

Pemodelan Sistem Perawatan Kesehatan Bergerak

Dian Andriana
Pusat Penelitian Informatika
dian@informatika.lipi.go.id

Abstrak

Makalah ini mendeskripsikan model formal dari sistem interaktif berdasarkan sebuah studi kasus pada sistem kesehatan bergerak. Model ini mendeskripsikan sebuah session interaktif melibatkan profesional medis yang menggunakan PDA untuk mengakses rekam medis pasien, dan terhubung pada server basis data dalam jaringan komunikasi nirkabel. Perilaku sistem dapat memunculkan masalah keselamatan bila dikaitkan dengan perilaku pemakai dalam pelayanan kesehatan. Akhirnya model sistem telah diverifikasi untuk menghindari masalah tersebut.

Kata kunci: metode formal, sistem interaktif, sistem perawatan kesehatan bergerak

1. Pendahuluan

Saat ini aplikasi komunikasi nirkabel tengah berkembang pesat baik dalam kualitas maupun kuantitas, contohnya telepon seluler dan PDA (Personal Data Assistant), baik dalam lingkup global maupun regional. Pemakai berkehendak untuk menggunakan berbagai fitur dalam konektivitas nirkabel, dan berharap untuk meningkatkan kualitas kehidupan, melalui penggunaan layanan SMS, internet, dan layanan berbasis lokasi.

Kebutuhan akan solusi untuk mengakses informasi secara efisien dengan menggunakan perangkat bergerak (*mobile*) / nirkabel terus meningkat, sebagai akibat dari peningkatan jumlah penggunaan perangkat tersebut. Pemakai berkeinginan untuk memperoleh informasi mutakhir dimana saja dan kapan saja dengan menggunakan perangkat genggam berukuran saku. Banyak perusahaan juga perlu mempublikasikan atau mempromosikan berbagai produk dan layanan. Sistem Informasi Medik dan Perawatan Kesehatan (*Medical Information and Health Care Systems*) telah berkembang dari aplikasi desktop ke aplikasi web, dan lebih jauh lagi ke aplikasi nirkabel. Para dokter dan pasien memerlukan untuk lebih sering menggunakan perangkat komunikasi bergerak. Oleh karena itu perangkat

komunikasi nirkabel/*mobile* berperan penting untuk mendukung para dokter dalam layanan medis mereka [1,8].

1.1 Sistem perawatan kesehatan bergerak

Teknologi *mobile* telah merambah industri-industri kesehatan. Sebagai contoh telah dipergunakan untuk mendistribusikan pengetahuan dan informasi medis. Di Finlandia, para dokter menjaga kemutakhiran pengetahuan dan informasi yang dimiliki untuk perawatan pasien dan memelihara tingkat kompetensi profesional menggunakan teknologi *mobile*. Sering kali mereka diharuskan menjawab pertanyaan kritis mengenai perawatan pasien. Sistem Perawatan Kesehatan Bergerak telah dikembangkan sedemikian sehingga elektronik, portabel, cepat, mudah digunakan, terhubung dengan basis data yang besar dan valid berisi pengetahuan medis dan rekam pasien (*patient records*) yang melayani para pasien dan dokter. Sistem tersebut telah dikembangkan berbasis EBMG (*Evidence Based Medical Guidelines*), menggunakan basis data XML dan berjalan di atas sistem operasi Symbian, Palm OS, dan Windows CE. Para dokter banyak mempergunakannya dengan Nokia 9210 Communicator, dengan kartu memori minimum 128 MB. Update dapat dilakukan

dengan mempertukarkan kartu memori atau melakukan download dari jaringan nirkabel GPRS/UMTS. Subsystem tersebut dapat melakukan *self-installing*, berisi *search engine*, program antarmuka pemakai, dan basis data utama [8].

Terdapat penelitian terdahulu tentang penggunaan perangkat bergerak dalam sebuah unit gawat darurat di Oregon USA. Penelitian tersebut dilaksanakan menggunakan beberapa buah PC tablet nirkabel berjalan bersama dengan PC kabel untuk registrasi pasien dan dokumentasi pusat perawatan. Unit tersebut menggunakan sistem informasi pasien terkomputerisasi EmSTAT yang diimplementasikan dengan PC kabel di setiap ruang rawat pasien, dan PC tablet nirkabel digunakan oleh perawat dengan alur kerja bergerak [1].

1.2 Usability

Terdapat beberapa isu *usability* dalam sistem bergerak [10], termasuk untuk implementasi dalam perawatan kesehatan:

- memperhatikan kesesuaian sistem untuk tujuan dan konteks yang spesifik,
- sebaik apakah sistem tersebut dan dimana masalahnya,
- bagaimana sistem harus bekerja dan tampilannya.
- Kriteria evaluasi dari sebuah sistem adalah :
- kecepatan eksekusi (secepat apakah pemakai dapat menyelesaikan tugasnya)
- kecepatan belajar (secepat apakah pemakai dapat belajar untuk menyelesaikan tugasnya)
- jumlah dan kerawanan error (seberapa sering error terjadi dan seberapa serius)
- kerja mental (apakah pemakai harus berpikir keras dan mengingat banyak informasi)
- fungsionalitas (berapa banyak tugas berbeda yang harus diselesaikan oleh pemakai)
- kualitas hasil (seberapa baik hasil dari kinerja tugas oleh pemakai)

- kehandalan (seberapa baik sistem tersebut dapat mengatasi situasi yang tidak biasa)
- kepuasan (sejauh mana pemakai menyukai penggunaan sistem)
- produktivitas (apakah sistem meningkatkan keluaran kerja / *work output* pemakai relatif terhadap masukan kerja / *work input*)
- Faktor-faktor yang harus dipertimbangkan dalam *usability* adalah sebagai berikut:
- faktor fisik (apakah ada kesulitan akan aksi pemakai secara fisik, dan apakah pesan sistem mudah diterima),
- faktor antarmuka (apakah pemakai dapat melakukan aksi dengan cara langsung dan alami, dan apakah sistem memberi informasi apa yang ingin diketahui oleh pemakai),
- faktor sosial (apakah sistem sesuai dengan konteks sosial dimana ia digunakan).

Usability adalah isu kunci dalam sistem bergerak [3,10,12,13] dan melibatkan pertimbangan keselamatan ketika sistem bergerak dipergunakan dalam implementasi medis. Para profesional medis memiliki pendekatan pragmatis dalam penggunaan sistem berbasis komputer: sebuah perangkat akan dipergunakan jika memenuhi kebutuhan, jika tidak, perangkat tersebut akan ditinggalkan dan akan cepat dicari penggantinya. Hal ini mungkin disebabkan karena bidang medis menuntut kecepatan dalam bertindak demi keselamatan [1].

1.3 Studi kasus *usability* dalam sistem perawatan kesehatan bergerak

Salah satu isu penting dalam *usability* adalah "*sebaik apakah suatu sistem, dan dimana masalahnya*" [10].

Sebelum memodelkan sistem perawatan kesehatan bergerak, dipelajari beberapa contoh skenario untuk menemukan masalah berkaitan dengan keselamatan. Masalah tersebut akan dipertimbangkan dalam pengembangan model.

Telah terdapat hasil penelitian terdahulu mengenai *usability* dalam sistem perawatan

kesehatan bergerak [11,2]. Penelitian tersebut menjelaskan dua contoh skenario dimana seorang dokter dan seorang perawat menemui seorang pasien lanjut usia di kediamannya. Dari skenario tersebut ditentukan pemakai dari sistem ini adalah para dokter dan perawat, dan mereka menggunakan perangkat PDA (*Personal Digital Assistant*). PDA tersebut digunakan untuk mengakses data pasien. Data pasien tersebut terletak dalam server basis data yang terhubung dengan PDA melalui jaringan komunikasi nirkabel yang aman (*secure*). Data pasien contohnya berupa rekam medis dan pengobatan terakhir.

Pada skenario pertama, perawat akan melakukan kunjungan pada pasien. Ia membutuhkan informasi mengenai pengobatan dan kondisi terakhir pasien tersebut. Dengan menggunakan PDAnya ia melakukan *login* ke jaringan komunikasi, dan mendapat akses ke basis data.

Pada skenario kedua, dokter sedang melakukan kunjungan pada pasien dan memutuskan untuk mengubah pengobatan. Sebuah record baru dari resep obat ditambahkan ke dalam basis data rekam medis pasien tersebut. Pada hari berikutnya pasien tersebut akan dikunjungi oleh perawat, perawat tersebut akan dapat mengetahui resep baru dalam rekam medik pasien.

Pada dua skenario diatas, pemakai sistem (yaitu dokter dan perawat) melakukan akses pada basis data rekam medik pasien. Secara umum aksi yang dilakukan terhadap data adalah membaca (*reading*), mengubah (*updating*), atau menghapus (*removing*). Dalam sistem medis, mengubah dan menghapus data memungkinkan terjadinya kesalahan yang berakibat terancamnya keselamatan. Jika kita mengasumsikan bahwa pemakai dapat menghapus data rekam medis yang salah, apakah mereka yakin bahwa mereka menghapus record yang salah, dan bukannya menghapus record yang benar? Dalam sistem medis kita tak dapat menampilkan banyak pesan konfirmasi yang dapat mengganggu

perhatian pemakai dari layanan perawatan pasien.

Agar penghapusan yang salah terhadap data dapat segera diperbaiki dengan mudah oleh pemakai, dispesifikasikan bahwa aksi penghapusan data sebagai pemberian tanda (*mark*) pada record yang akan dihapus. Record yang telah ditandai akan masih dapat dipanggil kembali oleh pemakai. Cara ini akan mencegah pemakai dari penghapusan total record yang salah.

Sebelum membaca, mengubah, atau menghapus record data pasien, pemakai dapat melakukan *browsing* atau pencarian (*searching*) ID pasien. Jika ID tersebut ditemukan, pemakai dapat melakukan aksi terhadap record pasien tersebut. Jika ID tidak ditemukan, pemakai dapat mengulangi *browsing* atau pencarian, atau menambah ID pasien baru.

Isu lainnya terkait sinkronisasi data. Jaringan komunikasi bergerak bersifat mudah terputus. Pemakai harus dapat bekerja secara terhubung (*online*) ke server atau terputus (*offline*) menggunakan data lokal dalam *cache* PDA. Data lokal dalam PDA pemakai harus disinkronisasi dengan data server ketika koneksi terhubung. Data lokal berisi data mengenai pasien tertentu yang dirawat oleh pemakai.

Juga terkait dengan sinkronisasi data, misalkan dokter melakukan perubahan data pasien, dan perawat akan membaca data pasien tersebut. Perawat tersebut perlu mengakses data mutakhir pasien sebelum melakukan kunjungan pasien. Setelah dokter selesai melakukan perubahan, ada kemungkinan perawat lupa melakukan pengambilan (*retrieve*) data mutakhir pasien. Sistem harus melakukan sinkronisasi secara periodik pada data lokal perawat dengan server basis data. Juga harus terdapat kesempatan bagi perawat untuk mengambil data ketika diperlukan.

2. Metode pemodelan sistem

Pengembangan model ini menggunakan notasi CSP dari Hoare [9] dan CWB-NC [5]. Sintax dari CWB-NC untuk CSP :

- “a -> X” berarti aksi a terjadi dan lalu X mulai;
- “X [] Y” adalah pilihan antara proses-proses X dan Y;
- “X [[S]] Y” adalah komposisi paralel dari proses-proses X and Y dengan sinkronisasi set S.

Model yang dibangun berdasarkan analisis formal pada interaksi manusia-komputer oleh Alan John Dix [7], dilanjutkan dengan pengecekan model pada sistem interaktif oleh Antonio Cerone, Peter Lindsay, dan Simon Connelly [4] serta justifikasi aturan disain berpedoman pada pemakai oleh Paul Curzon dan Ann E.Blandford [6]. Pengecekan model digunakan untuk memverifikasi apakah sistem memenuhi suatu properti (kriteria) tertentu.

2.1 Proses awal PDA and server

Berdasarkan studi kasus skenario di atas, dimodelkan sebuah sistem perawatan kesehatan yang mula-mula terdiri atas sebuah *server*, sebuah *PDA* sebagai perangkat pemakai, dan seorang pemakai (*User*) [2,11]. Pada server terdapat penyimpanan data dan berkomunikasi dengan *PDA* pemakai. Pemakai dapat bekerja *online* terhubung dengan server atau *offline* menggunakan data lokal dari *cache* *PDA*. Data dalam *cache* di-sinkronisasi secara periodik dengan data server jika koneksi *online*.

Aplikasi *PDA* dimodelkan sebagai sebuah proses *PDA* yang dimulai dengan aksi *login*. Setelah pemakai terautentikasi (aksi *valid_user*), keadaan (*state*) akan berganti menjadi keadaan *Data Synchronization*. Pada keadaan ini jika koneksi *online*, aplikasi *PDA* dapat mengirim permintaan untuk sinkronisasi data (*req_sync*) kepada Server atau menerima *broadcast* dari Server. Jika muncul permintaan untuk sinkronisasi (*req_sync*) pada proses *PDA*, aksi tersebut akan disinkronkan dengan proses Server dan memicu Server untuk menjalankan aksi *data_synchronize*. Data dalam *cache* pemakai disinkronkan dengan data Server.

Hasil sinkronisasi data ini akan mengubah *cache* (*cache_update*) lokal pada *PDA*, dan kemudian akan dilakukan *load_cache* oleh *PDA*. Jika Server tidak terhubung atau *offline* *PDA* hanya melakukan *load_cache*. Setelah aksi ini, proses *PDA* akan memasuki keadaan *Data Available*.

Pada keadaan *Data Available*, data akan tersedia bagi pemakai. Dengan data tersebut, pemakai dapat melakukan aksi-aksi seperti membaca (*read*) atau mengubah (*update*) data dari pasien tertentu. Aksi perubahan dapat berupa penambahan (*add*) rekam medis, penyuntingan (*edit*) rekam medis, penandaan untuk penghapusan (*mark for removal*) rekam medis, penghapusan tanda (*unmark for removal*), atau penambahan ID pasien baru (*add_new_ID*). Pemakai dapat pula mengubah ID pasien (*change_ID*) untuk membaca atau mengubah data pasien lain. Semua aksi perubahan akan dijalankan mula-mula pada *cache* lokal. Jika koneksi *online* *PDA* menghubungi Server untuk melakukan perubahan data Server (*server_update*), dan kemudian melakukan proses *Data Synchronization*. Sebelum aksi *server_update* dijalankan, jika Server *offline*, proses *PDA* akan kembali pada keadaan sebelumnya. Aksi-aksi pemakai ini berjalan tersinkron dengan proses Interaksi Pemakai yang dijelaskan pada subbagian berikutnya.

Pada keadaan *Data Available* pemakai juga dapat memerintahkan *PDA* untuk melakukan sinkronisasi (*user_sync*) data dengan Server. *PDA* juga dapat menjalankan sinkronisasi periodik (*periodic_sync*) dengan Server. Setelah terjadi aksi sinkronisasi atas perintah pemakai atau periodikal, *PDA* dapat menerima *broadcast* atau melanjutkan aksi permintaan untuk sinkronisasi (*req_sync*) dengan Server. Pemakai juga dapat memilih untuk *log out* dari aplikasi *PDA*, atau jika pemakai tidak melakukan aksi apapun dalam jangka waktu tertentu (*time out*), aplikasi akan me-*log out* pemakai.

Dalam proses Server, jika Server *online*, jika muncul aksi *req_sync*, Server akan menjalankan aksi *data_synchronize*. Atau, jika Server *online* dan terjadi perubahan

data Server (*server_update*) dan kemudian diikuti dengan aksi *broadcast* Server, Server juga akan menjalankan aksi sinkronisasi data. Jika Server *offline*, aksi-aksi *req_sync*, *data_synchronize*, *server update*, *broadcast* tidak terjadi.

Proses PDA berjalan paralel dengan proses Server dan tersinkron pada aksi-aksi *req_sync*, *broadcast*, *data_synchronize*, *server_update*, dan *is_offline*.

```

proc
PDA = login -> ((valid_user ->
DataSynchronization)
    [(invalid_user -> PDA)])
proc
DataSynchronization =
(req_sync -> data_synchronize ->
cache_update ->
    load_cache -> DataAvailable)
[(is_offline -> load_cache ->
DataAvailable)
[(broadcast -> data_synchronize
-> cache_update
-> load_cache -> DataAvailable)
[(logout -> PDA)
[(timeout -> logout -> PDA)
proc
DataAvailable =
(read -> DataAvailable)
[(add -> cache_update ->
((server_update ->
DataSynchronization)

[(is_offline ->
DataAvailable))]
[(edit -> cache_update ->
((server_update ->
DataSynchronization)

[(is_offline ->
DataAvailable))]
[(mark_remove -> cache_update ->
((server_update
-> DataSynchronization)

[(is_offline ->
DataAvailable))]
[(unmark_remove -> cache_update-
>((server_update
-> DataSynchronization)

[(is_offline ->
DataAvailable))]
[(add_new_ID -> cache_update ->
((server_update
-> DataSynchronization)

[(is_offline ->
DataAvailable))]

```

```

[(change_ID -> DataAvailable)
[(user_sync ->
DataSynchronization)
[(periodic_sync ->
DataSynchronization)
[(logout -> PDA)
[(timeout -> logout -> PDA)
proc
Server =
(server_update -> broadcast ->
Ready_Sync )
[(req_sync -> Ready_Sync )
[(offline -> Wait_Online)
proc
Wait_Online =
(online -> Server)
[(is_offline -> Wait_Online)
proc
Ready_Sync = (data_synchronize -
> Server)
proc
S1 = (PDA
[|{req_sync,broadcast,data_synchr
onize,server_update,
is_offline}|] Server)

```

2.2 Proses interaksi pemakai

Proses Interaksi Pemakai (*User Interaction*) berjalan tersinkron dengan proses PDA dan Server. Setelah pemakai *login* dan terautentikasi (*valid*), pemakai masuk dalam keadaan *Data Operation* sehingga dapat melakukan *browse*, *search*, *read*, *add* rekam medis, *edit* rekam medis, *mark for removal* rekam medis, *unmark for removal*, *add new ID* atau *change ID*. Pada keadaan ini pemakai juga dapat melakukan *logout*.

Aksi dalam keadaan *Data Operation* dalam proses Interaksi Pemakai berjalan paralel dan tersinkron dengan proses PDA pada keadaan *Data Available*, juga berjalan paralel dan tersinkron dengan proses Server. Sebagai contoh ketika proses Interaksi Pemakai menjalankan aksi *add*, aksi tersebut memicu proses PDA untuk menjalankan aksi *cache update*. Jika koneksi *online*, proses PDA akan menghubungi Server untuk melakukan *Server update*, dan kemudian Server melakukan *broadcast* dan aksi *data synchronization*.

Proses Interaksi Pemakai dimodelkan berjalan paralel dengan proses-proses PDA dan Server dan tersinkron pada aksi *login*, *valid_user*, *invalid_user*, *read*, *add*, *edit*,

mark_remove, *unmark_remove*, *add_new_ID*, *change_ID*, dan *logout*. Aksi pemakai *type to change ID* ditambahkan sebelum aksi *change ID* pada proses Interaksi Pemakai User Interaction untuk keperluan verifikasi sebagaimana dijelaskan pada subbagian berikutnya.

Pemakai dapat memerintahkan aksi pada proses PDA untuk sinkronisasi data (*user_sync*) dengan Server. PDA juga dapat menjalankan sinkronisasi data periodik (*periodic_sync*) dengan Server. Kedua aksi tersebut dijalankan secara asinkron dengan proses Interaksi Pemakai.

```

proc
UserInteraction = login ->
((valid_user ->
  DataOperation )

[](invalid_user ->
UserInteraction ))
proc
DataOperation =
  (browse -> BS_Result )
  [](search -> BS_Result )
  [](read -> DataOperation )
  [](add -> DataOperation )
  [](edit -> DataOperation )
  [](mark_remove ->
DataOperation )
  [](unmark_remove ->
DataOperation )
  [](type_to_change_ID ->
change_ID ->
DataOperation)
  [](logout ->
UserInteraction )
proc
BS_Result =
  (found_record ->
DataOperation)
  [](not_found_record -
>(Browse_Search
  [](add_new_ID ->
    DataOperation )))
  [](logout ->
UserInteraction )
proc
Browse_Search =
  ( browse -> BS_Result
)
  [](search -> BS_Result )
  [](logout ->
UserInteraction )
proc
S2 = (S1
  [|{login, valid_user,
invalid_user, read, edit,

```

```

mark_remove, unmark_remove,
add_new_ID,
change_ID,logout}||]
UserInteraction)

```

2.3 Hasil verifikasi properti sistem awal

Setelah memodelkan model awal, ingin diverifikasi bahwa sistem memenuhi sebuah properti. Pemakai harus dapat melakukan penggantian ID (*change ID*) tanpa menunggu aksi permintaan PDA untuk sinkronisasi data (*req_sync*), atau menunggu aksi data sinkronisasi (*data_synchronize*), pengubahan cache (*cache_update*), *broadcast*, dan pengubahan data server (*server_update*).

Aksi pemakai *wait* dimodelkan dapat terjadi sebagai akibat dari aksi-aksi *req_sync*, *data_synchronize*, *cache_update*, *broadcast*, dan *server_update*. Proses *UserWait* menggambarkan aksi *wait* terjadi akibat aksi-aksi *req_sync*, *data_synchronize*, *cache_update*, *broadcast* dan *server_update*.

Proses ini berjalan secara paralel dengan proses-proses PDA, Server, dan *UserInteraction* dan tersinkron pada aksi-aksi *req_sync*, *data_synchronize*, *cache_update*, *broadcast* dan *server_update*.

```

proc UserWait =
  (req_sync -> wait ->
UserWait )
  [](data_synchronize -> wait -
> UserWait )
  [](cache_update -> wait ->
UserWait )
  [](broadcast -> wait ->
UserWait )
  [](server_update -> wait ->
UserWait )
proc
S3 = (UserWait
  [|{req_sync,
data_synchronize,cache_update,bro
adcast,
server_update}||] S2)

```

Kemudian dua buah properti diverifikasi:

```

prop w1 = A G
  ({type_to_change_ID} ->
  (~{wait} W {change_ID}))
prop u1 = A G
  ({type_to_change_ID} ->
  (~{wait} U {change_ID}))

```

Kedua properti tidak terpenuhi dalam sistem, karena itu dilakukan modifikasi sistem sebagaimana dijelaskan pada subbagian berikutnya.

2.4 Modifikasi proses PDA

Berdasarkan diskusi pada hasil verifikasi properti awal, proses PDA dimodifikasi sebagai berikut. Jika keadaan *Data Synchronization* pada PDA diikuti aksi pada keadaan *Data Operation*, contohnya aksi data *add* muncul setelah aksi *data synchronize*, ia akan membatalkan (*abort*) aksi *cache update* pada proses PDA dan akan menyebabkan proses PDA menuju pada keadaan *Data Available*.

Pada PDA proses, baik keadaan *Data Synchronization* maupun *Data Available* dapat dibatalkan dengan aksi pemakai dalam proses *User Interaction* yaitu *read*, *add*, *edit*, *mark for removal*, *unmarking for removal*, atau *add a new patient ID*. Oleh karena itu ditambahkan aksi-aksi pemakai tersebut pada keadaan *Data Synchronization*, dan setiap aksi pemakai tersebut diikuti oleh aksi *abort*. Demikian pula aksi pemakai pada keadaan *Data Available* juga diikuti oleh aksi *abort*. Aksi *abort* akan membatalkan aksi Server untuk sinkronisasi data, dan menyebabkan kembali pada keadaan Server sebelumnya. Pada proses PDA aksi *abort* akan membatalkan proses *Data Synchronization* menyebabkan proses menuju pada keadaan *Data Available*. Ini dilakukan pada model dengan menjalankan proses PDA secara paralel dengan proses Server dan tersinkron pada aksi *abort*.

Sebagaimana telah dijelaskan, aksi *abort* ditambahkan pada proses-proses PDA dan Server, sehingga berjalan secara paralel dan tersinkron pada aksi-aksi *req_sync*, *broadcast*, *data_synchronize*, *server_update*, *is_offline*, dan *abort*.

Pada keadaan *Data Synchronization* maupun *Data Available*, pemakai dapat melakukan *logout*.

```
proc
PDA = login -> ((valid_user ->
DataSynchronization)
```

```

    [(invalid_user ->
PDA))
proc
DataSynchronization =
    (req_sync -> ((data_synchronize
->
((cache_update ->
    ((load_cache->
DataAvailable)

[(UserAbort)))

[(UserAbort)))
    [(UserAbort)))
[(is_offline -> ((load_cache ->
DataAvailable)
[(UserAbort)))
[(broadcast ->
((data_synchronize ->
((cache_update ->
((load_cache->
    DataAvailable)
    [(UserAbort)))

[(UserAbort)))
    [(UserAbort)))
[(logout -> PDA)
[(timeout -> logout -> PDA)
proc
Changes = (cache_update ->
Updated)
    [(change_ID -> abort ->
DataAvailable)
    [(abort -> DataAvailable)
proc
Updated = (is_offline ->
DataAvailable)
    [(server_update ->
DataSynchronization)
    [(change_ID -> abort ->
DataAvailable)
    [(abort -> DataAvailable)
proc
DataAvailable =
    (read -> DataAvailable)
    [(add -> ((cache_update ->
((is_offline ->
DataAvailable)
    [(server_update->
DataSynchronization)

[(UserAbort)))
    [(UserAbort)))
    [(edit -> ((cache_update ->
>
    ((is_offline->
    DataAvailable)
    [(server_update->
```

```

DataSynchronization)
[](UserAbort))
    [](UserAbort))
    [] (mark_remove ->
((cache_update ->
    ((is_offline ->
        DataAvailable)

[](server_update ->
    DataSynchronization)

[](UserAbort))
    [](UserAbort))
    [](unmark_remove ->
((cache_update ->
    ((is_offline ->
DataAvailable)

[](server_update->
    DataSynchronization)

[](UserAbort))
    [](UserAbort))
    [](add_new_ID ->
((cache_update ->
((is_offline ->
DataAvailable)

[](server_update ->
    DataSynchronization)

[](UserAbort))
    [](UserAbort))
    [] (change_ID ->
DataAvailable)
    [] (user_sync ->
DataSynchronization)
    [] (periodic_sync ->
DataSynchronization)
    [] (logout -> PDA)
    [] (timeout -> logout ->
PDA)
proc
Server =
    (server_update -> ((broadcast
-> Ready_Sync )
    [](abort ->
Server)))
    [](req_sync -> Ready_Sync )
    [](offline -> Wait_Online)
    [](abort -> Server)
proc
Wait_Online =
    (online -> Server)
    [](is_offline -> Wait_Online)
proc
Ready_Sync = (data_synchronize -
-> Server)
    [](abort -> Server)
proc
UserAbort =

```

```

(change_ID -> abort ->
DataAvailable)
    [](read -> abort ->
DataAvailable)
    [](add -> Changes )
    [](edit -> Changes )
    [](mark_remove -> Changes )
    [](unmark_remove -> Changes
)
    [](add_new_ID -> Changes )
proc
S1 = (PDA
[]{|is_offline,req_sync,data_sync
ronize,server_update,broadcast,ab
ort}|]Server)

```

2.5 Hasil verifikasi ulang properti

Verifikasi kembali dilakukan terhadap model yang telah dimodifikasi. Agar properti yang telah dijelaskan sebelumnya dapat terpenuhi oleh sistem, didefinisikan kondisi *Critical* dan *Non-Critical* dari sistem.

Sistem dimodelkan memasuki kondisi *Critical* setelah aksi pemakai *type to change ID* yang harus segera diikuti oleh aksi *change ID* karena berkaitan dengan keselamatan pasien, tanpa mengakibatkan pemakai menunggu karena aksi-aksi pemakai *req_sync*, *data_synchronize*, *cache_update*, *broadcast* dan *server_update*. Kondisi *Critical* - hingga kembali pada kondisi *Non-Critical* - tidak diijinkan diinterupsi oleh aksi-aksi ini. Dengan kata lain, aksi-aksi ini tidak diijinkan terjadi pada kondisi *Critical*. Di pihak lain, dalam kondisi *Non-Critical*, aksi-aksi ini diijinkan terjadi.

Mula-mula sistem berjalan dalam kondisi *Non-Critical*. Dalam kondisi ini aksi-aksi *req_sync*, *data_synchronize*, *cache_update*, *broadcast* dan *server_update* dapat terjadi. Jika pemakai melakukan aksi *type_to_change_ID*, sistem memasuki kondisi *Critical*. Setelah pemakai melakukan *change_ID*, sistem memasuki kondisi *Non-Critical*.

Pada kondisi *Critical* jika pemakai melakukan aksi *type_to_change_ID*, sistem kembali pada kondisi *Critical*. Setelah pemakai melakukan aksi *change_ID*, sistem memasuki kondisi *Non-Critical*.

Kemudian kondisi *Critical* dan *Non-Critical* dimodelkan sebagai sebuah proses

dimulai dengan kondisi *Non-Critical*, dan berjalan paralel dengan PDA, Server, dan proses Interaksi Pemakai, dan tersinkron pada aksi-aksi *type_to_change_ID*, *change_ID*, *req_sync*, *data_synchronize*, *cache_update*, *broadcast* dan *server_update*. Proses-proses *Critical* dan *Non-Critical* menggantikan proses *UserWait* yang tidak memenuhi properti sistem.

Aksi-aksi lain seperti *read*, *edit*, *browse*, *load cache*, dan *abort* tidak berakibat pada perubahan kondisi, sehingga tetap berjalan secara asinkron dengan kondisi *Critical* dan *Non-Critical*.

```

proc Critical =
  (type_to_change_ID ->
Critical)
  [] (change_ID -> NonCritical
)
proc NonCritical =
  (req_sync -> wait ->
NonCritical )
  [](data_synchronize -> wait ->
NonCritical )
  [](cache_update -> wait ->
NonCritical )
  [](broadcast -> wait ->
NonCritical )
  [](server_update -> wait ->
NonCritical )
  [](type_to_change_ID ->
Critical)
  [](change_ID -> NonCritical)
proc
S3 = (NonCritical
[|{type_to_change_ID,change_ID,req_sync,
data_synchronize,cache_update,broadcast,server_update
}|S2)

```

Kemudian kedua properti diverifikasi kembali, properti *u1* tidak terpenuhi oleh sistem, namun properti *w1* terpenuhi. Properti *w1* cukup untuk memverifikasi bahwa sesudah terjadi aksi *type to change ID*, tidak terjadi aksi *wait* hingga (*weak until*) terjadi aksi *change ID*.

3. Diskusi

Aksi-aksi *browse* dan *search* dapat ditambahkan pada proses *PDA* sehingga berjalan tersinkronisasi dengan aksi-aksi

yang sama pada proses *User Interaction*, serta meng-akibatkan pula terjadinya aksi *abort* pada proses *Data Synchronization*. Setelah itu dilakukan verifikasi ulang dan memberikan hasil yang sama, yaitu terpenuhinya properti *w1*.

Penelitian ini dapat dikembangkan lebih lanjut untuk properti yang lain dari sistem perawatan kesehatan bergerak, antara lain tentang keamanan dan pencegahan *deadlock*.

4. Kesimpulan

Telah dilakukan analisis formal pada sebuah sistem interaktif dengan memodelkan sebuah studi kasus sistem perawatan kesehatan bergerak yang terdiri atas proses PDA, Server, dan Interaksi Pemakai. Pengecekan model dilakukan agar sistem memenuhi sebuah properti yaitu keselamatan dalam aplikasi medis. Properti ini ditentukan berdasarkan perilaku pemakai dalam bidang medis yang menuntut kecepatan dalam bertindak karena terkait keselamatan. Penggantian ID untuk mengakses data pasien lain haruslah tidak terganggu oleh operasi yang dilakukan sistem seperti permintaan sinkronisasi, sinkronisasi data dengan server, pengubahan data lokal, pengubahan data server, dan *broadcast*.

Model awal ketiga proses tersebut telah dimodifikasi sehingga memenuhi properti tersebut. Adanya proses *Critical* dan *Non-Critical* memberi batasan (*restriction*) pada perilaku sistem sehingga memprioritaskan perilaku pemakai untuk melakukan penggantian ID. Perubahan disain sistem telah dilakukan dengan memperhitungkan perilaku pemakai sehingga memenuhi sebuah properti keselamatan.

5. Ucapan terima kasih

Kepada Kepala Pusat Penelitian Informatika Bapak Drs. Tigor Nauli, APU; dan para peneliti UNU-IIST khususnya Dr. Dang Van Hung dan Dr. Antonio Cerone, serta semua pihak yang memungkinkan terlaksananya penelitian ini.

7. Daftar pustaka

- [1] C.L. Andon. "Usability analysis of wireless tablet computing in an academic emergency department". Master's thesis, School of Medicine Oregon Health & Science University (Capstone Project), 2004. Available from:
<http://www.ohsu.edu/dmice/people/ms/2004/Andon-Capstone.pdf>
(Accessed 30.04.2006).
- [2] U.Arshad, et.al., "Exploiting mobile computing in health-care". In Michael Beigl, editor, *3rd International Workshop on Smart Appliances and Wearable Computing, Adjunct Workshop Proceedings Held at 23rd International Conference on Distributed Computing Systems (ICDCS 2003)*, pp. 19-24, University of Karlsruhe, 2003. Available from:
<http://www.cs.ucl.ac.uk/staff/c.mascolo/www/iwsawc.pdf>
(Accessed 30.04.2006).
- [3] J.I. Brevik et.al., "Medical informatics testing framework (mit-f)", 2003. Available from:
<http://www.sintef.no/upload/IKT/9013/EvacSys Evaluation Whitepaper.pdf>
(Accessed 30.04.2006).
- [4] A.Cerone, et.al., "Formal analysis of human-computer interaction using model-checking". In Bernhard Aichernig and Bernhard Beckert, editors, *Proceedings of the 3rd IEEE International Conference on Software Engineering and Formal Methods*, pp. 352-361, IEEE Comp. Soc., 2005.
- [5] R.Cleaveland, et.al., *The concurrency workbench of the new century.*, 2000. Available from:
<http://www.cs.sunysb.edu/~cwb>
(Accessed 30.04.2006).
- [6] P.Curzon & A.E.Blandford, "Formally Justifying User-centred Design Rules: a Case Study on Post-Completion Errors", *Integrated Formal Methods*, vol. 2999, pp. 461-480. Berlin, Germany: Springer-Verlag, 2004.
- [7] A. J. Dix, *Formal Methods for Interactive Systems*. Academic Press, 1991.
- [8] S. Han. "Understanding User Adoption of Mobile Technology: Focusing on Physicians in Finland, Second Edition". PhD thesis, Åbo Akademi University Turku Centre for Computer Science / Institute for Advanced Management Systems Research, 2005. Available from:
<http://iamsr.abo.fi/iamsrweb/iamsr/SavePublication?pubid=312&fileid=44>
(Accessed 30.04.2006).
- [9] C. A. R. Hoare. *Communicating Sequential Processes*. International Series in Computer Science. Prentice Hall, 1985.
- [10] A. Jameson. "Human computer interaction / interactive systems". In lecture slides: International University in Germany / German Research Center for Artificial Intelligence (DFKI), September 2001. Available from:
<http://www.dfki.de/~jameson/hci/slides/hci1.pdf>
(Accessed 30.04.2006).
- [11] S. Kalaja. "Security in mobile health care work. In Seminar on Network Security: Telecommunications Software and Multimedia Laboratory. Available from:
<http://citeseer.ist.psu.edu/kalaja01security.html>
(Accessed 30.04.2006).
- [12] H.Thimbleby, et.al., "Usability analysis with markov models," *ACM Trans. Comput.-Hum. Interact.*, 8(2):99-132, 2001.
- [13] H.W. Thimbleby. "Formulating usability". *ACM SIGCHI Bulletin*, 26(2), pp. 59-64, 1994. Available from:
<http://citeseer.ist.psu.edu/thimbleby98formulating.html>
(Accessed 30.04.2006).