**DESIGNING**

**GEOGRAPHICAL-BASED**

**DISTRIBUTED DATABASE**

**PROPOSAL TESIS**

**Disusun sebagai syarat kelulusan matakuliah**

**EL5090 Metodologi Penelitian / Tesis 1**

**OLEH**

**ARIS PRAWISUDATAMA**

**23215131**



**PROGRAM STUDI MAGISTER TEKNIK ELEKTRO**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2016**

**LEMBAR PENGESAHAN**

**DESIGNING**

**GEOGRAPHICAL-BASED**

**DISTRIBUTED DATABASE**

**PROPOSAL TESIS**

**OLEH**

**ARIS PRAWISUDATAMA**

**23215131**

**PROGRAM STUDI MAGISTER TEKNIK ELEKTRO**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

Telah disetujui sebagai Proposal Tesis

**di Bandung, pada tanggal ………………..**

|  |  |
| --- | --- |
| **Pembimbing I,** | **Pembimbing II,** |
| …………*(Nama)*  NIP……………… | ……………*(Nama)*  NIP……………… |

**DAFTAR ISI**

|  |  |
| --- | --- |
| **LEMBAR PENGESAHAN……………………………………………………..** | **ii** |
| **DAFTAR ISI……………………………………………………………………..** | **iii** |
| **DAFTAR GAMBAR ……………………………………………………………**  **DAFTAR TABEL……………………………………………………………….** | **iv**  **v** |
| 1. **Ringkasan Proposal……………………………………………………….** | **1** |
| 1. **Latar Belakang ……………………………………………………………** | **dst** |
| 1. **Rumusan Masalah…………………………………………………………** |  |
| 1. **Tujuan ……………………………………………………………………..** |  |
| 1. **Batasan Masalah ………………………………………………………….** |  |
| 1. **Studi Literatur ……………………………………………………………** |  |
| 1. **Metodologi ………………………………………………………………...** |  |
| 1. **Implikasi …………………………………………………………………..** |  |
| 1. **Sistematika Penulisan …………………………………………………….** |  |
| 1. **Penjadwalan ………………………………………………………………** |  |
| **DAFTAR REFERENSI …………………………………………………………** |  |

**DESIGNING**

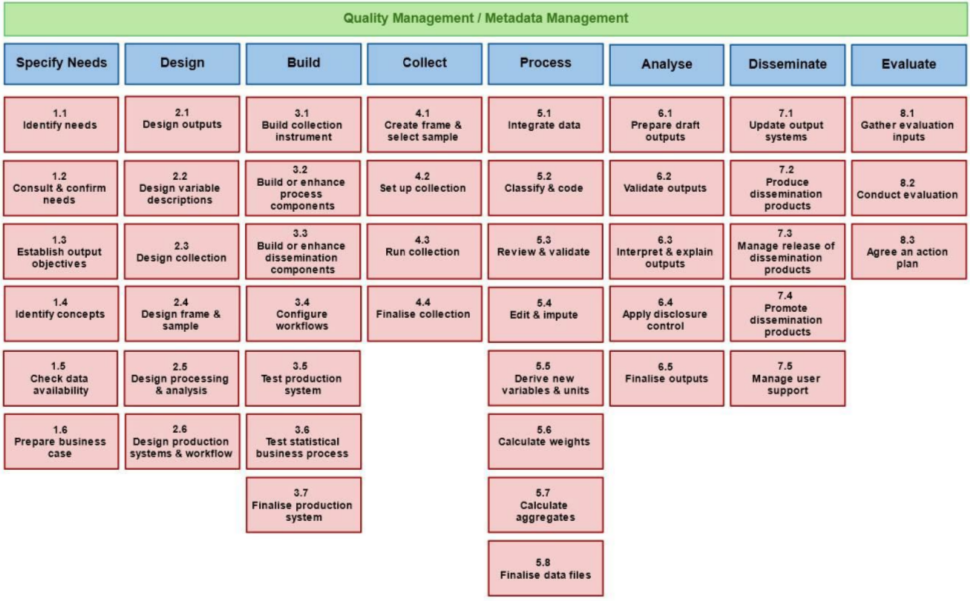
**GEOGRAPHICAL-BASED DISTRIBUTED DATABASE**

**FOR DATA ENTRY**

1. PENDAHULUAN

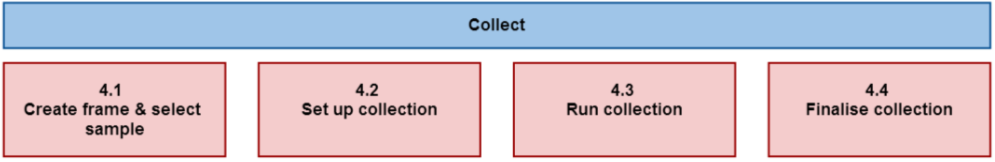
Badan Pusat Statistik (BPS) merupakan suatu lembaga pemerintah non-departemen yang bertanggung jawab dalam penyediaan statistik dasar. Dalam peranannya sebagai penyedia data, BPS melakukan pengumpulan data dengan 2 (dua) metode : primer dan sekunder. Pengumpulan data primer berarti BPS secara mandiri 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]](#footnote-2), 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.

Gambar 1. *Statistical Business Process Phases* dalam GSBPM

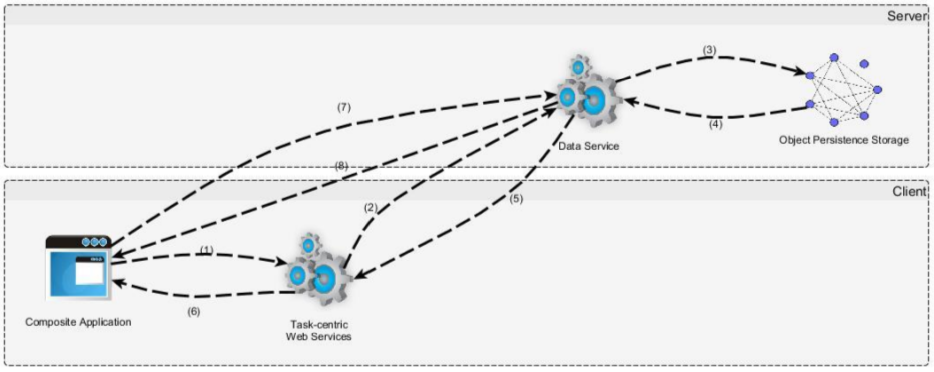
1. LATAR BELAKANG MASALAH

Dalam GSBPM terdapat *Collect Phase* yang didefinisikan sebagai kegiatan pengumpulan semua informasi (data dan metadata), menggunakan beberapa metode pengumpulan (termasuk ekstraksi dari register dan database statistik, administratif, maupun yang lain), dan memuatkannya ke dalam suatu *environtment* untuk pemrosesan lebih lanjut. *Collect Phase* dibagi menjadi 4 (empat) *sub-process* yang salah satunya merupakan *run collection sub-process.* Sub-proses *run collection* merupakan sub-proses dimana pengumpulan data dilakukan, termasuk didalamnya *manual entry data*.

Gambar 2. *Collect Phase* dalam GSBPM

Implementasi data entry manual di BPS dilakukan dalam bentuk pusat pengolahan (puslah) yang terdapat pada masing-masing satuan kerja (pusat, provinsi, kabupaten/kota). Desain masing-masing puslah adalah berupa sebuah jaringan komputer yang terdiri dari sebuah server dan sejumlah client. Suatu puslah juga didesain berdiri sendiri dan tidak terkoneksi dengan puslah yang lain membentuk sebuah *cluster*. Pola desain puslah seperti yang sekarang diterapkan mempunyai kelemahan dalam konsistensi data antara pusat-puslah dan *single point of failure.*

Dalam penelitiannya, Takdir[[2]](#footnote-3), menyimpulkan bahwa : 1. inkonsistensi data dapat diminimalisir dengan pengelolaan database secara terpusat, 2. penerapan pola terdistribusi, baik pada data (database), maupun pada proses (*web services*) terbukti dapat meningkatkan kinerja sistem yang bersifat data-intensif. Masalahnya, skema usulan Takdir hanya diuji dalam *environment* software virtualisasi Oracle VM VirtualBox® yang tidak mempertimbangkan *latency* dan *connection lost*. Pada faktanya, kondisi geografis Indonesia tidak memungkinkan terciptanya koneksi internet yang stabil yang menjamin keberlangsungan proses entry data.

Gambar 3. Skema Usulan, Takdir[[3]](#footnote-4)

1. BATASAN MASALAH

Sebagai *preliminary statement*, masalah koneksi seharusnya dapat diatasi dengan cara mengimplementasikan skema client-server terdistribusi, sebagaimana yang diusulkan Takdir, dalam sebuah jaringan lokal (LAN). Kemudian untuk menjamin konsistensi data, maka dapat diaplikasikan *middleware* yang diimplementasikan pada masing-masing database di setiap pusat pengolahan. *Middleware* ini akan terhubung dengan *middleware* pada pusat pengolahan yang lain dan membentuk sebuah *cluster*.

Solusi *distributed database* yang mendukung replikasi dan *sharding* sebenarnya sudah banyak digunakan pada NoSQL database, seperti Hadoop dan Cassandra, tapi belum banyak digunakan pada *relational database management system.* NoSQL database mempunyai sifat *non-relational[[4]](#footnote-5)*, sehingga tidak cocok digunakan untuk aplikasi *data entry* yang memerlukan *high-cardinality* untuk menjamin konsistensi data. Sementara itu, PostgreSQL sebagai salah satu contoh RDBMS sudah mendukung replication serta clustering, melalui beberapa *middleware*. Misalnya, Slony-I [[5]](#footnote-6) dapat melakukan replikasi database satu arah dari Master ke Slave. Contoh lain, Pgpool-II[[6]](#footnote-7), hanya dapat mereplikasi, tetapi tidak mendukung *sharding*.

Penelitian ini akan fokus kepada perancangan desain yang tepat dalam penerapan distributed relational database, dengan menggunakan *middleware*, untuk diterapkan pada *data entry system*.

1. STUDI LITERATUR

*High Performance Computing (HPC)* adalah sebuah bidang yang dulu hanya terbatas pada institusi yang memiliki anggaran yang besar dan memiliki *supercomputer*. Tetapi seiring dengan kebutuhan akan HPC dalam skala yang lebih kecil, maka berkembanglah apa yang sekarang kita kenal dengan *Cluster Computing*. Kemunculan *cluster platform* didorong oleh munculnya beberapa *academic project*, seperti Beowulf[[7]](#footnote-8), Berkeley NOW[[8]](#footnote-9), dan HPVM[[9]](#footnote-10). Teknologi lain, *Grid Computing,*  muncul di pertengahan 1990an, yang bertujuan memfasilitasi *users* untuk mengakses komputer yang sedang menganggur (*idle computers*) secara remote. Sementara itu, *Cloud Computing,* yang baru muncul diakhir tahun 2007, menyediakan sejumlah sumber daya (*computational resources*) yang dapat diakses melalui internet[[10]](#footnote-11).

Tantangan pada *Cluster, Grid,* dan *Cloud Computing[[11]](#footnote-12)* :

* *Cluster Computing*
  1. *Middleware*: Untuk menghasilkan *software environment* yang menghasilkan ilusi sebuah *cluster* sebagai suatu *single system*,
  2. Program: Aplikasi yang berjalan diatas cluster harus secara *explicit* menggabungkan pembagian tugas antar *nodes*,
  3. *Elasticity* : Variansi waktu respons ketika jumlah *request* berubah drastis,
  4. Scalability: Menambah sumber daya, sehingga mempengaruhi performa sistem,
* *Grid Computing*
  1. *Dinamicity* : *Recources* dimiliki dan dikelola oleh lebih dari satu organisasi, yang dapat keluar-masuk *grid* setiap saat, sehingga menyebabkan beban bagi *grid*,
  2. *Administration* : Untuk membentuk *resource pool* yang terpadu, sulit untuk menyelaraskan *policy* di *local administration* dan *global*,
  3. *Development* : Menulis *software* yang berjalan pada *grid platform* yang mampu untuk men-*decompose* dan mendistribusikan *request* ke *processing elements* dan meyusun solusi,
  4. *Accounting* : Menemukan cara untuk men-*support* berbagai infrastruktur keuangan, model ekonomi, dan model aplikasi yang berbeda,
  5. *Programming* : *Low-coupling* antar *node* dan *distributed nature* pada prosesnya, membuat pemrograman aplikasi menjadi kompleks,
  6. *Heterogeneity* : Menemukan cara untuk membuat *wide-area* pada pemrograman *data intensive* dan penjadwalan pada *resource* yang *heterogen*,
* *Cloud Computing*
  1. *Dynamic Scalability* : Jumlah *node* yang digunakan dapat di-*scale* secara naik dan turun, mengikuti waktu respon,
  2. *Multi-tenancy* : Ketika jumlah aplikasi yang berjalan di sebuah *node* bertambah, ini akan menurunkan *bandwith* yang memungkinkan turunnya peforma,
  3. *Querying and access* : Keamanan *query* dan akses adalah sebuah masalah yang masih terbuka,
  4. *Standardization* : Setiap organisasi memiliki API dan protokol sendiri, sehingga integrasi dan *interoperability*-nya menjadi sebuah tantangan,
  5. *Reliability* dan *fault-tolerance* : Diperlukan *tools* untuk melakukan *testing*, untuk menguji *fault-tolerance* dan kehandalan sistem,
  6. *Debugging* dan *profilling* : *Parallel* dan *remote debugging* adalah masalah klasik pada HPC,
  7. *Security* dan *privacy* : Pengguna tidak mempunyai pilihan dimana *data* akan disimpan,
  8. *Power* : Meskipun *cloud-computing* menawarkan beberapa tipe *service* yang sesuai dengan pengguna, *energy-aware resource management* tetaplah diperlukan.

Pembimbing,

Dr. I Bagus Baskara Nugraha

1. UNECE, 2013b. Generic Statistical Business Process Model, Version 5.0. [↑](#footnote-ref-2)
2. Takdir, PERANCANGAN POLA IMPLEMENTASI SOA UNTUK SISTEM APLIKASI DATA-INTENSIF TERDISTRIBUSI, Institut Teknologi Bandung, Bandung, 2015, Hal. 74 [↑](#footnote-ref-3)
3. Takdir, PERANCANGAN POLA IMPLEMENTASI SOA UNTUK SISTEM APLIKASI DATA-INTENSIF TERDISTRIBUSI, Institut Teknologi Bandung, Bandung, 2015, Hal. 54 [↑](#footnote-ref-4)
4. <http://nosql-database.org/>, diakses 6 February 2016 [↑](#footnote-ref-5)
5. <https://wiki.postgresql.org/wiki/Slony>, diakses 6 February 2016 [↑](#footnote-ref-6)
6. <https://wiki.postgresql.org/wiki/Pgpool-II>, diakses 6 February 2016 [↑](#footnote-ref-7)
7. “Beowulf.org | Mailing List and Archives,” accessed March 6, 2016, http://www.beowulf.org/. [↑](#footnote-ref-8)
8. T. E. Anderson, D. E. Culler, and D. Patterson, “A Case for NOW (Networks of Workstations),” *IEEE Micro* 15, no. 1 (February 1995): 54–64, doi:10.1109/40.342018. [↑](#footnote-ref-9)
9. “Scalable Clusters of Commodity Computers,” accessed March 6, 2016, http://cseweb.ucsd.edu/groups/csag/html/projects/clusters.html. [↑](#footnote-ref-10)
10. Ian Foster and Carl Kesselman, *The Grid 2: Blueprint for a New Computing Infrastructure* (Elsevier, 2003). [↑](#footnote-ref-11)
11. N. Sadashiv and S. M. D. Kumar, “Cluster, Grid and Cloud Computing: A Detailed Comparison,” in *2011 6th International Conference on Computer Science Education (ICCSE)*, 2011, 477–82, doi:10.1109/ICCSE.2011.6028683. [↑](#footnote-ref-12)