



Gambar 14 Tampak atas dari perangkat 7 Segments 4 Digits Common Cathode



Gambar 15 Tampak bawah dari perangkat 7 Segments 4 Digits Common Cathode

Dari perangkat di atas, setelah mencari datasheet di internet, sulit sekali ditemukan yang sesuai dengan kondisi barang yang ada. Sehingga saya melakukan pemeriksaan ulang pinout dari perangkat 7 Segments 4 Digits Common Cathode tersebut. Berikut adalah hasil pemeriksaan pinout beserta keterhubungannya dengan pin pada Arduino Nano:

No	Pin pada Arduino	7 Segments	Butuh Resistor?
1	D1	Digit 1	Tidak
2	D2	B	Ya
3	D3	A	Ya
4	D4	Digit 2	Tidak
5	D5	Digit 3	Tidak
6	D6	C	Ya
7	D7	Digit 4	Tidak
8	D8	G	Ya
9	D9	D	Ya
10	D10	Dot Point	Ya
11	D11	E	Ya
12	D12	F	Ya

Table 1 Pin Mapping Arduino Nano dan Perangkat 7 Seg 4 Dig Common Cathode

Setelah itu akan dilakukan proses penghubungan antara perangkat seven segments 4 digit dengan Arduino Nano menggunakan kabel jumper sesuai dengan pin mapping pada table 1. Perhatikan bahwa No 1 sampai No 12 merepresentasikan posisi pin pada gambar 14, sesuai dengan posisi kaki pin pada perangkat yang digunakan pada implementasi. Beberapa pin membutuhkan resistor, hal ini dikarenakan pada dasarnya segments pada perangkat ini terdiri dari LED, sehingga dibutuhkan pembatas arus agar LED pada perangkat ini tidak rusak karena overcurrent. Sementara ada yang tidak membutuhkan resistor dikarenakan pin tersebut merupakan pin digit yang mana akan langsung terhubung dengan base transistor sebagai switch untuk menyalakan dan mematikan satu digit pada perangkat, sehingga menjadi opsional untuk memberikan resistor pada base transistor.

- Perangkat Pushbuttons

No	Pin pada Arduino	Button	Butuh Resistor?
1	D0	Mode	Ya, Pullup
2	D13	Set	Ya, Pullup

Table 2 Pin Mapping Arduino Nano dan Perangkat Pushbutton

Pada implementasi perangkat keras untuk jam digital kali ini, terdapat dua pushbutton yang akan digunakan. Satu pushbutton berguna sebagai mode, yang secara umum dapat didefinisikan sebagai tombol untuk mengubah mode dan berpindah dari satu fitur ke fitur yang lain (clock, alarm, stopwatch). Satu pushbutton berguna sebagai set, yaitu secara umum digunakan untuk mengatur nilai jam:menit:detik pada clock, alarm dengan cara mengincrementnya sesuai dengan state yang tersedia. Tombol set ini juga digunakan untuk mereset, menyalakan dan memberhentikan stopwatch.

Perlu diperhatikan bahwa terdapat dua pin digital yang digunakan sebagai input pushbutton pada microcontroller. Yaitu pin digital D0 dan pin digital D13. Untuk menghindari floating, pada implementasinya digunakan resistor pullup yang terhubung dengan 5V. Sehingga ketika button ditekan, hasil input yang terbaca pada microcontroller bernilai LOW. Selain resistor eksternal, dapat digunakan juga alternatif berupa resistor pullup internal yang disediakan Arduino nano, namun karena Arduino

Nano sudah banyak dipakai pin nya, untuk mengurangi beban kerja microcontroller, digunakan resistor pullup eksternal sebagai pullup untuk pushbutton.

- Perangkat LED

No	Pin pada Arduino	LED	Butuh Resistor?
1	A0	Alarm	Ya, Current Limit
2	A1	Fitur/State	Ya, Current Limit
2	A2	Fitur/State	Ya, Current Limit

Table 3 Pin Mapping Arduino Nano dengan Perangkat LED

Seperti yang sudah dipaparkan pada perancangan spesifikasi, akan terdapat 3 buah LED. 1 buah LED berwarna merah untuk menunjukkan Alarm, dan 2 buah LED biru untuk menunjukkan state atau fitur apa yang sedang ditunjukkan oleh produk jam digital (apakah clock/alarm/stopwatch). Karena keterbatasan pin digital, maka pin analog A0 A1 dan A2 digunakan sebagai pin untuk mengeluarkan output digital untuk menyalakan atau mematikan LED.

Ketika alarm ditrigger atau waktu alarm = waktu clock pada saat ini, maka LED akan blinking selama 30 detik dengan setting defaultnya (memungkinkan untuk mengubah durasi blinking LED alarmnya dengan mengubah bagian program). Dua LED biru menunjukkan indicator fitur yang dapat memudahkan user untuk menentukan dalam mode apa jam digital tersebut berada (agar tidak membingungkan. Sesuai dengan perancangan, nyala LED biru diatur menjadi seperti berikut (merujuk pada perancangan pada bab 3):

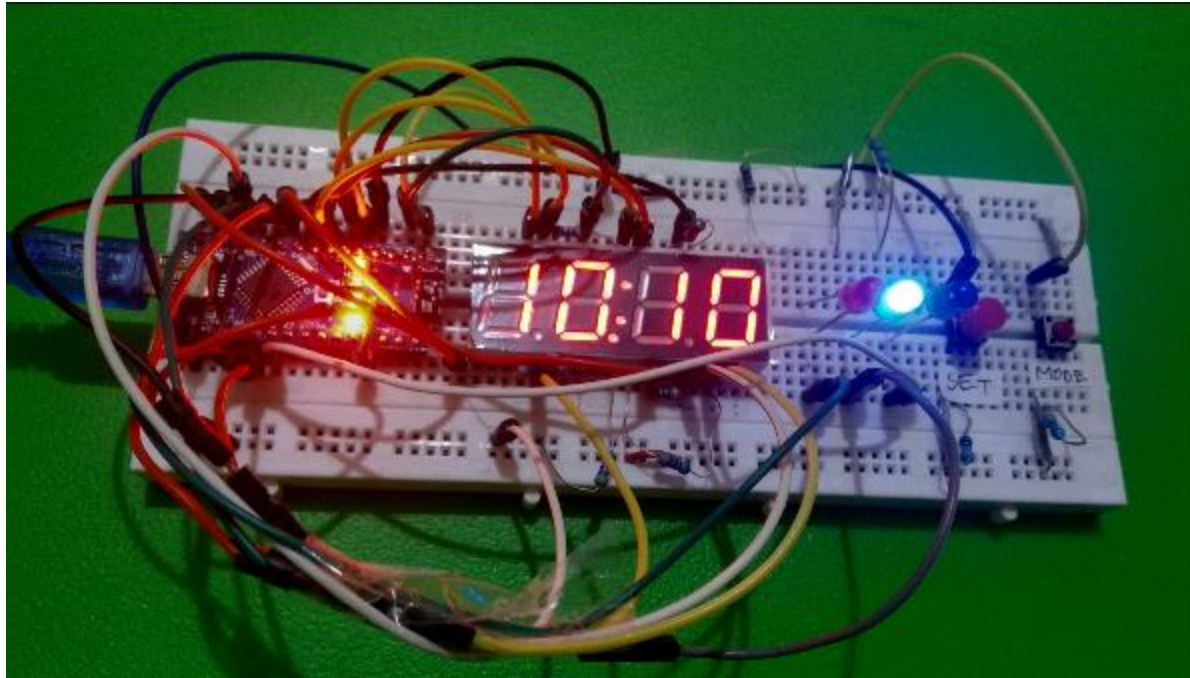
00 : State yang menampilkan menit:detik

01 : State yang menampilkan jam

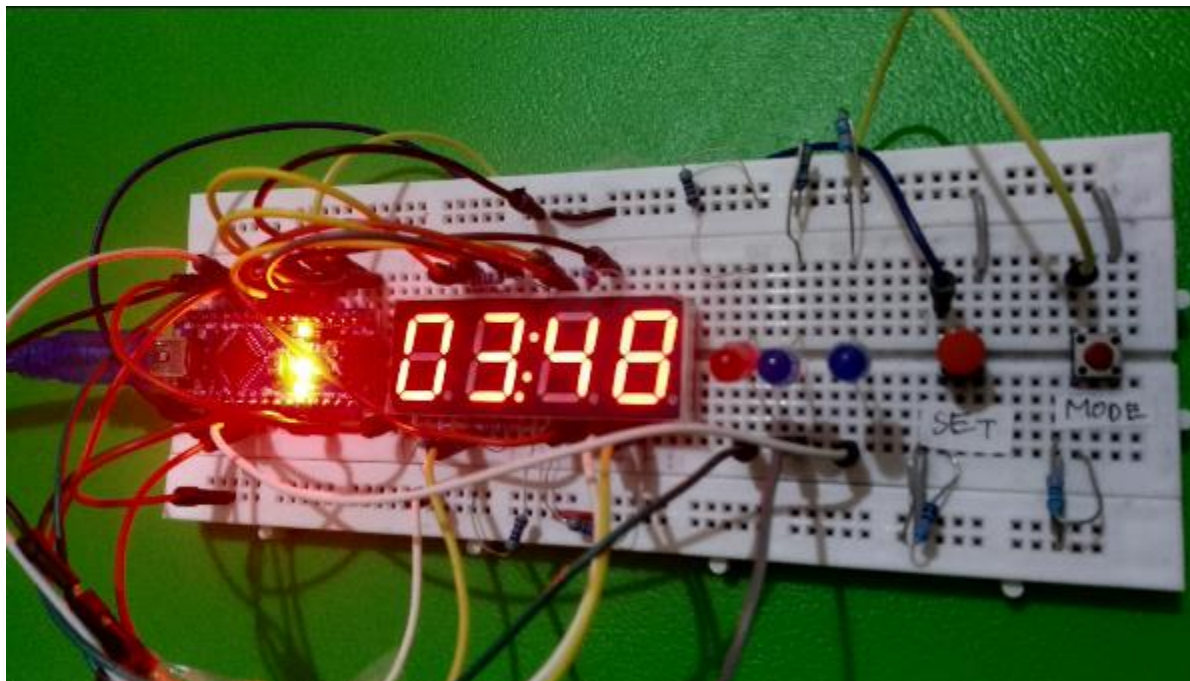
10 : State yang menampilkan Alarm

11 : State yang menampilkan Stopwatch

Secara keseluruhan, berikut adalah beberapa dokumentasi berupa foto dari penampakan alat jam digital yang sudah diimplementasikan:



Gambar 16 Foto Produk Jam Digital Ketika Berada dalam Fitur Alarm (nyala LED biru 10)



Gambar 17 Foto Produk Jam Digital Ketika Berada dalam Fitur Clock Yang Menunjukkan Menit Detik (Nyala LED Biru 00)

- Perbedaan Implementasi Perangkat Keras dan Perancangan

Secara garis besar, implementasi perangkat keras juga sudah sesuai dengan perancangan. Contohnya adalah dengan menggunakan 7 Segments 4 Digit, menggunakan microcontroller, dua push button untuk button mode dan button set, dan 3 buah LED (1 LED merah untuk penanda alarm sedang ditrigger, 2 buah LED untuk indicator state/fitur sesuai dengan kode binary pada perancangan).

Namun, masih terdapat beberapa perbedaan, yaitu:

1. 7 Segments 4 Digit yang digunakan adalah common cathode, sementara pada perancangan menggunakan common anode
2. 7 Segments 4 Digit yang digunakan sudah terdapat transistor NPN (common cathode) yang built in, sehingga tidak perlu lagi menambah transistor eksternal untuk mendrive digit seven segments
3. Perancangan Pin mapping dari arduino ke 7 segments 4 Digit berubah karena menyesuaikan perangkat yang diterima, namun untuk LED, secara umum sama, yaitu menggunakan pin analog untuk output digital.

5 Pengujian

Untuk tahap pengujian, akan dipaparkan pada video yang tertera pada link YouTube di bawah ini. Secara umum, pengujian dilakukan dengan mencoba seluruh fitur yang tersedia pada produk jam digital ini. Percobaan fitur dilakukan dengan menekan tombol mode dan tombol set yang tertera pada alat. Akan diperiksa kebenaran dan keterujian fitur dengan melihat perilaku digit yang dikeluarkan seven segments, dan melihat perilaku nyala LED merah untuk menguji alarm, dan nyala 2 LED biru untuk indicator fitur atau state.

Video pengujian dapat diakses pada link YouTube berikut: https://youtu.be/_bkrfAaPEXA

Untuk Source Code dan Dokumentasi dapat diakses pada link GitHub: <https://github.com/dhkar/digital-clock-7seg4dig>

Secara umum, menurut saya yang perlu diimprove dari jam digital yang sudah selesai diimplementasikan ini adalah, ditambahkan fitur untuk melihat jam pada alarm, stopwatch yang diperluas rangnya hingga menjadi skala jam, lalu alarm yang dapat berhenti ketika button mode/set di tekan (tidak duration based).

6 Referensi

[1] https://www.electronics-tutorials.ws/diode/diode_8.html

[2] <http://www.farnell.com/datasheets/2095876.pdf>

[3] <https://www.microcontrollertips.com/microcontroller-power-source-considerations-arduino-faq>

[4] <https://components101.com/microcontrollers/arduino-uno>

[5]

<https://www.sparkfun.com/products/11408#:~:text=The%20LEDs%20have%20a%20forward,max%20forward%20current%20of%2020mA.>

[6] <https://electronics.stackexchange.com/questions/67092/how-much-current-can-i-draw-from-the-arduinios-pins>

[7] <https://www.onsemi.com/pdf/datasheet/bc556b-d.pdf>

[8] Laboratorium Dasar Teknik Elektro. Petunjuk Praktikum Sistem Mikroprosesor. Sekolah Teknik Elektro Informatika, ITB. 2021

[9] <https://www.tokopedia.com/rajacell/clock-led-seven-segment-7-segmen-0-56-inch-red-common-cathode-merah>