

PERANCANGAN PERANGKAT LUNAK PENGUJIAN KARTU ADC PCI-1713 DAN DAC PCI-1720 SECARA SIMULTAN BERBASIS DELPHI 7 PADA PLATFORM WINDOWS XP 2003

Mohamad Amin HD
Pusat Rekayasa Perangkat Nuklir BATAN

ABSTRAK

Telah dibuat rancangan perangkat lunak untuk pengujian kartu ADC PCI-1713 dan DAC PCI-1720 secara simultan. Rancangan perangkat lunak ini terdiri dari diagram alir, diagram kelas, diagram sekuensial, diagram statecharts, diagram implementasi, dan form user interface. Hasil dari rancangan ini merupakan spesifikasi rancangan perangkat lunak yang akan diimplementasikan ke dalam bahasa pemrograman Borland Delphi versi 7.

Kata kunci : Perancangan perangkat lunak, Spesifikasi program, DAC PCI-1713, DAC PCI-1720

ABSTRACT

It has been design a software for evaluation of both electronic card ADC PCI-1713 simultaneously. The software design ed consist of a flow chart, a class diagram, asequential diagram, a state chart diagram, an implementation of the design is a specification set of the software which is designed to be implemented using Borland Delphi V7 Programming Codes.

Key word: Software design, Program specification, DAC PCI-1713, DAC PCI-1720

I. Pendahuluan

Suatu perangkat lunak yang digunakan untuk melakukan uji fungsi kartu ADC PCI-1713 dan DAC PCI-1720 telah disertakan oleh perusahaan pembuat kartu ketika kartu-kartu tersebut dibeli dan diserahkan ke pengguna. Walaupun demikian, perangkat lunak uji fungsi terhadap kartu yang disediakan oleh perusahaan tersebut bersifat sendiri-sendiri (*standalone*). Dengan kata lain, meskipun kedua kartu ADC PCI1713 dan DAC PCI1720 telah terpasang di dalam slot pci komputer secara bersama-sama, pengujian terhadap fungsi dari kedua kartu tersebut tidak dapat dilakukan secara bersamaan.

Perangkat lunak pengujian seperti itu memiliki kendala ketika dimodifikasi untuk digunakan secara simultan di dalam simulator reaktor daya yang

membutuhkan sinyal input dari kartu ADC PCI1713 untuk pengolahan sinyal, dan sinyal output dari kartu DAC PCI1713 untuk memberikan langkah aktuasi kepada simulator. Kendala tersebut adalah antara lain munculnya pesan kesalahan dan pesan kegagalan yang diakibatkan oleh penggunaan beberapa sumber daya dan perubah/variabel yang sama dari kedua kartu pada saat yang bersamaan.

Berdasarkan latarbelakang inilah, maka penulis termotivasi untuk melakukan perancangan perangkat lunak yang dapat digunakan untuk mengatasi kendala tersebut. Yakni, perancangan perangkat lunak yang dapat digunakan untuk mengirim sinyal keluaran ke kartu DAC PCI-1720 dan menerima sinyal masukan dari kartu ADC PCI-1713 pada saat kedua kartu tersebut terpasang di slot PCI komputer secara bersamaan.

Hasil dari perancangan ini dimaksudkan untuk di implementasikan

dalam bahasa pemrograman Borland Delphi 7 dan dapat dieksekusi di atas platform windows XP 2003.

Wilayah cakupan tulisan ini dibatasi pada perancangan perangkat lunaknya saja. Dengan kata lain, source code dari program yang terdiri dari interface dan implementasinya, serta hasil pengujian tidak dibahas di dalam tulisan ini. Perancangan perangkat lunak pengujian kartu ADC PCI-1713 dan DAC- PCI1720 menggunakan diagram alir (*flowchart*), diagram kelas (*class diagram*), diagram sekuensial (*sequential diagram*), diagram keadaan (*statecharts diagram*), diagram implementasi (*implelmentation diagram*), dan formulir (*form*) untuk antar muka pengguna (*user interface*).

Manfaat dari hasil perancangan ini adalah untuk memberikan informasi secara tidak langsung kepada staf-staf pengembangan *simulator test bed* mengenai cara merancang perangkat lunak untuk pengujian fungsi kartu ADC PCI1713 dan DAC PCI1720 yang dipasang di dalam slot-slot PCI komputer secara bersamaan.

Sebelum melanjutkan ke sesi berikutnya, penulis perlu menyatukan pemahaman tentang beberapa istilah yang digunakan dalam tulisan ini, antara lain. Istilah tersebut antara lain:

1. PCI (Peripheral Component Interconnect) yaitu suatu bentuk slot komputer standar industri dengan kecepatan alih data 32 bit per *clock*.
2. Slot PCI, yaitu tempat menancapkan kartu akuisisi data ke *motherboard* komputer.
3. Akuisisi data dalam tulisan ini dimaksudkan sebagai proses pembacaan atau penulisan data dari atau ke port-port kartu PCI.
4. *Source code*, yaitu kumpulan instruksi-instruksi dari bahasa pemrograman yang jika dieksekusi akan menjadi sebuah program aplikasi.
5. *User interface* yaitu formulir pengujian yang akan tampil ketika program dieksekusi (*running*). Formulir ini merupakan tempat berinteraksi antara pengguna

dengan komputer.

6. *Event*, yaitu sesuatu kejadian yang menyebabkan perangkat lunak melakukan aksi atau berpindah status.
7. *Port*, yaitu tempat penerimaan atau pengiriman data yang akan masuk dari atau keluar ke kartu PCI.
8. *Data* yaitu informasi berupa sinyal analog atau digital yang masuk ke atau keluar dari kartu PCI.

II. Metodologi

Metode perancangan perangkat lunak yang diuraikan dalam tulisan ini adalah sebagai berikut:

-Pertama-tama alur logika dari proses penciptaan formulir, inialisasi variabel-variabel yang akan digunakan untuk membuka, mengkonfigurasi, menulis/membaca, dan menutup kartu yang disertai dengan penutupan formulir disajikan dalam bentuk diagram alir.

-Kedua, penulis mencoba mengidentifikasi objek-objek beserta atribut dan operasi/*method* dari perangkat lunak tersebut menggunakan diagram kelas.

-Ketiga, urutan *event* mulai dari saat penciptaan formulir hingga pemusnahannya yang terjadi di dalam perangkat lunak pengujian diidentifikasi menggunakan diagram sekuensial.

-Keempat, *event* yang menyebabkan perangkat lunak melakukan aksi tertentu akan diidentifikasi menggunakan diagram keadaan.

-Kelima, untuk mengetahui nama kelompok-kelompok modul yang digunakan di dalam pengembangan perangkat lunak maka digunakan diagram implementasi.

-Yang terakhir, untuk mengetahui cara berinteraksi dengan komputer, maka antar muka pengguna dirancang menggunakan formulir yang disediakan oleh Borland Delphi 7.

III. Teori

Kartu ADC PCI-1713 adalah kartu yang digunakan untuk mengubah sinyal analog yang berasal dari instrumen

menjadi sinyal digital yang akan diolah oleh komputer. Sebaliknya, kartu DAC PCI-1720 merupakan kartu yang berfungsi untuk mengubah sinyal digital yang berasal dari komputer ke sinyal analog yang akan dikirim ke instrumen.

Pada dasarnya ada empat langkah berurutan yang mesti dilakukan dalam proses akuisisi data baik terhadap kartu ADC PCI-1713 maupun DAC PCI-1720.

Kempat langkah tersebut adalah:

1. Membuka kartu.
2. Inisialisasi/konfigurasi kartu.
3. Membaca/menulis dari/ke kartu.
4. Menutup kartu.

III.1. Membuka Kartu

Langkah pertama yang dilakukan untuk membuka kartu adalah dengan terlebih dahulu memeriksa ada atau tidak kartu yang terpasang di komputer. Fungsi yang digunakan untuk maksud tersebut adalah `DRV_DeviceGetList(DeviceList, MaxEntries, OutEntries)`, dengan `DeviceList` adalah parameter keluaran bertipe *long pointer* yang dikonversi ke *small integer*, tempat menyimpan daftar kartu yang terpasang di komputer. `MaxEntries` adalah parameter masukan bertipe *small integer* yang digunakan untuk membatasi jumlah maksimal kartu yang terpasang di komputer. `OutEntries` adalah parameter keluaran bertipe integer yang merupakan hasil konversi dari *long pointer* ke *small integer* untuk memverifikasi kesesuaian antara batas maksimal kartu yang terpasang dan jumlah kartu terpasang di komputer sebenarnya. Fungsi `DRV_DeviceGet-List(DeviceList, MaxEntries, OutEntries)` akan mengembalikan tiga nilai status, yaitu sukses, invalid, dan missing. Status sukses jika hasil verifikasi benar, invalid jika tidak menemukan kartu yang terpasang, dan missing jika proses registrasi kartu tidak benar.

Setelah proses pemeriksaan kartu berhasil, maka tahap berikutnya adalah memeriksa jumlah kartu yang

terpasang di komputer menggunakan fungsi

`DRV_DeviceGetNumOfList(NumOfDevices)`, di mana `NumOfDevices` adalah parameter keluaran bertipe *small integer* hasil konversi dari *pointer* ke *short*. Fungsi ini akan mengembalikan status sukses bila berhasil mengidentifikasi jumlah kartu yang terinstall, invalid jika kartu yang terinstall tidak ada, dan missing jika ada kesalahan registrasi kartu.

Bila jumlah kartu berhasil diidentifikasi, maka langkah selanjutnya adalah mengidentifikasi jenis kartu tersebut. Di dalam tulisan ini kartu yang diidentifikasi adalah kartu ADC PCI-1713, dan kartu DAC PCI-1720. Identifikasi jenis kartu dilakukan dengan merujuk ke `PT_DEVLIST` yang bertipe struktur, kemudian memeriksa atribut `SZDeviceName` yang menyimpan informasi nama kartu.

Kartu yang telah teridentifikasi selanjutnya dibuka dengan perintah `DRV_Device Open(DeviceNum, DriverHandle)`, dengan `DeviceNum` merupakan parameter masukan bertipe *unsigned long integer* yang digunakan untuk menangani nomor kartu, dan `DriverHandle` yang merupakan parameter keluaran bertipe *pointer* untuk menangani konfigurasi data ke atau dari kartu. Fungsi ini akan mengembalikan nilai sukses jika berhasil membuka kartu, `AllocatedFailed` jika gagal mengalokasikan memori, `DataLost` jika gagal membaca data, dan `FileFailed` jika kartu tidak bisa dibuka

III.2. Inisialisasi/Konfigurasi Kartu.

Setelah kartu berhasil dibuka, kartu tersebut kemudian dikonfigurasi berdasarkan pada tipe kartunya.

Proses konfigurasi kartu DAC PCI-1720 dilakukan dengan cara sebagai berikut:

1. Menyiapkan sebuah variabel yang merujuk (*pointer*) ke struct `PT_AOConfig (DAC)`
2. Menentukan kanal keluaran variabel tersebut.

3. Menentukan tegangan keluaran maksimum variabel tersebut
4. Mengatur tegangan keluaran minimum variabel tersebut.

Sementara itu, proses konfigurasi kartu ADC PCI-1713 dilakukan sebagai berikut:

1. Menyiapkan variabel yang merujuk (pointer) ke struct PT_AIConfig (ADC)
2. Mengatur kanal masukan dari variabel tersebut
3. Mengatur batas tegangan operasi variabel tersebut.

III.3. Membaca/Menulis dari/ke Kartu

Langkah menulis ke kartu DAC PCI-1720 baru dapat dilakukan setelah konfigurasi terhadap kartu telah berhasil dilaksanakan. Demikian pula langkah pembacaan dari kartu ADC PCI-1713.

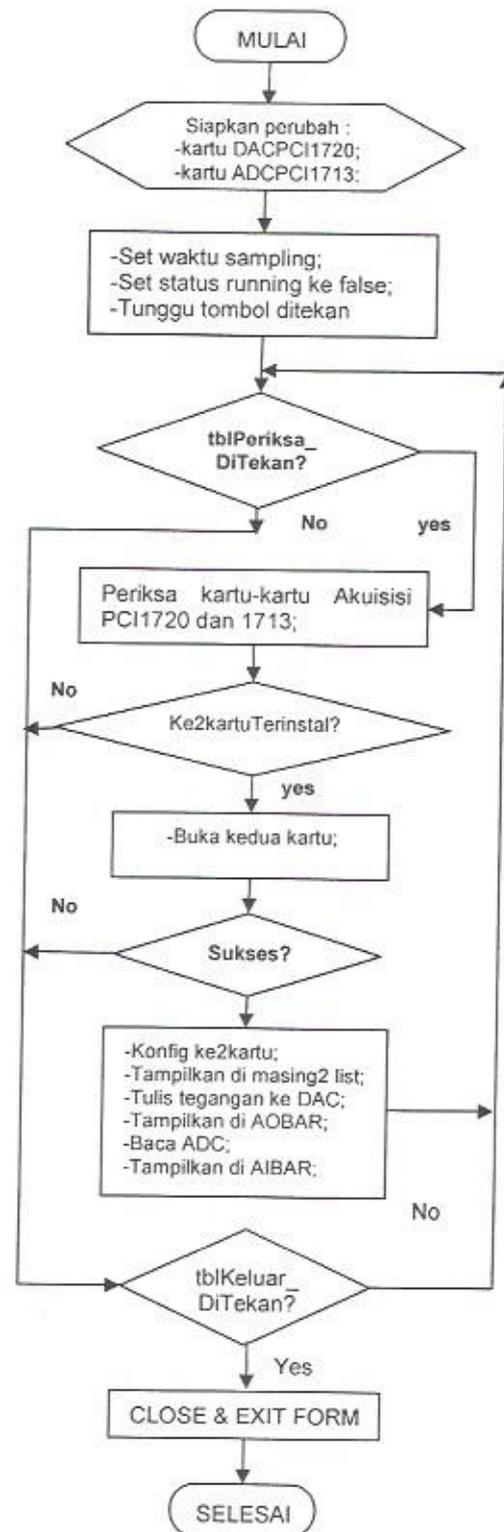
Proses penulisan ke kartu DAC PCI-1720 dilakukan dengan langkah-langkah berikut:

1. Menyiapkan variabel tipe struct AO_Config untuk menangani nomor kanal dan nilai tegangan keluaran.
2. Variabel tersebut kemudian ditugaskan untuk menyimpan nomor kanal serta nilai tegangan.
3. Variabel yang telah menyimpan nomor kanal dan nilai tegangan tersebut kemudian dimasukkan ke fungsi DRV_AOVoltageOut bersama variabel lain yang ditugaskan untuk menyimpan identitas kartu DAC PCI-1720 menggunakan perintah DRV_AOVoltageOut (handlePCI-1720, variabel)
4. Jika pengiriman berhasil, maka tegangan yang dikirim ke kartu akan dapat diukur pada kanal port hasil konfigurasi.
5. Jika tidak berhasil, maka akan muncul pesan kesalahan.

Untuk proses pembacaan data dari kartu ADC PCI-1713, maka dilakukan langkah-langkah sebagai berikut:

1. Menyiapkan variabel tipe struct PT_AIVoltageIn untuk menyimpan nomor kanal masukan, nilai penguatan, mode kontrol, dan nilai

tegangan operasi masukan.



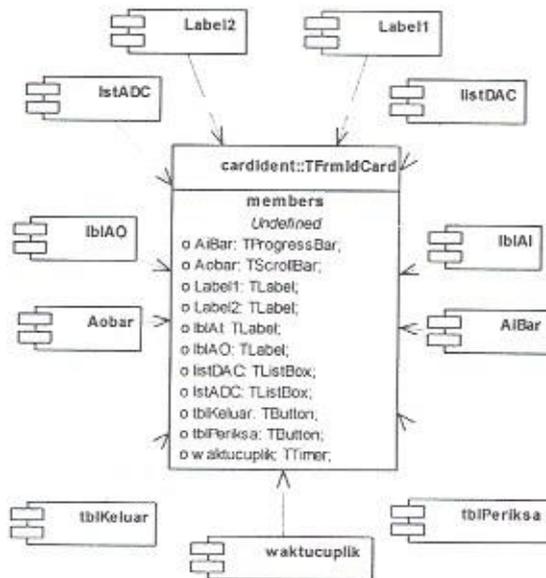
Gambar IV.1 Diagram Alir Perangkat Lunak Pengujian DAC dan ADC secara Simultan

Setelah itu, perancangan dilanjutkan menggunakan diagram kelas untuk melihat sejumlah atribut dan operasi yang terlibat di dalam pembuatan perangkat lunak pengujian tersebut. Hasil perancangan diagram kelas diperlihatkan oleh Gambar IV.2. Untuk melihat urutan event-event yang terlibat di dalam perangkat lunak pengujian serta efek yang dihasilkan oleh event-event tersebut digunakan diagram sekuensial. Rancangan diagram sekuensial ditunjukkan oleh Gambar IV.3.

Di dalam perangkat lunak pengujian ini diamati juga kemungkinan-kemungkinan terjadinya perpindahan keadaan akibat dari suatu event. Meskipun demikian, perpindahan atau perubahan keadaan yang diidentifikasi di dalam perancangan ini yaitu pada saat perangkat lunak melakukan pencuplikan dari kartu ADC PCI-1713. Hasil rancangan diagram keadaan terhadap proses pencuplikan

secara menyeluruh keterlibatan modul-modul atau komponen-komponen di dalam pembuatan perangkat lunak pengujian. Keterlibatan modul-modul atau komponen-komponen tersebut diperlihatkan oleh Gambar IV.5.

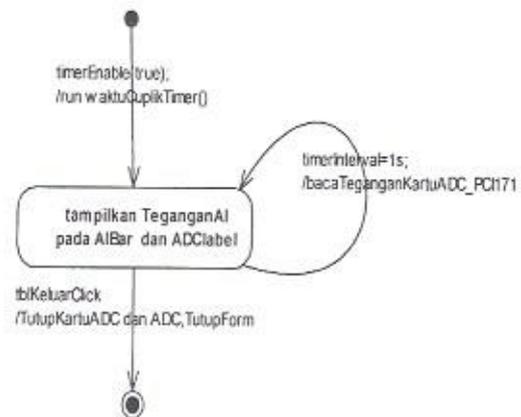
Rancangan user interface perangkat lunak pengujian ADC-PCI1713 dan DAC-PCI1720 yang merupakan hasil rancangan terakhir diperlihatkan oleh Gambar IV.6.



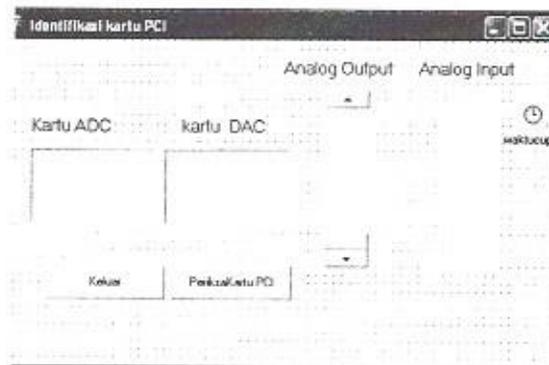
Gambar IV.4.
Diagram Implementasi perangkat lunak pengujian ADC dan DAC

diperlihatkan oleh Gambar IV.4.

Sebelum membuat rancangan *user interface*, maka terlebih dahulu dibuat diagram implementasi, untuk melihat



Gambar IV.5.
StateChart Proses Pencuplikan Data dari kartu ADC PCI-1713



Gambar IV.6.
Rancangan Antarmuka Pengguna

V. Pembahasan/Diskusi

Dari hasil perancangan perangkat lunak pengujian kartu ADC-PCI1713 dan

DAC-PCI1720 diperoleh spesifikasi rancangan sebagaimana diperlihatkan dalam bagian rancangan. Spesifikasi rancangan tersebut dalam bentuk:

1. Diagram alir,
2. Diagram kelas,
3. Diagram sekuensial,
4. Diagram statecharts,
5. Diagram implementasi, dan
6. User Interface.

V.1. Spesifikasi Diagram Alir

Diagram alir pada Gambar IV.1 memperlihatkan bahwa langkah pertama yang dilakukan untuk menguji keberadaan kartu ADC-PCI1713 dan DAC-PCI1720 secara simultan adalah dengan mempersiapkan variabel-variabel terlebih dahulu. Variabel-variabel tersebut dibagi dalam dua kelompok, yaitu variabel yang bersifat global dan variabel yang bersifat lokal. Kedua kelompok variabel tersebut tidak ditampilkan secara rinci di dalam gambar. Fungsi kedua kelompok variabel tersebut adalah untuk memeriksa, membuka, mengkonfigurasi, menulis/membaca, mengatur waktu pembacaan dan menutup kartu ADC dan DAC.

Setelah variabel-variabel tersebut dipersiapkan dan didefinisikan, maka tahap selanjutnya adalah melakukan proses pengesetan waktu pencuplikan dan status pengujian yang diikuti dengan pemunculan formulir *user interface*.

Formulir *user interface* tersebut berada pada posisi idle untuk menunggu event penekanan tombol periksa kartu atau tombol keluar. Jika tombol keluar tertekan, maka event keluar dalam kondisi benar, sehingga event tersebut akan memicu formulir *user interface* dari perangkat lunak pengujian yang sedang aktif tertutup. Pada saat yang bersamaan ditutup pula kartu ADC dan DAC. Berikutnya adalah jika tombol periksa kartu ditekan, maka perangkat lunak yang berada dalam kondisi idle tadi akan menampilkan nama kartu ADC dan DAC di listbox yang telah disediakan,

dilanjutkan dengan proses konfigurasi kartu DAC untuk keluaran dan ADC untuk masukan. Dalam kondisi seperti ini, perangkat lunak menandakan siap untuk melakukan pengujian terhadap kartu DAC maupun ADC.

Pengujian kartu DAC dilakukan dengan cara merubah-ubah posisi slider pada progress bar dari AOBAR kemudian mengukur keluaran pada nomor kanal keluaran/port yang telah ditentukan. Sementara itu, pengujian terhadap kartu ADC dilakukan dengan memberikan tegangan masukan dengan jangkauan seperti yang telah ditetapkan sebelumnya pada kanal masukan tertentu. Cara lain dalam melakukan pengujian kartu adalah dengan mengumpangkan balik tegangan keluaran dari kanal keluaran DAC ke kanal masukan ADC, sehingga ketika posisi AOBAR menunjukkan nilai tertentu AIBAR akan menampilkan nilai yang sama. Proses ini akan berlangsung secara berkala mengikuti interval waktu pencuplikan yang telah ditetapkan sebelumnya.

Proses pengujian berakhir ketika tombol keluar ditekan atau tanda silang pada ujung kanan atas form user interface diklik dengan mouse.

V.2. Diagram Kelas

Perancangan menggunakan diagram kelas digunakan untuk melihat atribut-atribut dan operasi-operasi yang kemungkinan terlibat di dalam membangun *source code* antar muka dan implementasi perangkat lunak pengujian. Perancangan menggunakan diagram kelas ini memudahkan identifikasi relasi antar objek-objek yang terlibat dalam membangun *source code*, demikian pula atribut-atribut dan operasi-operasi dari setiap objek dapat terlihat dengan jelas. Hal ini mempermudah dan mempercepat pembuatan program.

Berdasarkan Gambar IV.2 terlihat bahwa ada satu objek utama yang terlibat dalam perancangan perangkat lunak pengujian simultan terhadap kartu ADC-PCI1713 dan DAC-PCI1720. Objek utama tersebut adalah formulir *user*

interface, tempat melakukan interaksi pengujian antara penguji dan kartu yang akan diuji. Formulir *user interface* tersebut diberi nama *TFormPCICard* yang merupakan kelas turunan dari *TForm* bawaan Borland Delphi versi 7. *TForm* sendiri merupakan kelas turunan dari *TObject* bawaan Borland Delphi versi 7 juga.

TFormPCICard memiliki beberapa atribut dan operasi. Atribut-atribut yang terlihat di dalam diagram kelas tersebut adalah *tblKeluar* dan *tblPeriksaPCI* dengan tipe *TButton*, *Label1*, *Label2*, *lblAO*, *lblAI*, dengan tipe *TLabel*, *lstADC* dan *lstDAC* dengan tipe *TListBox*, *AOBAR* dengan tipe *TScrollBar*, *AIBAR* dengan tipe *TProgressBar*, dan *waktuCuplik* dengan Tipe *TTimer*. Beberapa tipe data primitif tidak ditampilkan di dalam diagram kelas tersebut, seperti tipe data string, integer, floating, dan lain-lain.

tblKeluar dirancang untuk ditempatkan di dalam formulir *user interface* agar pengguna dapat keluar dari formulir dengan cara menekan tombol tersebut sambil menutup kartu-kartu yang aktif.

tblPeriksaPCI merupakan tombol yang dirancang untuk memeriksa ada tidaknya kartu ADC dan DAC yang terpasang di komputer dan sekaligus mempersiapkan pengujian kepada pengguna. Jika kartu ADC terdeteksi, maka kartu tersebut akan ditampilkan di kotak list ADC. Demikian pula jika kartu DAC terdeteksi, maka kartu tersebut akan ditampilkan di kotak list DAC. Sebaliknya jika kartu tidak terdeteksi, maka akan muncul pesan kesalahan, dan formulir akan tertutup.

lblAO dan *lblAI*, *label1*, dan *label2* digunakan untuk menampilkan informasi masing-masing terhadap nilai AO, AI, kartu DAC, dan ADC.

Operasi-operasi yang terlihat di dalam diagram kelas yaitu, *tblPeriksaClick()*, *tblKeluarClick()*, *FormCreate()*, *AobarChange()*, dan *waktucuplik-Timer()*. Kelima operasi ini bertipe *method/ prosedur* karena tidak mengembalikan suatu nilai.

tblPeriksaClick() adalah prosedur

yang akan dikerjakan oleh perangkat lunak ketika tombol *periksaPCI* ditekan. Di dalam prosedur *tblPeriksaClick()* ini terletak *source code* untuk menginisialisasi dan menguji ketersediaan kartu.

tblKeluarClick() merupakan prosedur yang akan dikerjakan ketika tombol keluar di tekan. Di dalam prosedur ini perintah untuk menutup kartu-kartu dan formulir dituliskan.

AobarChange() yaitu prosedur yang digunakan untuk merubah-ubah nilai keluaran dari suatu kanal DAC. Di dalam prosedur inilah perintah penulisan nilai tegangan dan nomor kanal dituliskan.

FormCreate() yaitu prosedur yang digunakan untuk menginisialisasi perangkat lunak secara keseluruhan. Di dalam prosedur ini, interval waktu ditentukan, demikian juga dengan status timer diset ke kondisi salah.

Dengan demikian spesifikasi dari perangkat lunak yang dirancang menggunakan diagram kelas ini terlihat semakin spesifik, sehingga implementasi program menggunakan Borland Delphi versi 7 mulai dapat direalisasikan.

V.3. Diagram Sekuensial

Proses perancangan menggunakan diagram sekuensial dilakukan untuk memperjelas urutan dari suatu event yang terjadi di dalam perangkat lunak berdasarkan urutan waktu. Gambar IV.3 memperlihatkan diagram sekuensial dari perangkat lunak pengujian kartu ADC PCI-11713 dan DAC PCI-1720.

Pada gambar diagram sekuensial tersebut terlihat bahwa dalam membangun perangkat lunak pengujian kartu ADC dan DAC, maka yang perlu dikerjakan/diproses terlebih dahulu adalah membuat form user interface. Form user interface tersebut diberi nama *TFormldCard* yang merupakan turunan dari *TForm*. Di dalam *TFormldCard* ini di buat *lstADC*, *lstDAC*, *Label1*, *Label2*, *Aobar*, *AIBar*, *lblAO*, *lblAI*, *waktuCuplik* secara berurutan berdasarkan urutan waktu. Pada saat yang bersamaan nilai-nilai atribut dari masing-masing komponen ditetapkan.

Selanjutnya, dibuat tombol untuk memeriksa dan menampilkan kartu ADC dan DAC di listbox masing-masing. Kartu DAC PCI-1720 ditampilkan lebih dahulu, diikuti kartu ADC PCI-1713. Setelah itu, proses berlanjut pada konfigurasi kartu-kartu ADC dan DAC.

Setelah tombol periksa kartu ditekan, maka progres bar yang merupakan masukan untuk kartu DAC siap untuk diubah-ubah. Jika hal ini dilakukan maka akan terjadi perubahan juga pada label lblAO. Ketika interval waktu dari timer dicapai, maka perangkat lunak akan melakukan pencuplikan nilai tegangan ADC dan menampilkannya di AIBar dan pada label lblAI.

Tahap terakhir yaitu dibuat tombol untuk keluar dari bahasa pemrograman. Jika tombol ini ditekan, maka kartu-kartu yang telah dibuka akan ditutup bersamaan dengan penutupan form user interface.

V.4. Diagram Statecharts

Diagram statechart dibuat untuk melihat tingkah laku dari perangkat lunak pengujian kartu ADC dan DAC ketika melakukan pencuplikan data. Diagram statechart diperlihatkan oleh Gambar IV.4. Pada gambar tersebut terlihat bahwa perangkat lunak akan masuk ke keadaan pencuplikan ketika kondisi dari waktu cuplik diset ke enable. Bilamana interval timer pencuplikan telah mencapai nilai yang ditetapkan, dalam perancangan ini interval timer diset ke 1 detik, maka nilai tegangan yang berasal dari kartu ADC dibaca dan ditampilkan di AIBar dan lblAI.

Perangkat lunak selalu tetap dalam keadaan pencuplikan dengan interval waktu pencuplikan 1 detik, selama tombol keluar tidak ditekan.

Keadaan pencuplikan akan berakhir jika penguji menekan tombol klik, sehingga terjadi transisi ke keadaan keluar dari user interface pengujian, dengan terlebih dahulu menutup kartu-kartu ADC dan DAC yang aktif.

V.5. Diagram Implementasi

Diagram implementasi bertujuan untuk melihat komponen-komponen yang terlibat dalam pembentukan suatu perangkat lunak pengujian. Komponen-komponen ini diperlihatkan dengan simbol gambar seperti pada Gambar IV.5. Simbol gambar ters meskipun hanya berupa atribut atau data primitif.

Berdasarkan gambar Diagram implementasi diketahui bahwa ada sebelas kelompok source code implementasi yang terlibat dalam membangun perangkat lunak pengujian kartu ADC dan DAC secara simultan. Kesebelas kelompok source code tersebut berada didalam masing-masing komponen yaitu, di dalam 4 buah komponen label, 2 buah komponen listbox, dua buah komponen Button, sebuah komponen timer, sebuah komponen progress bar, dan sebuah komponen scrollbar.

V.6. User Interface

User interface merupakan form tempat berinteraksi antara penguji dan kartu yang akan diuji. Tanpa user interface, maka perangkat lunak pengujian akan bersifat abstrak. Oleh karena itu, form user interface merupakan bagian yang tak terpisahkan di dalam perancangan perangkat lunak pengujian kartu.

Form user interface diperlihatkan oleh Gambar IV.6. Dari gambar tersebut terlihat bahwa ketika perangkat lunak pertama kali di running, maka akan tampil form user interface dengan dua tombol, dua listbox, empat label, satu scrollbar, dan satu progress bar.

Ketika tombol periksa kartu ditekan maka nama kartu ADC akan muncul di listADC sementara itu DAC akan muncul di listDAC. Tampilnya kedua kartu tersebut menandakan bahwa kartu berhasil diidentifikasi dan Aobar yang merupakan komponen scrollbar siap diubah-ubah posisinya untuk memberikan tegangan ke kartu DAC. Reaksi atas pemberian tegangan ini akan terlihat dengan munculnya warna biru tua

dengan tinggi yang sama dengan tinggi perubahan posisi pada progress bar. Warna tersebut menunjukkan nilai tegangan masukan dari kartu ADC yang merupakan hasil umpan balik tegangan di kanal keluaran DAC ke kanal masukan ADC. Tampilan nilai pada progress bar akan diperbaharui secara periodik setiap 1 detik sekali.

Jika tombol keluar ditekan, maka kartu-kartu akan ditutup dan secara bersamaan form user interface pun ditutup.

VI. Penutup

VI.1 Kesimpulan

Dari pembahasan diketahui bahwa ada enam diagram yang digunakan untuk menentukan spesifikasi rancangan dari perangkat lunak yang digunakan untuk menguji kartu ADC dari PCI-1713 dan kartu DAC dari PCI-1720 secara simultan, yaitu: diagram alir yang lebih umum disebut *flowchart*, diagram kelas, diagram sekuensial, diagram statecharts, diagram implementasi, dan formulir *user interface*.

Ke enam alat bantu perancangan ini telah menunjukkan spesifikasi perangkat lunak pengujian kartu ADC dan DAC yang akan dibuat, sehingga bug-bug yang muncul ketika membangun *source code* dapat diteksi secara dini.

VI.2. Saran

Hasil rancangan ini hanya dapat dibangun untuk menguji kartu ADC dan DAC baik secara sendiri-sendiri maupun berdua. Namun kedua kartu harus dalam keadaan terpasang di slot PCI komputer.

Daftar Acuan Pustaka

1. Braude Eric J. "Software Design: From Programming To Architecture". John Wiley and Sons, Inc. 2004. ISBN:0-471-42920-1.
2. Quatrani Terry. "Visual Modeling With Rational Rose And UML". Addison-

Wesley. 1998, ISBN:0-201-31016-3

3. Anonim. "ADVANTECH, eAutomation, Device Driver V1.5 for Windows, User Manual". <http://www.advantech.com>.

3. Anonim. "ModelMaker Tools" BV Stenenkruis 27B, 6862 XG Oosterbeek, Netherlands

4. Anonim. Borland Delphi 7 help.

5. Anonim. Rational Enterprise Edition 2002 help.