

**RANCANGAN PERANGKAT LUNAK TREATMENT CONTROL UNIT  
BRAKITERAPI HDR IR-192 MENGGUNAKAN  
STATECHART DAN FLOWCHART**

Joko Triyanto, Mohamad Amin, Sukandar  
Pusat Rekayasa Fasilitas Nuklir – BATAN  
Gedung 71, Kawasan PUSPIPTEK Serpong, Tangerang Selatan 15314  
[triyanto123@gmail.com](mailto:triyanto123@gmail.com) ; [amin123@batan.go.id](mailto:amin123@batan.go.id)

**ABSTRAK**

**RANCANGAN PERANGKAT LUNAK TREATMENT CONTROL UNIT BRAKITERAPI HDR IR-192 MENGGUNAKAN STATECHART DAN FLOWCHART.** Makalah ini membahas tentang hasil rancangan perangkat lunak yang akan ditanamkan ke dalam mikrokontroler di bagian treatment control unit brakiterapi HDR-192. Rancangan ini merupakan rancangan awal untuk melihat secara umum sinkronisasi dari struktur, tingkah laku, serta proses-proses pada perangkat lunak sebelum diubah ke dalam rancangan rinci program yang akan ditanamkan ke dalam mikrokontroler. Perangkat lunak yang dirancang bersifat dinamis, sehingga metode yang digunakan dalam perancangan adalah statechart dan flowchart. Statechart untuk menggambarkan struktur dan tingkah laku perangkat lunak, sementara flowchart untuk melihat bentuk proses yang akan dilakukan oleh perangkat lunak. Rancangan perangkat lunak merupakan hasil dari deskripsi umum permasalahan yang berhubungan dengan tingkah laku sensor-sensor dan aktuator yang terhubung ke mikrokontroler. Hasil rancangan perangkat lunak dalam bentuk statechart dan flowchart disajikan di bagian hasil dan pembahasan makalah ini.

Kata kunci : rancangan, statechart, flowchart perangkat lunak, mikrokontroler.

**ABSTRACT**

**A SOFTWARE DESIGN OF TREATMENT CONTROL UNIT IN HDR IR-192 BRACYTHERAPY USING STATECHART AND FLOWCHART.** This paper discusses the results of a software design that will be embedded into a microcontroller in the treatment control unit of the HDR-192 brachytherapy. This design is a preliminary design to see in general view the synchronization of the structure, behavior, and processes in the software before it is converted into a detailed program design that will be embedded into the microcontroller. The software designed is dynamic, so the methods used in the design are statechart and flowchart. Statechart is to describe the structure and behavior of the software, while flowchart is to see the form of the process that will be carried out by the software. The software design is the result of a general description of the problems associated with the behavior of sensors and actuators connected to the microcontroller. The results of the software design are in the form of statechart and flowchart which are presented in the results and discussion section of this paper.

Keywords: design, statechart, software, flowchart, microcontroller.

**1. PENDAHULUAN**

Brakiterapi HDR IR-192 (*Brachytherapy*) merupakan salah satu mesin radioterapi yang digunakan untuk mengobati kanker atau tumor dengan cara meletakkan sumber radiasi dengan laju dosis tinggi secara langsung di dalam atau di dekat kanker atau tumor ganas tersebut. Mesin radioterapi tersebut merupakan sebuah

sistem yang secara garis besar tersusun dari 3 bagian utama [1-3]. Bagian pertama adalah subsistem yang disebut *Treatment Control Unit* (TCU). TCU ini merupakan sebuah unit kontrol terapi yang dikendalikan oleh mikroprosesor atau mikrokontroler. TCU memiliki beberapa fungsi. Fungsi-fungsi tersebut antara lain adalah fungsi untuk memilih lubang kanal yang akan digunakan untuk jalur keluar masuk sumber radiasi dari atau ke kontener ketika proses terapi berlangsung, fungsi untuk mengontrol pergerakan sling *dummy* maju-mundur dari atau ke kontener, fungsi untuk menggerakkan maju-mundur sling sumber dari atau ke kontener, fungsi untuk mengakselerasi pergerakan maju-mundur sling *dummy* atau sumber dari atau ke kontener, serta fungsi untuk membaca perintah dan mengirim status dari dan ke komputer master. Bagian selanjutnya adalah subsistem yang disebut *Treatment Control Panel* (TCP). TCP merupakan sebuah panel kontrol yang berisi sejumlah tombol dan lampu indikator. Tombol-tombol dan lampu indikator tersebut memiliki fungsi-fungsi antara lain; untuk menghidupkan TCU ke posisi *standby* atau posisi operasi, memulai proses terapi, mengaktifkan keadaan emergency, melakukan reset pada mikroprosesor/mikrokontroler, menampilkan indikator status daya, menampilkan indikator status letak posisi sumber radioaktif, menampilkan indikator terjadinya status alarm, status interupsi, atau status pembatalan (*override*). Bagian yang terakhir adalah subsistem komputer master yang dikenal dengan sebutan *Treatment Control Station* (TCS). Komputer master berfungsi untuk mengirim perintah moda terapi. Moda terapi berhubungan dengan besar, jumlah, dan lama laju dosis radiasi yang akan diberikan pada lokasi kanker atau tumor dari pasien. Penentuan jenis moda terapi yang digunakan untuk mengobati pasien ditentukan melalui hasil perhitungan di komputer lain yang disebut *Treatment Planning System* (TPS).

Saat ini, PRFN BATAN sedang dalam tahap mengembangkan prototip Brakiterapi HDR-IR192. Salah satu fokus kegiatan dalam pengembangan tersebut adalah perancangan subsistem TCU. TCU tersebut dikendalikan oleh sebuah kit mikrokontroler dari jenis ARM-Cortex@-M4 32 bit yang lebih umum dikenal dengan sebutan STM32F407VGT6 Discovery. Di satu sisi, salah satu alasan penggunaan kit mikrokontroler jenis STM32F407VGT6 sebagai unit pengendali di dalam TCU adalah ketersediaan sejumlah fitur yang memberikan kemudahan dalam melakukan hubungan antarmuka dengan perangkat keras yang terpasang disekitarnya. Di sisi lain, penggunaan mikrokontroler STM32F407VGT6 sebagai pengendali perangkat keras di dalam TCU memerlukan program atau perangkat lunak yang dapat melakukan sinkronisasi gerakan pengendalian terhadap semua perangkat keras yang terhubung dengannya dengan benar.

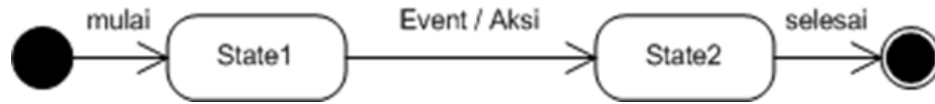
Pembuatan perangkat lunak yang dapat melakukan sinkronisasi gerakan pengendalian terhadap semua perangkat keras yang terhubung ke mikrokontroler STM32F407VGT6 Discovery harus diawali dengan kegiatan perancangan yang baik. Kegiatan perancangan tersebut bertujuan untuk melihat dan mengetahui bagaimana struktur dan tingkah laku program terhadap perangkat keras yang akan dikendalikan, apa saja fungsi-fungsi yang diperlukan untuk menggerakkan perangkat keras, serta untuk melihat bagaimana hubungan antara fungsi-fungsi yang akan melakukan gerakan sinkronisasi dari perangkat keras. Permasalahan ini memotivasi penulis untuk merancang program yang akan ditanamkan ke dalam mikrokontroler dan digunakan untuk mengendalikan gerakan dari perangkat keras tersebut.

Karena luasnya cakupan permasalahan yang berkaitan dengan perancangan program mikrokontroler di TCU, paper ini membatasi perancangan perangkat lunak pada program utama STM32F407VGT6 discovery dari *Treatment Control Unit* (TCU) Brakiterapi HDR IR-192. Metode rancangan menggunakan *statecharts* dan *flowchart*. Untuk memudahkan pemaparan, struktur penulisan paper ini disusun secara beurutan dari pendahuluan, dilanjutkan dengan dasar teori, deskripsi permasalahan, langkah kerja, hasil dan pembahasan, serta diakhiri dengan kesimpulan.

## 2. DASAR TEORI

### a. *Statechart*

*Statechart* adalah suatu simbol grafik yang digunakan untuk menggambarkan struktur dan tingkah laku dari suatu objek [4-7]. Komponen dari *statechart* terdiri dari simbol *state* berbentuk oval, simbol transisi berbentuk garis panah, simbol inisialisasi berbentuk lingkaran kecil berwarna hitam penuh, simbol selesai diwakili oleh lingkaran kecil berisi warna hitam tidak-penuh. (Gambar 1).

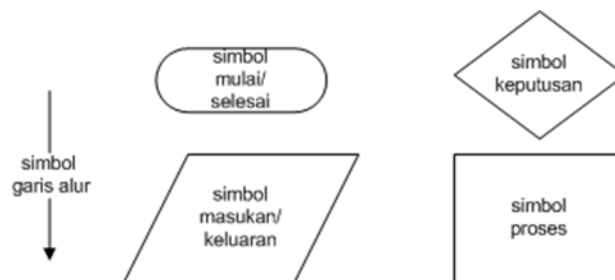


Gambar 1. Contoh simbol-simbol dari *statechart*.

*Statechart* umumnya digunakan untuk menggambarkan dinamika dari suatu objek. Dinamika dari suatu objek diawali oleh sebuah lingkaran kecil dengan warna hitam yang menunjukkan objek sedang melakukan inisialisasi dan sekaligus masuk ke dalam sebuah status awal atau mula-mula. Status mula-mula akan berpindah atau bertransisi ke status berikutnya jika dipicu oleh suatu *event*. *Event* tersebut dapat memicu aksi untuk mengaktifkan objek lainnya sehingga objek lain tersebut dapat berpindah status. Dinamika objek tersebut dapat berakhir jika suatu kondisi selesai terpenuhi.

### b. *Flowchart*

*Flowchart* adalah suatu bagan yang berisi simbol-simbol grafik dari suatu algoritma [7]. Algoritma sendiri merupakan kumpulan langkah-langkah perintah berhingga dan berurutan yang merepresentasikan spesifikasi rancangan program yang akan disandi untuk menyelesaikan suatu masalah tertentu menggunakan komputer. Beberapa dari simbol tersebut adalah antara lain; simbol berbentuk oval, simbol berbentuk jajaran genjang, simbol berbentuk persegi panjang, simbol berbentuk ketupat, serta garis alur (Gambar 2).



Gambar 2. Flowchart rancangan suatu program.

Bagan berbentuk oval adalah simbol yang merepresentasikan titik permulaan dan titik akhir dari suatu algoritma. Bagan berbentuk jajaran genjang merepresentasikan tentang sesuatu yang akan dimasukkan untuk diproses atau sesuatu yang akan ditampilkan sebagai hasil dari suatu proses. Selanjutnya, bagan berbentuk persegi panjang adalah simbol yang melambangkan tempat proses dari sesuatu yang telah dimasukkan (nilai dari suatu variabel) berlangsung. Bagan berbentuk belah ketupat menyatakan tempat untuk menguji benar atau salahnya nilai dari suatu kondisi. kemudian hasilnya ditampilkan.

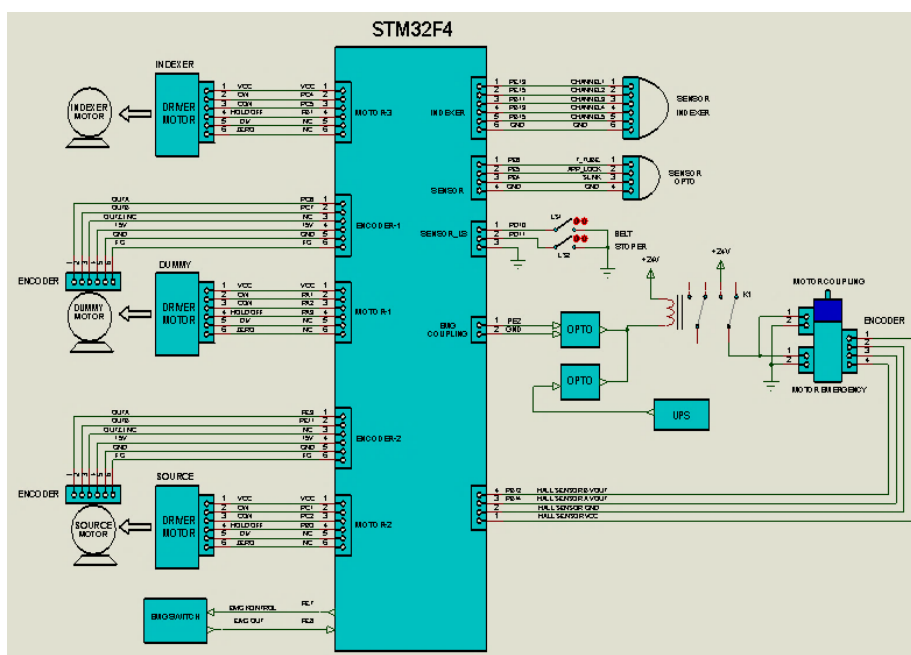
### 3. TATA KERJA

#### 3.1. Deskripsi Umum Permasalahan

Treatment Control Unit (TCU) adalah bagian dari sistem mesin braketerapi tempat mengendalikan sumber radiasi yang digunakan pada saat proses terapi berlangsung. Untuk mengendalikan sumber radiasi tersebut, TCU dilengkapi oleh mikrokontroler STM32F407VGT6 dan sejumlah komponen yang terhubung dengannya yang berfungsi sebagai aktuator dan sensor (Gambar 3). Aktuator-aktuator tersebut terdiri dari motor penggerak sling sumber, motor penggerak sling dummi, motor penggerak indeks, *switch* darurat, dan kopling darurat. Selanjutnya, sensor-sensor yang terhubung dengan mikrokontroler adalah *encoder* untuk motor sumber, *encoder* untuk motor dummi, sensor indeks, sensor opto, dan *encoder* motor darurat.

Permasalahan rancangan berhubungan dengan dinamika program yang akan dibuat dan ditanam di dalam mikrokontroler TCU. Dinamika program yang dimaksud adalah mikrokontroler bertugas mengatur dan mensinkronkan antara pergerakan motor-motor *stepper* dan pembacaan nilai sensor-sensor yang ada di dalam TCU. Pengaturan dan sinkronisasi tersebut dimulai ketika mesin braketerapi mulai dihidupkan.

Saat mesin braketerapi dihidupkan, mikrokontroler yang terletak di dalam TCU pertama-tama melakukan konfigurasi dan inialisasi terhadap motor-motor *stepper* yang berfungsi sebagai aktuator dan sensor-sensor yang digunakan untuk mengamati posisi dari pergerakan motor. Sebagai contoh, mikrokontroler melakukan konfigurasi keluar masuk untuk motor penggerak sling pembawa sumber radiasi dan melakukan inialisasi terhadap nilai posisi motor tersebut dengan memberi nilai 0. Pemberian nilai 0 tersebut adalah untuk memastikan bahwa sumber radiasi yang akan digunakan dalam melakukan terapi benar-benar berada di dalam kontener. Kepastian posisi nilai 0 dari motor penggerak sling sumber tersebut diperkuat oleh pembacaan nilai sensor status dari sling. Setelah posisi nilai sling sumber dipastikan bernilai 0, mikrokontroler melanjutkan konfigurasi dan inialisasi pada *encoder* sumber. Inialisasi *encoder* adalah dengan memberikan nilai awal 0 untuk mensinkronkan dengan posisi sling sumber. Proses inialisasi dan sinkronisasi berlaku untuk semua pasangan aktuator dan sensor.



Gambar 3. Deskripsi koneksi antar STM32F407VGT06, aktuator dan sensor-sensor.

Setelah inialisasi dan sinkronisasi selesai, mikrokontroler akan masuk ke mode awal *bacaRS232()*. Mode disebut juga *state*. Di mode awal, mikrokontroler menunggu perintah yang disebut *event* dari komputer *master* yang dikirim melalui komunikasi serial RS-485. Perintah tersebut akan menyebabkan mikrokontroler melakukan transisi dari *state* awal ke *state* tujuan. Pesan perintah/*event* yang mengakibatkan mikrokontroler melakukan transisi dari *state* awal ke *state* tujuan disajikan dalam bentuk tabulasi berikut; (Tabel 1).

Tabel 1. Pesan perintah/event yang menyebabkan transisi

No	State awal	State tujuan	Event	Penjelasan
1	<b>bacaRS232()</b>	<b>bacaRS232()</b>	-	Transisi/looping dari diri sendiri untuk menunggu <i>event</i>
2	<b>bacaRS232()</b>	<b>dispParam</b>	10	Transisi dari <i>state</i> awal ke tujuan terjadi akibat <i>event</i> 10
3	<b>bacaRS232()</b>	<b>mtrDummi</b>	20	Transisi dari <i>state</i> awal ke tujuan terjadi akibat <i>event</i> 20
4	<b>bacaRS232()</b>	<b>mtrSumber</b>	30	Transisi dari <i>state</i> awal ke tujuan terjadi akibat <i>event</i> 30
5	<b>bacaRS232()</b>	<b>mtrIndekser</b>	40	Transisi dari <i>state</i> awal ke tujuan terjadi akibat <i>event</i> 40
6	<b>bacaRS232()</b>	<b>mtrEmergency</b>	50	Transisi dari <i>state</i> awal ke tujuan terjadi akibat <i>event</i> 50

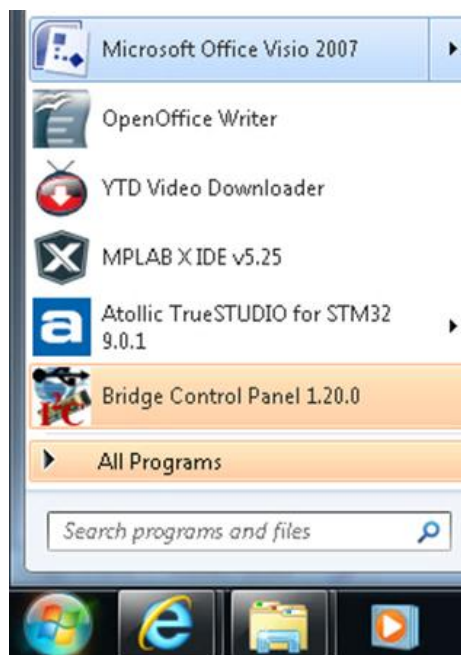
### 3.2. Alat dan Bahan

Peralatan yang digunakan dalam merancang struktur dan tingkah laku perangkat lunak TCU yaitu sebuah komputer laptop dengan *platform* minimal *Window 7 32*. Komputer *laptop* tersebut telah terpasang aplikasi *Microsoft Visio 2007* untuk menggambar *statechart* dan *flowchart*.

### 3.3. Langkah Kerja

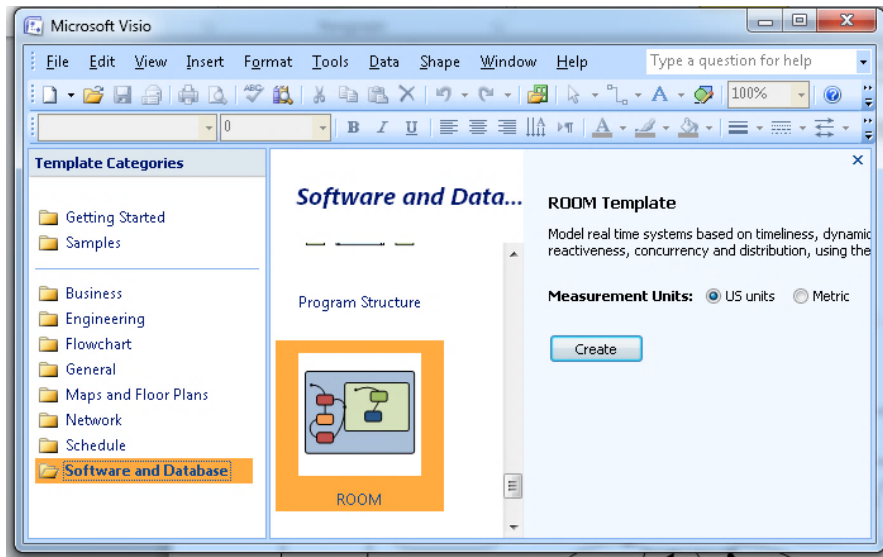
Langkah kerja Perancangan Perangkat Lunak TCU dengan Visio2017 yang terinstal pada *laptop* ber-*platform Windows 7, 32 bit Ultimate* adalah sebagai berikut:

- Pertama-tama, pengguna masuk ke Menu Start windows. Kemudian, pengguna memilih dan menjalankan Visio2017 dengan cara mengklik ganda.



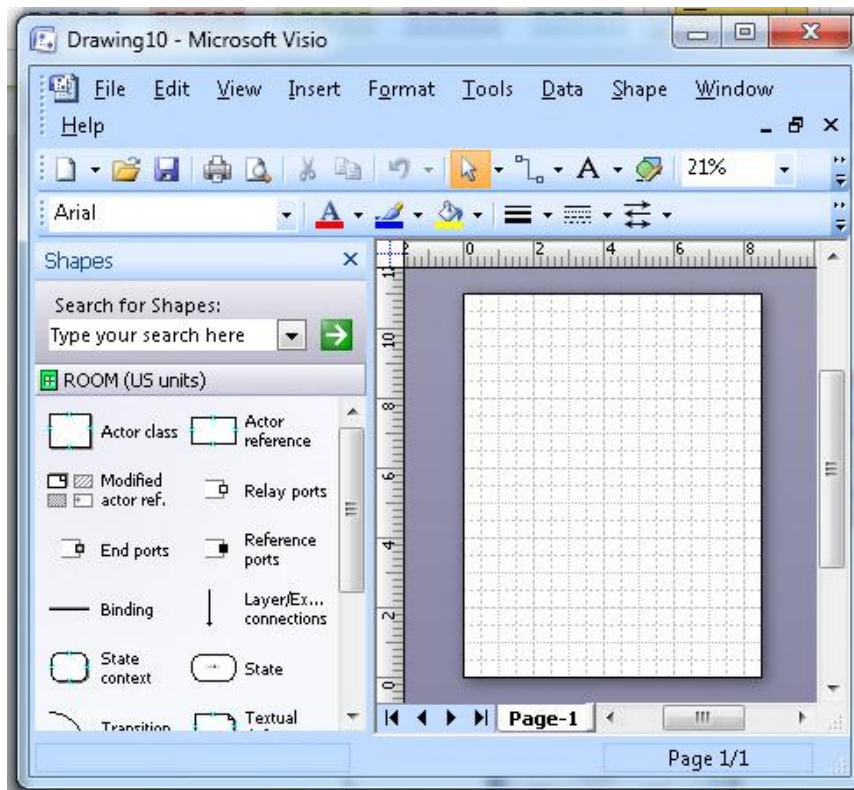
Gambar 4. Microsoft Office Visio 2007 pada Menu Start Windows 7-32 bit.

- b. Di *Template Categories* pada tampilan *editor visio 2007* (Gambar 5), Untuk menggambar *statechart*, pengguna memilih *Software and Database*. Kemudian, pengguna memilih *ROOM* pada bagian *software and database*, dan dilanjutkan dengan menekan tombol *create* pada daerah *ROOM template*.



Gambar 5. Bentuk tampilan dari layar *template ROOM*.

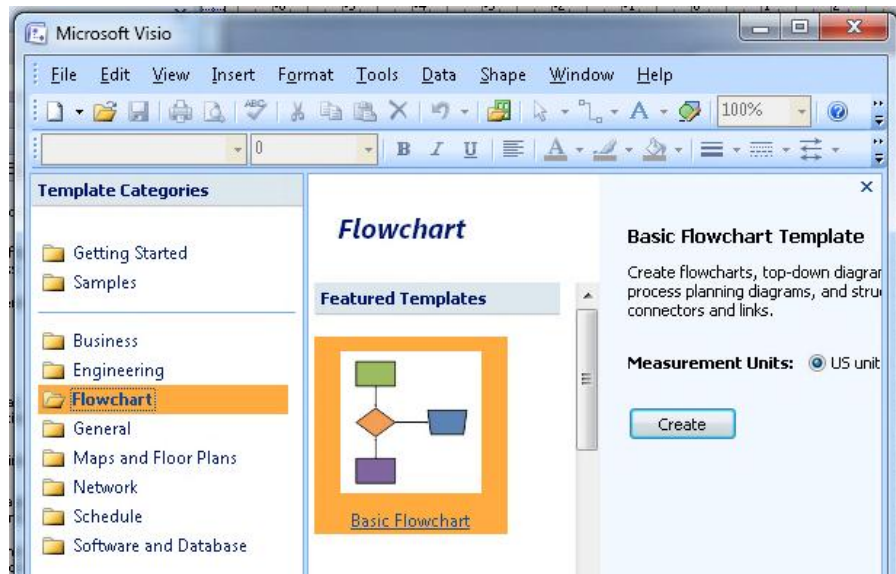
- c. Pada tahap ini, pengguna siap melakukan perancangan menggunakan simbol-simbol *statechart* yang terletak di sisi kiri layar kanvas menggambar (Gambar 6). Pengguna harus menarik simbol yang ingin digunakan ke dalam kanvas menggambar dengan cara *dragging*.



Gambar 6. Bentuk tampilan dari layar dokumen *statechart ROOM*.

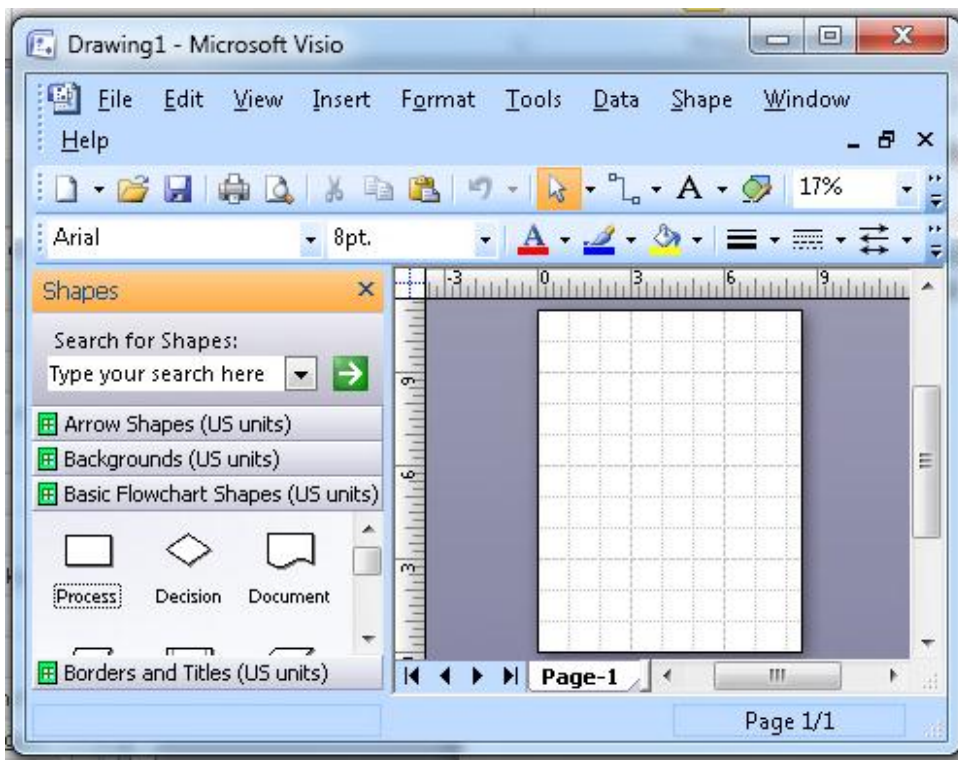


- d. Untuk menggambar *flowchart*, pengguna harus memilih *flowchart* di *Template Categories* (Gambar 7), dilanjutkan dengan memilih *basic flowchart*. Kemudian, pengguna menekan tombol *create* di sisi kanan tampilan layar *start Visio 2007*.



Gambar 7. Bentuk tampilan dari *template Basic Flowchart*.

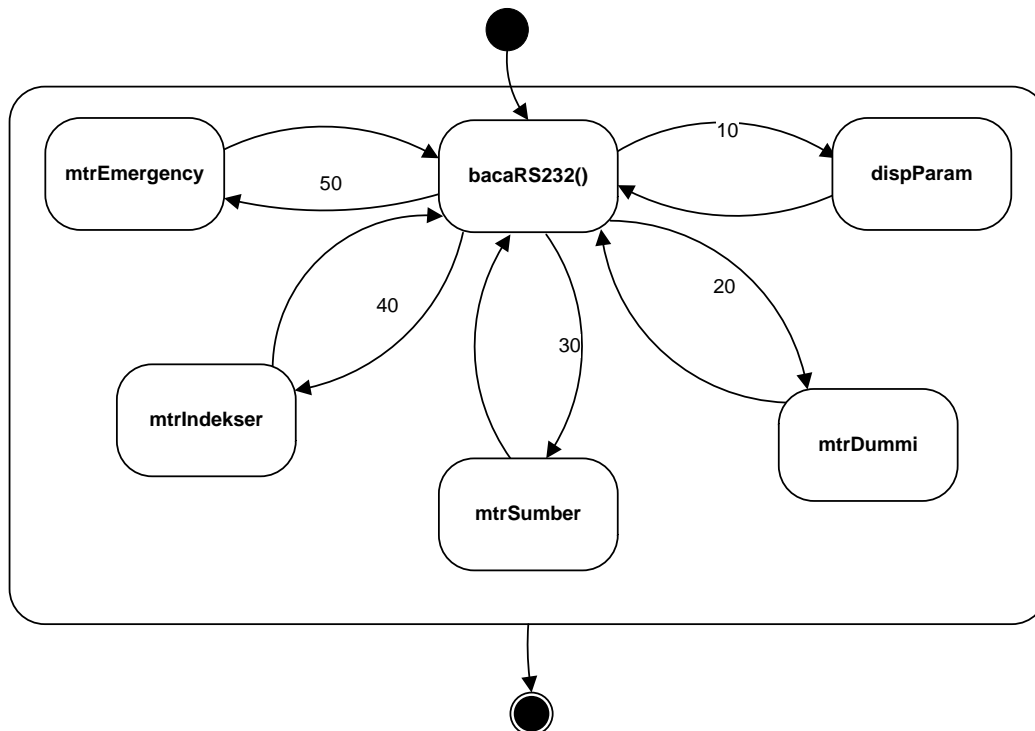
- e. Pada tahap ini, pengguna siap melakukan perancangan menggunakan simbol-simbol *flowchart* yang terletak di sisi kiri layar kanvas menggambar (Gambar 8). Pengguna harus menarik simbol yang ingin digunakan ke dalam kanvas menggambar dengan cara *dragging*.



Gambar 8. Bentuk tampilan dari layar dokumen *Basic Flowchart*.

#### 4. HASIL DAN PEMBAHASAN

Rancangan struktur dan tingkah laku perangkat lunak program yang akan ditanamkan ke dalam mikrokontroler di TCU diperlihatkan dalam Gambar 9. Gambar tersebut mendiskripsikan permasalahan sinkronisasi terhadap struktur, tingkah laku serta alur proses perangkat lunak dalam Tabel 1. Ketika bagian TCU dari brakiterapi dihidupkan, maka mikrokontroler pertama-tama akan melakukan konfigurasi terhadap perangkat keras yang terhubung dengannya, dan melakukan inisialisasi seluruh variabel yang digunakan oleh mikrokontroler. Setelah itu, mikrokontroler memasuki mode (*status*) yang disebut  **bacaRS232()**. Pada mode ini, mikrokontroler menunggu pesan dari komputer *master* di bagian *Treatment Control Station* (TCS).

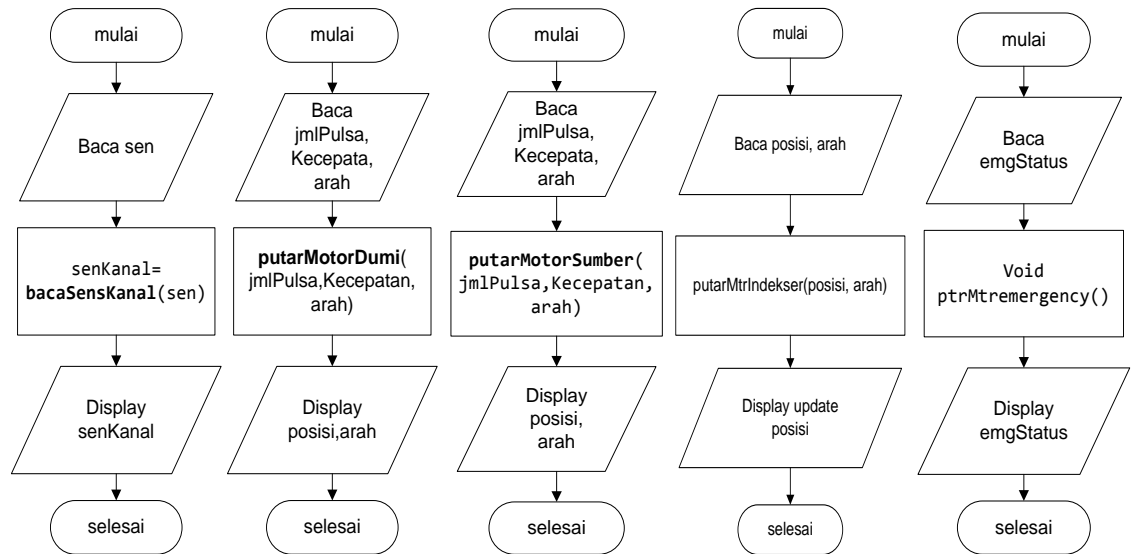


Gambar 9. Rancangan struktur dan tingkah laku program utama TCU.

Mikrokontroller akan berpindah mode dari mode  **bacaRS32()** ke mode  **disParam** ketika mikrokontroler menerima pesan instruksi bernilai 10 dari TCS. Di mode  **disParam**, mikrokontroler akan menjalankan proses  **bacaSensKanal(sen)** (Gambar 10.a). Proses ini membaca nilai variabel posisi dan arah yang terbaru dari semua aktuator, serta nilai *status* terbaru dari sensor-sensor yang terhubung ke mikrokontroler kemudian menyimpannya pada variabel *sen*. Nilai dari posisi, dan aktuator, serta nilai terbaru dari status sensor-sensor yang tersimpan di *sen* dikirim ke TCS sebelum mikrokontroler kembali ke mode  **bacaRS232()**.

Mikrokontroler akan berpindah dari mode  **bacaRS232()** ke mode  **mtrDummi** ketika mikrokontroler menerima pesan instruksi bernilai 20. Di mode  **mtrDummi**, mikrokontroler akan menjalankan proses  **putarMotorDummi(jmlPulsa, Kecepatan, arah)** (Gambar 10.b). Proses ini berfungsi untuk menggerakkan sling dummi keluar atau masuk kontener. Sling dummi digunakan untuk menguji alur dari kanal yang akan digunakan sebelum tindakan terapi dilakukan. Variabel nilai dari jumlah step pergerakan motor dummi, kecepatan, dan arah gerakan motor dummi diperbarui dan dikirim ke TCS sebelum mikrokontroler berpindah dari mode  **mtrDummi** ke mode  **bacaRS232()**.





- a. Proses pada mode **dispParam**      b. Proses pada mode **mtrDumi**      c. Proses pada mode **mtrSumber**      d. Proses pada mode **mtrIndekser**      e. Proses pada mode **mtrEmergency**

Gambar 10. Flowchart proses yang terjadi pada setiap mode ketika terjadi transisi.

Struktur dan tingkah laku mikrokontroler yang sama ketika mikrokontroler tersebut berpindah dari mode **bacaRS232()** ke mode **mtrSumber** karena mikrokontroler mendapat pesan bernilai 30 dari TCS. Di mode **mtrSumber**, mikrokontroler akan menjalankan proses **putarMotorsumber(jmlPulsa, Kecepatan, arah)** (Gambar 10.c). Proses ini berfungsi untuk menggerakkan sling sumber keluar atau masuk kontener. Sling sumber digunakan untuk tindakan terapi. Variabel nilai dari jumlah step pergerakan motor *dummy*, kecepatan, dan arah gerakan, serta *dwell time* dari motor sumber diperbarui dan dikirim ke TCS sebelum mikrokontroler berpindah dari mode **mtrSumber** ke mode **bacaRS232()**.

Mode **mtrIndekser** adalah mode dimana pengaturan kanal tempat keluar masuknya sling dilakukan dengan menggunakan perintah dari proses **putarMtrIndekser(posisi, arah)** (Gambar 10.d). Mode **bacaRS232()** akan berpindah ke mode **mtrIndekser** ketika *event* yang diterima oleh mikrokontroler bernilai 40. Variabel nilai posisi, dan arah motor indekser diperbaharui sebelum mikrokontroler kembali ke mode **bacaRS232()**.

Mode **mtrEmergency** adalah mode untuk menarik sling sumber jika terjadi *error* perangkat keras, sementara mikrokontroler bekerja secara normal. Mikrokontroler akan berpindah mode ini bila mode **bacaRS232()** menerima *event* 50. Di mode ini, mikrokontroler akan melakukan proses **putarMtrEmergency()** sebelum kembali ke mode **bacaRS232()**.

## 5. KESIMPULAN

Dari rancangan perangkat lunak menggunakan *statechart* dan *flowchart* dapat disimpulkan sebagai berikut:

- Mikrokontroler akan melakukan konfigurasi dan inisialisasi kemudian masuk ke mode **bacaRS232()** ketika pertama kali dihidupkan.
- Mikrokontroler akan berpindah dari mode **bacaRS232()** ke mode lainnya ketika menerima pesan perintah/*event* bernilai 10, 20, 30, 40, dan 50 dari TCS. Jika

- menerima nilai perintah/event selain nilai tersebut, mikrokontroler tetap pada mode **baRS232()**.
- c. Di dalam mode-mode transisi, mikrokontroler mengeksekusi proses yang berbeda sebelum kembali ke mode **baRS232()**.

## 6. DAFTAR PUSTAKA

- [1]. Anonim, " *Treatment Control Station & MicroSelectron HDR Classic. User Manual*", Nucletron, Part # 092.510ENG-07.
- [2]. Anonim, " *MicroSelectron -HDR V2 Service Training Manual. Nucletron*", Part # 190.000ENG-02.
- [3]. Jack Venselaar, José Pérez-Calatayud. " *A Practical Guide To Quality Control of Brachy- therapy Equipment*". ISBN 90-804532-8, ESTRO, 2004.
- [4]. Ian Sommerville. " *Software Engineering*". Addison-Wesley, Ninth Edition, 2011.
- [5]. Roger S. Pressman. " *Software Engineering. A Practitioner's Approach*", 7<sup>th</sup> Edition, McGraw Hill, 2010.
- [6]. Anonim, " *OOAD with UML. Object Oriented Analysis and Design Using UML*".
- [7]. Anonim, " *MS Visio. Tutorialspoints. Simply Easy Learning*" 2017, Dapat diakses di: [www.tutorialspoint.com](http://www.tutorialspoint.com).
- [8]. Peter Marwedel. " *Embedded System Design*", Springer, 2006.