

# **DESAIN DAN IMPLEMENTASI PENGUMPULAN DATA MENGUNAKAN *MOBILE DEVICE* BERBASIS *PROXY***

TESIS

Karya tulis sebagai salah satu syarat  
Untuk memperoleh gelar Magister dari  
Institut Teknologi Bandung

oleh

**ARIS PRAWISUDATAMA**

**23215131**



**Institut Teknologi Bandung**

**2016**

**DESAIN DAN IMPLEMENTASI PENGUMPULAN DATA  
MENGUNAKAN *MOBILE DEVICE* BERBASIS *PROXY***

TESIS

oleh

**ARIS PRAWISUDATAMA  
23215131**

**Program Studi Magister Teknik Elektro  
Opsi Layanan Teknologi Informasi  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung**

Menyetujui

Pembimbing I

---

**(Dr. I Gusti Bagus Baskara Nugraha)**

NIP. 197601242010121001

## **Abstrak**

### **Desain dan Implementasi Pengumpulan Data Menggunakan *Mobile Device* Berbasis *Proxy***

oleh

**Aris Prawisudatama**  
**23215131**  
**(Magister Teknik Elektro)**

Abstrak bahasa Indonesia

Kata Kunci: *Pengumpulan, Data, Mobile, Device, Proxy*

# **Abstract**

## **Design and Implementation Proxy-based Mobile Data Collection**

by

**Aris Prawisudatama**

**23215131**

**(Magister Teknik Elektro)**

English abstract

Keywords: *Data, Collection, Mobile, Device, Proxy*

**Engkau tak dapat meraih ilmu kecuali dengan enam hal yaitu cerdas, selalu ingin tahu, tabah, punya bekal dalam menuntut ilmu, bimbingan dari guru dan dalam waktu yang lama. (Ali bin Abi Thalib)**

*Untuk Ibunda, tercinta . . .*

## **Pedoman Penggunaan Tesis**

Dengan ini saya menyatakan (dan menjamin) bahwa penulisan tesis ini dilakukan secara mandiri dan disusun tanpa menggunakan bantuan yang tidak dibenarkan, sebagaimana lazimnya pada penyusunan sebuah tesis. Semua kutipan tulisan dan pemikiran orang lain yang digunakan di dalam penyusunan tesis, baik dari sumber yang dipublikasikan ataupun tidak (termasuk dari buku, artikel jurnal, catatan kuliah, tugas mahasiswa lainnya, dan lainnya), telah direferensikan dengan baik dan benar menurut kaidah akademik yang baku dan berlaku. Tesis ini belum pernah diajukan pada pendidikan program pascasarjana di perguruan tinggi lain, dan tindak plagiarisme akan dikenakan sanksi seperti yang tercantum dalam Peraturan Akademik dan Kemahasiswaan ITB.

Bandung, Februari 2016

Aris Prawisudatama

## **Kata Pengantar**

Mea adhuc accusata indoctum eu. At populo meliore eam. Esse eros error ut mea, eos in prodesset liberavisse. Omnes recusabo intellegebat cum te, numquam delectus inciderint ex eam. Ei sed timeam euismodi electram, mucius petentium at per, duo graece epicuri facilisis ea. Ad mea putent voluptatibus, vis imperdiet urbanitas ne.

Bandung, Februari 2016

Aris Prawisudatama

## Daftar Isi

Lembar Pengesahan . . . . .	i
Abstrak . . . . .	ii
Abstract . . . . .	iii
Pedoman Penggunaan Tesis . . . . .	v
Kata Pengantar . . . . .	vi
Daftar Isi . . . . .	vii
Daftar Tabel . . . . .	ix
Daftar Gambar . . . . .	x
Bab I PENDAHULUAN . . . . .	1
I.1 Latar Belakang . . . . .	1
I.2 Rumusan Masalah . . . . .	5
I.3 Tujuan Penelitian . . . . .	5
I.4 Batasan Masalah . . . . .	5
Bab II STUDI LITERATUR . . . . .	7
II.1 <i>Generic Statistical Business Process Model (GSBPM)</i> . . . . .	7
II.2 <i>Generic Statistical Information Model (GSIM)</i> . . . . .	7
II.3 <i>Common Statistical Production Architecture (CSPA)</i> . . . . .	7
II.4 Sistem Terdistribusi . . . . .	9
II.5 <i>Grid Computing</i> . . . . .	10
II.6 <i>Mobile Grid Computing</i> . . . . .	11
II.7 <i>Connectionless Grid Computing</i> . . . . .	12
II.8 Pola Implementasi SOA untuk Sistem Data-Intensif Terdistribusi . . . . .	13
Bab III METODOLOGI . . . . .	23
III.1 Metodologi . . . . .	23
III.2 Implementasi Metodologi . . . . .	25
Bab IV ANALISIS DAN PERANCANGAN . . . . .	27
IV.1 ANALISIS . . . . .	27
IV.1.1 Penyusunan Model Aplikasi . . . . .	27
IV.1.1.1 Pendekatan <i>Stand Alone</i> . . . . .	27
IV.1.1.2 Pendekatan <i>Client Server</i> . . . . .	29
IV.2 PERANCANGAN . . . . .	31
IV.2.1 Perancangan Model Aplikasi . . . . .	31
IV.2.2 Replikasi . . . . .	31



IV.2.2.1 Replikasi <i>Rule</i> Validasi . . . . .	32
IV.2.2.2 Replikasi Data . . . . .	32
IV.2.3 Sinkronisasi . . . . .	33
IV.2.3.1 Sinkronisasi <i>Rule</i> Validasi . . . . .	33
IV.2.3.2 Sinkronisasi Data . . . . .	33
IV.2.4 Routing . . . . .	34
IV.2.5 Pengelolaan Data . . . . .	34
Daftar Pustaka . . . . .	37
Lampiran 1 . . . . .	40

## **Daftar Tabel**

Tabel II.1	Response Time Pengujian Skenario I . . . . .	19
Tabel II.2	Total Traffic Size Pengujian Skenario I . . . . .	20
Tabel II.3	Response Time Pengujian Skenario II . . . . .	20
Tabel II.4	Total Traffic Size Pengujian Skenario II . . . . .	20

## Daftar Gambar

Gambar I.1	<i>Statistical Business Process Phases</i> dalam GSBPM . . . . .	2
Gambar I.2	Ilustrasi CAPI . . . . .	3
Gambar I.3	Skema Usulan, Takdir . . . . .	4
Gambar II.1	<i>Literature Map</i> . . . . .	8
Gambar II.2	<i>Statistical Business Process Phases</i> dalam GSBPM . . . . .	8
Gambar II.3	Kategori Objek pada GSIM . . . . .	9
Gambar II.4	Gambaran Sederhana Relasi Objek pada GSIM . . . . .	9
Gambar II.5	Layer pada CSPA . . . . .	10
Gambar II.6	Skenario Eksekusi Logik Workflow dan Web Services . . . . .	16
Gambar II.7	Sinkronisasi Data dan Web Service . . . . .	17
Gambar II.8	Proses Replikasi Data dan Web Service . . . . .	18
Gambar II.9	Skenario Deployment Web Service dan Data Terpusat . . . . .	19
Gambar II.10	Skenario Deployment Web Service Terdistribusi, Data Terpusat . . . . .	19
Gambar II.11	Skenario Deployment Web Service dan Data Terdistribusi . . . . .	21
Gambar II.12	Proses Replikasi Data dengan IMDG . . . . .	21
Gambar II.13	Network Utilization Proses Patching/Updating Web Service . . . . .	22
Gambar II.14	Grafik CPU Utilization Proses Updating Web Service . . . . .	22
Gambar III.1	Tahapan <i>Design Science Research Methodology (DSRM)</i> . . . . .	23
Gambar III.2	Artefak Hasil Penerapan Metode DSRM (Purao, 2002) . . . . .	24
Gambar III.3	Implementasi DSRM . . . . .	25
Gambar IV.1	Pendekatan <i>Stand Alone</i> . . . . .	28
Gambar IV.2	<i>Flowchart</i> Pendekatan <i>Stand Alone</i> . . . . .	28
Gambar IV.3	<i>Client Server</i> dengan RPC [23] . . . . .	29
Gambar IV.4	Pendekatan <i>Client Server</i> . . . . .	30
Gambar IV.5	<i>Flowchart</i> Pendekatan <i>Client Server</i> . . . . .	30
Gambar IV.6	Arsitektur Solusi . . . . .	32
Gambar IV.7	Skema Replikasi <i>Rule</i> Validasi . . . . .	32
Gambar IV.8	Skema Replikasi Data . . . . .	33
Gambar IV.9	Skema Sinkronisasi <i>Rule</i> Validasi . . . . .	34
Gambar IV.10	Skema Sinkronisasi Data . . . . .	34

## Bab I

### PENDAHULUAN

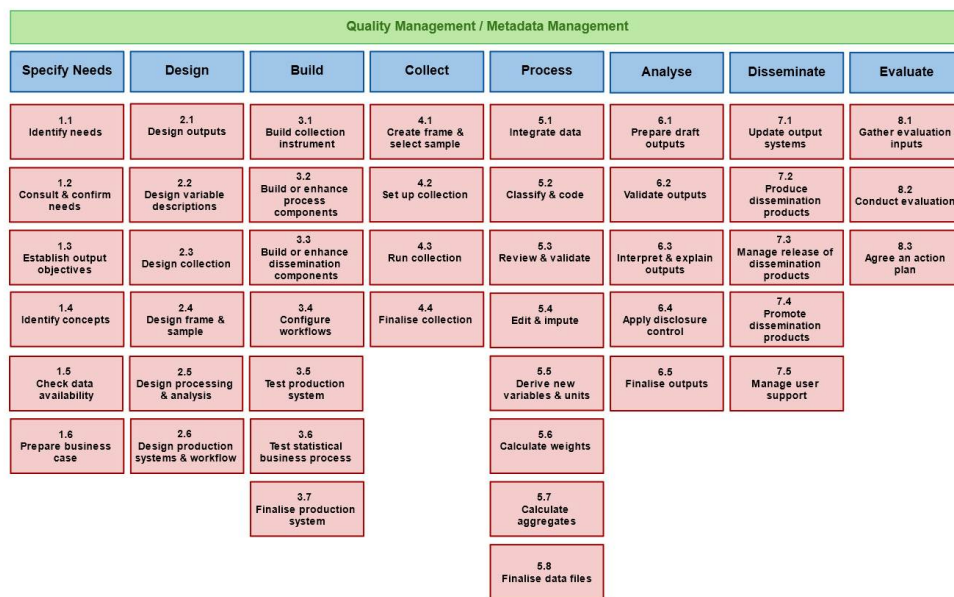
#### Latar Belakang

Badan Pusat Statistik (BPS) merupakan suatu lembaga pemerintah non-departemen yang bertanggung jawab dalam penyediaan statistik dasar. Dalam peranannya sebagai penyedia data, BPS melakukan pengumpulan data dengan 2 (dua) metode : primer dan sekunder. Pengumpulan data primer berarti BPS secara mandiri mengumpulkan data dengan menggunakan metode wawancara langsung dengan responden, baik responden individu, rumah tangga, maupun perusahaan. Sementara pengumpulan data sekunder berarti BPS memperoleh data dari pihak lain.

Dalam melakukan kegiatan perstatistikan, yang selanjutnya merujuk kepada pengumpulan data primer, BPS merujuk kepada *General Statistical Business Process Model* (GSBPM) [1], yang merupakan suatu standard arsitektur bisnis kegiatan perstatistikan yang dirumuskan oleh *United Nations Economic Commission for Europe* (UNECE). Dalam GSBPM, *Business Process Statistik* dibagi menjadi 7 (tujuh) phase : *Specify Needs, Design, Build, Collect, Process, Analyze, Disseminate, Evaluate*, dimana masing-masing phase dipecah menjadi beberapa sub-proses.

Pengumpulan dan pengolahan data dalam GSBPM tercakup dalam 3 (tiga) fase, yaitu : *Collect Phase, Process Phase, dan Analyze Phase*. *Collect phase* adalah fase dimana semua informasi (data dan metadata) dikumpulkan dengan menggunakan beberapa metode pengumpulan (termasuk ekstraksi dari register dan database statistik, administratif, maupun yang lain), dan memuatkannya ke dalam suatu *environment* untuk pemrosesan lebih lanjut. *Process phase* adalah fase dimana data dibersihkan dan dipersiapkan untuk tahap berikutnya, *analysis phase*. *Collect phase* dan *process phase* dapat dilakukan secara berulang dan paralel. Fase terakhir sebelum data siap untuk didesiminasikan adalah *analyze phase*. Pada tahap *analyze phase*, data ditransformasikan kedalam bentuk *statistical output* yang disesuaikan dengan kebutuhan (*fit for purpose*).

Kondisi saat ini, *process phase* dan *analyze phase* merupakan tahapan yang memiliki ketergantungan akan Teknologi Informasi dan Komunikasi (TIK) yang sangat



Gambar I.1: *Statistical Business Process Phases* dalam GSBPM

besar. *Process phase* merupakan tahapan dimana dilakukan input data hasil pendataan lapangan dari format kuesioner ke dalam format digital, termasuk didalamnya pengkodean, imputasi, validasi, dan penghitungan penimbang. Sementara *analyze phase* memerlukan keterlibatan software analisis yang membantu mentransformasikan data menjadi sebuah informasi. Adapun *collect phase*, meskipun saat ini masih menggunakan pengumpulan data dengan mengadopsi *paper questionnaire*, tetapi kedepannya akan dilakukan transformasi dengan menggunakan metode *Computer Assisted Personal Interviewing* (CAPI)<sup>1</sup>, meskipun feasibility-nya belum pernah diujicobakan<sup>2</sup>.

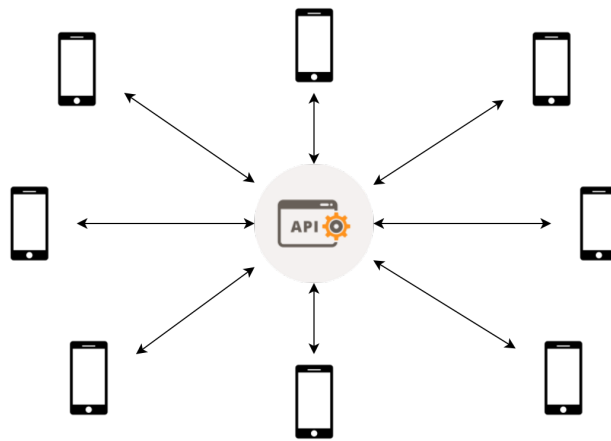
Penggunaan metode CAPI dalam pengumpulan data yang dilakukan BPS, sedikit banyak akan mengubah paradigma pengumpulan dan pengolahan data yang selama ini telah berjalan. Pengumpulan dan pengolahan data yang selama ini merupakan dua buah tahapan yang terpisah, dengan diterapkannya CAPI maka beberapa sub-proses dari *process phase*, seperti pengkodean dan validasi, dapat dilakukan secara terintegrasi dengan pengumpulan data. Metode CAPI sebenarnya bukanlah sebuah hal yang baru. Metode ini sudah ada sejak beberapa dekade terakhir [2]. Bahkan sebuah penelitian yang dilakukan oleh Gary Klein dkk menyatakan pengumpulan data dengan menggunakan metode CAPI berpotensi terjadi bias, terutama dalam

<sup>1</sup>Keterangan Dr. Said Mirza Pahlevi, M.Eng., Kepala Subdirektorat Pengembangan Basis Data, 24 Februari 2016

<sup>2</sup>Keterangan Dr. Muchammad Romzi, Kepala Subdirektorat Pengembangan Model Statistik, 4 Maret 2016

akurasi, *completeness*, dan *item omission* [3]. Akan tetapi, dengan semakin berkembangnya teknologi *mobile computing* yang dipadukan dengan penggunaan *Web service* [4], maka potensi bias dapat dikurangi dengan merancang *web service* yang digunakan untuk memvalidasi hasil pendataan.

Implementasi *Web service* pada pengumpulan data dengan menggunakan CAPI bukanlah tanpa kendala. Petugas pengumpulan data harus berpindah-pindah dari satu lokasi pendataan ke lokasi yang lain untuk mengunjungi responden. Dikarenakan keterbatasan infrastruktur seperti sinyal telekomunikasi dan daya tahan baterai *device*, seringkali tidak mudah bagi *device* untuk selalu terhubung dengan internet dan berkomunikasi dengan *Web service*. *Device* setiap saat dapat dengan mudah berubah dari *connected node* menjadi *disconnected node* dan sebaliknya.



Gambar I.2: Ilustrasi CAPI

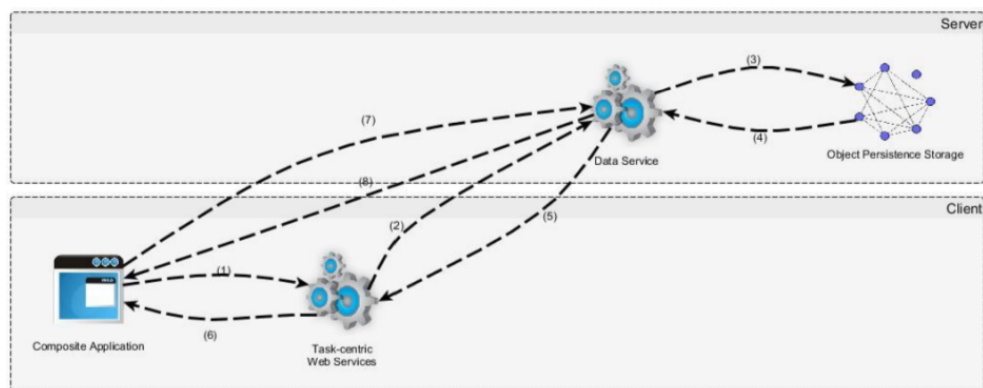
Gutwin dkk mengelompokkan tidak terhubungnya suatu *mobile node* dalam sebuah jaringan menjadi 3 (tiga) jenis jenis [5]:

- *Delay-based Interruption*, merupakan sebuah gap singkat (*short-term gap*) dalam pengiriman pesan. *Delay* dapat disebabkan oleh berbagai faktor, tetapi porsi terbesar penyebab *delay* adalah *transmission delay*, *contention delay*, dan *queuing delay* [6].
- *Network Outage*, merupakan kondisi dimana sebuah *mobile node* terputus dari jaringannya. Kondisi *network outage* dapat disebabkan oleh berbagai faktor, seperti: bencana (kebakaran di *Baltimore Howard Street Tunnel* or terputusnya *Mediterranean Cable*), kesalahan konfigurasi (*Pakistani Youtube routing*)

- *Explicit Departures*, merupakan kondisi *mobile node* keluar dari jaringan atau keluar dari aplikasi secara eksplisit

Brian DeRenzi dkk telah melakukan penelitian tentang cara pengumpulan data berbasis *mobile phone* pada lingkungan yang *highly disconnected* [7] dengan menggunakan *CAM Framework*. *CAM framework* [8] terbukti dapat digunakan dalam pengumpulan data dalam lingkungan yang *disconnected*, dan setelah device kembali ke *connected environment*, data yang terkumpul akan terupload ke server. Akan tetapi *CAM framework* memiliki kelemahan, antara lain : 1) *CAM* berbasis *fix-length text-based input*, yang membuatnya tidak cocok digunakan untuk pengumpulan data yang berbasis *data-intensive*; 2) Tidak terdapat *conflict resolution*, sehingga masih memungkinkan dua device atau lebih mengeksekusi data yang sama.

Sementara itu, untuk meningkatkan kinerja sistem yang bersifat data-intensif, Takdir dkk mengusulkan penggunaan pola terdistribusi berbasis SOA [9]. Pada implementasi pola terdistribusi berbasis SOA, digunakan metode *proxy* baik pada *workflow* (*Web service*) maupun *data-service*. Mekanisme yang digunakan dalam perancangan service pola terdistribusi mencakup 3 (tiga) hal : sinkronisasi, replikasi, dan *routing*. *Composite application* yang dijalankan pada sisi *client* akan melakukan replikasi data maupun *Web service*, kemudian data dan *Web service* tersebut digunakan secara lokal. Sementara itu, untuk menjamin konsistensi data, Takdir dkk mengakomodir mekanisme *sinkronisasi*.



Gambar I.3: Skema Usulan, Takdir

Berdasarkan hasil pengujian, pola distribusi berbasis SOA usulan Takdir dkk terbukti mampu memberikan hasil yang lebih baik, dengan penggunaan *resource* CPU dan memory yang lebih rendah. Akan tetapi, lingkup perancangan dan pengujian sistem hanya terbatas pada perangkat komputer (*desktop* maupun *laptop*). Pada pola terdistribusi, usulan Takdir dkk, *client* perlu untuk melakukan replikasi data

maupun *workflow* yang berupa *Web service* agar dapat berjalan. Sementara pada pendataan data berbasis CAPI, perangkat yang digunakan adalah *mobile device*, yang mempunyai arsitektur maupun *resource* yang berbeda jika dibandingkan dengan komputer. Proses replikasi data *workflow* bisa menjadi sesuatu yang *challenging* atau bahkan sulit dilakukan semenjak *mobile device* bukanlah sesuatu yang umum digunakan sebagai *service provider*. Oleh karena itu, penelitian ini akan berfokus kepada perancangan desain dan implementasi sistem terdistribusi yang dapat diimplementasikan pada perangkat *mobile*.

### **Rumusan Masalah**

Berdasarkan uraian latar belakang permasalahan diatas, maka dapat dirumuskan suatu permasalahan penelitian yaitu bagaimana merancang desain dan implementasi *mobile proxy* untuk pengumpulan data lapangan berbasis *mobile*.

### **Tujuan Penelitian**

Tujuan utama dari penelitian ini adalah untuk merancang desain dan implementasi *mobile proxy* untuk pengumpulan data lapangan berbasis *mobile*. Adapun tujuan khusus penelitian ini adalah :

- Merancang dan mengimplementasikan *mobile proxy* yang dapat digunakan pada *connected* maupun *disconnected environment*,
- Menganalisis kompatibilitas *mobile device* yang memenuhi spesifikasi rancangan *mobile proxy*,
- Melakukan ujicoba atas desain dan implementasi *mobile proxy* yang dirancang dengan studi kasus pengumpulan data lapangan,
- Menganalisis dan mengevaluasi hasil ujicoba sistem.

### **Batasan Masalah**

Batasan masalah dalam penelitian ini adalah :

- Penelitian ini hanya berfokus pada desain dan implementasi sistem pada *mobile device*,



- *Mobile device* yang digunakan dalam ujicoba terbatas hanya *mobile device* dengan sistem operasi *Android*,
- Perancangan GUI hanya digunakan sebagai bahan uji coba, dan bukan termasuk ke dalam poin inti penelitian.

## **Bab II**

### **STUDI LITERATUR**

Untuk mendukung penelitian ini, telah dilakukan studi literatur terkait topik penelitian. Penelitian terkait tersebut direpresentasikan dalam peta literatur Gambar II.1

#### ***Generic Statistical Business Process Model (GSBPM)***

GSBPM [1] mendefinisikan sekumpulan bisnis proses yang dibutuhkan untuk untuk kegiatan perstatistikan. GSBPM ditujukan untuk membantu institusi perstatistikan di dunia melakukan modernisasi pada kegiatan perstatistikan yang dijalankan. BPS juga telah mencanangkan untuk mengikuti bisnis proses pada GSBPM melalui program *Statistical Capacity Building-Change and Reform for the Development of Statistics (STATCAP-CERDAS)*

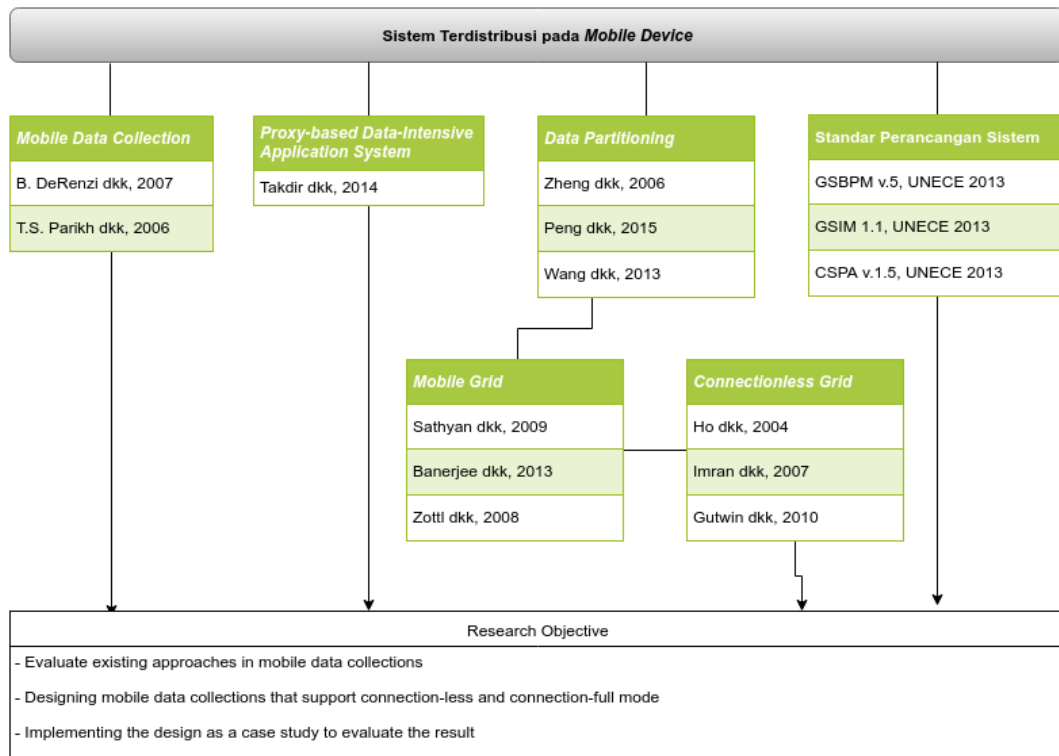
#### ***Generic Statistical Information Model (GSIM)***

GSIM merupakan *framework model* informasi statistik internasional pertama yang diciptakan. GSBPM merupakan *reference framework* untuk information objects pada kegiatan perstatistikan yang mendefinisikan penjelasan, pengelolaan dan penggunaan data dan metadata [10]. GSIM dibentuk untuk memenuhi kebutuhan informasi pada proses-proses yang ada pada GSBPM. Pada level atas, GSIM mengelompokkan objek informasi menjadi 4 kategori, yakni *Business*, *Exchange*, *Concepts*, dan *Structures*.

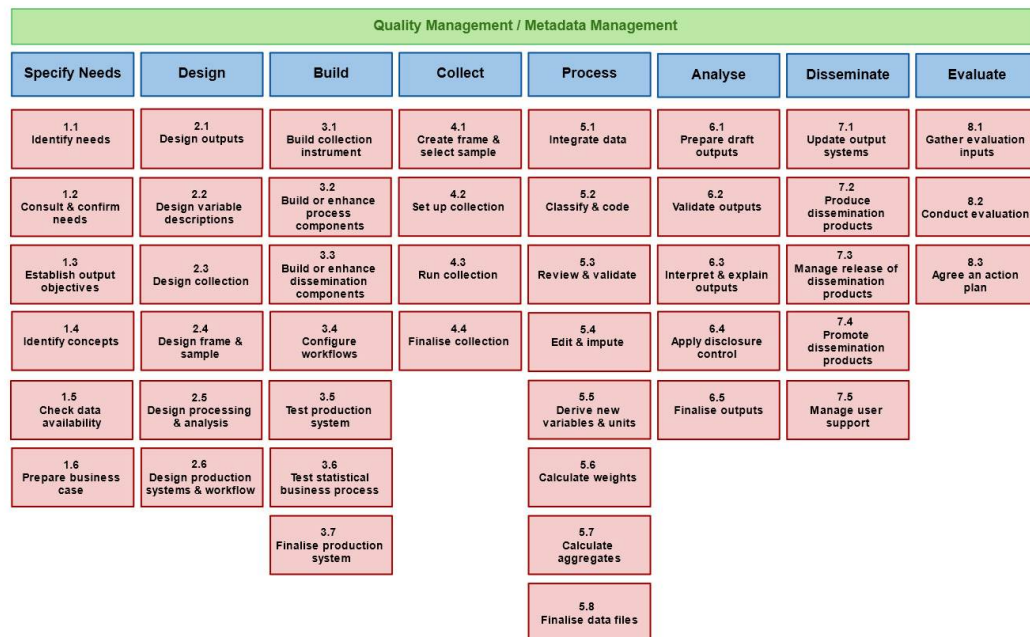
Selain itu, GSIM juga telah menetapkan spesifikasi teknis struktur data untuk tiap-tiap objek. Terdapat sebanyak 114 entitas yang telah didefinisikan oleh GSIM.

#### ***Common Statistical Production Architecture (CSPA)***

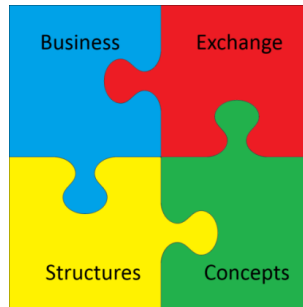
CSPA merujuk pada GSBPM sebagai arsitektur bisnis, GSIM sebagai arsitektur informasi, serta menggunakan pendekatan SOA (*Service Oriented Architecture*) untuk arsitektur aplikasi dan arsitektur teknologi. Saat ini CSPA merupakan *project* yang sedang aktif di UNECE. Project CSPA merupakan kolaborasi antara beberapa



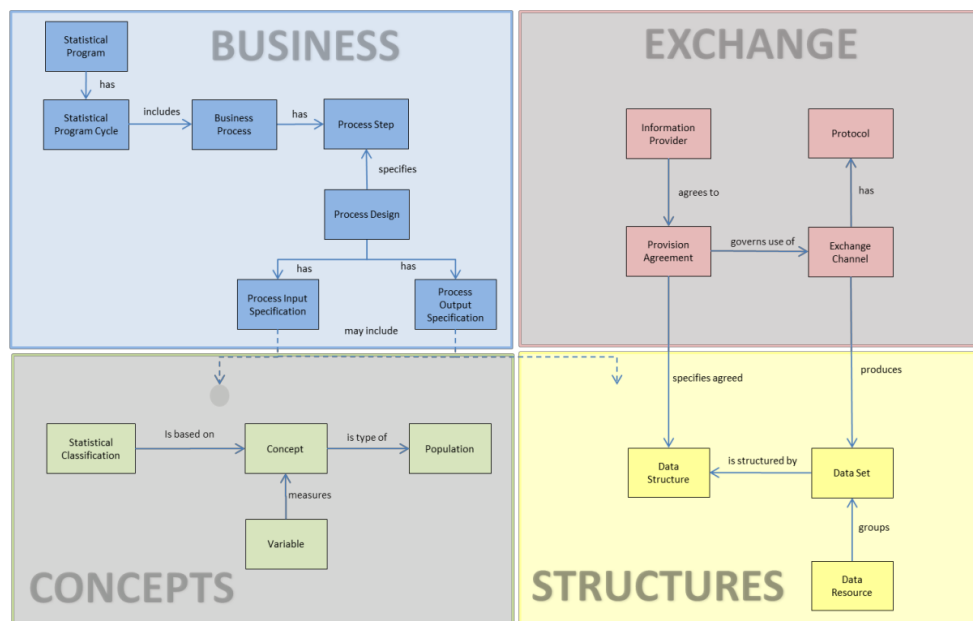
Gambar II.1: *Literature Map*



Gambar II.2: *Statistical Business Process Phases* dalam GSBPM



Gambar II.3: Kategori Objek pada GSIM

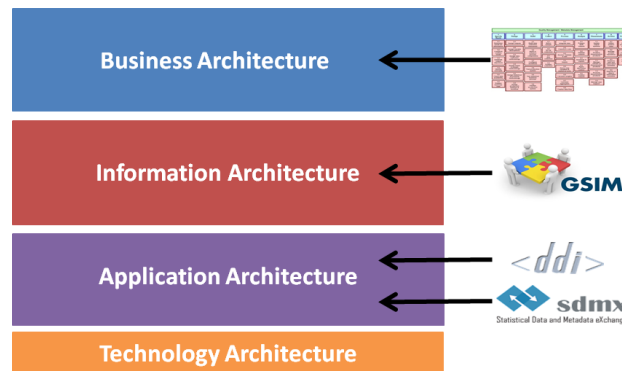


Gambar II.4: Gambaran Sederhana Relasi Objek pada GSIM

wakil di bidang TIK organisasi penyelenggara kegiatan perstatistikan dari berbagai negara guna merancang SOA untuk sistem perstatistikan [11]. Tahap awal pada CS-PA adalah mengidentifikasi kandidat layanan statistik (*statistical services*). Hingga saat ini, CSPA telah menghasilkan beberapa kandidat *statistical services* yang nantinya akan distandarisasi.

### Sistem Terdistribusi

Salah satu mekanisme untuk menangani masalah pada pendekatan centralized adalah dengan menggunakan pendekatan sebaliknya, yakni dengan menggunakan sistem terdistribusi. Sistem terdistribusi dapat diartikan sebagai sebuah jaringan komputer untuk menyelesaikan masalah komputasi (*computational problem*) dengan cara membagi *task* ke dalam beberapa komputer (Godfrey, 2006) [12]. Setiap kom-



Gambar II.5: Layer pada CSPA

puter dalam jaringan berkomunikasi dengan menggunakan metode *message passing* [13]. Keuntungan dari pemanfaatan sistem terdistribusi antara lain [9]:

- Mempercepat *response time* dan mengurangi *traffic jaringan*,
- Meningkatkan kecepatan akses dan *query* terhadap data,
- Meningkatkan system *availability* dan menghindari kegagalan sistem dengan menerapkan *single point of failure*.

Selain keuntungan, sistem terdistribusi juga memiliki beberapa tantangan, antara lain :

- Pembagian data (*data partitioning*) dan proses komputasi (*task*),
- Keselarasan data (*data synchronization*).

### ***Grid Computing***

*Grid computing* dapat dianggap sebagai sebuah sistem terdistribusi yang bersifat *non-interactive* antar *node*. Masing-masing *node* pada *grid computers* menjalankan proses komputasi (*task*) secara terpisah. Prinsip umum yang menjadi karakteristik *grid* [14]:

- Tersusun dari beberapa domain administrasi dan otonomi (*geographically dispersed*),
- *Heterogen* dan tersusun dari beragam teknologi yang bervariasi,

- *Scalable*, dapat berkembang dari hanya beberapa *integrated resources* menjadi ribuan bahkan jutaan, *Dynamic* dan *adaptable*, mampu mendeteksi *resource failure* dan memanfaatkan *available resource*.

Untuk mengakomodir *heterogen resources*, maka *grid* harus mengikuti prinsip berikut :

- Tidak menginterferensi domain administrasi dan otonomi yang sudah ada,
- Tidak perlu mengganti sistem operasi, protokol jaringan, atau *services*,
- Memungkinkan *remote sites* untuk bergabung atau meninggalkan jaringan kapanpun,
- Menyediakan sistem yang handal dan infrastruktur yang *fault tolerance* dengan *no single point of failure*,
- Menyediakan *support* untuk komponen yang heterogen,
- Menggunakan teknologi yang telah ada dan terstandarisasi,
- Menyediakan sinkronisasi.

### ***Mobile Grid Computing***

*Mobile grid computing* dapat didefinisikan sebagai sebuah sistem terdistribusi berbasis *grid computing* yang mengakomodir penggunaan *mobile device*. *Mobile device*, seperti *PDA* dan *smartphone*, biasanya diindikasikan dengan kekuatan pemrosesan yang lemah, kapasitas baterai yang terbatas, dan media penyimpanan yang terbatas. Menjalankan proses yang intensif pada *mobile device* memerlukan kemampuan pemrosesan yang lebih, media penyimpanan, penyelesaian proses dengan waktu seminimal mungkin, sehingga penggunaan *mobile grid* dapat menjadi salah satu solusi [15]. *Mobile grid computing* memungkinkan mobility dalam sebuah *grid framework*, sehingga memungkinkan *mobile device* dapat beroperasi secara halus (*seamless*), transparan, aman dan efisien.

Zotll dkk telah meneliti sebuah metode untuk menyimpan informasi dalam tiga tingkatan (*Three-tier*), (i) *base tier*, (ii) *information tier*, dan (iii) *organization tier* yang dinamakan IRMA. Metode IRMA diterapkan pada *mobile grid environment* dengan menggunakan *Erasure Code* [16]. *Erasure Code* (EC) [17] menggunakan skema

*replication* dan *parity coding*. Skema replikasi bekerja dengan menduplikasi data pada beberapa *node*, sementara skema *parity coding* tidak menduplikasi data, tetapi menggunakan *checksum* untuk me-recover dari *failure*. Pada *mobile computing*, replikasi adalah cara yang umum digunakan untuk menjamin ketersediaan (*persistency*). Beberapa strategi replikasi yang telah dikembangkan antara lain *uniform*, *proportional*, dan *square-root* [18]. Replikasi dapat menjamin *persistence* pada saat koneksi putus secara tiba-tiba (*unpredicted disconnected*). Berdasarkan hasil penelitian, metode IRMA dapat bekerja secara konsisten, persisten, dan efisien dalam jaringan *mobile grid* yang dinamis.

### ***Connectionless Grid Computing***

Pada sistem terdistribusi berbasis *mobile grid*, sebagian besar *node* berupa *mobile device*, seperti *PDA* dan *smartphone*. *Mobile device* mempunyai karakteristik yang *mobile*, dapat dengan mudah berpindah dari suatu lokasi ke lokasi yang lain. Lokasi dimana *mobile device* ini berada terkadang merupakan lokasi yang tidak terjangkau oleh sinyal telekomunikasi atau *wifi* area. Tidak terhubungnya suatu *mobile node* dalam sebuah jaringan dapat dikelompokkan dalam beberapa jenis [5]:

- *Delay-based Interruption*, merupakan sebuah gap singkat (*short-term gap*) dalam pengiriman pesan,
- *Network Outage*, merupakan kondisi dimana sebuah *mobile node* terputus dari jaringannya karena hilangnya jaringan komunikasi,
- *Explicit Departures*, merupakan kondisi *mobile node* keluar dari jaringan atau keluar dari aplikasi secara eksplisit

Ho dkk meneliti tentang penggunaan pendekatan *connection-less* untuk mengakomodir permasalahan komunikasi [19]. Pendekatan *connection-less* ini bekerja secara *ad-hoc* pada jaringan tanpa infrastruktur (*infrastructure-less network*), sehingga memungkinkan *mobile nodes* juga berperan sebagai *router* yang mengakomodir komunikasi dengan *mobile nodes* yang lain. Pendekatan *connection-less* juga diteliti oleh Imran dkk [20] dengan menggunakan *proxy-based checkpoint* dan *message logging* yang dinamakan *Mobile Host Proxies [MHP]*. Metode MHP menjadikan setiap *mobile device* sebagai *Mobile Host [MH]* dan *Mobile Service Station [MSS]* dan menyimpan *checkpoint* secara *asynchronous*.

## **Pola Implementasi SOA untuk Sistem Data-Intensif Terdistribusi**

Service Oriented Architecture (SOA) dapat dipandang dari dua perspektif yang berbeda : sebagai arsitektur bisnis (business architecture) dan arsitektur teknis (technical architecture). Dari sudut pandang bisnis, SOA menawarkan solusi dalam mengelola bisnis dengan menekankan pada layanan (services) sebagai unit/satuan terkecil yang perlu dirancang dan diorganisir. Sementara dari sudut pandang teknis, SOA memperkenalkan prinsip-prinsip dalam merancang dan mengembangkan perangkat lunak yang berorientasi service.

Service dalam konteks bisnis merupakan kumpulan aktifitas yang dijalankan oleh tiap-tiap unit kerja dalam suatu perusahaan. Dengan demikian, sebuah unit kerja dapat berperan sebagai penyedia layanan (service provider) bagi unit kerja lain yang membutuhkan (service consumer), dan sebaliknya. Hubungan antar penyedia dan pengguna layanan diatur dalam sebuah perjanjian (contract), yang berisi penjelasan dari layanan serta syarat dan ketentuan yang harus dipenuhi kedua belah pihak.

Service dalam konteks teknis, mengadopsi konsep service yang diterapkan dalam konteks bisnis. Akan tetapi, service dalam konteks teknis diinterpretasikan dalam fungsi-fungsi yang didukung oleh software, atau dikenal dengan logika bisnis (business logic). Logic business dikemas dalam bentuk komponen software dengan mekanisme tertentu sehingga dapat didistribusikan.

Thomas Erl memperkenalkan prinsip-prinsip dalam merancang software service, yang bertujuan untuk memudahkan integrasi sistem informasi, yaitu:

- **Standardize Service Contract**

Services yang terdapat pada sistem inventarisasi yang sama memiliki desain kontrak yang serupa.

- **Service Loose Coupling**

Service tidak memiliki ketergantungan terhadap lingkungan atau platform tertentu.

- **Service Abstraction**

Service provider hanya menyediakan informasi dasar, sementara teknis implementasi service tidak disertakan dalam kontrak.



- Service Reusability

Service tidak dirancang untuk keperluan tunggal, tetapi bersifat multi-purpose. Sebuah service merupakan enterprise resource yang dapat digunakan kembali.

- Service Statelessness

Meminimalisir penggunaan resources dengan tidak menyertakan pengelolaan state pada service.

- Service Discoverability

Service dilengkapi dengan metadata, sehingga dapat ditelusuri secara efektif.

- Service Composability

Sebuah service dapat tersusun dari penggabungan beberapa service.

SOA hanyalah merupakan sebuah arsitektur, sehingga diperlukan teknologi pendukung agar dapat direalisasikan. Web service merupakan salah satu teknologi yang dapat digunakan untuk merealisasikan konsep SOA menjadi sebuah sistem software. Web service menggunakan teknologi XML yang didesain sedemikian rupa sehingga mendukung komunikasi antar komponen software yang bersifat platform independent (W3C 2004).

XML sendiri merupakan dokumen teks yang mengikuti format dan spesifikasi tertentu sehingga mudah untuk diproses oleh mesin/komputer. Tujuan yang ingin dicapai dari XML (W3C 2006) antara lain :

- XML harus dapat digunakan pada jaringan Internet,
- XML harus dapat diaplikasikan secara luas,
- Adanya kemudahan pembuatan program (software) untuk memproses dokumen XML,
- Meminimalkan jumlah fitur yang tidak harus ada,
- Dokumen XML harus dapat terbaca dengan layak oleh manusia,
- Dokumen XML harus mudah dibuat.

Teknologi Web service mengadopsi XML dalam format Web Service Definition Language (WSDL). WSDL menyediakan informasi kepada service consumer berupa struktur data yang diperlukan, jenis operasi yang didukung, data binding, protokol komunikasi data, maupun endpoint. SOAP merupakan protokol komunikasi yang ditujukan untuk menangani pertukaran data antara service provider dan service consumer. Selain mendefinisikan struktur data, SOAP juga menyertakan nilai dari setiap atribut.

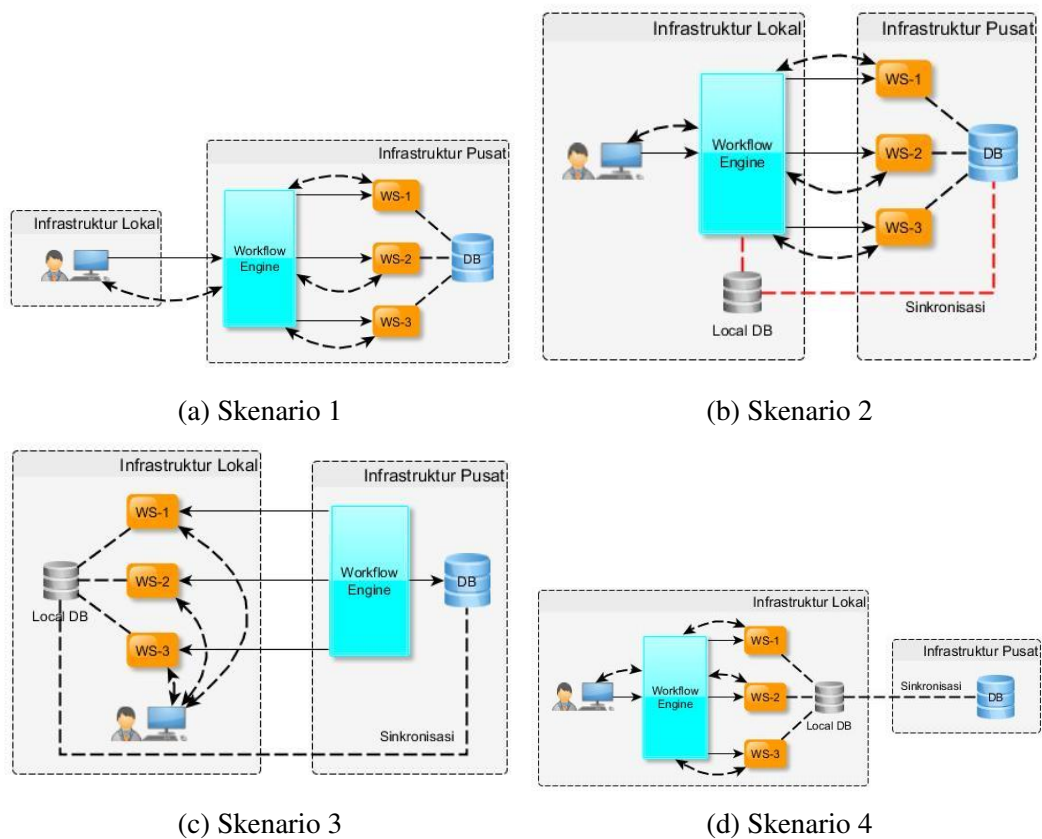
Pada penelitiannya, Takdir dkk telah mencoba menemukan pola implementasi SOA pada aplikasi yang bersifat data intensif. Aplikasi data intensif memiliki ciri : (i) tersusun dari sederetan operasi yang mengolah data dalam jumlah besar, dan (ii) adanya aliran data dengan volume yang besar antar operasi-operasi tersebut (Habich, Richly, Preissler, et al. 2008). Pola implementasi SOA rancangan Takdir dkk bertujuan untuk menjembatani gap antara desain business services dan software service. Proses perancangan pola implementasi menitikberatkan pada layanan statistik (statistical services) yang bersifat data intensif.

Perancangan pola implementasi dibagi menjadi dua, implementasi logik workflow dan implementasi web services. Mekanisme eksekusi logik workflow dan web services, sebagaimana dalam Gambar II.6, dalam pengujian dibagi dalam 4 (empat) skenario :

1. Pengeksekusian workflow dan web services dilakukan pada infrastruktur pusat,
2. Pengeksekusian workflow dilakukan pada infrastruktur lokal, sementara pengeksekusian web service dilakukan pada infrastruktur pusat,
3. Pengeksekusian workflow dilakukan pada infrastruktur pusat, dan web service pada infrastruktur lokal,
4. Pengeksekusian workflow dan web services dilakukan pada infrastruktur lokal.

Instilah infrastruktur pusat disini merujuk pada infrastruktur TIK yang ada di BPS Pusat, sementara istilah infrastruktur lokal merujuk pada infrastruktur TIK pada BPS Daerah.

Pada skenario yang melibatkan infrastruktur pusat dan infrastruktur lokal, terdapat mekanisme sinkronisasi pada database. Selain itu, pada skenario yang menempatkan web services pada repository lokal juga diperlukan sinkronisasi web services,



Gambar II.6: Skenario Eksekusi Logik Workflow dan Web Services

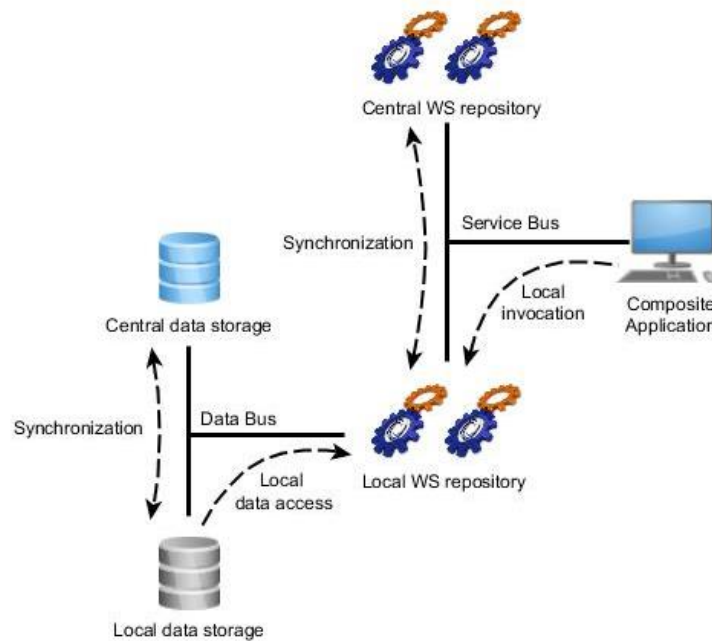
antara central repository dan local repository. Proses sinkronisasi, baik database maupun web services, antara infrastruktur pusat dan infrastruktur lokal dilakukan untuk menjaga konsistensi antara keduanya.

Selain mekanisme sinkronisasi, pada skenario yang melibatkan infrastruktur pusat dan daerah juga terdapat mekanisme replikasi. Replikasi, baik database maupun web services dilakukan dengan menggunakan pendekatan cache. Tujuan replikasi adalah untuk mendekatkan data maupun web service kepada service consumer. Replikasi data sangat tergantung pada replikasi web service, karena hanya data yang dibutuhkan oleh web service yang akan direplika.

Terdapat beberapa skenario deployment yang mengakomodir empat pendekatan eksekusi logik dan web service :

#### 1. Web service dan data terpusat

Pada skenario ini, web services (task-centric web service dan data service) dan database dipusatkan pada server seperti pada Gambar II.9 Skenario ini



Gambar II.7: Sinkronisasi Data dan Web Service

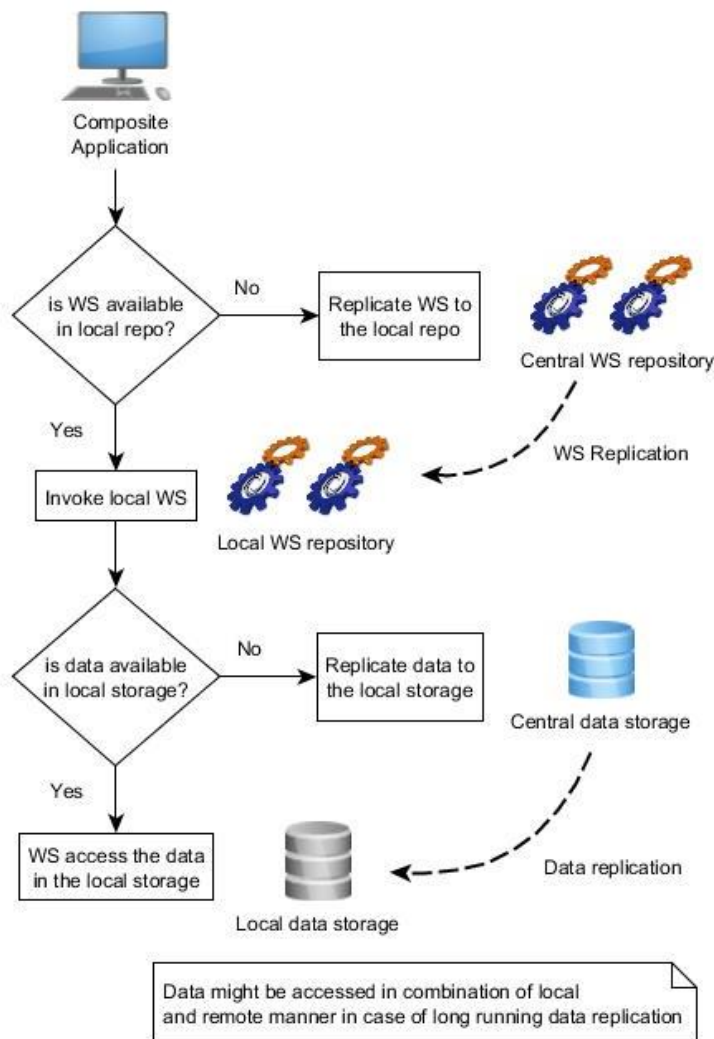
cocok untuk web service yang membutuhkan akses terhadap data yang bersifat global dan memiliki parameter dengan ukuran data yang kecil.

## 2. Web service terdistribusi, data terpusat

Logik task-centric web services direplika ke client untuk dieksekusi. Logik direpresentasikan dalam bentuk file \*.class untuk kemudian di-load ke run-time dengan Java Reflection API. Setelah file yang berisi logik direplika, maka client akan menjalankan servlet engine pada Apache CXF untuk melakukan hosting web service. Dalam hal ini, client juga berperan sebagai server (service provider) bagi dirinya sendiri, maupun client lain yang terkoneksi dengannya.

## 3. Web service dan data terdistribusi

Skenario ini ditujukan untuk memungkinkan sistem bekerja secara offline (disconnected) dimana semua komponen software yang dibutuhkan oleh client direplikasi, baik data, maupun web services. Komunikasi antarkomponen berjalan pada infrastruktur lokal, kecuali proses sinkronisasi data dengan server yang dilakukan secara asynchronous. DataObjek yang direplika ke object persistence datastore pada client adalah DataObject yang hanya dibutuhkan oleh task-centric web services yang telah direplika pada client. Setelah replikasi berlangsung, maka infrastruktur lokal dapat beroperasi meskipun kehilangan koneksi dengan infrastruktur pusat. Proses sinkronisasi data dapat

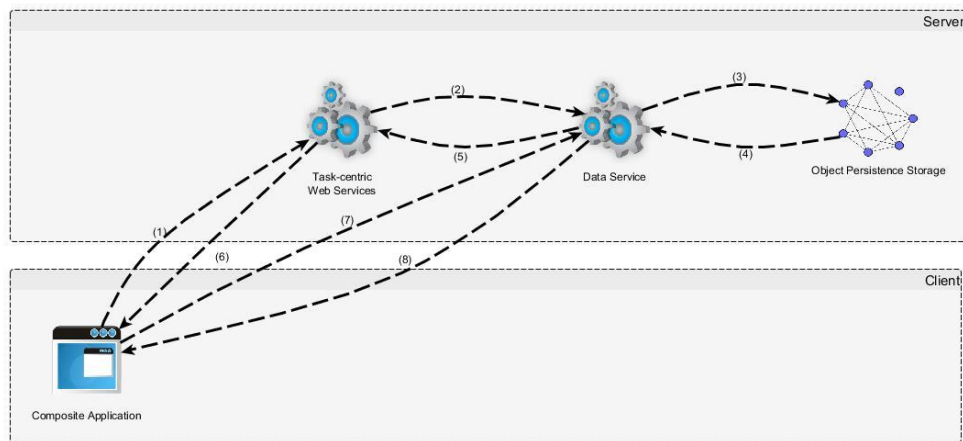


Gambar II.8: Proses Replikasi Data dan Web Service

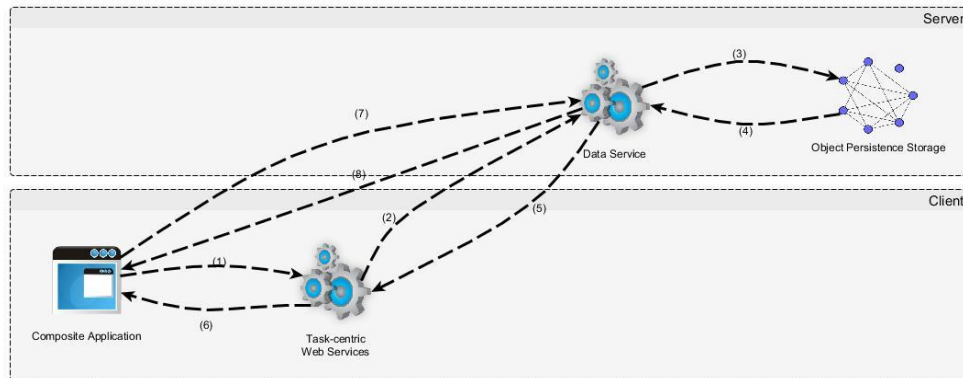
dilakukan secara asynchronous ketika koneksi ke infrastruktur pusat tersedia kembali. Gambar II.11 merupakan gambaran skenario tersebut.

Pada saat replikasi data dari infrastruktur pusat ke infrastruktur lokal sedang berlangsung, teknologi In-Memory Data Grids (IMDG) digunakan agar operasi di client dapat tetap berjalan sementara proses replikasi berjalan secara asynchronous. Hal ini juga berlaku untuk proses sinkronisasi data dari infrastruktur lokal ke infrastruktur pusat, maupun sebaliknya.

Pada tahap pengujian, data yang digunakan adalah data Kor SUSENAS 2009, yang terdiri dari 129 fields. Adapun variabel yang diamati dalam mengevaluasi kinerja sistem adalah : 1) Response Time, 2) Network utilization, dan 3) CPU utilization. Pengujian yang dilakukan pada environment virtual machine, dengan konfigurasi



Gambar II.9: Skenario Deployment Web Service dan Data Terpusat



Gambar II.10: Skenario Deployment Web Service Terdistribusi, Data Terpusat

yang telah disesuaikan dengan kondisi yang sebenarnya, diperoleh hasil sebagai berikut :

1. Skenario web service dan data terpusat
2. Skenario web service terdistribusi, data terpusat
3. Skenario web service dan data terdistribusi

Tabel II.1: Response Time Pengujian Skenario I

Jumlah Records	Fully Centralized	Distributed
5	00:00:04.627	00:00:02.484
10	00:00:08.320	00:00:03.205
100	00:02:00.142	00:00:12.648
200	00:04:01.497	00:00:22.653
500	00:10:02.997	00:01:50.549

Tabel II.2: Total Traffic Size Pengujian Skenario I

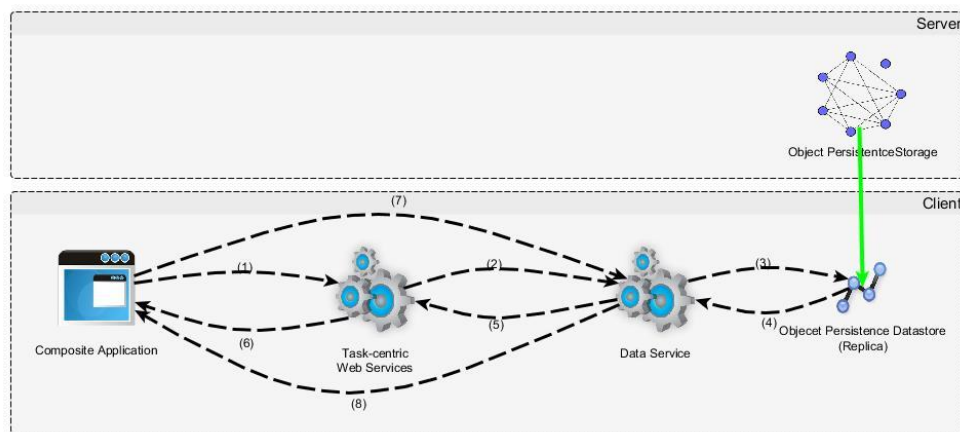
<b>Jumlah Records</b>	<b>Fully Centralized</b>	<b>Distributed</b>
5	748	7.64
10	1382.4	7.53
100	20449.28	7.76
200	40908.8	7.64
500	102256.64	7.64

Tabel II.3: Response Time Pengujian Skenario II

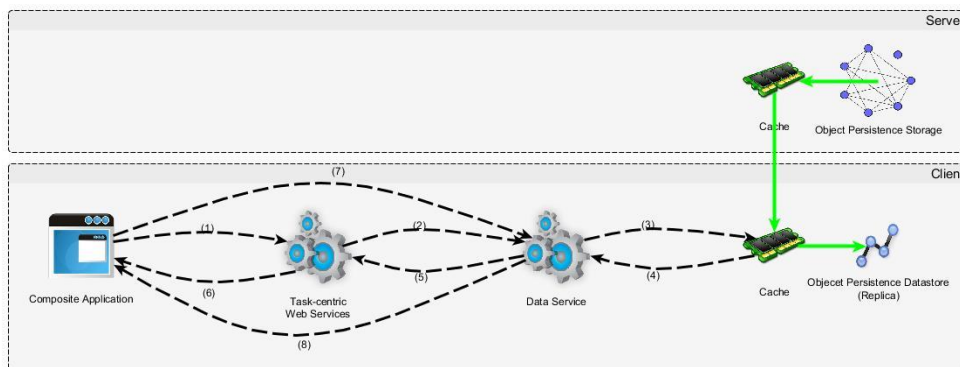
<b>Jumlah Records</b>	<b>Fully Centralized</b>	<b>Distributed</b>
5	00:00:14.470	00:00:09.530
10	00:00:14.872	00:00:10.745
100	00:01:15.759	00:00:23.183
200	00:02:22.225	00:00:37.554
500	00:05:39.980	00:02:43.895

Tabel II.4: Total Traffic Size Pengujian Skenario II

<b>Jumlah Records</b>	<b>Fully Centralized</b>	<b>Distributed</b>
5	365.62	449.88
10	692.19	934.07
100	10240	12400.64
200	20490.24	27084.8
500	51210.24	38840.32



Gambar II.11: Skenario Deployment Web Service dan Data Terdistribusi



Gambar II.12: Proses Replikasi Data dengan IMDG

(a) Response Time

Waktu yang dibutuhkan untuk melakukan updating web service (error correction service) adalah 00:00:04,647.

(b) Network Utilization

Total traffic yang dihasilkan sebesar 17.21 kilobytes. Ukuran tersebut merupakan file logik web service yang di-download oleh client.

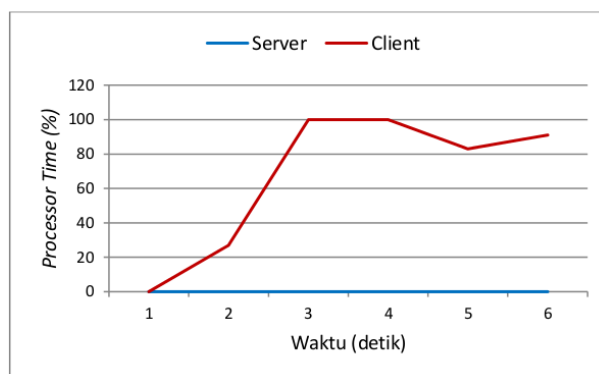
(c) CPU Utilization

Processor time yang dihasilkan mesin client pada Gambar II.14 merupakan proses untuk melakukan hosting web service yang di-download dari server. Proses tersebut menggunakan sekitar 50 persen processor time dari masing-masing CPU core.



Statistics Item	Current Value				
[-] Traffic	Bytes	Packets	Utilization	bps	pps
.....Total	17.21 KB	33	0.000%	0.000 bps	0
.....Broadcast	110.00 B	2	0.000%	0.000 bps	0
.....Multicast	1.75 KB	2	0.000%	0.000 bps	0
.....Average Packet Size	534.000 Bytes				
[-] Pkt Size Distribution	Bytes	Packets	Utilization	bps	pps
.....<=64	911.00 B	15	0.000%	0.000 bps	0
.....65-127	140.00 B	2	0.000%	0.000 bps	0
.....128-255	243.00 B	1	0.000%	0.000 bps	0
.....256-511	1.21 KB	4	0.000%	0.000 bps	0
.....512-1023	1.75 KB	2	0.000%	0.000 bps	0
.....1024-1517	1.13 KB	1	0.000%	0.000 bps	0
.....>=1518	11.86 KB	8	0.000%	0.000 bps	0

Gambar II.13: Network Utilization Proses Patching/Updating Web Service



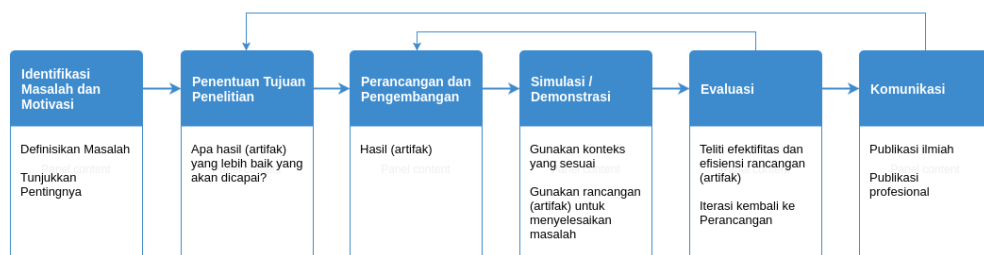
Gambar II.14: Grafik CPU Utilization Proses Updating Web Service

## Bab III

# METODOLOGI

### Metodologi

Metodologi penelitian yang akan dilakukan di dalam penelitian ini adalah melalui pendekatan *Design Science Research Methodology for Information Systems Research* (Peffers, et al, 2008) [21]. Metodologi penelitian terdiri dari enam tahapan seperti digambarkan pada Gambar III.1 yaitu: identifikasi masalah dan motivasi, penentuan tujuan dari penelitian, perancangan dan pengembangan solusi, simulasi atau demonstrasi, evaluasi, komunikasi.



Gambar III.1: Tahapan *Design Science Research Methodology (DSRM)*

#### 1. Identifikasi Masalah dan Motivasi

Langkah pertama dalam melakukan penelitian adalah dengan mengidentifikasi permasalahan sesungguhnya. Selain itu untuk melihat seberapa jauh pentingnya masalah tersebut terhadap sistem, hal ini dapat ditunjukkan pada dampak yang diperoleh apabila permasalahan tersebut tidak segera diatasi. Dengan mengetahui permasalahan, maka dapat diperoleh gambaran yang jelas terhadap masalah yang terjadi dan dapat menentukan batasan dari masalah tersebut. Tujuannya adalah untuk menentukan solusi seperti apa dan sejauhmana solusi tersebut sesuai dengan permasalahan yang terjadi.

#### 2. Menentukan Tujuan dari Penelitian

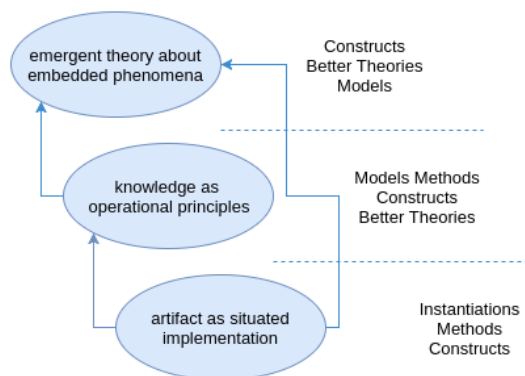
Setelah melakukan identifikasi masalah, langkah selanjutnya yang dilakukan pada fase ini adalah mendefinisikan tujuan dari sebuah solusi berdasarkan identifikasi masalah dan informasi (knowledge) mengenai kondisi permasalahan yang telah dilakukan pada langkah sebelumnya. Tujuan dapat bersifat

kuantitatif, misalnya, solusi yang diinginkan akan lebih baik daripada yang sekarang, atau kualitatif, misalnya, bagaimana metode baru ini diharapkan dapat mendukung solusi untuk masalah yang sampai sekarang belum bisa ditangani. Tujuan harus disimpulkan secara rasional dari spesifikasi masalah, kemampuan yang dibutuhkan untuk hal ini mencakup pengetahuan tentang keadaan masalah dan solusi yang akan dicapai.

### 3. Perancangan dan Pengembangan Solusi

Pelaksanaan tahap design and development pada DSRM adalah bertujuan untuk menghasilkan sebuah artefak baru yang bertujuan memberikan solusi terbaik dari problem yang dihadapi. Berikut beberapa jenis artefak yang dapat dihasilkan dari penerapan metode Design Science Research.

Dari Gambar III.2, Purao membagi keluaran hasil penerapan DSRM menjadi tiga jenis alternatif model artefak, yaitu: munculnya sebuah model teori baru, atau sebuah metode pengembangan yang baru, atau juga dapat berupa instansiasi baru dari sebuah metode yang sudah ada [22].



Gambar III.2: Artefak Hasil Penerapan Metode DSRM (Purao, 2002)

### 4. Pembuatan Simulasi / Demonstrasi

Tahap demonstration bertujuan untuk menunjukkan cara penggunaan artefak (metode) baru untuk memecahkan satu atau lebih kasus dari masalah. Ini bisa melibatkan penggunaannya dalam eksperimen, simulasi, studi kasus, bukti, atau kegiatan lain yang sesuai. Kemampuan yang dibutuhkan untuk demonstrasi ini termasuk pengetahuan yang efektif tentang bagaimana menggunakan metode untuk memecahkan masalah.

### 5. Evaluasi

Tahap evaluasi dilakukan guna mengamati dan mengukur seberapa baik artefak (metode) baru mendukung solusi penyelesaian masalah. Kegiatan ini

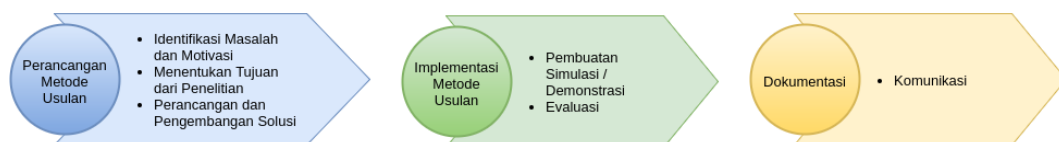
melibatkan pembandingan tujuan solusi dengan hasil yang didapatkan terhadap penerapan metode baru yang dihasilkan. Hasil evaluasi yang didapatkan akan menjadi masukan bagi peneliti untuk memutuskan apakah penelitian akan beralih kembali ke langkah ketiga (design and development) untuk mencoba untuk meningkatkan efektivitas dari artefak atau untuk melanjutkan ke tahap keenam (communication) dan meninggalkan perbaikan lebih lanjut untuk proyek berikutnya.

## 6. Komunikasi

Tahap communication bertujuan untuk memberikan kesempatan kepada peneliti untuk menjelaskan tentang permasalahan yang diangkat dan pentingnya permasalahan itu untuk diteliti. Tahap ini juga berguna untuk menjelaskan bagaimana artefak baru berhasil dibuat, langkah-langkah yang dilakukan dalam menghasilkan artefak baru serta kelebihanannya dalam memberikan solusi bagi permasalahan yang dihadapi. Communication dapat berupa dokumentasi publikasi penelitian ilmiah, peneliti mungkin menggunakan struktur penulisan karya tulis ilmiah dalam model communicationnya, seperti definisi masalah, tinjauan pustaka, pengembangan hipotesis, pengumpulan data, analisis, hasil, diskusi, dan kesimpulan.

## Implementasi Metodologi

Sebagaimana telah dijelaskan dalam bagian sebelumnya bahwa tahapan pelaksanaan riset terdiri dalam enam tahapan. Dalam pelaksanaannya, enam tahap dalam DSRM dibagi kedalam tiga aktivitas utama, yaitu: perancangan metode usulan, implementasi metode usulan, dan dokumentasi. Berikut tahapan pelaksanaan yang dilakukan dalam penelitian Desain dan Implementasi Pengumpulan Data Menggunakan *Mobile Device* Berbasis *Proxy*.



Gambar III.3: Implementasi DSRM

Perancangan metode usulan penelitian merupakan tahap awal yang dilakukan untuk menghasilkan sebuah metode baru. Tahap yang dilakukan pada fase perancangan metode usulan adalah : Identifikasi Masalah dan Motivasi, Menentukan Tujuan dari

Penelitian, dan Perancangan dan Pengembangan Solusi. Implementasi metode usulan dilakukan dengan melakukan simulasi/uji coba metode usulan dalam kaitannya dengan pengumpulan data di BPS. Fase implementasi metode usulan terdiri dari : Pembuatan Simulasi / Demonstrasi dan Evaluasi.

## Bab IV

### ANALISIS DAN PERANCANGAN

#### ANALISIS

Kegiatan perstatistikan di BPS dilakukan dengan mengacu kepada proses bisnis yang tertuang pada GSBPM. Dalam GSBPM, seperti pada Gambar II.2, setiap kegiatan perstatistikan dilakukan dalam 8 fase: *Specify needs*, *Design*, *Build*, *Collect*, *Process*, *Analyze*, *Disseminate*, dan *Evaluate*. Tahap *specify needs*, *design*, dan *build* merupakan tahapan-tahapan pada level perencanaan. Tahap *collect* dan *process* merupakan tahapan-tahapan pada level pelaksanaan. Sementara tahap *analyze*, *disseminate*, dan *evaluate* merupakan tahapan-tahapan paska pelaksanaan.

Pada tahap *collect*, data dikumpulkan dengan menggunakan media kuesioner. Sebagian besar kuesioner yang digunakan dalam pengumpulan data di BPS masih menggunakan media kuesioner kertas, meskipun beberapa *survey* telah diujicobakan untuk menggunakan *mobile device*. Sementara pada tahap *process*, kuesioner yang telah terisi pada tahap *collect* dilakukan *editing*, *coding*, dan imputasi.

#### Penyusunan Model Aplikasi

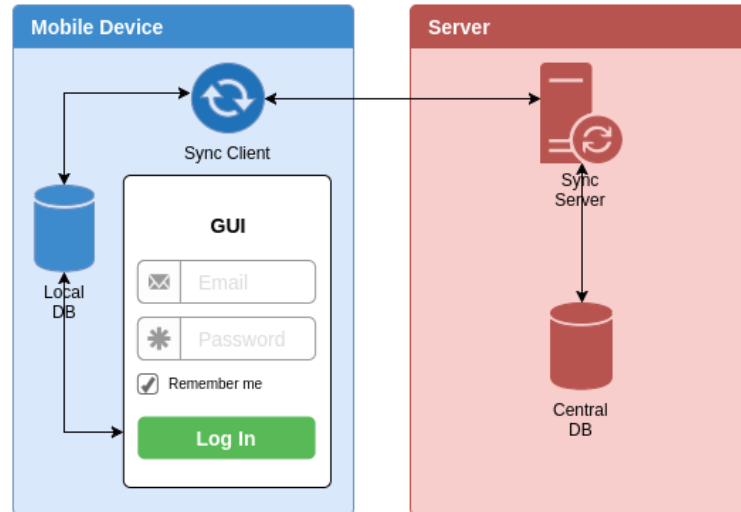
Berdasarkan hasil analisis dan studi literatur dari sejumlah pendekatan yang diusulkan, penggunaan *mobile device* sebagai media pengumpulan data dapat diimplementasikan dalam dua pendekatan:

1. Pendekatan *stand alone*
2. Pendekatan *client server*

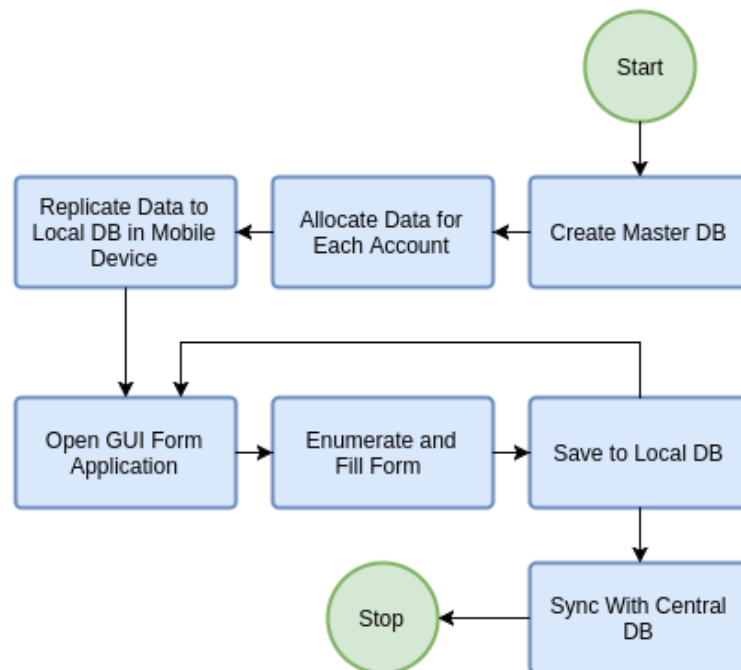
#### Pendekatan *Stand Alone*

Pendekatan *stand alone* merupakan model dimana semua *resources* ditempatkan pada *machine* tunggal. *User interface*, *services*, dan *functions* semuanya berada pada *machine* yang sama. Dalam keterkaitannya dengan pengumpulan data, dimana terdapat dua komponen utama yang harus terinstal pada *device* yaitu *user interface* dan

data, maka keduanya terletak pada *device* yang sama. Gambar IV.1 memberikan ilustrasi cara kerja pendekatan *stand alone*, sementara Gambar IV.2 menunjukkan alur kerja dari pendekatan *stand alone*.



Gambar IV.1: Pendekatan *Stand Alone*



Gambar IV.2: *Flowchart Pendekatan Stand Alone*

Keuntungan dari pendekatan *stand alone* adalah *latency* yang terjadi pada saat *client application* mengakses data menjadi lebih kecil dikarenakan letak data berada pada *local machine*. Selain itu, pendekatan ini juga tidak memerlukan jaringan sebagai media komunikasi, sehingga sangat cocok digunakan pada *disconnected environment*, dimana untuk kasus pengumpulan data lapangan kondisi ini akan sering

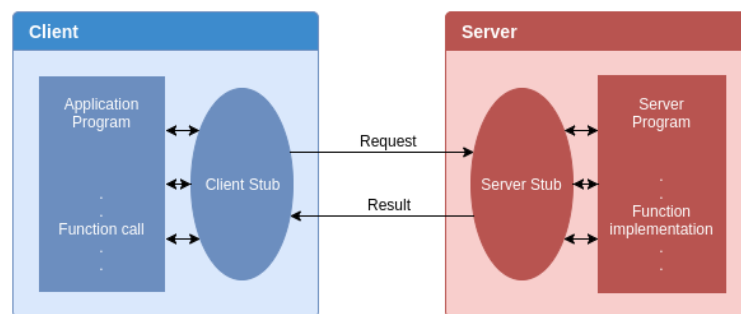
dijumpai.

Sementara itu, pendekatan *stand alone* memiliki sejumlah kelemahan, yaitu dalam hal data transfer dan pembaharuan aplikasi. Pendekatan *stand alone* tidak menggunakan media komunikasi untuk melakukan transfer data, sehingga diperlukan mekanisme transfer data manual untuk menggabungkan keseluruhan data. Selain itu, pembaharuan aplikasi juga harus dilakukan secara manual, sehingga perbaikan *rule* validasi tidak akan dapat diakomodir secara otomatis.

Selain itu, pendekatan ini juga dapat digunakan pada *disconnected environment*, dimana untuk kasus pengumpulan data lapangan kondisi ini akan sering dijumpai. Sementara itu, kelemahan pendekatan *stand alone* ada pada letak eksekusi validasi yang terdapat pada sisi *client*, sehingga perbaikan *rule* validasi harus diikuti dengan pembaharuan aplikasi.

### Pendekatan *Client Server*

Pendekatan *client server* merupakan pendekatan yang telah banyak diadopsi [23] [24] [25]. Secara abstrak, pendekatan *client server* menghubungkan 2 (dua) bagian, *client* dan *server*. *Server* menyediakan fungsi-fungsi, sementara program aplikasi (*client*) melakukan eksekusi terhadap fungsi-fungsi tersebut [23]. Gambar IV.3 menunjukkan arsitektur *client server Remote Procedure Call (RPC)*.

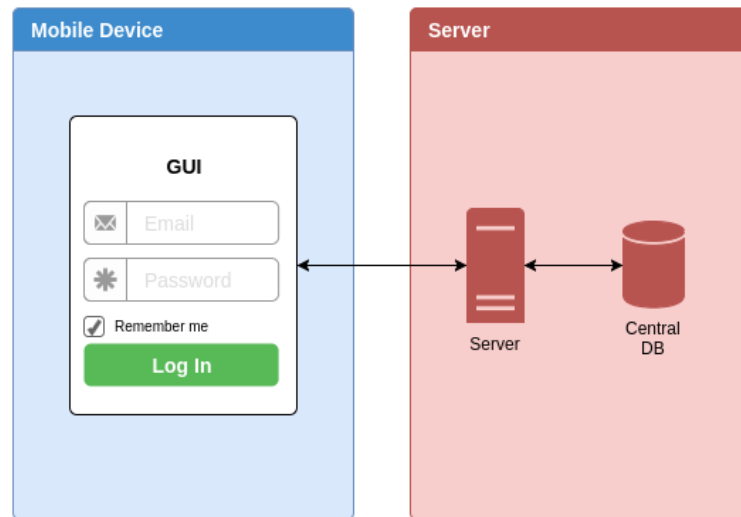


Gambar IV.3: *Client Server* dengan RPC [23]

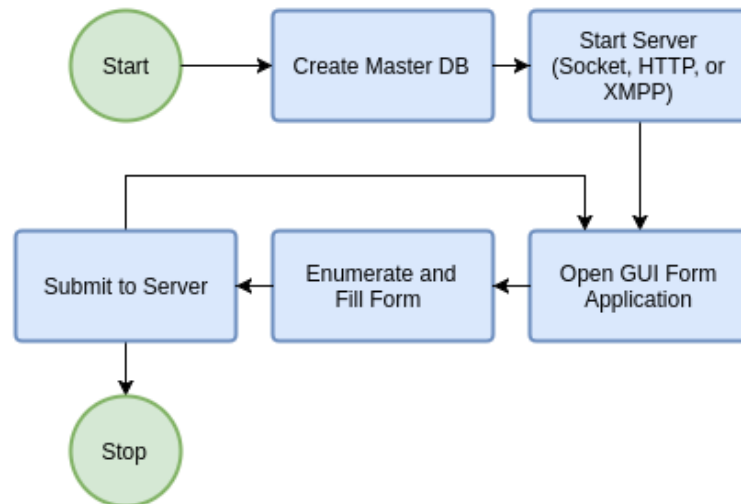
Implementasi pendekatan *client server* dalam konteks pengumpulan data menggunakan *mobile device* sebagai *client*. Pada pendekatan ini data diletakkan berdekatan dengan *server*, sehingga *device* perlu mengaksesnya secara *remote*. Gambar IV.4 memberikan ilustrasi penggunaan pendekatan *client server* dalam pengumpulan data lapangan. Sementara Gambar IV.5 menunjukkan alur kerja dari pendekatan *client server*.



Kelemahan yang dimiliki pendekatan *client server* adalah, untuk dapat digunakan, *client application* harus selalu terkoneksi dengan *central server* dimana data berada, sehingga pendekatan ini tidak dapat digunakan pada *disconnected environment*. Selain itu, setiap kali *client application* melakukan komunikasi dengan *central server*, maka sejumlah data akan ditransmisikan, sehingga diperlukan kapasitas dan kapabilitas jaringan yang cukup baik kuota maupun *transfer rate*.



Gambar IV.4: Pendekatan *Client Server*



Gambar IV.5: *Flowchart* Pendekatan *Client Server*

Sebagaimana telah diketahui, pengumpulan data merupakan suatu kegiatan yang memerlukan mobilitas. Petugas perlu berkeliling untuk mengunjungi responden secara *door-to-door*. Pendekatan *client-server* memerlukan ketergantungan yang tinggi terhadap sinyal komunikasi, sementara kondisi lapangan tidak selalu mendukung tersedianya sinyal komunikasi. Di sisi lain, pendekatan *stand alone* memiliki kelemahan dalam proses transfer data yang harus dilakukan secara manual.

Berdasarkan analisis terhadap kebutuhan dalam pengumpulan data dengan menggunakan *mobile device*, maka desain yang dirancang harus mempertimbangkan kriteria berikut:

1. Mendukung *disconnected environment*

Pengumpulan data memerlukan penyisiran lapangan dengan kondisi geografis yang bervariasi. Kondisi sinyal telekomunikasi pada masing-masing wilayah pendataan juga bervariasi, dari yang bagus sampai yang tidak terdapat sinyal komunikasi. Desain yang dirancang harus dapat mengakomodir pengumpulan data pada wilayah yang tidak terdapat sinyal komunikasi.

2. Mendukung pembaharuan secara otomatis

Kuesioner elektronik yang terinstal pada *device* merupakan representasi dari variabel sensus/survei yang akan dikumpulkan. Dalam pengisiannya, ada aturan *rule* validasi yang harus diikuti, sehingga pembaharuan *rule* validasi diperlukan ketika ada perbaikan konsep dan definisi.

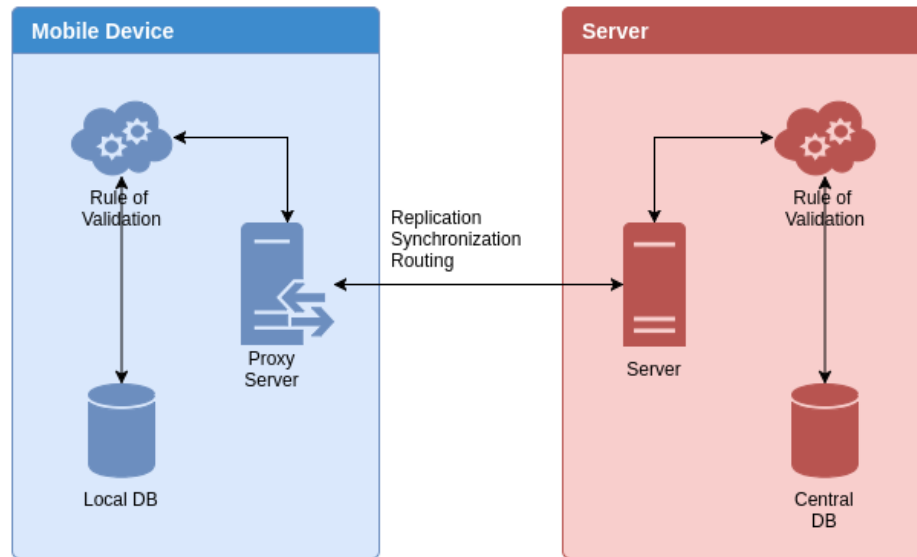
## PERANCANGAN

### Perancangan Model Aplikasi

Secara garis besar, terdapat 2 (dua) jenis *object* yang menjadi poin utama dari solusi yang akan dirancang, yaitu: data dan *rule* validasi. Agar mendukung *disconnected environment*, maka diperlukan adanya mekanisme replikasi data. Sementara agar data senantiasa *up-to-date*, diperlukan juga adanya mekanisme sinkronisasi. Secara umum, solusi yang akan dirancang akan memiliki arsitektur seperti pada Gambar IV.6.

### Replikasi

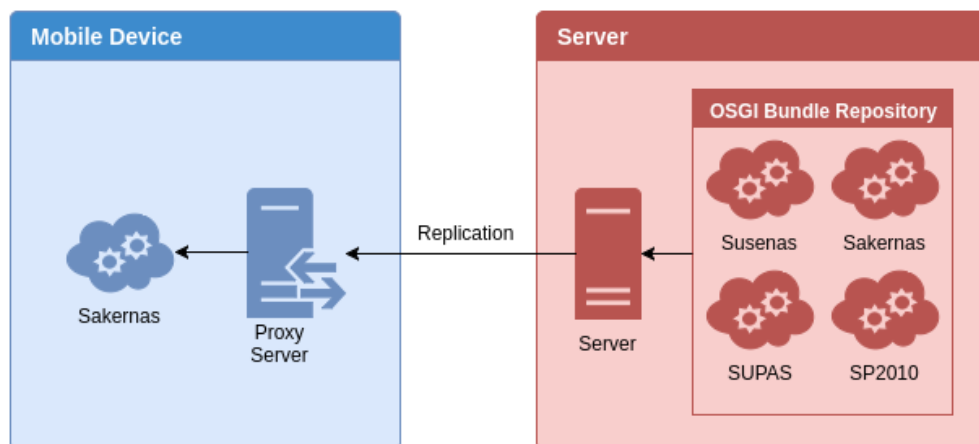
Replikasi merupakan suatu cara untuk memastikan agar baik *device* maupun *server* mengeksekusi objek yang sama. Replikasi dilakukan satu arah, yaitu dari *server* ke *device*. Terdapat 2 (dua) jenis mekanisme replikasi yang dirancang, yaitu: replikasi *rule* dan replikasi data.



Gambar IV.6: Arsitektur Solusi

### Replikasi Rule Validasi

*Rule* validasi ada implementasi dari *logic workflow*, seperti yang tertuang pada GS-BPM fase *Build* subproses *Configure Workflows*. Replikasi *rule* validasi bertujuan agar validasi isian kuesioner dapat dijalankan secara lokal. Implementasi *rule* validasi dikemas dalam format *OSGI Bundle*. Gambar IV.7 menunjukkan skema replikasi *rule* validasi.

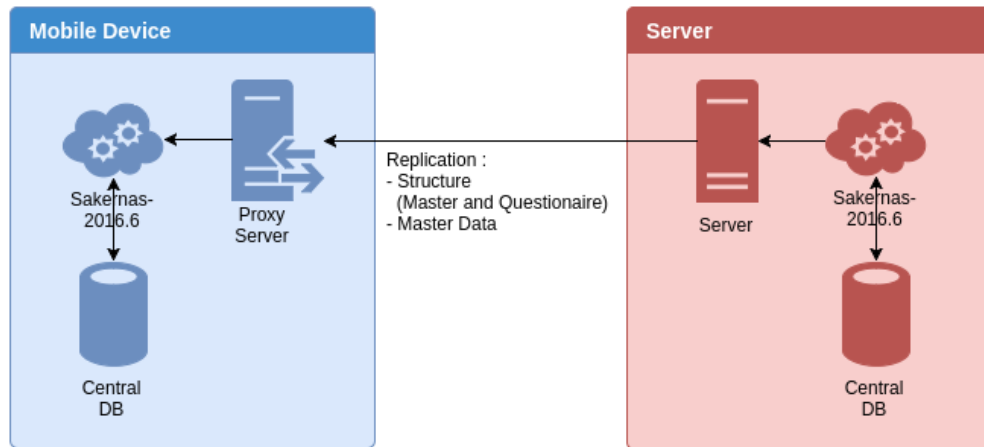


Gambar IV.7: Skema Replikasi Rule Validasi

### Replikasi Data

*Rule* validasi selain memuat *logic workflow* juga memuat *object* yang merupakan representasi dari tabel di *database* yang disebut dengan *Object Relational Mapping*

(ORM). Tidak semua data diperlukan, sehingga replikasi harus dilakukan secara selektif. ORM inilah yang akan menentukan tabel dan data apa saja yang akan di replikasi. Gambar IV.8 menunjukkan skema replikasi data.



Gambar IV.8: Skema Replikasi Data

## Sinkronisasi

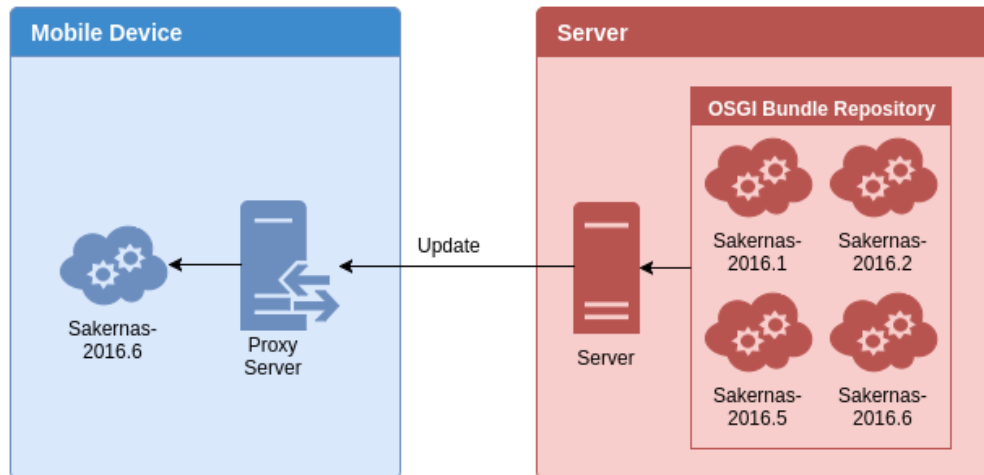
Seperti halnya replikasi, sinkronisasi juga bertujuan untuk memastikan *device* dan *server* mengeksekusi objek yang sama. Akan tetapi, sinkronisasi berjalan sepanjang waktu, baik melalui *scheduler* maupun *trigger*. Selain itu, sinkronisasi dilakukan dua arah, dari *server* ke *device* maupun dari *device* ke *server*, tergantung *object* yang sedang dieksekusi. Terdapat 2 (dua) jenis mekanisme sinkronisasi yang dirancang, yaitu: sinkronisasi rule dan sinkronisasi data.

### Sinkronisasi Rule Validasi

Sinkronisasi *rule* validasi menggunakan implementasi yang sama dengan replikasi *rule* validasi. Perbedaanya, sinkronisasi *rule* validasi menggunakan *trigger* yang berupa versi. Setiap kali *client application* menemukan pembaharuan versi *rule* validasi pada server, maka *client application* secara otomatis akan memperbarui *rule* validasinya. Gambar IV.9 menunjukkan skema sinkronisasi *rule* validasi.

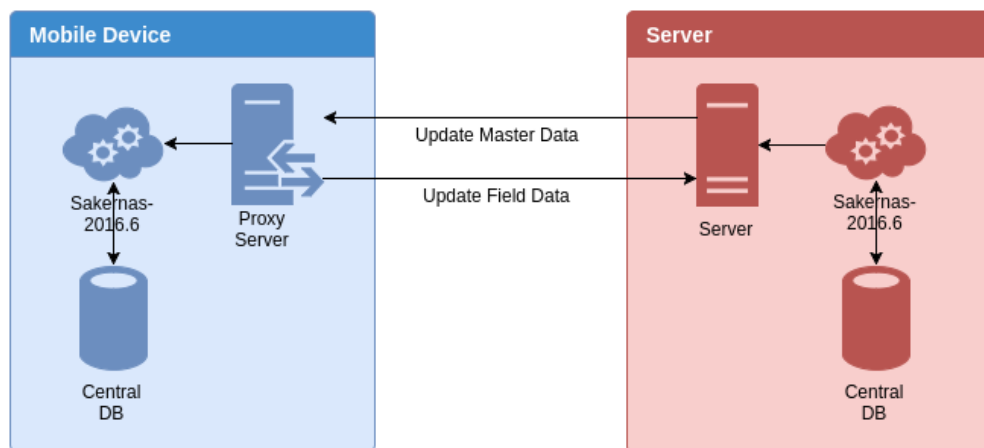
### Sinkronisasi Data

Sebagaimana yang telah dijabarkan dalam analisis, terdapat 2 (dua) jenis data yang dikelola, yaitu: data *master* dan data lapangan. Arah sinkronisasi data menyesuaikan jenis data-nya. Data master pada *server* diasumsikan lebih *up-to-date*, sehingga



Gambar IV.9: Skema Sinkronisasi *Rule* Validasi

sinkronisasi dilakukan dari arah *server* ke *device*. Sebaliknya, data lapangan pada *device* diasumsikan lebih *up-to-date*, sehingga sinkronisasi dilakukan dari arah *device* ke *server*. Gambar IV.10 menunjukkan skema sinkronisasi data.



Gambar IV.10: Skema Sinkronisasi Data

## Routing

Proses *routing* menggunakan sebuah *proxy server*. *Proxy server* berperan sebagai *Application-Level Gateway (ALG)* yang menyediakan fungsi *security*, *filtering*, dan *packet forwarding*.

## Pengelolaan Data

Sesuai dengan model aplikasi yang akan dirancang, yaitu pendekatan *client server* dengan *proxy*, terdapat 2 (dua) jenis penyimpanan data, yaitu pada sisi *client* dan

sisi *server*. Agar kedua sisi penyimpanan data tersebut merepresentasikan data yang sama, maka terdapat mekanisme pengelolaan data yang harus diikuti.

Dalam kegiatan pengumpulan data lapangan, terdapat 2 (dua) jenis data yang digunakan:

1. Data *master*

Data *master* merupakan data yang digunakan sebagai referensi oleh variabel pendataan. Data *master* bersifat *non-volatile*, yang artinya tidak mudah berubah. Perubahan data master hanya dapat dilakukan pada *phase* design pada GSBPM. Contoh data *master* antara lain: data master wilayah, data master status perkawinan, data master status dalam rumah tangga, dll.

2. Data lapangan

Data lapangan merupakan data yang isinya merupakan representasi hasil pendataan. Data ini bersifat transaksional, artinya data dapat dengan mudah ditambah, dihapus, maupun diupdate. Contoh data lapangan antara lain: data sakernas semester I 2016, data susenas semester I 2016, data usaha rumah tangga pertanian ST2016, dll.

Secara umum, variabel yang sejenis yang digunakan pada beberapa sensus maupun survei mempunyai konsep dan definisi yang sama, misalnya: Variabel jenis kelamin pada Sakernas <sup>1</sup>, Susenas <sup>2</sup>, SUPAS <sup>3</sup>, dll, mempunyai konsep dan definisi, serta atribut yang sama, yaitu: 1 - laki-laki, 2 - perempuan. Begitu pula variabel-variabel yang lain, digunakan secara bersamaan oleh beberapa sensus dan survei.

Dari hasil analisis berdasarkan jenis-jenis data dan karakteristik variabel yang digunakan, maka dapat dirangkum dalam beberapa poin berikut:

1. *Server* memuat seluruh data *master* yang digunakan pada seluruh kegiatan sensus dan survei.
2. *Client* memuat hanya data *master* yang digunakan oleh kegiatan tertentu sana.
3. Data *master* bersifat *non-volatile*, sehingga diasumsikan data master pada *server* lebih *up-to date* dibandingkan data master pada *client*. Oleh karena itu, replikasi data *master* dilakukan dari *server* ke *client*.

---

<sup>1</sup><https://sirusa.bps.go.id/index.php?r=variabel/view&id=11482>

<sup>2</sup><https://sirusa.bps.go.id/index.php?r=variabel/view&id=12526>

<sup>3</sup><https://sirusa.bps.go.id/index.php?r=variabel/view&id=11341>

4. Data lapangan bersifat transaksional, artinya data lapangan pada *client* diasumsikan kondisinya lebih *up-to date* dibandingkan data lapangan pada *server*. Oleh karena itu, replikasi data lapangan dilakukan dari *client* ke *server*.

## Daftar Pustaka

- [1] *GSBPM v5.0 - Generic Statistical Business Process Model - UNECE Statistics Wikis.*
- [2] *Redesigning a Questionnaire for Computer Assisted Data Collect - cp940020.pdf.*
- [3] G. Klein and M. G. Sobol, "Bias in computer-assisted surveys," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 26, pp. 566–571, Sept. 1996.
- [4] R. Tergujeff, J. Haajanen, J. Leppanen, and S. Toivonen, "Mobile SOA: Service Orientation on Lightweight Mobile Devices," in *IEEE International Conference on Web Services, 2007. ICWS 2007*, pp. 1224–1225, July 2007.
- [5] C. Gutwin, T. N. Graham, C. Wolfe, N. Wong, and B. de Alwis, "Gone but Not Forgotten: Designing for Disconnection in Synchronous Groupware," in *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, CSCW '10*, (New York, NY, USA), pp. 179–188, ACM, 2010.
- [6] X. M. Zhang, Y. Zhang, F. Yan, and A. V. Vasilakos, "Interference-based topology control algorithm for delay-constrained mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 742–754, 2015.
- [7] B. DeRenzi, Y. Anokwa, T. Parikh, and G. Borriello, "Reliable Data Collection in Highly Disconnected Environments Using Mobile Phones," in *Proceedings of the 2007 Workshop on Networked Systems for Developing Regions, NSDR '07*, (New York, NY, USA), pp. 4:1–4:5, ACM, 2007.
- [8] T. S. Parikh and E. D. Lazowska, "Designing an Architecture for Delivering Mobile Information Services to the Rural Developing World," in *Proceedings of the 15th International Conference on World Wide Web, WWW '06*, (New York, NY, USA), pp. 791–800, ACM, 2006.
- [9] Takdir and A. I. Kistijantoro, "Multi-layer SOA implementation pattern with service and data proxies for distributed data-intensive application system," in *2014 International Conference on ICT For Smart Society (ICISS)*, pp. 37–41, Sept. 2014.
- [10] *GSIM Specification - Generic Statistical Information Model - UNECE Statistics Wikis.*
- [11] *CSPA v1.5 - Common Statistical Production Architecture - UNECE Statistics Wikis.*
- [12] *A primer on distributed computing.*



- [13] G. R. Andrews, *Foundations of Multithreaded, Parallel, and Distributed*. Addison Wesley, Oct. 1999.
- [14] M. Baker, R. Buyya, and D. Laforenza, "Grids and Grid technologies for wide-area distributed computing," *Software: Practice and Experience*, vol. 32, pp. 1437–1466, Dec. 2002.
- [15] J. Sathyan and M. Rijas, "Job Management in Mobile Grid Computing," in *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '09, (New York, NY, USA), pp. 422–426, ACM, 2009.
- [16] J. Zottl, G. Haring, and G. Kotsis, "A Three-tier Information Management Architecture for Mobile Grid Environments," in *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '08, (New York, NY, USA), pp. 243–247, ACM, 2008.
- [17] H. Weatherspoon and J. Kubiatowicz, "Erasure Coding Vs. Replication: A Quantitative Comparison," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, (London, UK, UK), pp. 328–338, Springer-Verlag, 2002.
- [18] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-peer Networks," in *Proceedings of the 16th International Conference on Supercomputing*, ICS '02, (New York, NY, USA), pp. 84–95, ACM, 2002.
- [19] Y. H. Ho, A. H. Ho, K. A. Hua, and G. L. Hamza-Lup, "A connectionless approach to mobile ad hoc networks," in *Ninth International Symposium on Computers and Communications, 2004. Proceedings. ISCC 2004*, vol. 1, pp. 188–195 Vol.1, June 2004.
- [20] N. Imran, I. Rao, Y.-K. Lee, and S. Lee, "A Proxy-based Uncoordinated Checkpointing Scheme with Pessimistic Message Logging for Mobile Grid Systems," in *Proceedings of the 16th International Symposium on High Performance Distributed Computing*, HPDC '07, (New York, NY, USA), pp. 237–238, ACM, 2007.
- [21] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.
- [22] S. Purao, "Design research in the technology of information systems: Truth or dare," *GSU Department of CIS Working Paper*, pp. 45–77, 2002.
- [23] H. Schuster, S. Jablonski, T. Kirsche, and C. Bussler, "A client/server architecture for distributed workflow management systems," in *Parallel and Dis-*

*tributed Information Systems, 1994., Proceedings of the Third International Conference on*, pp. 253–256, IEEE, 1994.

- [24] M. Bertocco, F. Ferraris, C. Offelli, and M. Parvis, “A client-server architecture for distributed measurement systems,” *IEEE transactions on instrumentation and measurement*, vol. 47, no. 5, pp. 1143–1148, 1998.
- [25] M. J. Callaghan, J. Harkin, E. McColgan, T. M. McGinnity, and L. P. Maguire, “Client–server architecture for collaborative remote experimentation,” *Journal of Network and Computer Applications*, vol. 30, no. 4, pp. 1295–1308, 2007.

## **Lampiran 1**