

27558/14/10



ITS
Institut
Teknologi
Sepuluh Nopember

RSIF
009.6
Sal
9-1
2009

TUGAS AKHIR - CI1599

ARSITEKTUR REMOTE DATA ACCESS (RDA) UNTUK APLIKASI MONITORING KESEHATAN

SITTI SALMIAH
NRP 5105 100 150

Dosen Pembimbing I
Royana Muslim I, S.Kom, M.Kom.

Dosen Pembimbing II
Darlis Herumurti, S.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2009

PERPUSTAKAAN ITS	
Tgl. Terima	11-8-2009
Terima Dari	H
No. Induk	764



TUGAS AKHIR - CI 559

ARSITEKTUR REMOTE DATA ACCESS (RDA) UNTUK APLIKASI MONITORING KESEHATAN

SITI LALIAH

NRS 5105 100 450

Dosen Pembimbing I
Rovyan Muzni, S.Kom, M.Kom

Dosen Pembimbing II
Dalis Herumuti, S.Kom

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2009

PERFORSKAN	
	No. Induk
	Tanggal Terbit
	Tempat Terbit



ITS
Institut
Teknologi
Sepuluh Nopember

UNDERGRADUATE THESES - CI1599

**A REMOTE DATA ACCESS (RDA)
ARCHITECTURE FOR HOME-MONITORING
HEALTH-CARE APPLICATION**

**SITTI SALMIAH
NRP 5105 100 150**

**First Supervisor
Royana Muslim I, S.Kom, M.Kom.**

**Second Supervisor
Darlis Herumurti, S.Kom.**

**DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institute of Technology Sepuluh Nopember
Surabaya 2009**



UNDERGRADUATE THESES - C1899

A REMOTE DATA ACCESS (RDA)
ARCHITECTURE FOR HOME-MONITORING
HEALTH-CARE APPLICATION

SITTI SALMAH
HRP 2102 100 150

First Supervisor
Karyana Muslim, S.Kom, M.Kom.

Second Supervisor
Datis Herumuti, S.Kom.

DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2009

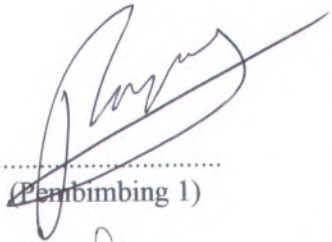
**ARSITEKTUR REMOTE DATA ACCESS (RDA)
UNTUK APLIKASI MONITORING KESEHATAN**

Diajukan Guna Memenuhi Sebagai Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer
Pada
Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya

Oleh :
SITTI SALMIAH
NRP. 5105 100 150

Disetujui oleh Pembimbing Tugas Akhir :

Royyana Muslim I, S.Kom, M.Kom.
NIP: 132320036



.....
(Pembimbing 1)

Darlis Herumurti, S.Kom.
NIP. 132306430



.....
(Pembimbing 2)

**SURABAYA
JUNI 2009**

ARTICLE 18 BAYBORN DATA ACCESS (BDA)

Bayborn Energy Services, Inc. (BES) is a subsidiary of

Bayborn Energy Services, Inc. (BES) is a subsidiary of
Bayborn Energy Services, Inc. (BES) is a subsidiary of
Bayborn Energy Services, Inc. (BES) is a subsidiary of

ARTICLE 18 BAYBORN DATA ACCESS (BDA)

Bayborn Energy Services, Inc. (BES) is a subsidiary of

Bayborn Energy Services, Inc. (BES) is a subsidiary of
Bayborn Energy Services, Inc. (BES) is a subsidiary of
Bayborn Energy Services, Inc. (BES) is a subsidiary of

Bayborn Energy Services, Inc. (BES) is a subsidiary of

ARTICLE 18 BAYBORN DATA ACCESS (BDA)

KATA PENGANTAR

Segala puji syukur penulis panjatkan ke hadirat Allah SWT atas segala limpahan rahmat, karunia, hidayah dan pertolongan-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini yang berjudul:

“ARSITEKTUR REMOTE DATA ACCESS (RDA) UNTUK APLIKASI MONITORING KESEHATAN”

Tidak lupa pada kesempatan ini penulis hendak menyampaikan penghormatan dan terima kasih yang sebesar-besarnya kepada pihak-pihak yang telah memberikan dukungan secara langsung maupun tidak langsung antara lain kepada :

1. Bapak, Ibu, Kakak dan Adik di Kendari, atas semua tuntunan, bantuan moril dan materiil serta doanya setiap saat. Mohon maaf jika selama kuliah di Surabaya saya jarang memberi kabar.
2. Bapak Royyana Muslim I, S.Kom.,M.Kom. dan Bapak Darlis Herumurti, S.Kom. atas kesabarannya memberikan bantuan dan bimbingan hingga tugas akhir ini selesai.
3. Bapak Imam Kuswardayan, S.Kom. atas semua bantuan dan kesabarannya memberikan pencerahan dalam pengerjaan tugas akhir ini.
4. Bapak Ir. Muhammad Husni, M.Kom. atas semua ilmu yang diajarkan tentang jaringan komputer serta dukungannya sehingga saya bisa mengambil tugas akhir di bidang NCC.
5. Bapak Yudhi Purwananto, S.Kom.,M.Kom. selaku Kajar Teknik Informatika yang selalu memahami kebutuhan mahasiswa, serta izin penggunaan lab NCC, lab RPL, dan PDA milik jurusan untuk keperluan tugas akhir ini.
6. Bapak Fajar Baskoro,S.Kom.,M.T. selaku dosen wali yang selalu memberikan kemudahan dalam proses perwalian dan bantuan jika terdapat masalah dalam pengambilan mata kuliah.

7. Semua admin & kru NCC: Fahim, Firman'07, Tosa'06, Ibnu'06, Syifa'07, Retno'06, Dian'06, Ardi'06, Doni'06, Mas Wahyu'04, Mas Yono'04, atas semua bantuan dan kesabarannya. Untuk Firman, maaf ya Dek, kompi-mu aku kuasai dulu selama pengerjaan tugas akhir ini.
8. Semua admin & kru RPL: Hendri, Fitrah, Ateng, Mas Bayu Krisna'04 atas bantuan teknis dan pinjaman database servernya. Untuk Hendri, terima kasih banyak sudah bersedia membantu, mohon maaf sudah banyak merepotkan.
9. Teman-teman pengajian: Mbak Zulaeha dan Mbak Rifa, terima kasih atas semua bantuannya.
10. Ratri, Fahrur, Arya, dan Mirza atas semua pencerahannya, Adit atas pinjaman PDA-nya, serta Aphanx atas laptopnya.
11. Endang, Shintami, Ajeng, Mbak Maul'03, Dian, Deni, Army, Erny, yang menjadi teman terbaik tempat berbagi cerita dan pengalaman.
12. Teman-teman kelompok BDL-ku: Nuci, Sari, Leska'06, Diah'06 dan Irma'06.
13. Ganbatteam2 Crew: Ruktin, Wanda, Wiwing, Rizky.
14. Teman seperjuangan NCC yang sidang Tahap I: Nyoman, BMW, Arie, Bayu, Wicux.
15. Semua teman-teman 2005, Mbak Erin'04, Mbak Yanti'04, Mbak Riza, serta adek-adek 2006, 2007.
16. Teman-teman GW 19A: Wasa, Ulil, Nanik, Ocha atas semua sharing dan dukungannya selama pengerjaan TA ini.
17. Semua pihak yang tidak bisa disebutkan satu per satu yang telah memberikan dukungan selama pengerjaan tugas akhir ini.

Akhir kata, manusia diciptakan dengan sempurna oleh Allah SWT. Namun ciptaan-Nya tidak mampu membuat karya yang sempurna. Karenanya penulis sangat mengharapkan saran dan kritik yang membangun dari pembaca.

Surabaya, Juni 2009

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	v
ABSTRAK.....	vii
ABSTRACT.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Permasalahan.....	2
1.3. Batasan Masalah.....	3
1.4. Tujuan Pembuatan Tugas Akhir.....	4
1.5. Metodologi Pembuatan Tugas Akhir.....	4
1.6. Sistematika Penulisan.....	5
BAB II DASAR TEORI.....	7
2.1. SQL Server Mobile.....	7
2.1.1. Fitur-Fitur SQL Server Mobile.....	7
2.1.2. Penggunaan SQL Server Mobile.....	8
2.1.3. Arsitektur SQL Server Mobile.....	9
2.1.3.1. Development Environment.....	9
2.1.3.2. Client-Server Environment.....	10
2.2. Remote Data Access (RDA).....	13
2.2.1. Penggunaan RDA.....	13
2.2.2. Arsitektur RDA.....	14
2.2.2.1. SQL Server Mobile Database Engine.....	15
2.2.2.2. SQL Server Mobile Client Agent.....	16
2.2.2.3. SQL Server Mobile Server Agent.....	16
2.2.3. Cara Kerja RDA.....	17
2.2.4. Propagate Data.....	19
2.2.4.1. Pull Data dari Server ke Client.....	19
2.2.4.2. Push Data dari Client ke Server.....	20

2.3. HyperText Transfer Protocol (HTTP)	21
BAB III PERANCANGAN PERANGKAT LUNAK	23
3.1. Deskripsi Umum Sistem	23
3.2. Perancangan Arsitektur Aplikasi	23
3.2.1. Arsitektur-1	24
3.2.2. Arsitektur-2	25
3.3. Perancangan Database Aplikasi	26
3.3.1. Database ServerMedicalRecord	26
3.3.2. Database PatientMedicalRecord	27
3.3.2.1. Database Patient01	28
3.3.2.2. Database Patient02	29
3.4. Diagram Use Case	31
3.4.1. Admin Server	31
3.4.2. Monitoring User	32
3.4.3. Pasien-1 (IP-Address Statis)	32
3.4.4. Pasien-2 (IP-Address Dinamis)	33
3.5. Diagram Activity	33
3.5.1. Create File Database (.sdf)	34
3.5.2. Update Database SQL Server	34
3.5.3. Update Database SQL Server Mobile	35
3.5.4. Pull Data	36
3.5.5. Push Data	37
3.5.6. Update Database Server melalui Web	38
BAB IV IMPLEMENTASI PERANGKAT LUNAK	39
4.1. Implementasi Arsitektur	39
4.1.1. Arsitektur-1	40
4.1.2. Arsitektur-2	42
4.2. Implementasi Database	44
4.2.1. Database ServerMedicalRecord	44
4.2.2. Database PatientMedicalRecord	45
4.2.2.1. Database Patient01	45
4.2.2.2. Database Patient02	46
4.3. Implementasi Program	48
4.3.1. Broker Server	48
4.3.1.1. Aplikasi Web	48

4.3.1.2. SQL Server Mobile Server Agent	51
4.3.1.3. Server Program	55
4.3.2. Mobile Device Monitoring-User	59
4.3.2.1. Update Database SQL Server Mobile	61
4.3.2.2. Propagate Data	63
4.3.3. Home-PC Pasien-1 (IP-Address Statis).....	67
4.3.4. Home-PC Pasien-2 (IP-Address Dinamis)	70
4.3.4.1. Update Database SQL Server Mobile	72
4.3.4.2. Propagate Data	72
BAB V UJI COBA DAN EVALUASI.....	75
5.1. Uji Coba Fungsionalitas Arsitektur-1	76
5.1.1. Uji Coba Pull Data	77
5.1.2. Uji Coba Push Data.....	81
5.2. Uji Coba Fungsionalitas Arsitektur-2	84
5.2.1. Uji Coba Pull Data	85
5.2.2. Uji Coba Push Data.....	90
5.3. Uji Coba Performa	95
5.3.1. Uji Coba Jumlah Client.....	95
5.3.2. Uji Coba Kapasitas Memory	97
BAB VI PENUTUP	99
6.1. Kesimpulan	99
6.2. Saran	100
DAFTAR PUSTAKA	101
LAMPIRAN.....	103

4.1.1	SQL Server Mobile Server Agent	21
4.1.2	Server Program	22
4.1.3	Mobile Device Monitoring	29
4.1.3.1	Lighting Database SQL Server Mobile	31
4.1.3.2	Propagate Data	33
4.1.3.3	Home-PC Edition-1 (IP-Address Status)	37
4.1.3.4	Home-PC Edition-2 (IP-Address Changes)	70
4.1.4	Mobile Outlook: SQL Server Mobile	72
4.1.5	Program Data	77
4.2	COBA DAN EVALUASI	79
4.2.1	COBA FUNGSIANALISA (ANALISA)	79
4.2.1.1	Uji Coba Full Data	77
4.2.1.2	Uji Coba Full Data	81
4.2.1.3	COBA FUNGSIANALISA (ANALISA)	84
4.2.1.4	Uji Coba Full Data	85
4.2.1.5	Uji Coba Full Data	90
4.2.1.6	Uji Coba Performa	93
4.2.1.7	Uji Coba Hasil Cetak	97
4.2.1.8	Uji Coba Hasil Alarm	97
BAB VI PENUTUP		
6.1	Kesimpulan	99
6.2	Saran	100
DAFTAR PUSTAKA		
101		
102		

DAFTAR GAMBAR

Gambar 2.1 Relasi antar komponen pada development environment	9
Gambar 2.2 Relasi antar komponen pada client-server environment	10
Gambar 2.3 Arsitektur Remote Data Access (RDA)	15
Gambar 3.1 Arsitektur-1 dengan IP-address statis pada home-PC pasien.....	24
Gambar 3.2 Arsitektur-2 dengan IP-address dinamis pada home-PC/notebook pasien	25
Gambar 3.3 Diagram CDM database ServerMedicalRecord.....	27
Gambar 3.4 Diagram CDM database Patient01	29
Gambar 3.5 Diagram CDM database Patient02	30
Gambar 3.6 Diagram use case admin server.....	31
Gambar 3.7 Diagram use case monitoring-user.....	32
Gambar 3.8 Diagram use case pasien-1 (IP-address statis).....	33
Gambar 3.9 Diagram use case pasien-2 (IP-address dinamis).....	33
Gambar 3.10 Diagram activity create file database (.sdf)	34
Gambar 3.11 Diagram activity update data SQL Server	35
Gambar 3.12 Diagram activity update data SQL Server Mobile....	35
Gambar 3.13 Diagram activity pull data.....	36
Gambar 3.14 Diagram activity push data	37
Gambar 3.15 Diagram activity update database melalui web.....	38
Gambar 4.1 Implementasi arsitektur-1	41
Gambar 4.2 Implementasi arsitektur-2	43
Gambar 4.3 Diagram PDM database ServerMedicalRecord	44
Gambar 4.4 Diagram PDM database Patient01.....	46
Gambar 4.5 Diagram PDM database Patient02.....	47
Gambar 4.6 Tampilan halaman monitoring-side web server.....	49
Gambar 4.7 Konfigurasi web.config pada monitoring side web server	49
Gambar 4.8 Tampilan halaman patient-side web server.....	50
Gambar 4.9 Connection string pada patient-side web server	51
Gambar 4.10 Connection string ke database login	51
Gambar 4.11 Tampilan url SQL Server Mobile Server Agent	54
Gambar 4.12 Source code fungsi CekDatabase().....	56
Gambar 4.13 Source code fungsi GetRDAObject()	56

Gambar 4.14 Source code create database	59
Gambar 4.15 MainForm program MonitoringUser	60
Gambar 4.16 MonitoringUserForm program MonitoringUser	60
Gambar 4.17 Source code load data SQL Server Mobile	61
Gambar 4.18 Source code add data SQL Server Mobile	62
Gambar 4.19 Source code save data SQL Server Mobile	62
Gambar 4.20 Source code edit data SQL Server Mobile	62
Gambar 4.21 Source code delete data SQL Server Mobile	63
Gambar 4.22 Connection string Patient01.sdf pada program MonitoringUser	64
Gambar 4.23 Connection string Patient02.sdf pada program MonitoringUser	64
Gambar 4.24 Source code fungsi DeleteTable()	65
Gambar 4.25 Source code fungsi ShowErrors()	66
Gambar 4.26 Source code pull data	66
Gambar 4.27 Source code push data	67
Gambar 4.28 MainForm program Pasien01	68
Gambar 4.29 PasienForm program Pasien01	69
Gambar 4.30 Source code load data SQL Server	69
Gambar 4.31 Source code save data SQL Server	70
Gambar 4.32 MainForm program Pasien02	71
Gambar 4.33 PasienForm program Pasien02	72
Gambar 4.34 Connection string pada program Pasien02	73
Gambar 5.1 Uji coba arsitektur-1	76
Gambar 5.2 Form Pasien01 pada home-PC	78
Gambar 5.3 Form Pasien01 pada home-PC setelah load data	79
Gambar 5.4 FormPasien01 pada mobile device sebelum pull data	79
Gambar 5.5 Form Pasien01 pada home-PC setelah edit data	80
Gambar 5.6 FormPasien01 pada mobile device ketika pull data	81
Gambar 5.7 FormPasien01 pada mobile device setelah pull data	81
Gambar 5.8 Form MonitoringUser01 awal pada mobile device	82
Gambar 5.9 Form MonitoringUser01 awal pada home-PC	82
Gambar 5.10 Form MonitoringUser01 pada mobile device setelah edit data	83
Gambar 5.11 Form MonitoringUser01 pada mobile device ketika push data	83
Gambar 5.12 Form MonitoringUser01 pada home-PC setelah push data	84
Gambar 5.13 Uji coba arsitektur-2	85

Gambar 5.14 Form Pasien02 pada mobile device	86
Gambar 5.15 Form Pasien02 pada mobile device setelah load data	86
Gambar 5.16 Form Pasien02 awal pada home-PC	87
Gambar 5.17 Form Pasien02 pada home-PC setelah edit data	88
Gambar 5.18 Push data Pasien02 pada home-PC.....	89
Gambar 5.19 Form Pasien02 pada mobile device ketika pull data .	89
Gambar 5.20 Form Pasien02 pada mobile device setelah pull data	90
Gambar 5.21 Form MonitoringUser02 awal pada mobile device ...	91
Gambar 5.22 Form MonitoringUser02 awal pada home-PC	91
Gambar 5.23 Form MonitoringUser02 pada mobile device setelah edit data.....	92
Gambar 5.24 Form MonitoringUser02 pada mobile device setelah save data	92
Gambar 5.25 Form MonitoringUser02 pada mobile device ketika push data	93
Gambar 5.26 Pull data Pasien02 pada home-PC	94
Gambar 5.27 Form MonitoringUser02 pada home-PC setelah pull data	95
Gambar 5.28 Grafik perbandingan waktu transfer terhadap jumlah client	96
Gambar 5.29 Grafik perbandingan waktu transfer terhadap kapasitas memory	98

Gambar 2.29 Grafik perbandingan waktu transfer terpanjang kepada memori	88
Gambar 2.28 Grafik perbandingan waktu transfer terpendek dari memori	89
Gambar 2.27 Form Monitoring server pada home-PC setelah tutup	90
Gambar 2.26 Form data Parameter pada home-PC	91
Gambar 2.25 Form Monitoring server pada mobile device pada saat	92
Gambar 2.24 Form Monitoring server pada mobile device setelah	93
Gambar 2.23 Form Monitoring server pada mobile device pada saat	94
Gambar 2.22 Form Monitoring server pada home-PC	95
Gambar 2.21 Form Monitoring server pada mobile device	96
Gambar 2.20 Form Parameter pada mobile device setelah tutup	97
Gambar 2.19 Form Parameter pada mobile device pada saat	98
Gambar 2.18 Form data Parameter pada home-PC	99
Gambar 2.17 Form Parameter pada home-PC setelah tutup	100
Gambar 2.16 Form Parameter pada mobile device pada saat	101
Gambar 2.15 Form Parameter pada mobile device setelah	102
Gambar 2.14 Form Parameter pada mobile device	103

DAFTAR TABEL

Tabel 2.1 Action SQL Server Mobile Client Agent.....	16
Tabel 2.2 Action SQL Server Mobile Server Agent.....	17
Tabel 2.3 Data Propagation pada RDA.....	18
Tabel 4.1 Application Settings untuk PATIENT01LocalCS	55
Tabel 4.2 Application Settings untuk PATIENT02LocalCS	55
Tabel 4.3 Application Settings untuk PATIENT01ServerCS.....	57
Tabel 4.4 Application Settings untuk PATIENT02ServerCS.....	57
Tabel 4.5 Application Settings untuk PATIENT01OleDbCS.....	58
Tabel 4.6 Application Settings untuk PATIENT02OleDbCS.....	58
Tabel 5.1 Uji coba berdasarkan jumlah client	96
Tabel 5.2 Uji coba berdasarkan kapasitas memory.....	97
Tabel 6.1 Data Type Mapping	103
Tabel 6.2 Data Type Mapping - lanjutan	104

DAFTAR TABEL

Tabel 2.1. skema SQL Server Mobile Client Agent.....	16
Tabel 2.2. skema SQL Server Mobile Server Agent.....	17
Tabel 2.3. Data Propagation pada WMS.....	41
Tabel 4.1. Application Settings untuk PATIENTWORKS.....	55
Tabel 4.2. Application Settings untuk PATIENTWORKS.....	55
Tabel 4.3. Application Settings untuk PATIENTWORKS.....	57
Tabel 4.4. Application Settings untuk PATIENTWORKS.....	57
Tabel 4.5. Application Settings untuk PATIENTWORKS.....	58
Tabel 4.6. Application Settings untuk PATIENTWORKS.....	58
Tabel 5.1. Uji coba berdasarkan jumlah client.....	96
Tabel 5.2. Uji coba berdasarkan kapasitas memory.....	97
Tabel 6.1. Data Type Mapping.....	103
Tabel 6.2. Data Type Mapping - Inverted.....	104

BAB I

PENDAHULUAN

Pada bagian pendahuluan ini dijelaskan mengenai latar belakang, permasalahan, serta tujuan dari pembuatan Tugas Akhir ini. Selain itu, bagian ini juga akan menjelaskan batasan masalah, metodologi pembuatan Tugas Akhir dan sistematika pembahasan keseluruhan Tugas Akhir.

1.1. Latar Belakang

Di sebagian besar negara di dunia, persentase jumlah penduduk yang menderita penyakit kronis seperti diabetes, hipertensi, asma dan penyakit jantung, terus bertambah seiring dengan pertambahan jumlah penduduk. Perawatan terhadap pasien penderita penyakit kronis tersebut selama ini hanya dilakukan secara manual, yaitu pasien melakukan pengukuran kesehatan mereka sendiri di rumah seperti pengukuran glukosa darah pada penderita diabetes, tekanan darah pada penderita hipertensi, denyut jantung pada penderita penyakit jantung, dan *peak flow* pada penderita asma. Hasil pengukuran tersebut selanjutnya dicatat oleh pasien pada buku rekaman medik, yang selanjutnya dalam jangka waktu tertentu harus dikonsultasikan dengan juru rawat.

Seiring dengan perkembangan teknologi, rekaman medik pasien tersebut akan lebih efisien jika dikembangkan menjadi aplikasi monitoring kesehatan yang tidak lagi dilakukan secara manual. Aplikasi ini dapat menyediakan komunikasi yang lebih murah antara pasien dan juru rawat. Pasien cukup menyimpan setiap rekaman mediknya di database aplikasi pada home-PC, yang selanjutnya dapat diakses oleh juru rawat melalui internet. Dan sebaliknya, juru rawat juga dapat mengirim pesan yang berisi diagnosa dan terapi yang tepat kepada pasien juga melalui internet. Salah satu metode untuk menerapkan aplikasi monitoring kesehatan tersebut adalah dengan *remote data access* (RDA).

pada mobile device, dan Microsoft SQL Server Compact Edition 3.1 sebagai database server pada home-PC (IP-address dinamis).

4. Uji coba dan evaluasi

Pada tahap ini dilakukan uji coba terhadap sistem yang dibuat, tujuannya untuk menemukan kesalahan-kesalahan yang mungkin terjadi serta melakukan perbaikan.

5. Penyusunan paper dan buku tugas akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan untuk mengerjakan tugas akhir ini serta hasil dari implementasi aplikasi yang telah dibuat.

1.6. Sistematika Penulisan

Sistematika penulisan yang digunakan untuk menyelesaikan Tugas Akhir adalah sebagai berikut:

BAB I : PENDAHULUAN

Bab ini berisi penjelasan mengenai latar belakang, permasalahan, batasan masalah, tujuan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

BAB II : DASAR TEORI

Bab ini membahas tentang beberapa dasar teori penunjang yang berhubungan dengan pokok bahasan dalam tugas akhir ini. Dasar teori yang digunakan melingkupi teori yang menjelaskan tentang komponen-komponen penyusun aplikasi dan teori yang lain yang menunjang aplikasi.

BAB III : PERANCANGAN PERANGKAT LUNAK

Bab ini membahas desain dan perancangan perangkat lunak. Desain perangkat lunak meliputi desain arsitektur aplikasi,

desain database, desain use case untuk tiap actor, dan desain activity untuk tiap use case.

BAB IV : IMPLEMENTASI

Bab ini menjelaskan tentang implementasi pembuatan aplikasi berdasarkan hasil rancangan yang telah dibuat sebelumnya.

BAB V : UJI COBA DAN EVALUASI

Bab ini berisi penjelasan tentang uji coba dan evaluasi aplikasi. Uji coba dan evaluasi dilakukan demi kelayakan sistem dan keberhasilan sistem agar sesuai dengan tujuannya dibangun.

BAB VI : PENUTUP

Bab ini berisi kesimpulan yang dapat diambil dari pelaksanaan Tugas Akhir ini, serta saran untuk pengembangan sistem.

BAB II DASAR TEORI

2.1. SQL Server Mobile

Microsoft SQL Server 2005 Mobile Edition (SQL Server Mobile 3.0), yang versi sebelumnya dikenal dengan nama *SQL Server for Windows CE* (SQL Server CE 2.0), merupakan *compact relational database* dengan berbagai macam fitur yang didesain untuk aplikasi *smart device* dan Tablet PC. Smart device meliputi semua device yang berjalan pada platform Microsoft Windows CE 5.0, Microsoft Mobile Pocket PC 2003, Microsoft Mobile Version 5.0 Pocket PC, maupun Microsoft Mobile Version 5.0 Smart Phone.^[3]

Versi terbaru dikenal dengan SQL Server Compact 3.5 SP1 yang mendukung .NET Framework 3.5 seperti halnya Windows Mobile 2003, 5.0, dan 6.0. Tujuan utama SQL Server Mobile adalah aplikasi yang tidak terus-menerus terhubung ke jaringan (*occasionally-connected*) dan aplikasi dengan *embedded-database*.

Database SQL Server Mobile terletak pada file .sdf yang dapat mencapai ukuran 4GB. File .sdf tersebut dapat dienkripsi dengan enkripsi 128-bit untuk keamanan data. Aplikasi yang menggunakan database SQL Server Mobile tidak membutuhkan keseluruhan *path* ke file .sdf pada *connection string* ADO.NET, tetapi cukup ditentukan dengan string `|DataDirectory|<database_name>.sdf` yang menunjukkan directory letak file database .sdf berada.^[4]

2.1.1. Fitur-Fitur SQL Server Mobile

SQL Server Mobile umumnya digunakan sebagai bagian dari *deployment* aplikasi Microsoft .NET Compact Framework, tetapi juga dapat diinstal secara terpisah pada smart device. *Remote data access* (RDA) dan *merge replication* memastikan

bahwa data dari database SQL Server dapat di-*deliver* ke SQL Server Mobile pada device. Data tersebut dapat di-update secara *offline* dan kemudian disinkronisasikan kembali ke server.

SQL Server Mobile diintegrasikan dengan Microsoft .NET Compact Framework dengan menggunakan Microsoft Visual Studio .NET. Hal ini menyederhanakan pengembangan aplikasi database pada device. Dengan menggunakan *data provider* SQL Server Mobile untuk *manage* code dengan *common language runtime*, para pengembang *software mobile* dapat membangun aplikasi *extensible* dengan kemampuan manajemen data secara offline.

Fitur-fitur SQL Server Mobile meliputi:

- Compact Database Engine dan *robust query optimizer*.
- Mendukung akses *multi-user* ke database SQL Server Mobile.
- Mendukung *merge replication* dan *remote data access* (RDA).
- Integrasi *tool* SQL Server Mobile dengan Microsoft SQL Server 2005.
- Integrasi dengan Visual Studio 2005.
- *Setup and connectivity wizard*.
- Kemampuan untuk membuat database SQL Server Mobile pada komputer desktop.
- .NET Compact Framework Data Provider untuk SQL Server Mobile (System.Data.SqlServerCe).
- Mendukung ADO.NET dan OLE DB Provider untuk SQL Server Mobile.
- Subset sintaks SQL.

2.1.2. Penggunaan SQL Server Mobile

SQL Server Mobile menawarkan solusi bagi skenario akses data pada mobile device yang tidak setiap saat terhubung ke *back-end server*. Skenario mobile enterprise sering dibutuhkan

Remote data access memungkinkan bagi juru rawat sebagai monitoring-user untuk melakukan sinkronisasi data antara data medical record pasien pada home-PC dan database lokal pada mobile device-nya. Dengan adanya dukungan koneksi *broadband* saat ini yang semakin meningkat maka aplikasi monitoring kesehatan seperti ini diharapkan dapat menjadi salah satu solusi untuk memberikan pelayanan kesehatan yang lebih cepat serta dengan biaya yang lebih terjangkau.

Pada topologi aplikasi remote data access biasa, home-PC yang berada di tempat pasien harus terus-menerus terhubung ke internet untuk menyediakan layanan bagi monitoring-user. Karena home-PC terus-menerus terhubung ke internet, maka home-PC akan terancam oleh berbagai macam intrusi dan akses ilegal dari internet. Resiko keamanan tersebut akan tetap ada selama home-PC menyediakan service secara langsung bagi monitoring device melalui internet. Untuk menghindari resiko keamanan tersebut, maka diperkenalkan sebuah arsitektur remote data access untuk aplikasi monitoring kesehatan yang menggunakan sebuah 'broker server' antara mobile device monitoring-user dan home-PC pasien. Dengan adanya broker server, maka home-PC pasien tidak perlu terus-menerus terhubung ke internet untuk menyediakan layanan bagi monitoring-user sehingga dapat mengurangi resiko keamanan home-PC akibat intrusi dan akses ilegal dari internet.

1.2. Permasalahan

Permasalahan yang terjadi dalam penyelesaian tugas akhir ini adalah:

1. Bagaimana mengimplementasikan remote data access (RDA) untuk aplikasi monitoring kesehatan?
2. Bagaimana melakukan sinkronisasi data antara database pada home-PC pasien dan database pada mobile device monitoring-user?
3. Bagaimana mengimplementasikan arsitektur remote data access (RDA) untuk aplikasi monitoring kesehatan dengan

broker server sebagai perantara antara home-PC pasien dan mobile device monitoring-user?

4. Bagaimana implementasi arsitektur jika home-PC pasien memiliki IP-address statis?
5. Bagaimana implementasi arsitektur jika home-PC pasien memiliki IP-address dinamis?

1.3. Batasan Masalah

Batasan masalah dalam pembuatan tugas akhir ini adalah sebagai berikut :

1. Data kesehatan yang digunakan merupakan hasil pengukuran kesehatan yang dilakukan oleh pasien secara terpisah. Data ini selanjutnya dimasukkan ke database pada home-PC.
2. Protokol komunikasi yang digunakan antara SQL Server Mobile pada mobile device dan SQL Server pada home-PC dan broker server adalah HTTP (Hypertext Transfer Protocol).
3. Mobile device, home-PC dan broker server yang digunakan berbasis Windows dan mendukung koneksi internet.
4. Sistem Operasi yang digunakan pada PC adalah Windows XP dan Windows Server 2003.
5. Sistem Operasi yang digunakan pada mobile device adalah Windows Mobile 2003 dan Windows Mobile 5.0.
6. Skenario aplikasi yang digunakan adalah 1 monitoring user dan 2 pasien. Pasien dibedakan menjadi 2 yaitu 1 pasien dengan IP-address statis dan 1 pasien dengan IP-address dinamis.
7. Sinkronisasi data yang dilakukan hanya antara database SQL Server 2005 dan SQL Server Mobile 3.0/SQL Server CE 3.1.
8. Microsoft IIS yang digunakan sebagai web server adalah versi 5.1.
9. Program hanya dapat dijalankan pada .NET Framework 2.0 pada PC dan .NET Compact Framework 2.0 pada mobile device.

10. Program tidak dapat dijalankan pada PC dengan tipe processor x64.
11. Implementasi aplikasi dilakukan pada lingkungan intranet.

1.4. Tujuan Pembuatan Tugas Akhir

Tugas akhir ini bertujuan untuk mengimplementasikan arsitektur remote data access (RDA) dengan broker server untuk aplikasi monitoring kesehatan. Pembuatan tugas akhir ini diharapkan dapat memberikan sistem keamanan yang baik bagi komunikasi antara juru rawat sebagai monitoring user dan pasien yang membutuhkan perawatan kesehatan.

1.5. Metodologi Pembuatan Tugas Akhir

Pada bagian ini dikemukakan secara urut metodologi proses pada pengerjaan tugas akhir ini.

1 . Studi literatur

Pada tahap ini dilakukan pencarian literatur tentang SQL Server Mobile dan remote data access (RDA).

2 . Analisa dan perancangan sistem

Pada tahap ini dilakukan analisa kebutuhan sistem dan kemampuan dukungan teknologi terhadap permasalahan. Pada tahap ini juga ditentukan arsitektur sistem yang akan diimplementasikan.

3 . Implementasi

Pada tahap ini dilakukan implementasi sistem menggunakan sistem operasi Windows XP pada server dan home-PC, Windows Mobile 2003 Second Edition pada mobile device, Framework ASP.NET 2.0, Microsoft IIS 5.1 sebagai web server, Microsoft SQL Server 2005 sebagai database server pada broker server dan home-PC (IP-address statis), Microsoft SQL Server Mobile 3.0 sebagai database server

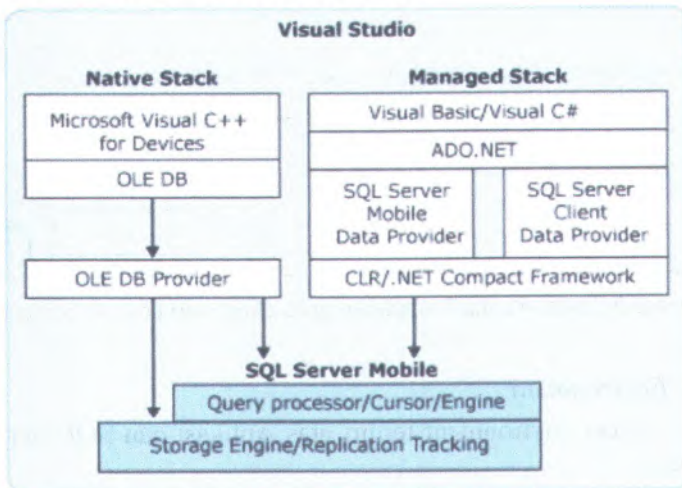
untuk bekerja dengan data jika konektivitas tidak tersedia. SQL Server Mobile menunjukkan skenario tersebut dengan menyediakan sebuah *rich relational store* yang dapat disinkronisasikan dengan SQL Server jika koneksi telah tersedia.

Misalnya, aplikasi database terdistribusi bagi mobile user dengan menggunakan SQL Server Mobile. Aplikasi ini digunakan bagi mobile user yang tidak selalu terhubung ke back-end server. Mobile user dapat melakukan perubahan pada data secara offline kemudian melakukan *propagation* perubahan data tersebut ke database asal maupun lokasi lain jika terhubung ke jaringan. Sinkronisasi dapat dilakukan pada device yang dapat mengakses server Microsoft Internet Information Service (IIS).

2.1.3. Arsitektur SQL Server Mobile

Arsitektur Microsoft SQL Server 2005 Mobile Edition (SQL Server Mobile) mencakup development environment dan client-server environment.

2.1.3.1. Development Environment



Gambar 2.1 Relasi antar komponen pada development environment

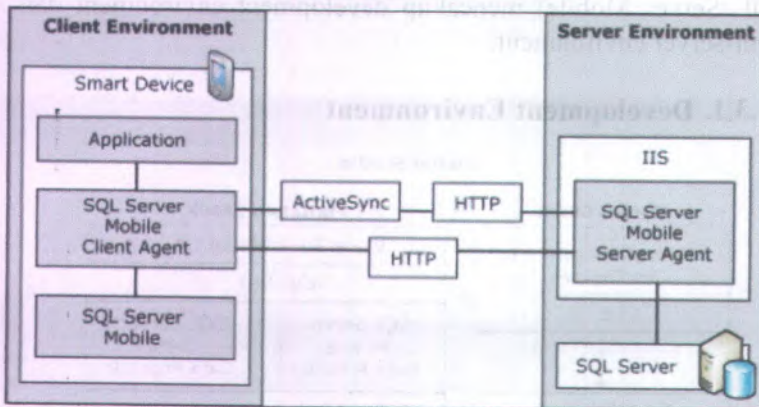
Gambar 2.1 di atas menunjukkan relasi antar komponen pada development environment.

Visual Studio 2005 menyediakan *development platform* bagi pembuatan aplikasi. Dengan menggunakan Visual Basic atau Visual C#, dapat dikembangkan aplikasi bagi device yang dapat mengakses database pada SQL Server dan SQL Server Mobile.

2.1.3.2. Client-Server Environment

Microsoft SQL Server 2005 Mobile Edition (SQL Server Mobile) bergantung pada beberapa komponen pada *client-server environment* untuk melakukan pertukaran data antara device dan instance Microsoft SQL Server.

Gambar 2.2 berikut menunjukkan relasi antara komponen-komponen client server.



Gambar 2.2 Relasi antar komponen pada client-server environment

Client Environment

Client environment terdiri atas Aplikasi dan SQL Server Mobile.

• Aplikasi

Aplikasi ini dikembangkan dengan menggunakan .NET Compact Framework dan Microsoft Visual Studio 2005, dengan bahasa pemrograman Microsoft Visual Basic atau Microsoft Visual C#, atau dengan menggunakan Microsoft Visual C++ for Device.

• SQL Server Mobile

SQL Server Mobile mencakup semua tool untuk melakukan setup, konfigurasi, konektivitas, serta akses dan modifikasi data; query processor; database storage engine; dan *programming* API yang digunakan untuk mengembangkan aplikasi yang dapat mengakses data SQL Server Mobile. Selain itu juga mencakup:

◆ SQL Server Mobile Database Engine

SQL Server Mobile Database Engine *manage* penyimpanan data SQL Server Mobile pada device. Database Engine dapat melakukan track terhadap semua record yang di-insert, di-update, maupun di-delete dengan *me-maintain* sejumlah kecil informasi *change tracking* untuk tiap record. *Tracking functionality* dapat di-set *enable* jika aplikasi menggunakan salah satu dari dua solusi konektivitas: merge-replication atau remote data access (RDA).

◆ SQL Server Mobile Client Agent

SQL Server Mobile Client Agent merupakan komponen utama bagi konektivitas pada device. Library ini mengimplementasikan objek SQL Server Mobile berikut:

- objek Replication
- objek RemoteDataAccess
- objek Engine.

Dengan menggunakan objek di atas pada aplikasi, maka koneksi ke SQL Server dapat dikontrol dengan program.

Server Environment

Server environment terdiri atas SQL Server Mobile Server Agent, Microsoft Internet Information Service (IIS), dan SQL Server.

• SQL Mobile Server Agent

SQL Server Mobile Server Agent meng-handle HTTP request yang dibuat oleh SQL Server Mobile Client Agent. SQL Server Mobile Server Agent terhubung ke SQL Server dan mengembalikan data dan schema information ke SQL Server Mobile Client Agent melalui HTTP.

• Microsoft Internet Information Service (IIS)

IIS menyediakan kemampuan Web server yang terintegrasi. IIS digunakan untuk *host* dan *manage* halaman web pada internet maupun intranet, *host* dan *manage* FTP-site, dan *routing news* maupun *mail* dengan menggunakan Network News Transport Protocol (NNTP) dan Simple Mail Transfer Protocol (SMTP). Jika digunakan dengan SQL Server Mobile, IIS menyediakan protokol dimana *device* dapat terhubung ke server untuk melakukan transfer dan pertukaran data dengan menggunakan RDA maupun *replication*.

• Microsoft SQL Server

SQL Server merupakan suatu set komponen yang bekerja secara bersamaan yang menggabungkan penyimpanan dan analisis data yang dibutuhkan oleh sebagian besar Web site dan sistem *data-processing* enterprise. SQL Server Mobile mendukung sintaks SQL yang familiar dan menyediakan sebuah *development model* dan API (Application Programming Interface) yang sesuai dengan SQL Server 2000 maupun versi setelahnya.^[3]

2.2. Remote Data Access (RDA)

Remote data access (RDA) pada SQL Server Mobile memungkinkan aplikasi untuk mengakses data dari tabel database *remote* SQL Server dan menyimpan data tersebut pada tabel database SQL Server Mobile. Aplikasi selanjutnya dapat melakukan *read* dan *update* tabel database lokal SQL Server Mobile. SQL Server Mobile juga dapat melakukan track terhadap semua perubahan yang dilakukan pada tabel lokal. Aplikasi selanjutnya dapat melakukan update pada record yang telah diubah dari tabel lokal ke tabel SQL Server.

Pada SQL Server Mobile, melakukan *propagate* data dari tabel SQL Server ke tabel lokal SQL Server Mobile disebut dengan *pull data*. Data yang telah diubah pada tabel lokal SQL Server Mobile dapat di-propagate kembali ke tabel SQL Server dan metode ini disebut dengan *push data*.

2.2.1. Penggunaan RDA

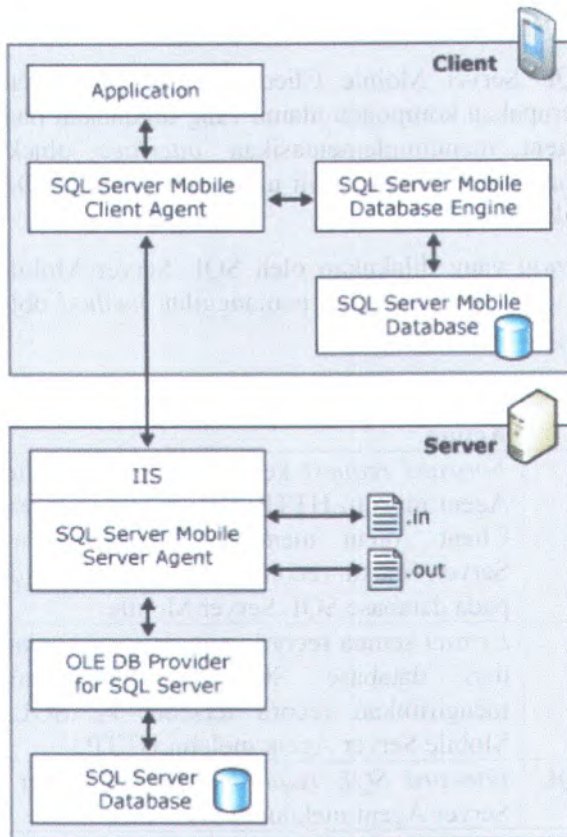
Aplikasi SQL Server 2005 Mobile yang berjalan pada mobile device dapat menggunakan remote data access (RDA) untuk:

1. *Download data*. Aplikasi dapat melakukan *pull* data dari database pada SQL Server ke database pada SQL Server Mobile.
2. *Capture dan upload data*. Aplikasi dapat melakukan *capture* data yang berasal dari *user-entered form*, *global positioning system*, *barcode reader*, dan sebagainya. Aplikasi kemudian menyimpan *captured data* tersebut pada database SQL Server Mobile pada device. Secara periodik, aplikasi akan melakukan *push* data tersebut dari database SQL Server Mobile ke database SQL Server.
3. *Download, update, dan upload data*. Aplikasi dapat melakukan *pull* data dari database SQL Server ke database SQL Server Mobile. Aplikasi kemudian dapat melakukan *update* database SQL Server Mobile. Secara periodik, aplikasi dapat melakukan *push* data yang telah di-*update* tersebut dari database SQL Server Mobile ke database SQL Server.
4. *Submit SQL statement*. Aplikasi dapat melakukan *submit SQL statement* untuk di-*execute* pada database SQL Server. Metode ini berguna jika device selalu terhubung ke jaringan.

2.2.2. Arsitektur RDA

Remote data access (RDA) menggunakan tiga komponen pada Microsoft SQL Server 2005 Mobile Edition (SQL Server Mobile) yaitu: SQL Server Mobile Database Engine, SQL Server Mobile Client Agent, dan SQL Server Mobile Server Agent.

RDA menggunakan Microsoft Internet Information Service (IIS) sebagai mekanisme komunikasi antara database SQL Server pada server dan database SQL Server Mobile pada device. SQL Server Mobile Database Engine, SQL Server Mobile Client Agent, dan SQL Server Agent ditunjukkan pada Gambar 2.3 berikut.



Gambar 2.3 Arsitektur Remote Data Access (RDA)

2.2.2.1. SQL Server Mobile Database Engine

SQL Server Mobile Database Engine mengatur *data store* SQL Server pada device. Pada operasi *pull* data yang di-*track*, Database Engine melakukan *track* terhadap semua record database yang di-insert, di-update, atau di-delete dengan cara menyimpan informasi sejumlah kecil perubahan tracking untuk tiap record. Jika SQL Server mendukung indexing pada data, maka RDA juga mendukung indexing pada data lokal.

2.2.2.2. SQL Server Mobile Client Agent

SQL Server Mobile Client Agent, yang berada pada device, merupakan komponen utama yang digunakan pada RDA. Client Agent mengimplementasikan *interface* objek RDA. Aplikasi memanggil *interface* ini untuk mengontrol RDA secara programatik.

Action yang dilakukan oleh SQL Server Mobile Client Agent sebagai respon terhadap pemanggilan *method* objek RDA dijelaskan pada tabel berikut.

Tabel 2.1 Action SQL Server Mobile Client Agent

Method	Action
Pull	<i>Forward request</i> ke SQL Server Mobile Server Agent melalui HTTP. Ketika SQL Server Mobile Client Agent menerima recordset dari SQL Server, maka recordset tersebut akan disimpan pada database SQL Server Mobile.
Push	<i>Extract</i> semua record yang di-insert/update/delete dari database SQL Server Mobile dan mengirimkan record tersebut ke SQL Server Mobile Server Agent melalui HTTP.
SubmitSQL	<i>Forward SQL request</i> ke SQL Server Mobile Server Agent melalui HTTP.

2.2.2.3. SQL Server Mobile Server Agent

SQL Server Mobile Server Agent, berada pada komputer yang menjalankan IIS, berfungsi untuk menangani HTTP request yang dibuat oleh SQL Server Mobile Client Agent. Agent ini menggunakan file *temporary messages* (*.in dan *.out) untuk mengatur pertukaran data antara SQL Server dan SQL Server Mobile.

Action yang dilakukan oleh SQL Server Mobile Server Agent sebagai respon terhadap pemanggilan *method* objek RDA dijelaskan pada Tabel 2.2 berikut.

Tabel 2.2 Action SQL Server Mobile Server Agent

Method	Action
Pull	Menerima <i>request</i> dari SQL Server Mobile Client Agent, membuat koneksi ke SQL Server melalui Provider OLE DB untuk SQL Server, kemudian melakukan <i>invoke</i> terhadap statemen SQL client. SQL Server Mobile Server Agent mengembalikan <i>recordset</i> yang dihasilkan ke SQL Server Mobile Client Agent melalui HTTP.
Push	Menerima semua <i>record</i> yang di- <i>insert</i> , di- <i>update</i> , dan di- <i>delete</i> dari SQL Server Mobile Client Agent, membuat koneksi ke SQL Server melalui OLE DB, dan melakukan insert, update dan delete record pada database SQL Server. Jika terjadi error, maka SQL Server Mobile Server Agent akan mengirim report error tersebut ke SQL Server Mobile Client Agent melalui HTTP.
SubmitSQL	Menerima <i>request</i> SQL dari SQL Server Mobile Client Agent melalui HTTP, membuat koneksi ke SQL Server melalui OLE DB, dan meng- <i>invoke</i> statement SQL client tersebut. Jika terjadi error, maka SQL Server Mobile Server Agent akan mengirim report error tersebut ke SQL Server Mobile Client Agent melalui HTTP.

2.2.3. Cara Kerja RDA

Remote Data Access memungkinkan aplikasi untuk mengakses data yang terdapat pada *remote database* SQL Server. Client pertama kali menginisiasikan *data propagation*. Data dari suatu tabel di-pull dari server ke client. Kemudian, perubahan



data pada client dapat di-*push* dari client ke server. Untuk meng-update agar data pada client sesuai dengan perubahan data pada server maka tabel tersebut harus di-*drop* kemudian di-*pull* kembali dari server.

Tabel 2.3 Data Propagation pada RDA

Method	Definisi
Pull	Extract data dari database SQL Server dan menyimpan data tersebut pada database SQL Server Mobile.
Push	Mengirim perubahan dari database SQL Server Mobile ke database SQL Server yang telah ada sebelumnya.

Aplikasi dapat menyediakan SQL query yang mengembalikan suatu rowset. Rowset yang dihasilkan ditransmisikan dari device dimana rowset disimpan pada sebuah tabel. Semua perubahan yang dibuat oleh aplikasi secara opsional dapat di-track. Jika terdapat request dari aplikasi, row yang telah di-update dikirim kembali ke server, yang selanjutnya di-apply pada database SQL Server. Rowset yang dihasilkan dapat di-filter secara horizontal maupun vertikal, tetapi pendefinisian filter ditentukan pada query SQL dari device.

RDA tidak membutuhkan konfigurasi pada server. Sehingga RDA tidak dapat mengubah schema database server. RDA merupakan solusi yang tepat bagi kebutuhan bisnis yang mencegah perubahan schema data pada server. Client harus dikonfigurasi untuk RDA secara individual melalui aplikasi yang berjalan pada device. Oleh karena itu, koneksi ini merupakan solusi terbaik bagi aplikasi yang tidak melayani banyak client atau dimana tidak terdapat banyak *client-code* yang dibutuhkan untuk *me-manage connected-data* pada aplikasi.^[3]

2.2.4. Propagate Data

Pada RDA, *propagate* data mencakup pull data dari server ke client, push data dari client ke server serta *tracking change* dan error jika dibutuhkan. Untuk melakukan propagate data dari client ke server, objek RDA dapat dikonfigurasi melalui program. Untuk mendefinisikan program RDA pada development environment Visual Studio 2005, data provider yang digunakan adalah .NET Compact Framework Data Provider for SQL Server Mobile, yang menyediakan class `System.Data.SqlServerCe.SqlCeRemoteDataAccess`.

2.2.4.1. Pull Data dari Server ke Client

Remote data access umumnya diinisiasikan dengan melakukan pull data dari Microsoft SQL Server ke database client SQL Server Mobile. Jika data di-pull dari server ke client, maka operasi pull diinisiasi oleh SQL Server Mobile. SQL Server Mobile meng-extract data dari database SQL Server dan menyimpan data tersebut pada database SQL Server Mobile.

Untuk melakukan pull data dari server, hal-hal yang perlu ditentukan ketika melakukan setting up dan inisiasi objek adalah:

1. dari database mana data tersebut akan di-pull;
2. autentikasi web server dibutuhkan untuk melakukan koneksi ke web server;
3. autentikasi SQL Server yang digunakan;
4. tabel yang di-copy dan di-propagate ke client;
5. option (pilihan) pada operasi pull mencakup *tracking* dan *error table*.

Aplikasi dapat melakukan track terhadap perubahan yang dibuat pada tabel SQL Server Mobile dengan melakukan setting pilihan tracking yang sesuai. SQL Server Mobile melakukan track terhadap semua data yang di-insert, update, dan delete. Aplikasi

selanjutnya dapat melakukan method push untuk melakukan propagate data kembali ke tabel awal SQL Server.

RDA_TRACKOPTION mendefinisikan apakah SQL Server Mobile harus melakukan track terhadap semua perubahan yang dilakukan pada tabel yang di-pull. TRACKINGON atau TRACKINGON_INDEXES digunakan jika ingin melakukan update tabel yang di-pull pada mobile device kemudian melakukan push perubahan record kembali ke tabel SQL Server. Jika menggunakan TRACKINGON, maka constraint PRIMARY KEY disertakan pada tabel yang di-pull. Jika menggunakan TRACKINGON_INDEXES, maka constraint PRIMARY KEY dan index tambahan pada tabel SQL Server disertakan pada tabel yang di-pull. Pada kedua option tersebut, SQL Server Mobile melakukan track untuk setiap record yang di-insert, update, dan delete pada *local table*.

2.2.4.2. Push Data dari Client ke Server

Push data dari client ke server meliputi propagate data dari tabel SQL Server Mobile kembali ke tabel SQL Server. Aplikasi harus telah membuat *local table* SQL Server Mobile dengan memanggil method *pull* dengan melakukan set on pada *tracking option*.

Method pull dan push yang di-track oleh RDA menggunakan *optimistic concurrency control*. SQL Server tidak me-lock semua data yang di-pull. Jika aplikasi melakukan push, record perubahan yang dibuat pada database lokal SQL Server Mobile akan di-apply pada database SQL Server. Ini dapat menyebabkan perubahan yang dibuat oleh user database SQL Server yang lain akan hilang.

RDA_BATCHOPTION mendefinisikan apakah SQL Server harus mengumpulkan (batch) record yang akan dikirim ke tabel SQL Server. Setting default adalah BATCHINGOFF, yaitu record akan di-apply ke tabel SQL Server sebagai *individual*

transaction. Keberhasilan tiap transaksi tidak dipengaruhi oleh transaksi yang lain. BATCHINGON mendefinisikan bahwa semua record akan dikirim sebagai satu transaksi tunggal. Dalam kasus ini, semua record yang dikirim harus berhasil agar method push berhasil. Jika salah satu record gagal dikirim, maka operasi push juga akan gagal, dan tidak ada record yang di-apply pada tabel SQL Server.

Baik BATCHINGON maupun BATCHINGOFF, keduanya melakukan return terhadap semua error pada error table, bukan hanya error pertama yang terjadi. Misalnya, terdapat lima record yang akan di-push dan transaksi tiga record dari kelima record tersebut gagal; jika BATCHINGON yang digunakan maka tidak ada record yang akan di-apply dan ketiga record yang gagal tersebut akan disimpan pada error table, sedangkan jika menggunakan BATCHINGOFF maka tiga record yang gagal tersebut juga akan disimpan pada error table dan dua record yang lain akan di-apply pada tabel SQL Server.

2.3. HyperText Transfer Protocol (HTTP)

HTTP digunakan sebagai protokol komunikasi data pada World Wide Web (WWW). Protokol ini menjelaskan jenis message apa yang akan dikirim oleh client ke server serta response apa yang akan di-return. Setiap komunikasi terdiri atas request ASCII, yang diikuti dengan response.^[2]

HTTP merupakan suatu standar request-response antara client dan server. Client merupakan end-user, sedangkan server merupakan web site. Client membuat suatu request HTTP, dengan menggunakan web browser atau end-user tool lainnya, yang ditujukan kepada user agent. Server yang memberikan response, yang menyimpan atau membuat resource seperti file HTML dan image, disenut sebagai origin server.

Umumnya, client HTTP menginisiasikan request. Client membuat koneksi TCP (Transmission Control Protocol) ke port tertentu pada host (default-nya adalah port 80). Server HTTP melakukan listening pada port tersebut dan menunggu client yang mengirim *request message*. Setelah menerima request, server kemudian mengirimkan kembali sebuah *status line*, seperti "HTTP/1.1 200 OK", yang diikuti dengan message. Message yang dikirim kembali dapat berupa resource yang di-request, error message, maupun informasi lainnya. Resource yang diakses oleh HTTP diidentifikasi menggunakan URI (Uniform Resource Identifiers), atau yang lebih khusus dengan URL (Uniform Resource Locators), dengan menggunakan `http:` maupun *https URI scheme*.

Pada versi HTTP/0.9 dan 1.0, koneksi di-close setelah sebuah single request/response. Pada HTTP/1.1 diperkenalkan mekanisme keep-alive, dimana suatu koneksi dapat digunakan kembali untuk lebih dari satu request. *Persistent connection* yang seperti ini dapat mengurangi *lag time*, sehingga dapat meningkatkan kecepatan koneksi, karena client tidak perlu lagi melakukan re-negosiasi koneksi TCP setelah request pertama dikirim.

Protokol HTTP di-handle oleh web server. Jika web server menerima HTTP request, maka web server akan merespon dengan sebuah HTTP response, seperti mengirim kembali halaman HTML. Untuk memproses suatu request, web server dapat memberikan respon dengan suatu halaman HTML statis atau image, mengirim *redirect*, atau men-delegasikan *dynamic response generation* ke program lainnya seperti CGI script, JSP (Java Server Pages), servlets, ASP (Active Server Pages), atau teknologi server-side lainnya.^[3]

BAB III

PERANCANGAN PERANGKAT LUNAK

3.1. Deskripsi Umum Sistem

Tugas akhir ini dirancang untuk mengimplementasikan arsitektur remote data access (RDA) untuk aplikasi monitoring kesehatan dengan menggunakan broker server sebagai perantara antara pasien dan monitoring user. Dengan adanya broker server maka home-PC atau notebook pasien tidak perlu terhubung ke internet secara terus-menerus untuk menyediakan service bagi monitoring user. Untuk melakukan sinkronisasi data antara database monitoring user dan database pasien maka monitoring-user cukup mengakses broker server.

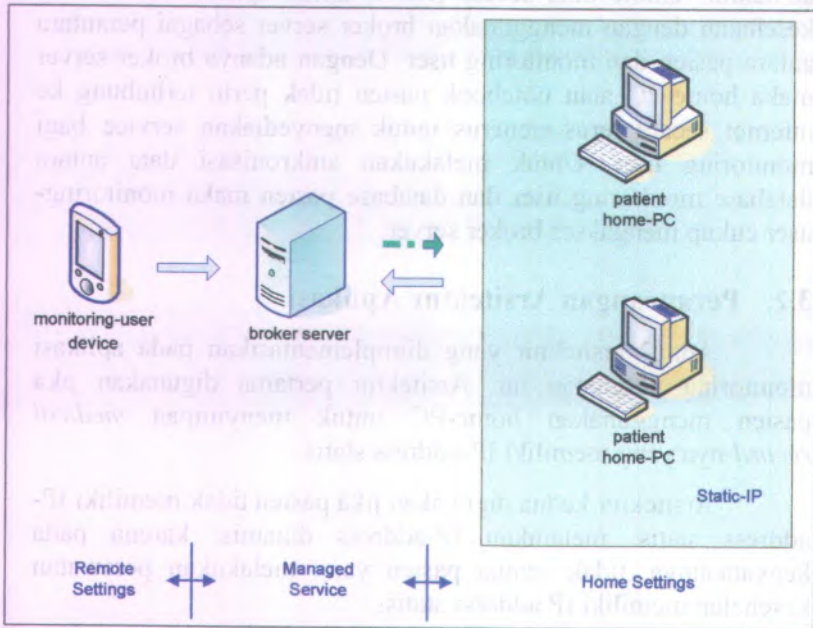
3.2. Perancangan Arsitektur Aplikasi

Ada 2 arsitektur yang diimplementasikan pada aplikasi monitoring kesehatan ini. Arsitektur pertama digunakan jika pasien menggunakan home-PC untuk menyimpan *medical record*-nya yang memiliki IP-address statis.

Arsitektur kedua digunakan jika pasien tidak memiliki IP-address statis, melainkan IP-address dinamis, karena pada kenyataannya, tidak semua pasien yang melakukan perawatan kesehatan memiliki IP address statis.

3.2.1. Arsitektur-1

Pada arsitektur ini, pasien menggunakan database SQL Server pada home-PC untuk menyimpan medical record-nya. Sinkronisasi data dilakukan antara database SQL Server Mobile pada mobile device monitoring user dan database SQL Server pada home-PC pasien. Arsitektur ini ditunjukkan seperti pada Gambar 3.1.

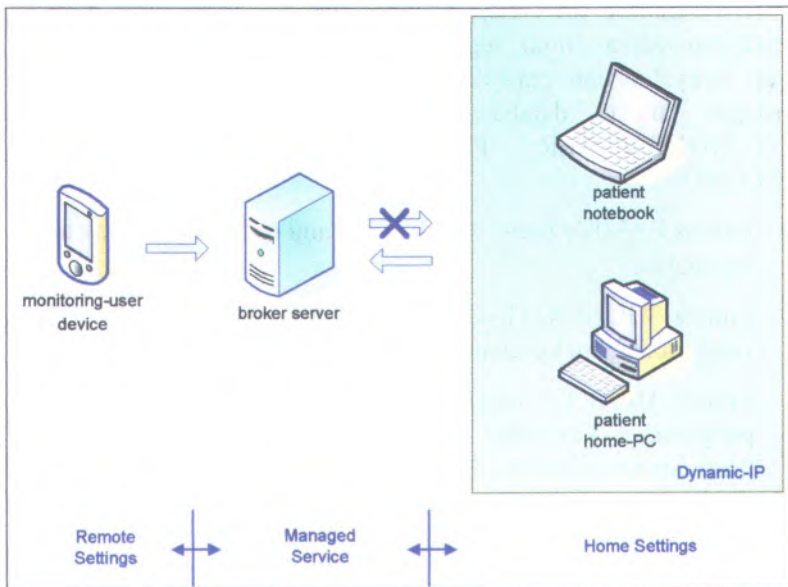


Gambar 3.1 Arsitektur-1 dengan IP-address statis pada home-PC pasien

3.2.2. Arsitektur-2

Pada arsitektur ini, pasien dapat menggunakan home-PC maupun notebook yang menggunakan database SQL Server Mobile untuk menyimpan data medical record-nya. Broker server dilengkapi dengan database SQL Server untuk menyimpan data sinkronisasi pasien.

Sinkronisasi data dilakukan 2 tahap. Tahap pertama, pasien melakukan sinkronisasi data antara database SQL Server Mobile pada home-PC dengan database SQL Server pada broker server. Tahap kedua, monitoring user dapat melakukan sinkronisasi data antara database SQL Server Mobile pada mobile device dengan database SQL Server pada broker server. Arsitektur ini ditunjukkan seperti pada Gambar 3.2.



Gambar 3.2 Arsitektur-2 dengan IP-address dinamis pada home-PC/notebook pasien

3.3. Perancangan Database Aplikasi

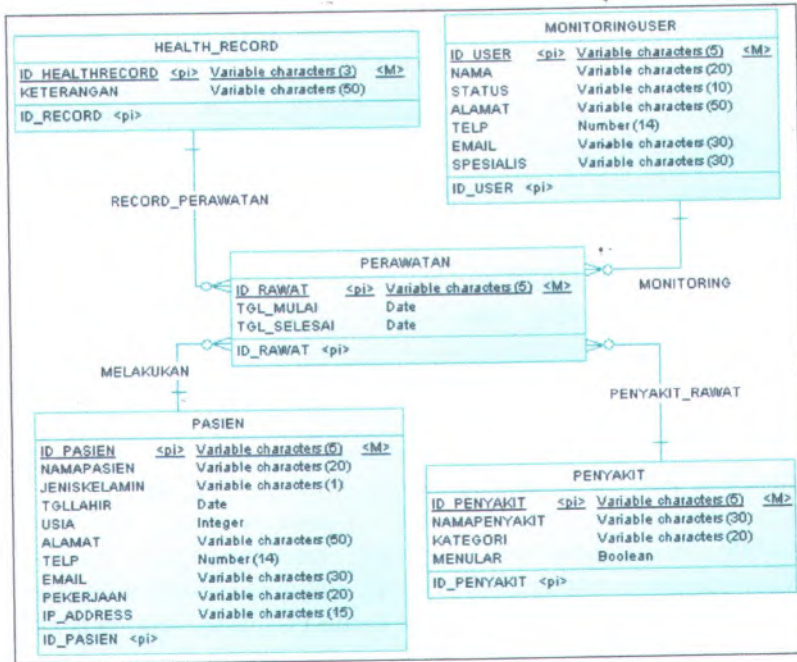
Ada 2 macam database yang digunakan pada aplikasi ini yaitu:

- ServerMedicalRecord merupakan database pada broker server. Database ini berisi data monitoring kesehatan secara umum.
- PatientMedicalRecord merupakan database yang menyimpan data medical record dan hasil pengukuran perawatan kesehatan pasien, misal pada pasien yang menderita penyakit pada home-PC/notebook pasien.

3.3.1. Database ServerMedicalRecord

Database ServerMedicalRecord.mdf terletak pada broker server. Database ini menyediakan sistem informasi monitoring kesehatan secara umum, seperti infor tentang pasien, monitoring-user, penyakit dan perawatan kesehatan. Ada 5 entitas yang terdapat pada database ini, antara lain: PASIEN, MONITORINGUSER, PERAWATAN, PENYAKIT dan RECORD.

- Entitas PASIEN berisi data pasien yang melakukan perawatan kesehatan.
- Entitas MONITORINGUSER berisi data monitoring-user yang memantau kesehatan pasien.
- Entitas HEALTH_RECORD berisi informasi tentang jenis pengukuran kesehatan yang dilakukan oleh pasien, misal pengukuran tekanan darah (blood pressure) memiliki ID 'BPR', pengukuran kadar kolesterol (cholesterol rate) memiliki ID 'CRT', dan sebagainya.
- Entitas PENYAKIT berisi informasi tentang penyakit.
- Entitas PERAWATAN berisi informasi tentang perawatan yang dilakukan oleh pasien.



Gambar 3.3 Diagram CDM database ServerMedicalRecord

3.3.2. Database PatientMedicalRecord

Database ini terletak pada home-PC atau notebook pasien. Database ini menyimpan data detail pasien dan record hasil pengukuran kesehatan pasien. Data detail pasien terdiri atas 4 entitas yaitu PASIEN, RIWAYAT_PENYAKIT, KELUARGA, dan MONITORINGUSER. Sedangkan data record hasil pengukuran kesehatan pasien bervariasi antara pasien satu dengan yang lain, tergantung kebutuhan pengukuran kesehatan pasien masing-masing.

- Entitas PASIEN berisi data detail pasien yang melakukan perawatan kesehatan.

- Entitas MONITORINGUSER berisi data monitoring-user yang memantau kesehatan pasien.
- Entitas KELUARGA berisi data keluarga pasien yang bersangkutan.
- Entitas RIWAYAT_PENYAKIT berisi informasi riwayat penyakit pasien.

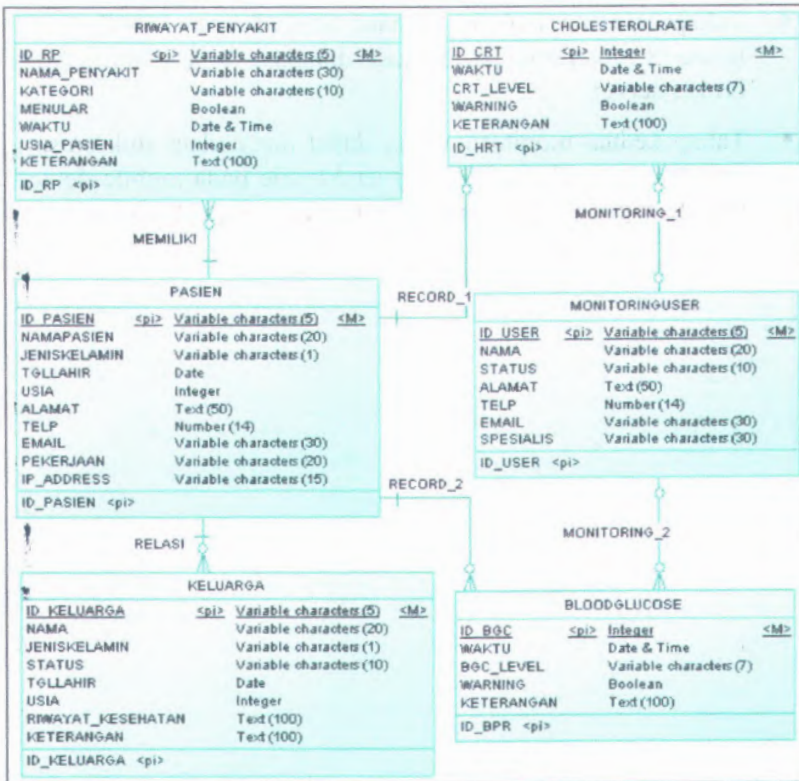
Pada tugas akhir ini, digunakan 2 database pasien sebagai PatientMedical Record, yaitu Patient01 dan Patient02.

3.3.2.1. Database Patient01

Database Patient01, merupakan database medical record pasien-1. File database SQL Server Patient01.mdf berada pada home-PC pasien yang memiliki IP-address statis. Database mobile device monitoring-user yang menyimpan data medical record pasien-1 adalah Patient01.sdf. Sinkronisasi data dilakukan antara database SQL Server pada home-PC pasien dan database SQL Server Mobile pada mobile device monitoring user.

Pada database ini, data record hasil pengukuran pasien terdiri atas 2 entitas yaitu CHOLESTEROLRATE dan BLOODGLUCOSE.

- Entitas CHOLESTEROLRATE berisi data hasil pengukuran kadar kolesterol pasien.
- Entitas BLOODGLUCOSE berisi data hasil pengukuran kadar gula darah pasien.

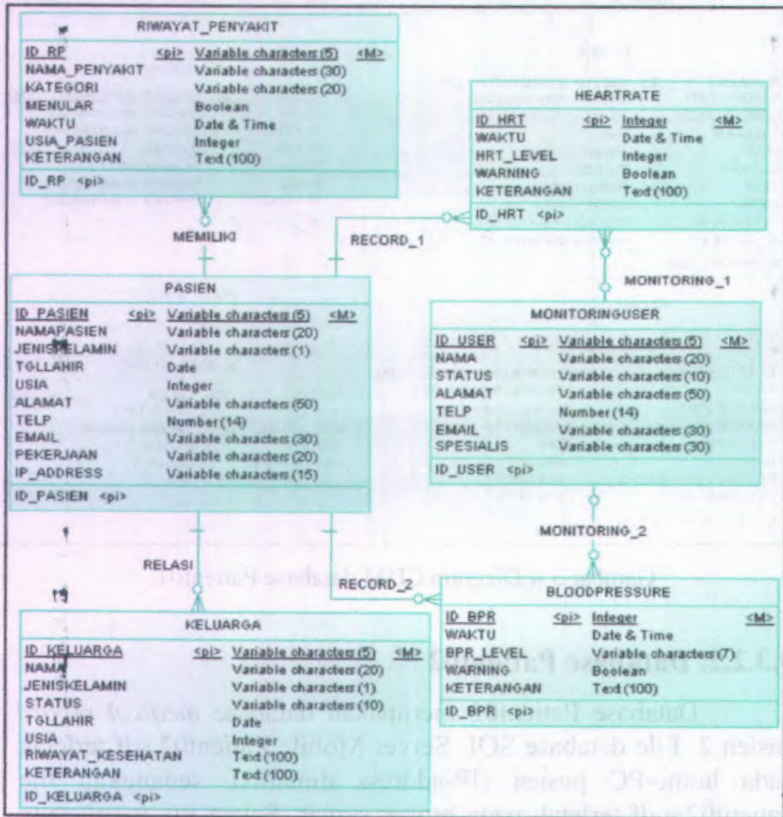


Gambar 3.4 Diagram CDM database Patient01

3.3.2.2. Database Patient02

Database Patient02 merupakan database *medical record* pasien 2. File database SQL Server Mobile Patient02.sdf terletak pada home-PC pasien (IP-address dinamis), sedangkan file Patient02.mdf terletak pada broker server. Selain itu, monitoring user juga memiliki file database Patient02.sdf pada mobile device yang menyimpan data sinkronisasi dari database Patient02.mdf. Sinkronisasi dilakukan 2 tahap, yaitu:

- Tahap pertama antara database SQL Server Mobile pada home-PC/notebook pasien dan database SQL Server pada broker server.
- Tahap kedua, monitoring user dapat melakukan sinkronisasi data antara database SQL Server Mobile pada mobile device monitoring-user dan database SQL Server pada broker server.



Gambar 3.5 Diagram CDM database Patient02

Pada database ini, data record hasil pengukuran pasien terdiri atas 2 entitas yaitu BLOODPRESSURE dan HEARTRATE.

- Entitas BLOODPRESSURE berisi data hasil pengukuran tekanan darah pasien.
- Entitas HEARTRATE berisi data hasil pengukuran denyut jantung pasien.

3.4. Diagram Use Case

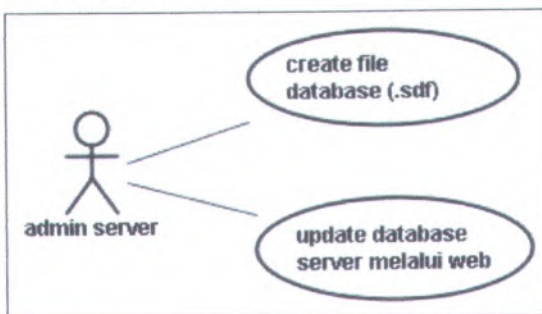
Diagram use case pada tugas akhir ini dibagi menjadi 4 berdasarkan actor yang terlibat, yaitu: admin server, monitoring user, pasien 1 (IP-address statis), dan pasien 2 (IP-address dinamis).

3.4.1. Admin Server

Admin server bertugas menangani computer program dan web yang berada pada broker server. Computer program berfungsi untuk membuat (create) file database SQL Server Mobile (.sdf) dari file SQL Server (.mdf). File database (.sdf) yang dibuat pada tugas akhir ini antara lain:

- Patient01.sdf dari file Patient01.mdf
- Patient02.sdf dari file Patient02.mdf

Diagram use case untuk admin server ditunjukkan seperti pada Gambar 3.6 di bawah ini:



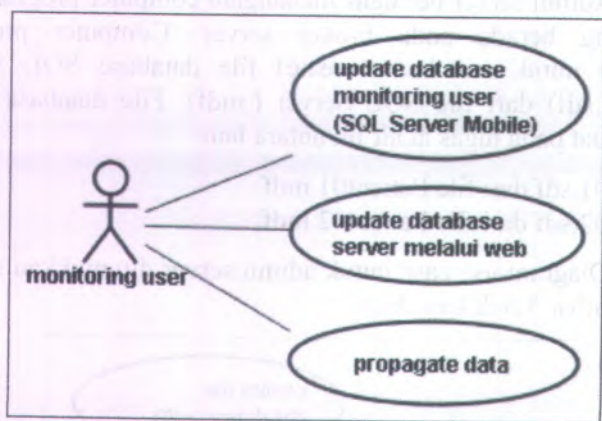
Gambar 3.6 Diagram use case admin server

3.4.2. Monitoring User

Monitoring user adalah dokter atau juru rawat yang bertugas memantau kondisi kesehatan pasien. Secara berkala monitoring-user mengakses database pasien melalui web. Selanjutnya pasien dapat melakukan *propagate* data (*pull* dan *push*) antara database pasien pada home-PC atau server dan database lokal monitoring-user pada mobile device.

Pada tugas akhir ini, monitoring user bertugas memonitoring kondisi 2 pasien, yaitu pasien-1 dan pasien-2, sehingga pada mobile device terdapat 2 file database yaitu Patient01.sdf dan Patient02.sdf.

Diagram use case untuk monitoring-user ditunjukkan seperti pada Gambar 3.7 di bawah ini:

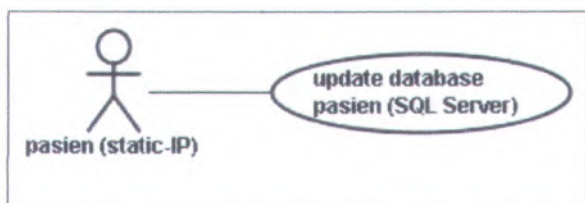


Gambar 3.7 Diagram use case monitoring-user

3.4.3. Pasien-1 (IP-Address Statis)

Diagram use case untuk pasien-1 (IP-address statis) ditunjukkan seperti pada Gambar 3.8. Pasien cukup melakukan update database SQL Server pada home-PC. Karena memiliki IP-address statis yang dapat dikenali oleh monitoring-user, maka monitoring-user dapat mengakses database pasien secara remote

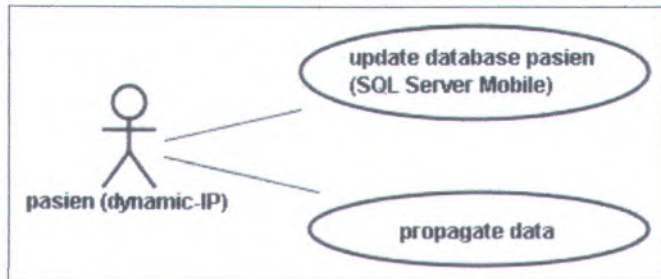
dan melakukan *propagate* data jika home-PC terhubung ke jaringan.



Gambar 3.8 Diagram use case pasien-1 (IP-address statis)

3.4.4. Pasien-2 (IP-Address Dinamis)

Diagram use case untuk pasien-2 (IP-address dinamis) ditunjukkan seperti pada Gambar 3.9. Sama seperti monitoring-user, pasien juga melakukan update database SQL Server Mobile pada home-PC. Setelah itu pasien dapat melakukan *propagate data* (pull atau push) antara database lokal dan database SQL Server pada broker server.



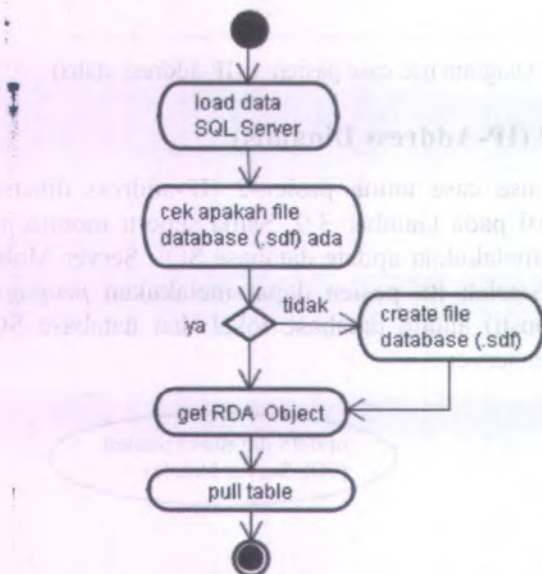
Gambar 3.9 Diagram use case pasien-2 (IP-address dinamis)

3.5. Diagram Activity

Diagram activity menjelaskan state aktivitas untuk tiap use case yang telah digambarkan sebelumnya pada diagram use case.

3.5.1. Create File Database (.sdf)

Diagram activity untuk use case create file database ditunjukkan seperti pada Gambar 3.10 di bawah. File database (.sdf) yang dihasilkan berisi data yang sama dengan database SQL Server. File database ini selanjutnya dipindahkan ke mobile device monitoring-user.

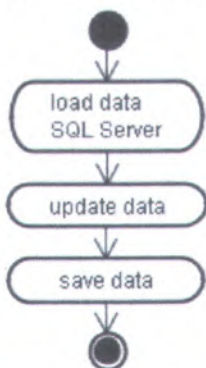


Gambar 3.10 Diagram activity create file database (.sdf)

3.5.2. Update Database SQL Server

Update database SQL Server dilakukan oleh admin server dan pasien-1. Admin server melakukan update database ServerMedicalRecord.mdf dan Patient02.mdf pada broker server, sedangkan pasien-1 melakukan update database medical record-nya, Patient01.mdf, pada home-PC.

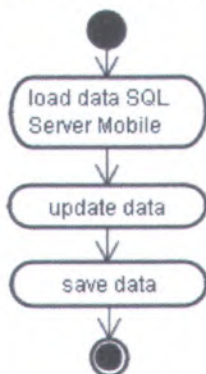
Diagram activity update database SQL Server ditunjukkan seperti pada Gambar 3.11 di bawah ini.



Gambar 3.11 Diagram activity update data SQL Server

3.5.3. Update Database SQL Server Mobile

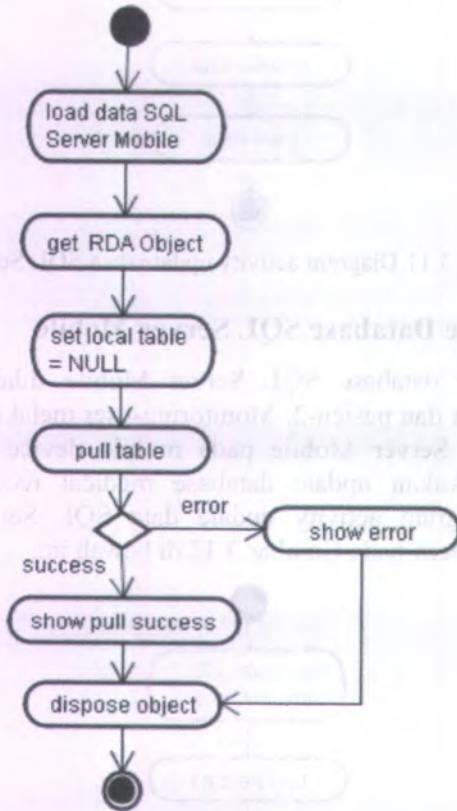
Update database SQL Server Mobile dilakukan oleh monitoring-user dan pasien-2. Monitoring-user melakukan update database SQL Server Mobile pada mobile device, sedangkan pasien-2 melakukan update database medical record-nyapada home-PC. Diagram activity update data SQL Server Mobile ditunjukkan seperti pada Gambar 3.12 di bawah ini.



Gambar 3.12 Diagram activity update data SQL Server Mobile

3.5.4. Pull Data

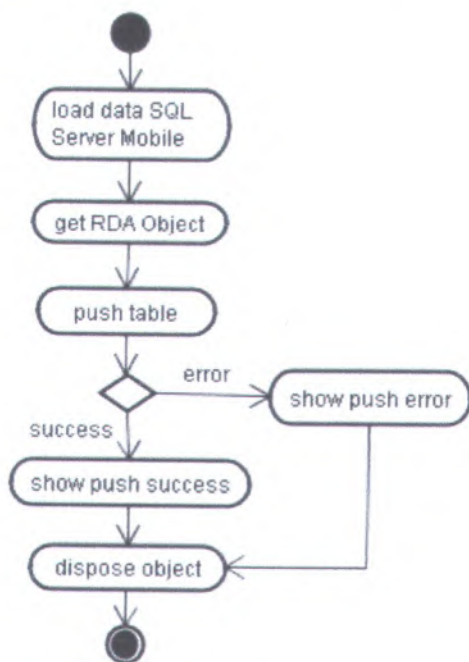
Pull data dilakukan oleh monitoring-user. Pull data dilakukan dari database SQL Server pasien ke database SQL Server Mobile pada mobile device. Diagram activity pull data ditunjukkan seperti pada Gambar 3.13 di bawah ini.



Gambar 3.13 Diagram activity pull data

3.5.5. Push Data

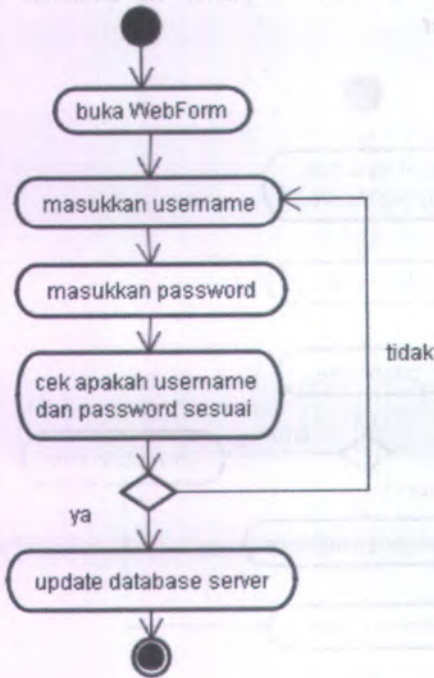
Pull data dilakukan oleh monitoring-user dan pasien-2. Monitoring-user melakukan push data dari database SQL Server Mobile pada mobile device ke database SQL Server pasien, sedangkan pasien-2 melakukan push data dari database SQL Server Mobile pada home-PC pasien ke database SQL Server pada broker server.



Gambar 3.14 Diagram activity push data

3.5.6. Update Database Server melalui Web

Monitoring-user dapat meng-update database server dan database pasien melalui web. Akses ke database pasien juga dilakukan sebelum melakukan pull data, untuk mengetahui apakah data pasien telah di-update atau belum.



Gambar 3.15 Diagram activity update database melalui web

BAB IV IMPLEMENTASI PERANGKAT LUNAK

Pada bab ini akan dibahas tentang implementasi dari perancangan perangkat lunak yang telah dilakukan pada bab 3 yang meliputi implementasi arsitektur aplikasi dan database yang digunakan.

4.1. Implementasi Arsitektur

Secara umum, arsitektur aplikasi terdiri atas 3 komponen yaitu mobile device monitoring-user, broker server, dan home-PC atau notebook pasien. terdiri atas 2 bagian utama yaitu komponen sinkronisasi data dan aplikasi web.

Mobile device monitoring-user terdiri atas beberapa komponen seperti arsitektur RDA awal, yaitu *device program* (.NET Program), database SQL Server Mobile, SQL Server Mobile Client Agent (sqlcesa30.dll), dan SQL Server Mobile Database Engine. Database SQL Server Mobile menyimpan sinkronisasi dari database pasien.

Broker server berfungsi sebagai perantara antara monitoring-user dan pasien. Broker server terdiri atas komponen sinkronisasi data, SQL Server Mobile Server Agent (sqlcesa30.dll), aplikasi web, dan server program (.NET Program) untuk melakukan create file database SQL Server Mobile (.sdf) yang akan digunakan oleh monitoring-user dan pasien-2. Pada tugas akhir ini, broker server menggunakan IP-address 10.151.32.32.

Home-PC maupun notebook pasien terdiri atas *computer program* (.NET Program) dan database yang menyimpan data *medical record* pasien. Computer program berfungsi sebagai interface bagi pasien untuk menyimpan data pada database. Pasien dapat berkonsultasi ke juru rawat untuk mengetahui jika

terdapat ketidaknormalan pada data record hasil pengukuran kesehatan, misalnya level maksimum dan minimum pada kadar kolesterol. Selanjutnya pada database dibuat trigger untuk menampilkan warning jika kadar kolesterol pasien berada di luar level normal.

Pada tugas akhir ini, diimplementasikan 2 arsitektur:

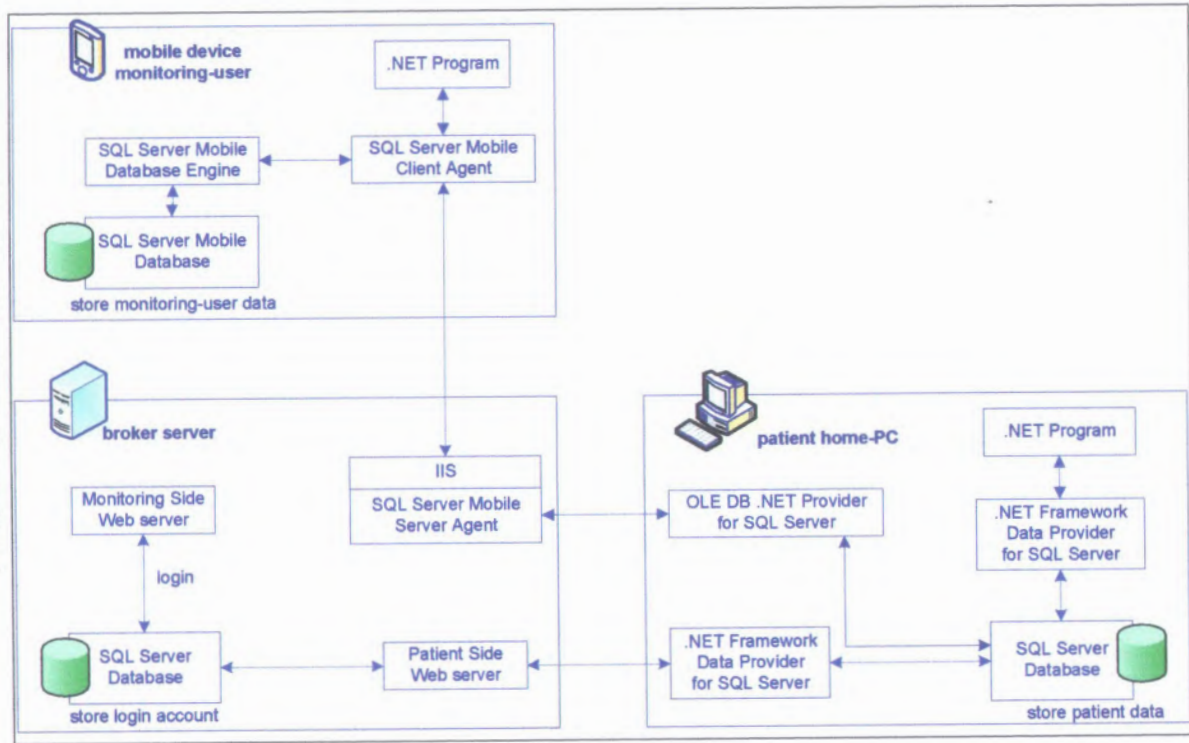
- Arsitektur-1, merupakan arsitektur utama, digunakan jika pasien memiliki IP-address statis. Pada implementasi ini, arsitektur-1 digunakan oleh pasien-1.
- Arsitektur-2, merupakan arsitektur tambahan, digunakan jika pasien memiliki IP-address dinamis. Pada implementasi ini arsitektur-2 digunakan oleh pasien-2.

4.1.1. Arsitektur-1

Pada arsitektur ini, pasien menyimpan data medical record nya pada database SQL server pada home-PC yang memiliki IP-address statis. Home-PC yang diimplementasikan pada tugas akhir ini memiliki IP-address 10.151.31.4. Untuk melakukan propagate data terhadap database pasien maka monitoring user terlebih dahulu harus menjalankan program pada mobile device-nya, dan mengecek melalui web-browser apakah data pada home-PC pasien dapat diakses. Jika data dapat diakses, maka monitoring user dapat mengirim request ke broker server. Oleh broker server (SQL Server Mobile Server Agent), request ini selanjutnya akan di-forward ke home-PC pasien.

Setelah menerima request, computer program pada home-PC pasien akan mengirim response sesuai dengan command yang diterima. Response yang dikirim berupa data yang dikirim kembali ke broker server dan mobile device monitoring-user melalui protokol HTTP. Jadi sinkronisasi data dilakukan antara database SQL Server Mobile pada mobile device monitoring-user dan database SQL Server pada home-PC pasien. Gambar 4.1 menunjukkan implementasi arsitektur-1.

Gambar 4.1 Implementasi arsitektur-1

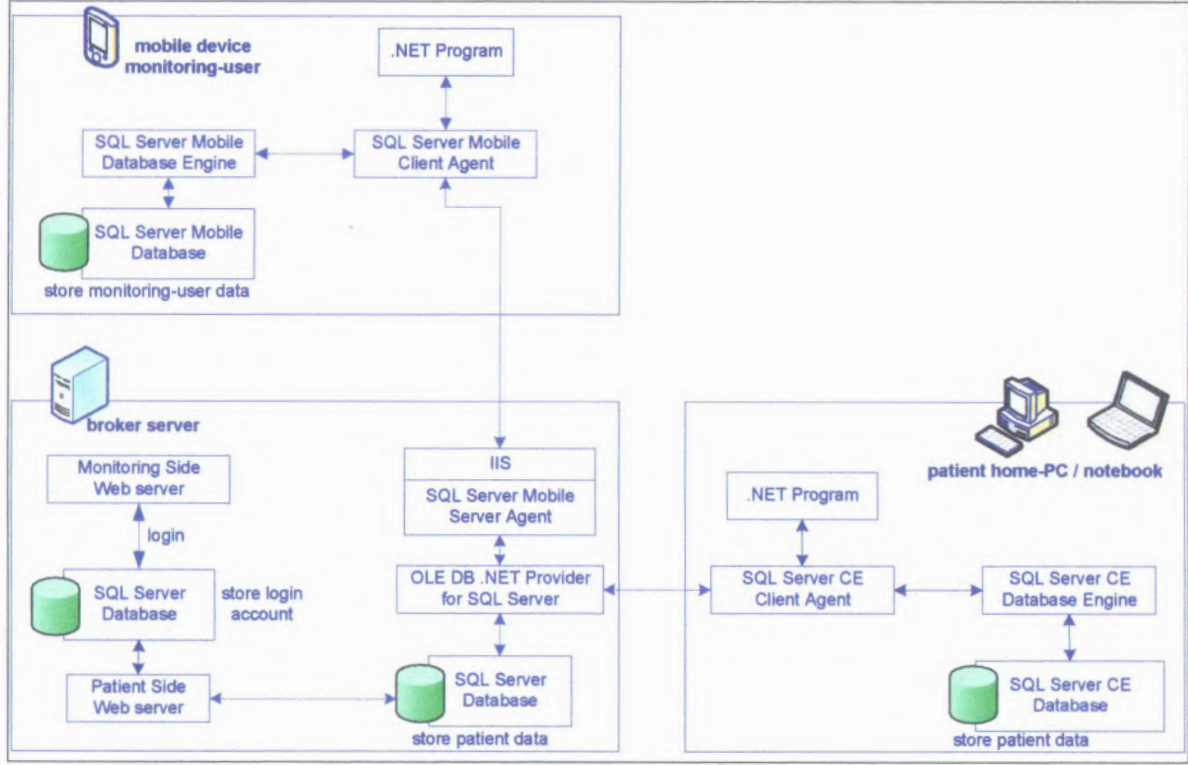


4.1.2. Arsitektur-2

Arsitektur-2 digunakan jika pasien tidak memiliki IP-address statis melainkan IP-address dinamis. Pada arsitektur ini, pasien menyimpan data medical record-nya pada database SQL Server Compact pada notebook/PC pasien yang memiliki IP-address dinamis. Sinkronisasi data dilakukan 2 tahap yaitu antara home-PC pasien dengan broker server dan antara mobile device monitoring-user dengan broker server.

Setelah melakukan update database medical record pada home-PC, pasien harus melakukan push data dari database SQL Server Compact (Patient02.sdf) di home-PC ke database SQL Server (Patient02.mdf) pada broker server. Selanjutnya, monitoring-user dapat mengakses database Patient02.mdf tersebut untuk mengetahui hasil pengukuran kesehatan pasien yang terbaru. Monitoring-user juga dapat melakukan pull dan push data antara database Patient02.mdf ke database Patient02.sdf yang berada pada mobile device monitoring-user. Gambar 4.2 menunjukkan implementasi arsitektur-2.

Gambar 4.2 Implementasi arsitektur-2

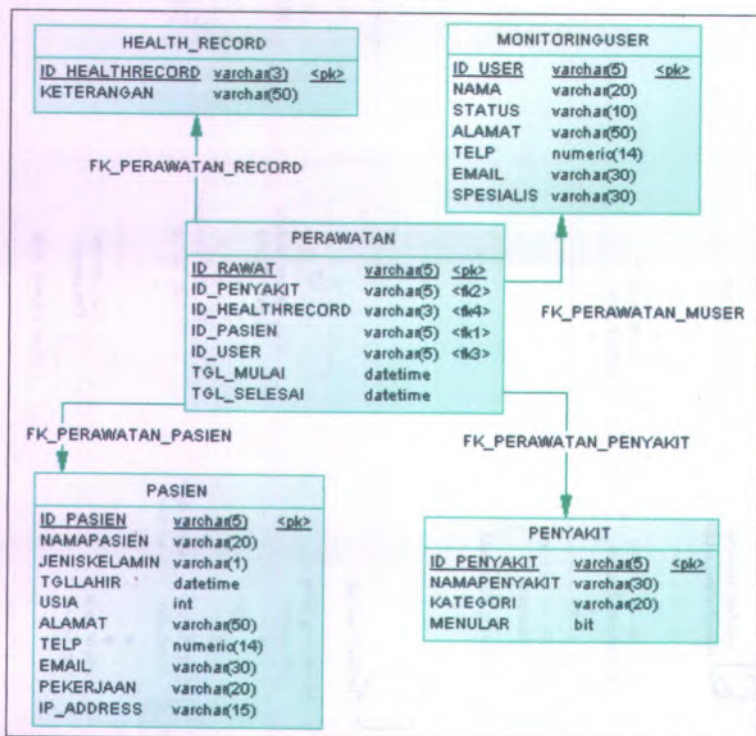


4.2. Implementasi Database

Implementasi database aplikasi berisi penjelasan diagram Physical Data Model (PDM) dari tiap-tiap diagram Conceptual Data Model (CDM) yang telah dijelaskan sebelumnya pada bab 3.3.

4.2.1. Database Server MedicalRecord

Diagram PDM database Server MedicalRecord ditunjukkan seperti pada Gambar 4.3. Database ini terdiri atas 5 tabel yaitu PASIEN, MONITORINGUSER, PENYAKIT, HEALTH_RECORD, dan PERAWATAN.



Gambar 4.3 Diagram PDM database Server MedicalRecord

Tabel PERAWATAN memiliki foreign key dari keempat tabel lainnya, sehingga pada tabel ini dapat diketahui detail perawatan kesehatan yang dilakukan oleh pasien. Database ini terletak pada broker server dengan nama file ServerMedicalRecord.mdf.

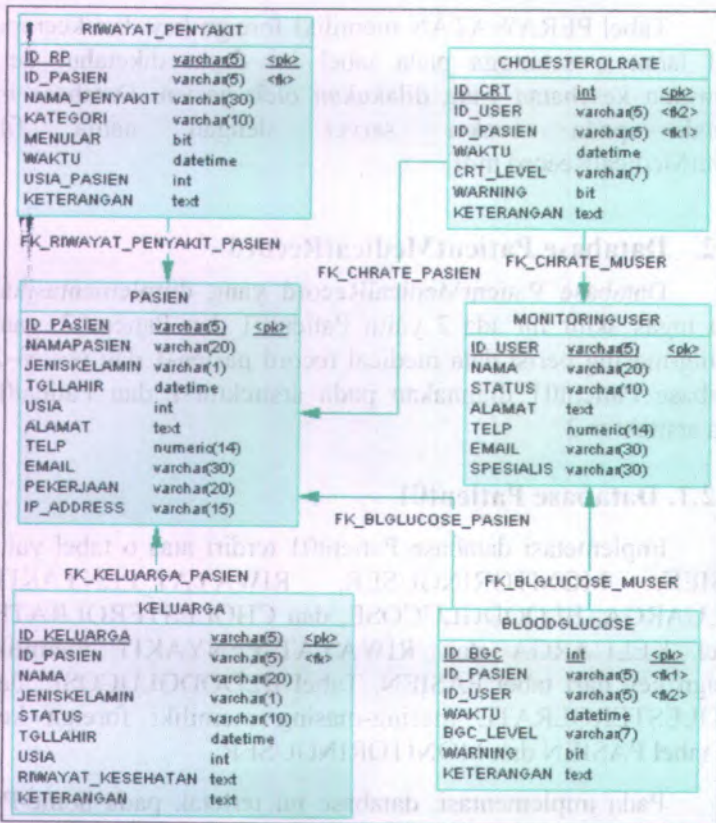
4.2.2. Database PatientMedicalRecord

Database PatientMedicalRecord yang diimplementasikan pada tugas akhir ini ada 2 yaitu Patient01 dan Patient02, yang masing-masing berisi data medical record pasien-1 dan pasien-2. Database Patient01 digunakan pada arsitektur-1 dan Patient02 pada arsitektur-2.

4.2.2.1. Database Patient01

Implementasi database Patient01 terdiri atas 6 tabel yaitu PASIEN, MONITORINGUSER, RIWAYAT_PENYAKIT, KELUARGA, BLOODGLUCOSE, dan CHOLESTEROLRATE. Tabel KELUARGA dan RIWAYAT_PENYAKIT memiliki foreign key dari tabel PASIEN. Tabel BLOODGLUCOSE, dan CHOLESTEROLRATE masing-masing memiliki foreign key dari tabel PASIEN dan MONITORINGUSER.

Pada implementasi, database ini terletak pada home-PC pasien (IP-address statis) dengan nama file Patient01.mdf, serta database sinkronisasinya pada mobile device monitoring-user dengan nama file Patient01.sdf. Diagram PDM (Physical Data Model) database ini ditunjukkan seperti pada Gambar 4.4.

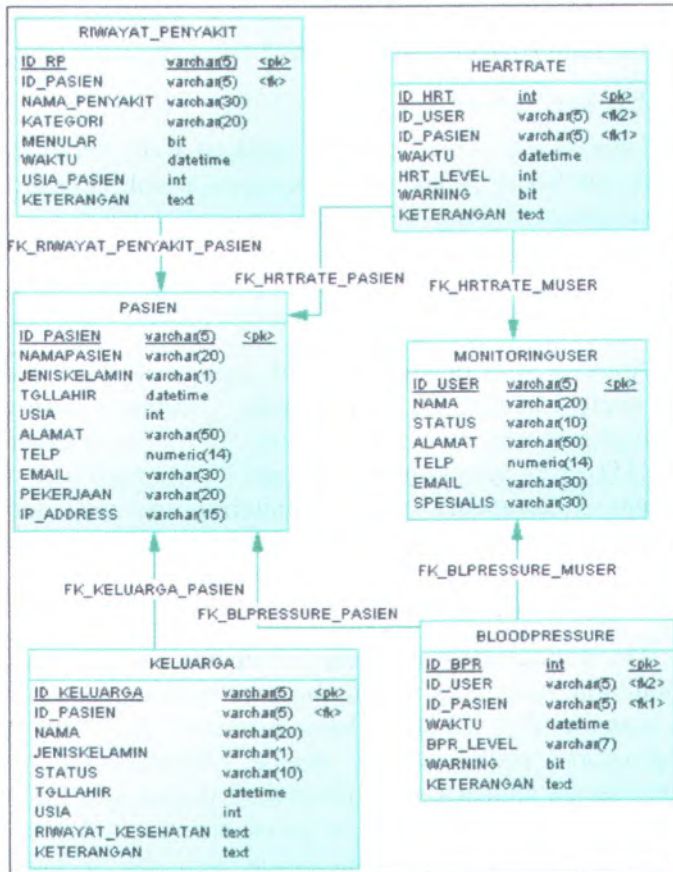


Gambar 4.4 Diagram PDM database Patient01

4.2.2.2. Database Patient02

Database Patient02 terdiri atas 6 tabel; 4 tabel data yaitu PASIE, MONITORINGUSER, RIWAYAT_PENYAKIT, dan KELUARGA, serta 2 tabel hasil pengukuran kesehatan pasien yaitu BLOODPRESSURE dan HEARTRATE. Sama seperti Patient01, tabel KELUARGA dan RIWAYAT_PENYAKIT juga memiliki foreign key dari tabel PASIE. Tabel BLOODPRESSURE, dan HEARTRATE masing-masing

memiliki foreign ke dari tabel PASIEN dan MONITORINGUSER. Diagram PDM (Physical Data Model) database Patient02 ditunjukkan seperti pada Gambar 4.5.



Gambar 4.5 Diagram PDM database Patient02

Pada implemetasi ini, database ini terletak pada home-PC pasien(IP-address dinamis) dengan nama file Patient02.sdf, pada broker server dengan nama file Patient02.mdf, serta pada mobile device monitoring-user dengan nama file Patient02.sdf.

4.3. Implementasi Program

Implementasi program berisi penjelasan program yang diimplementasikan pada tiap-tiap komponen aplikasi yaitu pada broker server, mobile device monitoring-user, home-PC pasien-1 (IP-address statis) dan home-PC pasien-2 (IP-address dinamis).

4.3.1. Broker Server

Pada broker server, terdapat aplikasi web, library SQL-Server Mobile Server Agent sebagai komponen sinkronisasi data, dan server-program yang merupakan program tambahan untuk melakukan create file database SQL Server Mobile (.sdf).

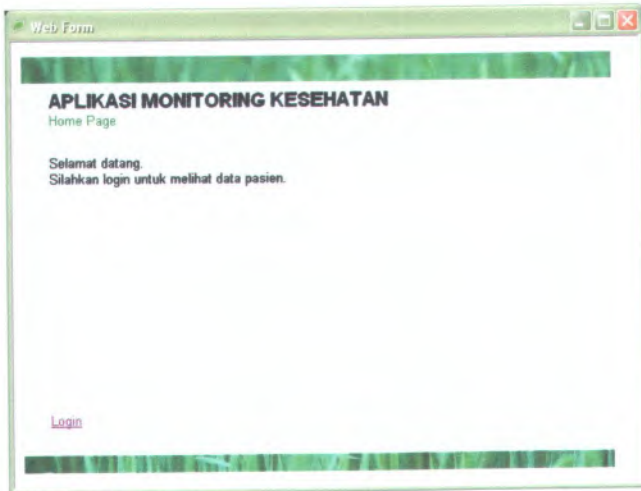
4.3.1.1. Aplikasi Web

Aplikasi web digunakan untuk mengecek data pasien. Pada implementasi ini digunakan virtual directory MonitoringKesehatan dengan internet-url <http://10.151.32.32/MonitoringKesehatan/>. Aplikasi web ini terdiri atas 3 komponen yaitu: monitoring-side web server, patient-side web server, dan database SQL Server.

Monitoring-side web server

Monitoring-side web server, merupakan service interface bagi monitoring-user. Jika monitoring-user ingin mengecek data pasien, maka terlebih dahulu ia harus membuka WebForm pada device-program pada mobile device. WebBrowser pada WebForm secara otomatis akan membuka halaman aplikasi web. Monitoring-user selanjutnya dapat melakukan login ke sistem agar dapat mengakses data patient-side web dan melakukan request terhadap data pasien. Selanjutnya monitoring-user dapat melakukan login agar dapat mengakses data pasien.

Gambar 4.6 menunjukkan tampilan halaman monitoring-side web server.



Gambar 4.6 Tampilan halaman monitoring-side web server

Gambar 4.7 di bawah menunjukkan konfigurasi pada file `web.config` untuk mencegah akses anonymous user, sehingga hanya user yang telah melakukan login yang dapat mengakses data pasien.

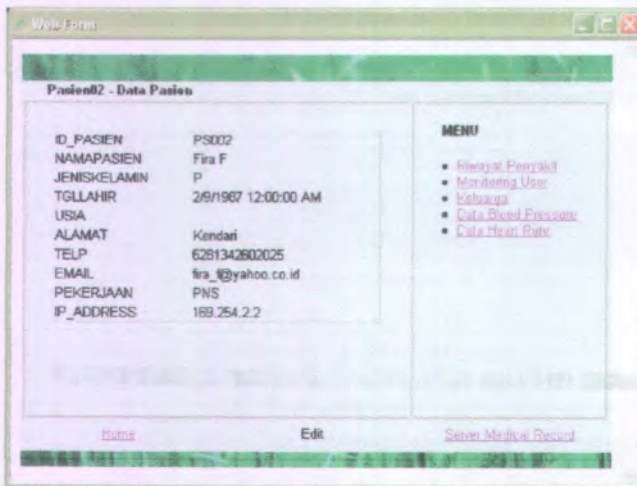
```
<configuration
xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <system.web>
    <authorization>
      <deny users="?" />
    </authorization>
  </system.web>
</configuration>
```

Gambar 4.7 Konfigurasi `web.config` pada monitoring side web server

Patient-side web server

Patient-side web server, berfungsi meng-*handle* koneksi ke database SQL Server pada broker server maupun home-PC dengan menggunakan .NET Framework Data Provider for SQL Server. Halaman web yang menampilkan data pasien hanya dapat

diakses oleh user yang telah melakukan login. Tampilan halaman patient-side web server ditunjukkan seperti Gambar 4.8.



Gambar 4.8 Tampilan halaman patient-side web server

Gambar 4.9 berikut menunjukkan connection-string ke database. Karena halaman web menampilkan data dari 3 database sehingga diimplementasikan 3 connection-string yaitu `ServerMedicalRecordConnectionString` ke database `ServerMedicalRecord`, `PATIENT01ConnectionString` ke database `medical record pasien-1`, dan `PATIENT02ConnectionString` ke database `medical record pasien-2`.

```
<add name="ServerMedicalRecordConnectionString"
connectionString="Data Source=10.151.32.32;
Initial Catalog=ServerMedicalRecord;
Persist Security Info=True;
User ID=sa;Password=bismillah"
providerName="System.Data.SqlClient" />

<add name="PATIENT01ConnectionString"
connectionString="Data Source=10.151.31.4;
Initial Catalog=PATIENT01;
Persist Security Info=True;
User ID=sa;Password=bismillah"
```

```

providerName="System.Data.SqlClient" />
<add name="PATIENT02ConnectionString"
connectionString="Data Source=10.151.32.32;
Initial Catalog=PATIENT02;
Persist Security Info=True;
User ID=sa;Password=bismillah"
providerName="System.Data.SqlClient" />

```

Gambar 4.9 Connection string pada patient-side web server

Database SQL Server

Database SQL Server yang berfungsi untuk menyimpan informasi account login. Komponen ini juga mengatur request queue dan response queue antara monitoring-side web server dan patient-side web server melalui login. Pada implementasi ini, digunakan file database memberDB.mdf. Gambar 4.10 menunjukkan connection-string ke ke database login (memberDB.mdf).

```

<remove name="LocalSqlServer" />
<add name="LocalSqlServer" connectionString="Data
Source=localhost;Initial Catalog=memberDB;Persist Security
Info=True;User ID=sa;Password=bismillah"
providerName="System.Data.SqlClient" />

```

Gambar 4.10 Connection string ke database login

4.3.1.2. SQL Server Mobile Server Agent

SQL Server Mobile Server Agent berfungsi sebagai komponen untuk melakukan sinkronisasi data. Pada implementasi ini, SQL Server Mobile Server Agent (sqlcesa30.dll) diletakkan pada virtual directory HealthMonitoring, dan dapat diakses dengan url <http://10.151.32.32/HealthMonitoring/sqlcesa30.dll>. Library ini digunakan ketika melakukan inisialisasi objek RDA.

Langkah-langkah konfigurasi komponen SQL Server Mobile Server Agent (sqlcesa30.dll) ada 2 tahap yaitu Pembuatan Share Directory dan Sinkronisasi Web:

Pembuatan Share Directory:

1. Buat folder baru pada komputer server, misal **RDADData** dengan path **C:\Program Files\Microsoft SQL Server 2005 Mobile Edition\Server\RDADData**.
2. Klik kanan pada folder lalu pilih **Properties**.
3. Pilih tab **Sharing**. Pada bagian *enable sharing*, pilih **Share this folder**. Masukkan **Share name** untuk folder, misal **RDADData**.
4. Selanjutnya, pilih tab **Web Sharing**. Pilih **Share this folder**, lalu masukkan nama **Alias** untuk folder, misal **RDADData**.
5. Klik **OK**.

Sinkronisasi Web

1. Pada **Start Menu**, jalankan **Configure Web Synchronization Wizard** (**Conwiz.exe**). Sebelum itu, pastikan komponen sinkronisasi (**Sqlce30setupen.msi**) telah terinstal pada komputer server.
2. Pada bagian **Subscriber Type** pilih **SQL Server Mobile Edition**, lalu klik **Next**.
3. Pada bagian **Web Server**, masukkan nama komputer yang menjalankan IIS, misal **DIPONEGORO**. Pilih **Create a new virtual directory** untuk membuat virtual directory baru yang dapat diakses melalui web. Pilih **Configure an existing virtual directory** jika virtual directory telah ada. Pada implementasi ini digunakan **Create a new virtual directory**.
4. Pada bagian **Select the Web site**, klik nama komputer lalu pilih **Default Web Site**. Klik **Next**.
5. Pada **Virtual Directory Information**, masukkan nama **Alias** untuk virtual directory yang akan dibuat. Pada implementasi ini digunakan nama **HealthMonitoring**. Pada bagian **Path**,

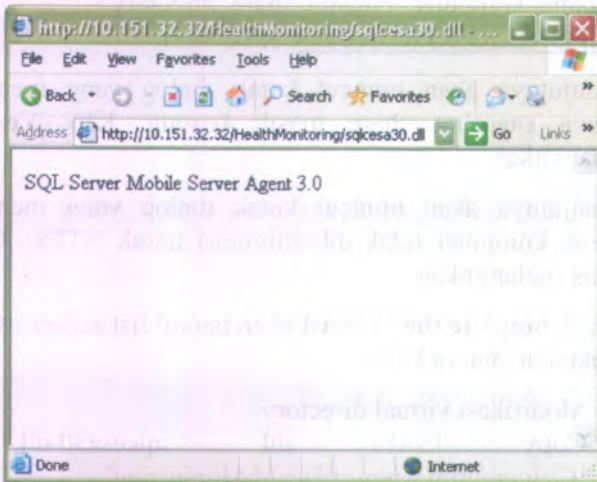
klik **Browse...** untuk menentukan path virtual directory, yaitu **C:\Inetpub\wwwroot\HealthMonitoring**. Klik **Next**.

6. Selanjutnya akan muncul kotak dialog. Pilih **Yes** untuk melakukan copy dan register SQL Mobile Server Agent.
7. Pada **Secure Communications**, pilih default **Do not require secure channel (SSL)**. Klik **Next**.
8. Pada **Client Authentication**, pilih **Clients will be authenticated**, sehingga hanya user yang telah melakukan login yang dapat mengakses virtual directory. Klik **Next**.
9. Pada **Authenticated Access**, beri tanda centang pada **Integrated Windows authentication**. Klik **Next**.
10. Pada **Directory Access** tentukan group atau user name yang dapat mengakses virtual directory. Klik **Next**
11. Pada **Snapshot Address Access**, masukkan path share directory yang telah dibuat sebelumnya dengan format \\<nama_komputer>\<nama_share_directory>, misal [\DIPONEGORO\RDADData](#). Klik **Next**.
12. Selanjutnya akan muncul kotak dialog yang menyatakan bahwa snapshot share masih kosong. Klik **Yes** untuk melanjutkan.
13. Selanjutnya akan muncul kotak dialog yang menyatakan bahwa komputer tidak dikonfigurasi untuk NTFS. Klik **OK** untuk melanjutkan.
14. Pada **Complete the Wizard** akan tampil list action yang akan dilakukan, antara lain:
 - Modifikasi virtual directory.
 - Copy ISAPI dll 'sqlcesa30.dll' ke 'C:\Inetpub\wwwroot\HealthMonitoring'
 - Set NTFS permission.
 - Virtual directory 'HealthMonitoring' akan dikonfigurasi pada server 'DIPONEGORO'.

- 'HealthMonitoring' digunakan sebagai alias.
- 'C:\Inetpub\wwwroot\HealthMonitoring' digunakan sebagai physical path bagi virtual directory.
- 'Integrated Windows' digunakan sebagai authentication method.
- Akses ke virtual directory tidak membutuhkan *secure channel*.
- Client dapat menggunakan internet-url <http://DIPONEGORO/HealthMonitoring/sqlcesa30.dll>.

Klik **Finish**.

15. Untuk mengecek apakah sinkronisasi web yang telah dilakukan berhasil maka pada web browser ketikkan internet-url yang telah dibuat seperti pada langkah 14. Nama komputer dapat diganti dengan IP komputer. Pada implementasi ini digunakan IP-address 10.151.32.32. Gambar 4.11 berikut menunjukkan tampilan web browser jika sinkronisasi web berhasil.



Gambar 4.11 Tampilan url SQL Server Mobile Server Agent

4.3.1.3. Server Program

Server-program merupakan program tambahan yang berfungsi untuk melakukan create database SQL Server Mobile (.sdf) dari database SQL Server (.mdf). File database (.sdf) yang dibuat pada tugas akhir ini adalah:

- Patient01.sdf dari file Patient01.mdf;
- Patient02.sdf dari file Patient02.mdf.

File Patient01.sdf selanjutnya akan diletakkan pada mobile device monitoring-user. Sedangkan file Patient02.sdf diletakkan pada mobile device monitoring-user dan home-PC pasien-2. Berikut ini langkah-langkah untuk melakukan create file database.

1. Mendefinisikan local connection string ke database SQL Server Mobile dan membuat fungsi-fungsi yang akan digunakan. Karena ada 2 file database yang akan dibuat, maka dibutuhkan 2 local connection string. Application settings untuk local connection string pada file Patient01.sdf ditunjukkan pada Tabel 4.1.

Tabel 4.1 Application Settings untuk PATIENT01LocalCS

Name	PATIENT01LocalCS
Type	(Connection string)
Scope	Application
Value	Data Source = ".\PATIENT01.sdf"

Application settings untuk local connection string pada file Patient02.sdf ditunjukkan seperti pada Tabel 4.2.

Tabel 4.2 Application Settings untuk PATIENT02LocalCS

Name	PATIENT02LocalCS
Type	(Connection string)
Scope	Application
Value	Data Source = ".\PATIENT02.sdf"

2. Langkah kedua adalah mendefinisikan fungsi CekDatabase() untuk masing-masing file database yang akan dibuat. Gambar 4.12 di bawah menunjukkan source code fungsi tersebut. Jika file database tidak ada maka fungsi ini akan melakukan create file sesuai dengan connection string pada Application Settings.

```
Public Function CekDatabase() As Boolean
    Dim createdDatabase As Boolean = False
    Using conn As New SqlConnection
        conn.ConnectionString = _
            My.Settings.PATIENT01LocalCS
        If Not System.IO.File.Exists(conn.Database) Then
            createdDatabase = True
            Using engine As New _
                SqlConnection(My.Settings.PATIENT01LocalCS)
                engine.CreateDatabase()
            End Using
        End If
    End Using
    Return createdDatabase
End Function
```

Gambar 4.12 Source code fungsi CekDatabase()

3. Langkah ketiga adalah mendefinisikan fungsi GetRDAObject() untuk melakukan inialisasi objek RDA. Gambar 4.13 di bawah menunjukkan source code fungsi ini.

```
Public Function GetRDAObject() As _
    SqlCeRemoteDataAccess
    Dim rda As New SqlCeRemoteDataAccess()
    rda.InternetUrl = _
        "http://10.151.32.32/HealthMonitoring/sqlcesa30.dll"
    rda.InternetLogin = "mia"
    rda.InternetPassword = "bismillah"
    rda.LocalConnectionString = _
        My.Settings.PATIENT01LocalCS
    Return rda
End Function
```

Gambar 4.13 Source code fungsi GetRDAObject()

4. Langkah keempat adalah mendefinisikan server connection string yang menghubungkan program dengan database SQL Server. Sama seperti local connection string, disini juga dibuat 2 server connection string untuk masing-masing file database. Tabel 4.3 berisi deskripsi Application Settings untuk server connection string pada file Patient01.sdf.

Tabel 4.3 Application Settings untuk PATIENT01ServerCS

Name	PATIENT01ServerCS
Type	(Connection string)
Scope	Application
Value	Data Source=10.151.31.4; Initial Catalog=PATIENT01; Persist Security Info=True; User ID=sa;Password=rpl

Application settings untuk server connection string pada file Patient02.sdf ditunjukkan seperti pada Tabel 4.4.

Tabel 4.4 Application Settings untuk PATIENT02ServerCS

Name	PATIENT02ServerCS
Type	(Connection string)
Scope	Application
Value	Data Source=DIPONEGORO; Initial Catalog=PATIENT02; Persist Security Info=True; User ID=sa;Password=bismillah

5. Langkah kelima adalah melakukan load data SQL server berdasarkan server connection string.
6. Langkah keenam adalah mendefinisikan OleDb connection string ke database SQL Server untuk melakukan method pull. Tabel 4.5 berisi deskripsi Application Settings untuk connection string pada file Patient01.sdf.

Tabel 4.5 Application Settings untuk PATIENT01OleDbCS

Name	PATIENT01OleDbCS
Type	(Connection string)
Scope	Application
Value	Provider=SQLOLEDB; Data Source=10.151.31.4; Persist Security Info=True; Password=rpl;User ID=sa; Initial Catalog=PATIENT01

Application settings untuk OleDb connection string pada file Patient02.sdf ditunjukkan seperti pada Tabel 4.6.

Tabel 4.6 Application Settings untuk PATIENT02OleDbCS

Name	PATIENT0102eDbCS
Type	(Connection string)
Scope	Application
Value	Provider=SQLOLEDB;Data Source=DIPONEGORO; Persist Security Info=True; Password=bismillah;User ID=sa; Initial Catalog=PATIENT02

- Jika data telah di-load ke form, maka untuk melakukan create database, dilakukan pemanggilan fungsi CekDatabase() lalu GetRDAObject(). Setelah itu, dilakukan pemanggilan method pull dengan menggunakan OleDb connection string ke database SQL Server. Gambar 4.14 menunjukkan source code create file database.

```
Private Sub CreateButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CreateButton.Click
    Windows.Forms.Cursor.Current = Cursors.WaitCursor
    Dim rda As SqlCeRemoteDataAccess = Nothing
    Try
        CekDatabase1()
        rda = GetRDAObject1()
        rda.Pull("DataPasien", "SELECT * FROM PASIEN", My.Settings.PATIENT01OleDbCS, RdaTrackOption.TrackingOn)
    End Try
End Sub
```

```

    MessageBox.Show("Create database success!!")
Catch ex As Exception
    MessageBox.Show(ex.ToString())
Finally
    rda.Dispose()
    Windows.Forms.Cursor.Current = Cursors.Default
End Try
End Sub

```

Gambar 4.14 Source code create database

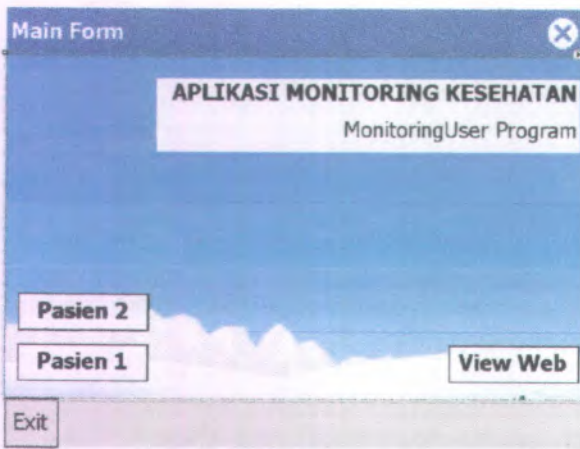
8. Melakukan pull data dari semua tabel pada database Patient01.mdf dan Patient02.mdf.

4.3.2. Mobile Device Monitoring-User

Pada mobile device monitoring-user terdapat device-program MonitoringUser.exe serta 2 database SQL Server Mobile, Patient01.sdf dan Patient02.sdf, yang masing-masing merupakan sinkronisasi dari database medical record pasien-1 dan pasien-2. Sesuai dengan diagram use case, monitoring-user dapat melakukan:

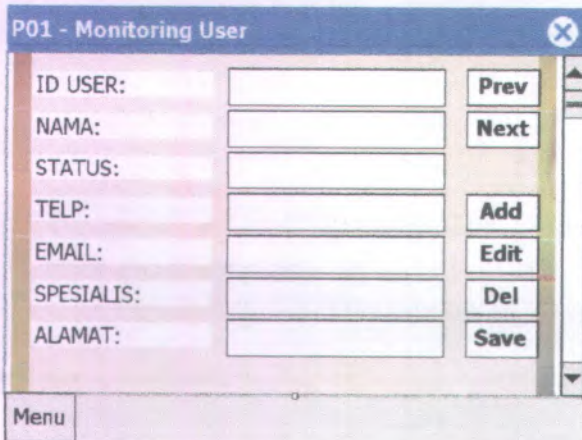
- update database SQL Server Mobile, Patient01.sdf dan Patient02.sdf, yang terdapat pada mobile device;
- akses database pasien dan database ServerMedicalRecord melalui web;
- propagate data (pull dan push).

Gambar 4.15 berikut menunjukkan interface MainForm pada program MonitoringUser pada mobile device.



Gambar 4.15 MainForm program MonitoringUser

Gambar 4.16 berikut menunjukkan interface MonitoringUserForm pada program MonitoringUser. Form ini akan menampilkan data monitoring user.



Gambar 4.16 MonitoringUserForm program MonitoringUser

4.3.2.1. Update Database SQL Server Mobile

Monitoring user dapat melakukan update terhadap database lokal pada mobile device. Update data dapat berupa add data, edit data, dan delete data. Untuk melihat dan melakukan update data, maka terlebih dahulu harus dilakukan load data dari tabel database ke form. Gambar 4.17 berikut menunjukkan source code untuk melakukan load data dari tabel ke form:

```
Private Sub LoadDataMenu_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Try
        If PATIENT01_MonitoringUser_DataSetUtil.
            DesignerUtil.IsRunTime Then
            Me.DataMonitoringUserTableAdapter.Fill
                (Me.PATIENT01_MonitoringUser_DataSet.
                    DataMonitoringUser)
            End If
        Catch
            MessageBox.Show(ex.ToString())
        End Try
    End Sub
```

Gambar 4.17 Source code load data SQL Server Mobile

Gambar 4.18 berikut menunjukkan source code untuk melakukan add data.

```
Private Sub AddButton_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles AddButton.Click
    Dim msg1 As String = "Anda ingin menambah data?"
    Dim msg2 As String = "Setelah menambah data klik Save
    untuk menyimpan perubahan."
    Dim title As String = "Pasien01 - Data Monitoring User"
    Dim style As MsgBoxStyle
    Dim response As MsgBoxResult
    style = MsgBoxStyle.Exclamation Or
    MsgBoxStyle.DefaultButton1 Or MsgBoxStyle.YesNo
    response = MsgBox(msg1, style, title)

    If response = MsgBoxResult.Yes Then
        response = MsgBox(msg2, MsgBoxStyle.OkOnly, title)
        Try
            Me.DataMonitoringUserBindingSource.AddNew()
        Catch ex As Exception
            MessageBox.Show(ex.ToString())
        End Try
    End If
End Sub
```

```

End Try
End If
End Sub

```

Gambar 4.18 Source code add data SQL Server Mobile

Setelah melakukan add data, maka monitoring-user harus melakukan save data untuk menyimpan perubahan pada tabel database lokal. Gambar 4.19 berikut menunjukkan source code save data.

```

Private Sub SaveButton_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles SaveButton.Click
Try
    Me.PATIENT01_MonitoirngUser_DataSet.AcceptChanges()
    Me.DataKeluargaBindingSource.EndEdit()
Me.DataMonitoringUserTableAdapter.Update(Me.PATIENT01_Monito
ringUser_DataSet.DataMonitoringUser)
    MessageBox.Show("Data telah disimpan!")
Catch ex As Exception
    MessageBox.Show(ex.ToString())
End Try
End Sub

```

Gambar 4.19 Source code save data SQL Server Mobile

Untuk melakukan edit data, dapat dilakukan pada text box secara langsung, lalu menyimpan perubahan data yang telah di-edit pada tabel database. Gambar 4.20 berikut menunjukkan source code edit data.

```

Private Sub EditButton_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles EditButton.Click
Try
    Me.DataMonitoringUserBindingSource.EndEdit()
Me.DataMonitoringUserTableAdapter.Update(Me.PATIENT01_Monito
ringUser_DataSet.DataMonitoringUser)
    Me.PATIENT01_MonitoirngUser_DataSet.AcceptChanges()
    MessageBox.Show("Data telah di-edit!")
Catch ex As Exception
    MessageBox.Show(ex.ToString())
End Try
End Sub

```

Gambar 4.20 Source code edit data SQL Server Mobile

Monitoring-user juga dapat menghapus data pada database lokal. Gambar 4.21 berikut menunjukkan source code delete data.

```
Private Sub DelButton_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles DelButton.Click
    Dim msg As String = "Anda ingin menghapus data ini?"
    Dim title As String = "Pasien01 - Data Monitoring User"
    Dim style As MsgBoxStyle
    Dim response As MsgBoxResult
    style = MsgBoxStyle.Exclamation Or
MsgBoxStyle.DefaultButton2 Or MsgBoxStyle.YesNo
    response = MsgBox(msg, style, title)
    If response = MsgBoxResult.Yes Then
        Try
            Me.DataMonitoringUserBindingSource.RemoveCurrent()
            Me.DataMonitoringUserTableAdapter.Update(Me.PATIENT01_Monito
ringUser_DataSet.DataMonitoringUser)
            MessageBox.Show("Data telah dihapus!")
        Catch ex As Exception
            MessageBox.Show(ex.ToString())
        End Try
    End If
End Sub
```

Gambar 4.21 Source code delete data SQL Server Mobile

4.3.2.2. Propagate Data

Untuk melakukan propagate data (pull dan push), yang pertama kali dilakukan adalah mendefinisikan connection string dan fungsi-fungsi yang digunakan. Pada mobile device monitoring-user terdapat 2 file database yaitu database pasien-1 (Patient01.sdf) dan database pasien-2 (Patient02.sdf). Untuk melakukan propagate data dibutuhkan 2 connection string untuk tiap database, local connection string ke file database pada mobile device, dan OleDb connection string ke back-end server.

Connection-string untuk file database Patient01.sdf pada program MonitoringUser ditunjukkan seperti pada Gambar 4.22. Pada mobile device, file database terletak pada folder `Program Files\MonitoringUser`

```
Public localCS1 As String = "Data Source = '\Program
Files\MonitoringUser\PATIENT01.sdf';"

Public OleDbCS1 As String = "Provider=SQLOLEDB; Data
Source=10.151.31.4;Persist Security
Info=True;Password=rpl;User ID=sa;Initial Catalog=PATIENT01"
```

Gambar 4.22 Connection string Patient01.sdf pada program MonitoringUser

Connection-string untuk file database Patient02.sdf pada program MonitoringUser ditunjukkan seperti pada Gambar 4.23 berikut.

```
Public localCS2 As String = "Data Source = '\Program
Files\MonitoringUser\PATIENT02.sdf';"

Public OLEDBCS2 As String = "Provider=SQLOLEDB;Data
Source=DIPONEGORO;Persist Security
Info=True;Password=bismillah;User ID=sa;Initial
Catalog=PATIENT02"
```

Gambar 4.23 Connection string Patient02.sdf pada program MonitoringUser

Ada 2 fungsi yang digunakan yaitu GetRDAObject() untuk inialisasi objek RDA dan DeleteTable() untuk mengeset tabel = NULL sebelum melakukan pull data. Karena terdapat 2 database sehingga diimplementasikan 2 fungsi GetRDAObject() dan 2 fungsi DeleteTable(). Fungsi GetRDAObject() sama seperti pada program Server. Sedangkan source code fungsi DeleteTable() seperti terlihat pada Gambar 4.24 berikut.

```
Public Function DeleteTable _
    (ByVal tableName As String) As Boolean
    Dim tableFound As Boolean = False
    Using conn As New SqlCeConnection
        Dim cmd As New SqlCeCommand()
        conn.ConnectionString = localCS1
        cmd.Connection = conn
        cmd.CommandText = "SELECT COUNT (*) FROM
INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = @table_name"
        cmd.Parameters.Add("@table_name", tableName)
        conn.Open()
```

```

If cmd.ExecuteNonQuery() = 1 Then
    tableFound = True
    cmd.Parameters.Clear()
    cmd.CommandText = String.Format("DROP TABLE {0}",
tableName)
    cmd.ExecuteNonQuery()
End If
End Using
Return tableFound
End Function

```

Gambar 4.24 Source code fungsi DeleteTable()

Fungsi ShowErrors() juga diimplementasikan pada program MonitoringUser. Fungsi ini menggunakan class sqlCeException untuk melakukan catch terhadap exception SQL Server Mobile. Gambar 4.25 menunjukkan source code fungsi ini.

```

Public Sub ShowErrors(ByVal e As SqlCeException)
Dim errorCollection As SqlCeErrorCollection = e.Errors
Dim bld As New StringBuilder()
Dim inner As Exception = e.InnerException

If Not inner Is Nothing Then
    MessageBox.Show(("Inner Exception: " & inner.ToString()))
End If

Dim err As SqlCeError

' Enumerate each error to a message box.
For Each err In errorCollection
    bld.Append((ControlChars.Cr & " Error Code: " &
err.HResult.ToString("X")))
    bld.Append((ControlChars.Cr & " Message : " &
err.Message))
    bld.Append((ControlChars.Cr & " Minor Err.: " &
err.NativeError))
    bld.Append((ControlChars.Cr & " Source : " &
err.Source))

' Retrieve the error parameter numbers for each error.
Dim numPar As Integer
For Each numPar In err.NumericErrorParameters
    If 0 <> numPar Then
        bld.Append((ControlChars.Cr & " Num. Par. : " &
numPar))
    End If

```

```

Next numPar

' Retrieve the error parameters for each error.
Dim errPar As String
For Each errPar In err.ErrorParameters
    If [String].Empty <> errPar Then
        bld.Append((ControlChars.Cr & " Err. Par. : " &
errPar))
    End If
Next errPar

MessageBox.Show(bld.ToString())
bld.Remove(0, bld.Length)
Next err
End Sub

```

Gambar 4.25 Source code fungsi ShowErrors()

Pull Data

Source code pull data ditunjukkan seperti pada Gambar 4.26. Pada gambar tersebut pull data dilakukan dari tabel MONITORINGUSER pada database SQL Server ke tabel DataMonitoringUser pada database SQL Server Mobile.

```

Private Sub PullMenu_Click(ByVal sender As _
System.Object, ByVal e As System.EventArgs) _
Handles PullMenu.Click
Windows.Forms.Cursor.Current = Cursors.WaitCursor
Dim rda As SqlCeRemoteDataAccess = Nothing
Try
    rda = GetRDAObject1()
    DeleteTable1("DataMonitoringUser")
    rda.Pull("DataMonitoringUser", "SELECT * FROM
MONITORINGUSER", OLEDBCS1, RdaTrackOption.TrackingOn)
    MessageBox.Show("Data has been pulled!")
Catch ex As Exception
    MessageBox.Show(ex.ToString())
Finally
    rda.Dispose()
    Windows.Forms.Cursor.Current = Cursors.Default
End Try
End Sub

```

Gambar 4.26 Source code pull data

Push Data

Source code push data ditunjukkan seperti pada Gambar 4.27. Pada gambar tersebut push data dilakukan dari tabel DataMonitoringUser pada database SQL Server Mobile (Patient01.sdf) ke tabel MONITORINGUSER pada database SQL Server.

```
Private Sub PushMenu_Click _
    (ByVal sender As System.Object, ByVal e As _
    System.EventArgs) Handles PushMenu.Click
    Windows.Forms.Cursor.Current = Cursors.WaitCursor
    Dim rda As SqlCeRemoteDataAccess = Nothing
    Try
        rda = GetRDAObject1()
        rda.Push("DataMonitoringUser", OLEDBCS1,
        RdaBatchOption.BatchingOff)
    Catch ex As Exception
        MessageBox.Show(ex.ToString())
    Finally
        rda.Dispose()
        Windows.Forms.Cursor.Current = Cursors.Default
    End Try
End Sub
```

Gambar 4.27 Source code push data

Pull dan push data juga dapat dilakukan pada tabel-tabel yang lain pada database Patient01.sdf dan Patient02.sdf.

4.3.3. Home-PC Pasien-1 (IP-Address Statis)

Pada home-PC pasien-1 terdapat computer-program, Pasien01.exe, dan database SQL Server Patient01.mdf. Database Patient01.mdf berisi data medical record pasien-1. Program Pasien01 berfungsi sebagai interface bagi pasien untuk menyimpan data medical record-nya pada database.

Sesuai dengan diagram use case dan activity, pasien-1 hanya melakukan update database medical record-nya (SQL Server) pada home-PC. Pasien juga dapat mengakses aplikasi web melalui WebForm. Update data dapat berupa add data, edit data,

dan delete data. Sebelum melakukan update, pasien harus melakukan load data, dan setelah melakukan update pasien dapat melakukan save data ke database.

Gambar 4.28 berikut menunjukkan interface MainForm pada program Pasien01. Button ViewWeb digunakan untuk membuka WebForm yang langsung menampilkan halaman aplikasi web jika koneksi tersedia.



Gambar 4.28 MainForm program Pasien01

Gambar 4.29 berikut menunjukkan tampilan PasienForm pada program Pasien01. Pada BindingNavigator, terdapat ToolStripButton LoadData untuk melakukan load data dari database ke form.

Gambar 4.29 PasienForm program Pasien01

Edit data dilakukan secara langsung pada TextBox yang bersangkutan. Begitu pula add data dilakukan dengan mengklik icon AddNewItem (+), serta delete data dilakukan dengan mengklik icon DeleteItem (X) pada BindingNavigator. Gambar 4.30 menunjukkan source code load data pada program Pasien01.

```
Private Sub LoadDataButton_Click(ByVal sender As _
    System.Object, ByVal e As System.EventArgs) _
    Handles LoadDataButton.Click
    Try
        Me.PASIENTableAdapter.Fill _
            (Me.PATIENT01_Pasien_DataSet.PASIEN)
    Catch
        MessageBox.Show(ex.ToString())
    End Try
End Sub
```

Gambar 4.30 Source code load data SQL Server

Gambar 4.31 berikut menunjukkan source save data pada program Pasien01.

```

Private Sub PASIENBindingNavigatorSaveItem_Click _
    (ByVal sender As System.Object, ByVal e As _
    System.EventArgs) Handles _
    PASIENBindingNavigatorSaveItem.Click
Try
    Me.Validate()
    Me.PASIENBindingSource.EndEdit()
    Me.PASIENTableAdapter.Update _
    (Me.PATIENT01_Pasien_DataSet.PASIEN)
Catch
    MessageBox.Show(ex.ToString())
End Try
End Sub

```

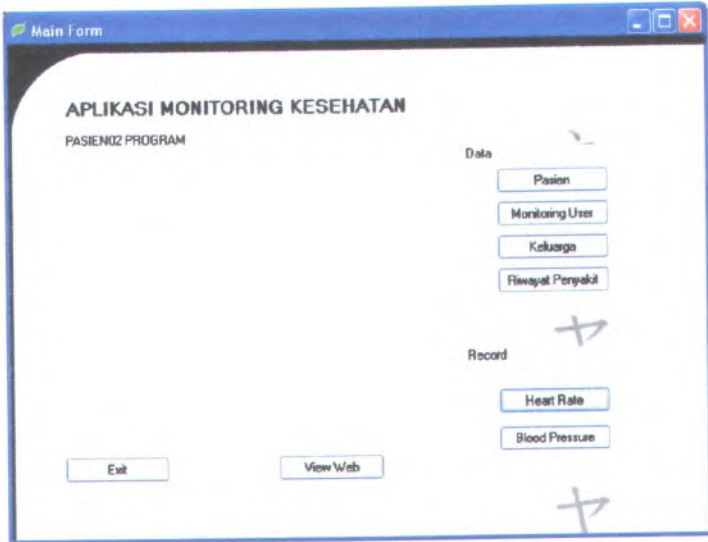
Gambar 4.31 Source code save data SQL Server

4.3.4. Home-PC Pasien-2 (IP-Address Dinamis)

Pada home-PC pasien-2 juga terdapat computer program, Pasien02.exe. Berbeda dengan pasien-1, pada pasien-2 data medical record disimpan pada database SQL Server Mobile, Patient02.sdf.

Sama seperti monitoring-user, pasien-2 juga melakukan update database SQL Server Mobile dan propagate data (pull dan push) dari database SQL Server Mobile pada home-PC pasien ke database SQL Server pada broker server. Selain itu pasien juga dapat mengakses aplikasi web melalui WebForm.

Interface MainForm pada program Pasien02 ditunjukkan seperti pada Gambar 4.32.



Gambar 4.32 MainForm program Pasien02

Interface PasienForm pada program Pasien02 ditunjukkan seperti pada Gambar 4.33. Berbeda dengan program Pasien01, pada program Pasien02, selain ToolStripButton LoadData pada BindingNavigator, juga terdapat button PullData untuk melakukan pull data dan button PushData untuk melakukan push data.

Gambar 4.33 PasienForm program Pasien02

4.3.4.1. Update Database SQL Server CE

Sama seperti program Pasien01, update data pada program Pasien01 dapat berupa add data, edit data dan delete data. Edit data dilakukan secara langsung pada TextBox yang bersangkutan. Add data dilakukan dengan mengklik icon AddNewItem (+), serta delete data dilakukan dengan mengklik icon DeleteItem (X) pada BindingNavigator. Source code untuk melakukan load dan save data sama seperti program Pasien01.

4.3.4.2. Propagate Data

Seerti pada program Server dan MonitoringUser, untuk melakukan propagate data (pull dan push), yang pertama kali dilakukan adalah mendefinisikan connection string dan fungsi-fungsi yang digunakan, yaitu fungsi GetRDAObject() untuk

melakukan inialisasi objek RDA, DeleteTable() untuk mengeset nilai tabel ke NULL, method pull dan push data.

Untuk melakukan propagate data dibutuhkan 2 connection string ke database yaitu: local connection string ke file database SQL Server Mobile pada mobile device, dan OleDb connection string ke back-end server. Gambar 4.34 menunjukkan connection string pada program Pasien02. File database PATIENT02.sdf terletak di folder 'D:\Pasien02\`.

```
Public localCS As String = "Data Source =
'D:\Pasien02\PATIENT02.sdf';"

Public OleDbCS As String = "Provider=SQLOLEDB; Data
Source=10.151.32.32;Persist Security
Info=True;Password=bismillah;User ID=sa;Initial
Catalog=PATIENT02"
```

Gambar 4.34 Connection string pada program Pasien02

Source code fungsi GetRDAObject() sama seperti program Server, source code fungsi DeleteTable(), method pull dan push data sama seperti pada program MonitoringUser.

Halaman ini sengaja dikosongkan

BAB V

UJI COBA DAN EVALUASI

Pada bab ini akan dibahas mengenai uji coba fungsional dan evaluasi perangkat lunak untuk kedua arsitektur remote data access yang telah diimplementasikan. Uji coba dilakukan untuk melihat apakah fungsi-fungsi dasar dari perangkat lunak ini berjalan sebagaimana mestinya atau tidak. Hasil uji coba ditunjukkan dengan gambar-gambar dari *Graphical User Interface* (GUI) perangkat lunak.

Uji coba dilakukan pada Laboratorium NCC (Net Centric Computing) dan RPL (Rekayasa Perangkat Lunak) ITS. Spesifikasi komputer untuk broker server dan mobile device yang digunakan adalah:

1. Broker server

IP address : 10.151.32.32 (Lab NCC)
Computer Name : DIPONEGORO
Processor : Intel x86 2,4 GHz
Memory : 2 GB
OS Name : Microsoft Windows XP Professional
OS Version : 5.1.2600 Service Pack 3 Build 2600
Database : Microsoft SQL Server 2005
Web server : Microsoft IIS 5.1

2. Mobile device monitoring user

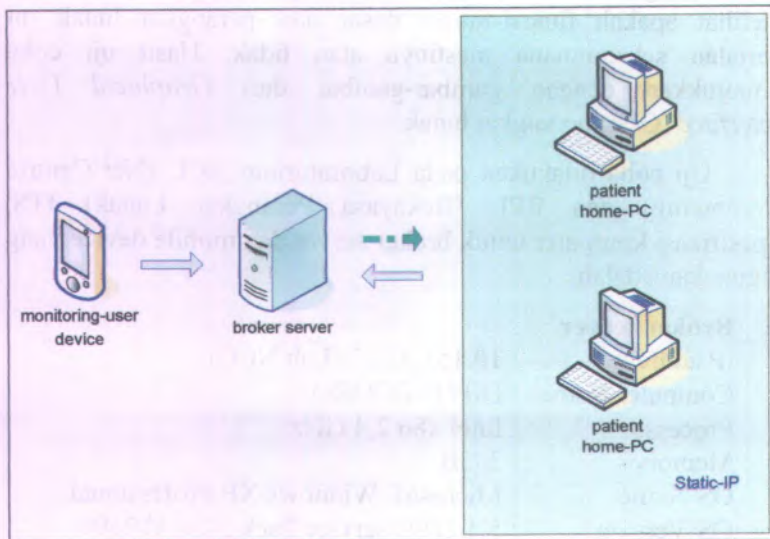
Product ID : PocketPC
Model ID : HP iPAQ hx4700
Processor Type : PXA270
System ROM : 128 MB
System RAM : 64 MB
OS Name : Microsoft Windows Mobile 2003
OS Version : Windows CE 4.21
Database : Microsoft SQL Server 2005 Mobile
Edition 3.0



Sedangkan untuk home-PC pasien disesuaikan dengan arsitektur masing-masing.

5.1. Uji Coba Fungsionalitas Arsitektur-1

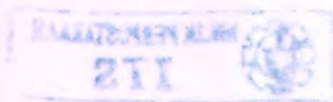
Skenario uji coba untuk arsitektur-1 ditunjukkan seperti pada Gambar 5.1.



Gambar 5.1 Uji coba arsitektur-1

Karena home-PC pasien-1 memiliki IP-address statis (10.151.31.4) maka pasien tidak perlu melakukan propagate data (pull dan push). Propagate data dilakukan oleh monitoring-user pada program MonitoringUser pada mobile device.

Sesuai dengan implementasi arsitektur-1, untuk melakukan propagate data, monitoring-user mengirim request dari mobile device (SQL Server Client Agent) ke broker server (SQL Server Mobile Server Agent). Jika request telah diterima oleh broker server, maka SQL Server Mobile Server Agent akan melakukan forward request tersebut ke OleDb .Net Provider for



SQL Server pada home-PC pasien. Request ini selanjutnya diteruskan ke database SQL Server. Jika request telah diterima oleh SQL Server, maka response akan dikirim kembali sesuai dengan jalur sebelumnya ke mobile device monitoring-user.

Spesifikasi komputer untuk home-PC pasien-1 untuk uji coba adalah:

IP address	: 10.151.31.4 (Lab RPL)
Computer Name	: ANDALUSIA
Processor	: Intel x86 2,6 GHz
Memory	: 2 GB
OS Name	: Microsoft Windows Server 2003
OS Version	: 5.2.3790 Service Pack 1 Build 3790
Database	: Microsoft SQL Server 2005

5.1.1. Uji Coba Pull Data

Pada uji coba ini, data yang di-pull adalah data pada tabel PASIEN dari database SQL Server pada home-PC pasien ke database SQL Server Mobile pada mobile device monitoring-user.

Gambar 5.2 menunjukkan tampilan Form Pasien01 pada home-PC sebelum data di-load.

The screenshot shows a web browser window with the title "Data Pasien". The address bar shows "of 0" and a "Load Data" button. The main content area is divided into two columns. The left column contains a form with the following fields: ID PASIEN, NAMAPASIE, JENISKELAMIN, TGLLAHIR (a dropdown menu showing "Sunday, June 14, 2009"), USIA, ALAMAT, TELP, EMAIL, PEKERJAAN, and IP ADDRESS. The right column contains a "Data" section with four buttons: "Pasien", "Monitoring User", "Keluarga", and "Rimwayat Penyakit". Below the "Data" section is a "Hasil Pengukuran" section with two buttons: "Cholesterol Rate" and "Blood Glucose". At the bottom right of the window, there is a decorative graphic of a stethoscope.

Gambar 5.2 Form Pasien01 pada home-PC

Gambar 5.3 menunjukkan tampilan Form Pasien01 pada home-PC setelah data di-load. Data pasien di-load dari database SQL Server pada home-PC pasien kemudian ditampilkan ke Form Pasien01.

The screenshot shows a web browser window with the title "Data Pasien". The address bar shows "of 1" and "Load Data". The main content area is divided into two columns. The left column contains a form with the following fields and values:

ID PASIEN:	PS001
NAMAPASIEN:	Yuyun Masriana
JENISKELAMIN:	
TGLLAHIR:	Friday, August 14, 1987
USIA:	
ALAMAT:	Kendari
TELP:	
EMAIL:	yuyun_m@yahoo.co.id
PEKERJAAN:	PNS
IPADDRESS:	10.151.31.4

The right column contains two sections of buttons. The "Data" section has buttons for "Pasien", "Monitoring User", "Keluarga", and "Riwayat Penyakit". The "Hasil Pengukuran" section has buttons for "Cholesterol Rate" and "Blood Glucose".

Gambar 5.3 Form Pasien01 pada home-PC setelah load data

Gambar 5.4 berikut menunjukkan tampilan Form Pasien01 pada program MonitoringUser pada mobile device yang menampilkan data pasien awal. Data yang ditampilkan sama seperti data pada program Pasien01 pada home-PC pasien.

The screenshot shows a mobile application window with the title "P01 - Pasien". The status bar at the top right shows the time "11:08". The main content area displays the patient data in a list view:

ID PASIEN:	PS001
NAMA PASIEN:	Yuyun Masriana
JENIS KELAMIN:	
TGL LAHIR:	8/14/87 12:00:00 AM
USIA:	
ALAMAT:	Kendari
TELP:	
EMAIL:	yuyun_m@yahoo.co.id

A "Menu" button is visible at the bottom left of the form.

Gambar 5.4 FormPasien01 pada mobile device sebelum pull data

Selanjutnya pada home-PC pasien, data pasien di-edit, nama pasien diganti dari “Yuyun Masriana” menjadi “Yuyun M.”, seperti yang ditunjukkan pada Gambar 5.5.

Data Pasien	
ID PASIEN:	PS001
NAMAPASIAN:	Yuyun M.
JENISKELAMIN:	
TGLLAHIR:	Friday, August 14, 1987
USIA:	
ALAMAT:	Kendari
TELP:	
EMAIL:	yuyun_m@yahoo.co.id
PEKERJAAN:	PNS
IP ADDRESS:	10.151.31.4

Data	
Pasien	
Monitoring User	
Keluarga	
Riwayat Penyakit	

Hasil Pengukuran	
Cholesterol Rate	
Blood Glucose	

Gambar 5.5 Form Pasien01 pada home-PC setelah edit data

Pada mobile device, dilakukan pull data untuk meng-update database lokal. Gambar 5.6 berikut menunjukkan tampilan form pada mobile device ketika proses pull data.

P01 - Pasien	
ID PASIEN:	PS001
NAMA PASIEN:	Yuyun Masriana
JENIS KELAMIN:	
TGL LAHIR:	
USIA:	
ALAMAT:	
TELP:	
EMAIL:	yuyun_m@yahoo.co.id

Dialog Box: Pull data berhasil! (ok)

Gambar 5.6 FormPasien01 pada mobile device ketika pull data

Gambar 5.7 berikut menunjukkan hasil setelah pull data dilakukan. Sekarang nama pasien berubah menjadi "Yuyun M."

P01 - Pasien	
ID PASIEN:	PS001
NAMA PASIEN:	Yuyun M.
JENIS KELAMIN:	
TGL LAHIR:	8/14/87 12:00:00 AM
USIA:	
ALAMAT:	Kendari
TELP:	
EMAIL:	yuyun_m@yahoo.co.id

Gambar 5.7 FormPasien01 pada mobile device setelah pull data

5.1.2. Uji Coba Push Data

Untuk uji coba push data, digunakan tabel MONITORINGUSER. Push data dilakukan dari database SQL Server Mobile pada mobile device monitoring-user ke database SQL Server pada home-PC pasien. Gambar 5.8 berikut

menunjukkan tampilan awal Form MonitoringUser01 pada mobile device.

The screenshot shows a mobile application window titled "P01 - Monitoring User". The window contains a form with the following fields and buttons:

ID USER:	U0101	Prev
NAMA:	Alini K.	Next
STATUS:	Dokter	
TELP:		Add
EMAIL:	alini_k@yahoo.co.id	Edit
SPECIALIS:	Cardiovascular	Del
ALAMAT:	Kendari	Save

At the bottom of the form, there is a "Menu" button.

Gambar 5.8 Form MonitoringUser01 awal pada mobile device

Gambar 5.9 berikut menunjukkan tampilan Form MonitoringUser01 pada home-PC.

The screenshot shows a desktop application window titled "Data Monitoring User". The window contains a form with the following fields and buttons:

ID USER:	U0101
NAMA:	Alini K.
STATUS:	Dokter
ALAMAT:	Kendari
TELP:	
EMAIL:	alini_k@yahoo.co.id
SPECIALIS:	Cardiovascular

On the right side of the form, there are two sections:

- Data**
 - Pasien
 - Monitoring User
 - Keluarga
 - Riwayat Penyakit
- Hasil Pengukuran**
 - Cholesterol Rate
 - Blood Glucose

At the bottom of the window, there is a decorative graphic of a stethoscope.

Gambar 5.9 Form MonitoringUser01 awal pada home-PC

Selanjutnya data Monitoring User pada mobile device di-edit. Nama monitoring-user diganti dari “Alini K.” menjadi “Alini Kusuma” seperti pada Gambar 5.10 berikut.

ID USER:	U0101	Prev
NAMA:	Alini Kusuma	Next
STATUS:	Dokter	
TELP:		Add
EMAIL:	alini_k@yahoo.co.id	Edit
SPELIALIS:	Cardiovascular	Del
ALAMAT:	Kendari	Save

Gambar 5.10 Form MonitoringUser01 pada mobile device setelah edit data

Selanjutnya dilakukan push data ke database SQL Server pada home-PC pasien-1, seperti pada Gambar 5.11 berikut.

ID USER:	U0101	Prev
NAMA:	Alini Kusuma	Next
STATUS:	Dokter	
TELP:		Add
EMAIL:	alini_k@yahoo.co.id	Edit
SPELIALIS:	Cardiovascular	Del
ALAMAT:	Kendari	Save

Gambar 5.11 Form MonitoringUser01 pada mobile device ketika push data

Gambar 5.12 berikut menunjukkan tampilan Form MonitoringUser01 pada home-PC pasien setelah proses push data dan reload data.

The screenshot shows a web browser window titled "Data Monitoring User". The address bar shows "of 3" and a "Load Data" button. The main content area contains a form with the following fields and values:

ID USER:	U0101
NAMA:	Alini Kusuma
STATUS:	Dokter
ALAMAT:	Kendari
TELP:	
EMAIL:	alini_k@yahoo.co.id
SPECIALIS:	Cardiovascular

On the right side of the form, there are two sections:

- Data:** Contains four buttons: "Pasien", "Monitoring User", "Keluarga", and "Riwayat Penyakit".
- Hasil Pengukuran:** Contains two buttons: "Cholesterol Rate" and "Blood Glucose".

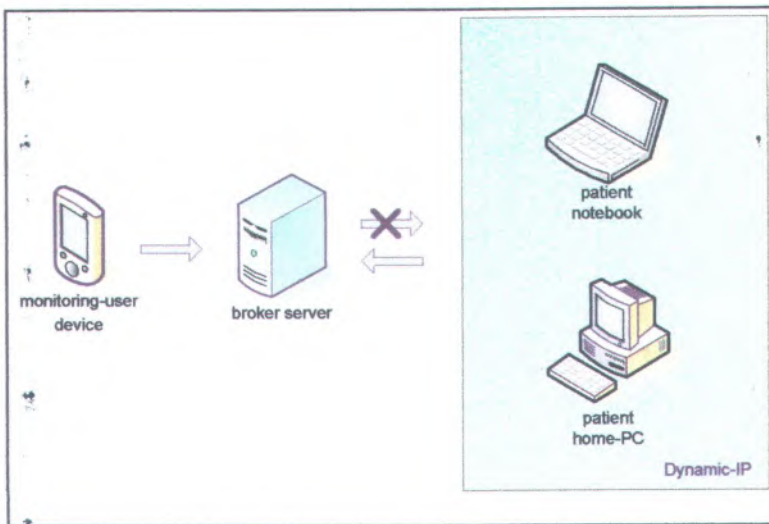
At the bottom right of the form, there is a decorative graphic of a stethoscope.

Gambar 5.12 Form MonitoringUser01 pada home-PC setelah push data

5.2. Uji Coba Fungsionalitas Arsitektur-2

Pada arsitektur 2, propagate data (pull dan push) dilakukan 2 tahap. Tahap 1 antara database SQL Server Mobile pada home-PC pasien dan database SQL Server pada broker server. Tahap 2 antara database SQL Server Mobile pada mobile device monitoring-user dengan database SQL Server pada broker server.

Skenario uji coba untuk arsitektur-2 ditunjukkan seperti pada Gambar 5.13 dibawah ini:



Gambar 5.13 Uji coba arsitektur-2

Spesifikasi komputer untuk home-PC pasien-2 untuk uji coba adalah:

IP address	: dinamis (Lab NCC)
Processor	: Intel x86 2,4 GHz
Memory	: 2 GB
OS Name	: Microsoft Windows Server 2003
OS Version	: 5.2.3790 Service Pack 1 Build 3790
Database	: Microsoft SQL Server Compact Edition 3.1

5.2.1. Uji Coba Pull Data

Pada uji coba ini, monitoring-user akan melakukan pull data tabel PASIEN. Gambar 5.14 berikut menunjukkan tampilan PasienForm pada program MonitoringUser sebelum data di-load.

ID PASIEN:	
NAMA PASIEN:	
JENIS KELAMIN:	
TGL LAHIR:	
USIA:	
ALAMAT:	
TELP:	
EMAIL:	

Menu

Gambar 5.14 Form Pasien02 pada mobile device

Gambar 5.15 berikut menunjukkan tampilan Form Pasien02 pada program MonitoringUser pada mobile device setelah data di-load.

ID PASIEN:	PS002
NAMA PASIEN:	Fira Febrianti
JENIS KELAMIN:	P
TGL LAHIR:	2/9/87 12:00:00 AM
USIA:	
ALAMAT:	Kendari
TELP:	6281342602025
EMAIL:	fira_f@yahoo.co.id

Menu

Gambar 5.15 Form Pasien02 pada mobile device setelah load data

Gambar 5.16 berikut menunjukkan tampilan awal Form Pasien02 pada program Pasien02 pada home-PC pasien-2.

The screenshot shows a web browser window with the title "Data Pasien". The address bar shows "of 1" and navigation icons. The main content area is divided into two sections. On the left is a form with the following fields and values:

ID PASIEN:	PS002
NAMAPASIEIN:	Fira Febrianti
JENISKELAMIN:	P
TGLLAHIR:	Monday, February 09, 1987
USIA:	
ALAMAT:	Kendari
TELP:	6281342602025
EMAIL:	fira_f@yahoo.co.id
PEKERJAAN:	PNS
IP ADDRESS:	169.254.2.2

On the right is a sidebar with the title "Data". It contains several buttons:

- Pasien
- Monitoring User
- Keluarga
- Riwayat Penyakit
- Record
- Heart Rate
- Blood Pressue

Handwritten blue 'X' marks are present over the "Record", "Heart Rate", and "Blood Pressue" buttons.

Gambar 5.16 Form Pasien02 awal pada home-PC

Selanjutnya data pasien pada home-PC di-edit. Nama pasien diganti dari "Fira Febrianti" menjadi "Fira F." seperti yang ditunjukkan pada Gambar 5.17 berikut.

ID PASIEN:	PS002
NAMA PASIEN:	Fira F.
JENISKELAMIN:	P
TGL LAHIR:	Monday, February 09, 1987
USIA:	
ALAMAT:	Kendari
TELP:	6281342602025
EMAIL:	fira_f@yahoo.co.id
PEKERJAAN:	PNS
IP ADDRESS:	169.254.2.2

Data

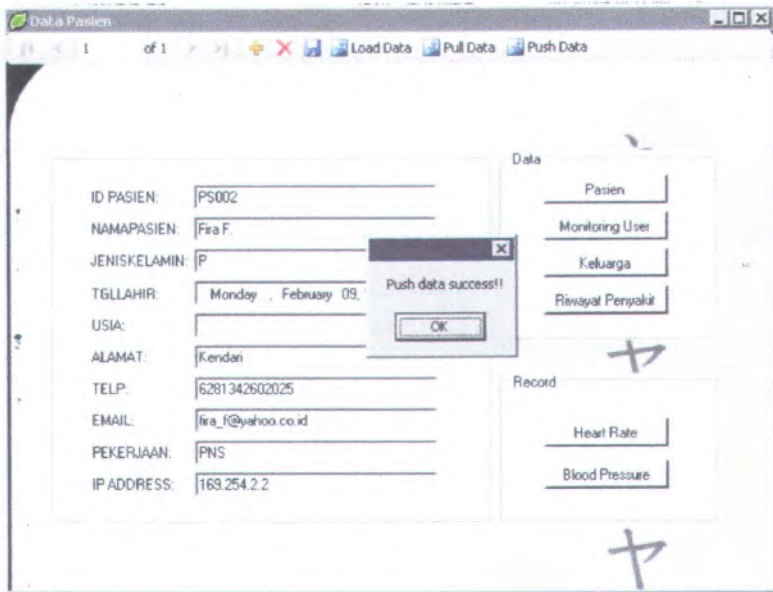
- [Pasien](#)
- [Monitoring User](#)
- [Keluarga](#)
- [Rawat Perakit](#)

Record

- [Heart Rate](#)
- [Blood Pressure](#)

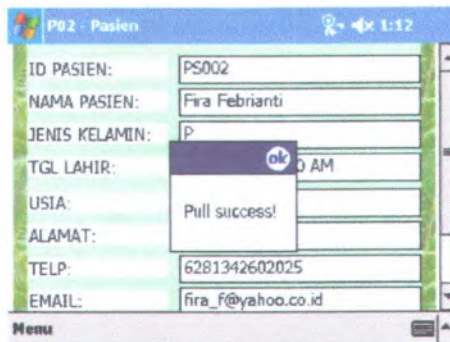
Gambar 5.17 Form Pasien02 pada home-PC setelah edit data

Selanjutnya untuk melakukan sinkronisasi dengan database pada mobile device monitoring-user, maka pasien harus melakukan push data dari database SQL Server Mobile pada home-PC ke database SQL Server pada broker server, seperti yang ditunjukkan pada Gambar 5.18 berikut.



Gambar 5.18 Push data Pasien02 pada home-PC

Selanjutnya untuk melakukan sinkronisasi data, maka monitoring-user harus melakukan pull data dari database SQL Server pada broker server ke database SQL Server Mobile pada mobile device, seperti yang ditunjukkan pada Gambar 5.19 berikut.



Gambar 5.19 Form Pasien02 pada mobile device ketika pull data

Setelah data pasien02 di-load maka nama pasien sekarang sudah berubah menjadi "Fira F." sama seperti data pasien pada home-PC. Tampilan Pasien02Form sekarang ditunjukkan seperti pada Gambar 5.20 berikut.

ID PASIEN:	PS002
NAMA PASIEN:	Fira F.
JENIS KELAMIN:	P
TGL LAHIR:	2/9/87 12:00:00 AM
USIA:	
ALAMAT:	Kendari
TELP:	6281342602025
EMAIL:	fira_f@yahoo.co.id

Gambar 5.20 Form Pasien02 pada mobile device setelah pull data

5.2.2. Uji Coba Push Data

Uji coba push data dilakukan antara data monitoring-user pada database SQL Server Mobile pada mobile device dan database SQL Server pada home-PC pasien-2. Gambar 5.21 berikut menunjukkan tampilan MonitoringUser02Form pada mobile device.

ID USER:	U0201	Prev
NAMA:	Gita Fauziah	Next
STATUS:	Dokter	
ALAMAT:	Kendari	Add
TELP:		Edit
EMAIL:	gita_f@yahoo.co.id	Del
SPESIALIS:	Penyakit dalam	Save

Menu

Gambar 5.21 Form MonitoringUser02 awal pada mobile device

Gambar 5.22 berikut menunjukkan tampilan Form MonitoringUser02 pada home-PC pasien-2. Data monitoring-user pada home-PC pasien sama dengan data pada mobile device.

ID USER:	U0201
NAMA:	Gita Fauziah
STATUS:	Dokter
ALAMAT:	Kendari
TELP:	
EMAIL:	gita_f@yahoo.co.id
SPESIALIS:	Penyakit dalam

Data

- Pasien
- Monitoring User
- Keluarga
- Riwayat Penyakit

Record

- Heart Rate
- Blood Pressure

Gambar 5.22 Form MonitoringUser02 awal pada home-PC

Selanjutnya, pada mobile device, data monitoring user di-edit. Nama monitoring-user diganti dari “Gita Fauziah” menjadi “Gita F.” seperti yang ditunjukkan pada Gambar 5.23 berikut.

ID USER:	U0201	Prev
NAMA:	Gita F.	Next
STATUS:	Dokter	
ALAMAT:		Add
TELP:		Edit
EMAIL:		Del
SPESIALIS:	Penyakit dalam	Save

Gambar 5.23 Form MonitoringUser02 pada mobile device setelah edit data

Data yang telah di-edit selanjutnya disimpan ke database seperti yang ditunjukkan pada Gambar 5.24 berikut.

ID USER:	U0201	Prev
NAMA:	Gita F.	Next
STATUS:	Dokter	
ALAMAT:	Kendari	Add
TELP:		Edit
EMAIL:	gita_f@yahoo.co.id	Del
SPESIALIS:	Penyakit dalam	Save

Gambar 5.24 Form MonitoringUser02 pada mobile device setelah save data

Selanjutnya, pada mobile device, dilakukan push data ke database SQL Server pada broker server seperti yang ditunjukkan pada Gambar 5.25 berikut.

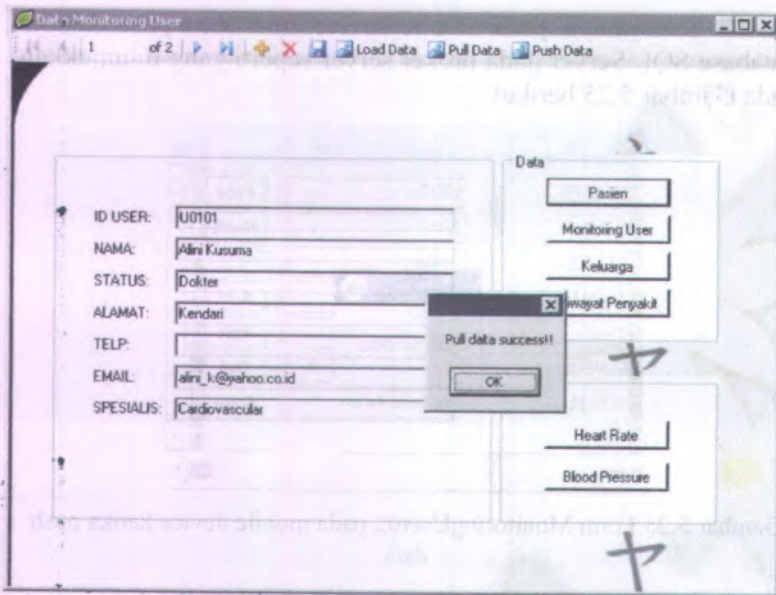
The screenshot shows a mobile application window titled "P02 - Monitoring User" with a status bar at the top showing signal strength, Wi-Fi, and the time 2:00. The form contains the following fields and buttons:

- ID USER: U0201 (with "Prev" button)
- NAMA: Gita F. (with "Next" button)
- STATUS: Dokter
- ALAMAT: (with "Add" button)
- TELP: (with "Edit" button)
- EMAIL: (with "Del" button)
- SPESIALIS: Penyakit dalam (with "Save" button)

A modal dialog box is displayed in the center with the text "Push success!" and an "ok" button.

Gambar 5.25 Form MonitoringUser02 pada mobile device ketika push data

Pada home-PC pasien-2 dilakukan pull data untuk melakukan sinkronisasi data dengan mobile device, seperti yang ditunjukkan pada Gambar 5.26 berikut.



Gambar 5.26 Pull data Pasien02 pada home-PC

Selanjutnya, pada home-PC, data di-load ulang untuk melihat hasil sinkronisasi data, seperti yang ditunjukkan pada Gambar 5.27.

Move next

ID USER: U0201
 NAMA: Gita F
 STATUS: Dokter
 ALAMAT: Kendari
 TELP:
 EMAIL: gita_f@yahoo.co.id
 SPESIALIS: Penyakit dalam

Data

Pasien
 Monitoring User
 Keluarga
 Riwayat Penyakit

Record

Heart Rate
 Blood Pressure

Gambar 5.27 Form MonitoringUser02 pada home-PC setelah pull data

Sekarang, data monitoring-user pada home-PC pasien sama dengan data monitoring-user pada mobile device.

5.3. Uji Coba Performa

Uji coba performa dilakukan pada arsitektur-1. Ada 2 skenario yang dilakukan pada uji coba ini. Skenario pertama dilakukan dengan menambah jumlah client yang melakukan sinkronisasi data. Skenario kedua dilakukan dengan mengurangi kapasitas memory pada broker server.

5.3.1. Uji Coba Jumlah Client

Uji coba ini dilakukan untuk jumlah client (pasien) 1, 2 dan 3. Pada uji coba ini, monitoring-user melakukan pull dan push data untuk 1 client (pasien), 2 client dan 3 client sekaligus. Uji coba ini juga dapat menunjukkan pengaruh jumlah data yang

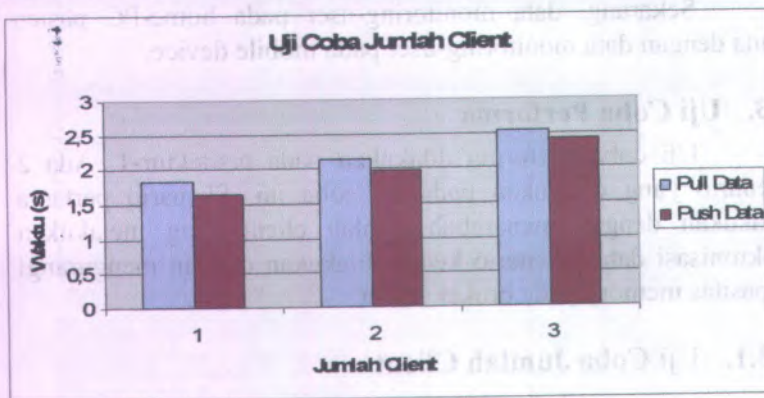
ditransfer terhadap waktu sinkronisasi. Jika jumlah client 1, maka data yang ditransfer hanya 1 tabel data, untuk 2 client maka data yang ditransfer adalah 2 tabel data, dan untuk 3 client maka data yang ditransfer adalah 3 tabel data.

Untuk tiap jumlah client, uji coba dilakukan sebanyak 5 kali, kemudian diambil waktu rata-rata untuk melakukan operasi pull dan push data. Hasil uji coba dengan penambahan jumlah client ditunjukkan seperti pada Tabel 5.1.

Tabel 5.1 Uji coba berdasarkan jumlah client

Jumlah Client	Waktu rata-rata (s)	
	Pull	Push
1	1,82	1,64
2	2,13	1,96
3	2,54	2,41

Dari Tabel 5.1 diatas, dapat dibuat grafik yang menunjukkan perbandingan waktu transfer data berdasarkan jumlah client, seperti yang ditunjukkan pada Gambar 5.28.



Gambar 5.28 Grafik perbandingan waktu transfer terhadap jumlah client

Dari hasil uji coba berdasarkan jumlah client ini, dapat dilihat bahwa dengan bertambahnya jumlah client (pasien) yang harus diakses oleh monitoring-user melalui mobile device, dan bertambahnya jumlah data yang harus ditransfer, maka waktu untuk melakukan transfer data juga semakin bertambah, sehingga performa sistem akan menurun.

5.3.2. Uji Coba Kapasitas Memory

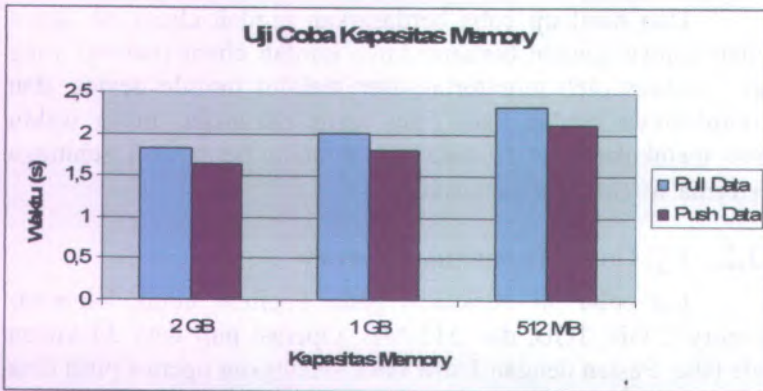
Uji coba ini dilakukan pada 1 client untuk kapasitas memory 2 GB, 1GB, dan 512 MB. Operasi pull data dilakukan pada tabel Pasien dengan 1 row data, sedangkan operasi push data dilakukan pada tabel MonitoringUser dengan 2 row data.

Sama seperti uji coba jumlah client, untuk uji coba kapasitas memory, masing-masing dilakukan sebanyak 5 kali, kemudian diambil waktu rata-rata untuk melakukan operasi pull dan push data. Hasil uji coba berdasarkan kapasitas memory pada broker server ditunjukkan seperti pada Tabel 5.2.

Tabel 5.2 Uji coba berdasarkan kapasitas memory

Kapasitas Memory	Waktu rata-rata (s)	
	Pull	Push
2 GB	1,82	1,64
1 GB	2,01	1,81
512 MB	2,33	2,11

Dari Tabel 5.2, dapat dibuat grafik yang menunjukkan perbandingan waktu transfer terhadap kapasitas memory broker server, seperti yang ditunjukkan pada Gambar 5.29.



Gambar 5.29 Grafik perbandingan waktu transfer terhadap kapasitas memory

Dari hasil uji coba berdasarkan kapasitas memory pada broker server ini, dapat dilihat bahwa performa sistem juga dipengaruhi oleh spesifikasi memory pada broker server. Jika kapasitas memory yang digunakan pada broker server semakin kecil, maka waktu untuk melakukan transfer data akan semakin bertambah, sehingga akan menurunkan performa sistem.

BAB VI PENUTUP

6.1. Kesimpulan

Dari hasil pengamatan selama perancangan, implementasi, dan proses uji coba perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

- Aplikasi monitoring kesehatan dengan remote data access (RDA) dapat diimplementasikan dengan menggunakan broker server antara mobile device monitoring-user dan home-PC pasien.
- Pada implementasi remote data access (RDA) dengan broker server, komponen SQL Server Mobile Server Agent (sqlcesa30.dll) diletakkan pada broker server, sedangkan komponen OleDb .Net Provider for SQL Server dan database SQL Server diletakkan pada home-PC pasien.
- Performa sistem dipengaruhi oleh jumlah client (pasien) yang harus diakses oleh monitoring-user. Jika jumlah client (pasien) semakin banyak, maka waktu transfer data akan semakin lambat, sehingga akan menurunkan performa sistem. Sebaliknya, jika jumlah client (pasien) semakin sedikit, maka waktu transfer data akan semakin cepat, sehingga akan meningkatkan performa sistem.
- Performa sistem juga dipengaruhi oleh kapasitas memory pada broker server. Jika kapasitas memory semakin kecil, maka waktu transfer data akan semakin lambat, sehingga akan menurunkan performa sistem. Sebaliknya, jika kapasitas memory besar, maka waktu transfer data akan semakin cepat sehingga akan meningkatkan performa sistem.

6.2. Saran

Untuk pengembangan perangkat lunak selanjutnya, disarankan untuk meningkatkan kualitas program dan database baik pada home-PC, server, mobile device, serta aplikasi web.

DAFTAR PUSTAKA

- [1] Lin C-H, Young S-T, Kuo T-S. 2007. "A remote data access architecture for home-monitoring health-care applications". **Proceeding Of International Conference on Medical Engineering and Physics**. Elsevier.
- [2] Tanenbaum, Andrew S. 2003. **Computer Networks, Fourth Edition**. Prentice Hall.
- [3] Microsoft Corporation. 2005. **SQL Server 2005 Mobile Edition Books Online**.
- [4] Tanpa nama. 2009. **SQL Server Compact**, <URL:http://en.wikipedia.org/wiki/SQL_Server_Compact>
- [5] Tanpa nama. 2009. **Hypertext Transfer Protocol**, <URL:http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol>.

DAFTAR PUSTAKA

- [1] The 2007 World 2-1-4 - I-8-2007-74 report has access available in home-learning healthcare applications. *Proceeding Of International Conference on Medical Engineering and Health Care*
- [2] *Proceeding Volume 2 2007 Computer Networks*. Fourth Edition, Prentice Hall
- [3] *Microsoft Corporation 2002 SQL Server 2002 database*. Fifth Edition, O'reilly
- [4] *SQL Server 2002*. O'reilly, 2002
- [5] *SQL Server 2002*. O'reilly, 2002

LAMPIRAN

Data Type Mapping

Microsoft SQL Server 2005 Mobile Edition (SQL Server Mobile) tidak mendukung keseluruhan tipe data SQL Server. SQL Server Mobile harus menerjemahkan tipe data yang tidak didukung menjadi tipe data yang dapat digunakan

Tabel 6.1 berikut menunjukkan Data Type Mapping yang dilakukan ketika proses sinkronisasi data antara SQL Server dan SQL Server Mobile.

Tabel 6.1 Data Type Mapping

SQL Server Data Type	SQL Server Mobile Data Type
bigint	bigint
binary(n)	binary(n)
bit	bit
character(n) <i>synonym: char(n)</i>	national character(n) <i>or</i> ntext
character varying (n) <i>synonym: varchar(n)</i>	national character varying <i>or</i> ntext
character varying (MAX) <i>synonym: varchar(MAX)</i>	ntext
Computed columns	<i>not supported</i>
datetime	datetime
decimal	<i>not supported, use numeric</i>
double precision	double precision
float	float
image	image
integer <i>synonym: int</i>	integer
money	money
national character(n) <i>synonym: nchar(n)</i>	national character(n)

Tabel 6.2 Data Type Mapping - lanjutan

SQL Server Data Type	SQL Server Mobile Data Type
national character varying(n) <i>synonym: nvarchar(n)</i>	national character varying(n)
national character varying(MAX) <i>synonym: nvarchar(MAX)</i>	ntext
ntext	ntext
numeric <i>synonym: decimal, dec</i>	numeric
real	real
smalldatetime	datetime
smallint (int 32)	smallint (int 32)
smallmoney	money
sql_variant	ntext
text	text
timestamp	<i>not supported</i>
tinyint	tinyint
uniqueidentifier	uniqueidentifier
varbinary(n)	varbinary(n)
varbinary(MAX)	image
varchar	<i>see character varying</i>
XML	ntext

BIODATA PENULIS



Sitti Salmiah, biasa dipanggil Mia atau Sitti, dilahirkan pada tanggal 3 Juni 1987 di kota Kendari, propinsi Sulawesi Tenggara. Penulis merupakan anak ketiga dari 4 (empat) bersaudara yang mengalami masa kecil, TK, SD, SMP, dan SMA di Kendari, Sulawesi Tenggara.

Selama berkuliah di Jurusan Teknik Informatika - Institut Teknologi Sepuluh Nopember Surabaya, penulis mengidentifikasi diri sebagai akademisi dan pengurus kajian Islam di jurusan (SITC / Studi Islam Teknik Computer-Informatika). Database dan pemrograman VB.Net adalah dua hal yang digemari oleh penulis pada bidang IT. Selain itu, penulis juga sangat menggemari seni dan bahasa (art & linguistic).

Penulis menyelesaikan tugas akhir dalam bidang NCC (Net Centric Computing). Topik tugas akhir penulis tentang remote data access (RDA), yaitu basis data terdistribusi antara mobile device dan komputer server. Komunikasi dengan penulis dapat dilakukan melalui email atau YM sitti_salmiah@yahoo.com.



BIBLIOTEKA

Ein weiterer wichtiger Punkt ist die
Einbindung der Mitarbeiter in den
Entscheidungsprozess. Dies ist
besonders wichtig, wenn es um
strategische Entscheidungen geht.
Die Mitarbeiter sollten nicht nur
informiert, sondern auch einbezogen
werden.

Ein weiterer wichtiger Punkt ist die
Einbindung der Mitarbeiter in den
Entscheidungsprozess. Dies ist
besonders wichtig, wenn es um
strategische Entscheidungen geht.
Die Mitarbeiter sollten nicht nur
informiert, sondern auch einbezogen
werden.

Ein weiterer wichtiger Punkt ist die
Einbindung der Mitarbeiter in den
Entscheidungsprozess. Dies ist
besonders wichtig, wenn es um
strategische Entscheidungen geht.
Die Mitarbeiter sollten nicht nur
informiert, sondern auch einbezogen
werden.