

# DESAIN DATABASE YANG SKALABEL UNTUK BERBAGAI APLIKASI BERBASIS GIS

Edhy Sutanta<sup>1)</sup>, Rosalia Arum Kumalasanti<sup>2)</sup>

<sup>1,2)</sup> Jurusan Teknik Informatika, Fakultas Teknologi Industri, IST AKPRIND Yogyakarta  
email: <sup>1)</sup>edhy\_sst@akprind.ac.id, <sup>2)</sup>rosaliaarum@akprind.ac.id

## *Abstract*

*The new approach in GIS based system has made a change from spatial data division and attribute to become database structure, so that the displayed information by GIS is taken from data which storages in the database. Most part of GIS application which has been developed recently for a certain purpose of specific object by applying a specific database design. As the result, the development of the data in the GIS applications only can only be executed by detaching the database then adjust them based on the development needed. This paper discuss the alternative solution, in form of a scaled database design framework to be applied in GIS based applications.*

**Keywords:** database, GIS applications, scalable database.

## 1. PENDAHULUAN

Aplikasi *Geographic Information System* (GIS) atau Sistem Informasi Geografis (SIG) terus berkembang dari waktu ke waktu. Aplikasi GIS pertama kali diterapkan oleh Departemen Energi, Pertambangan dan Sumber Daya, Kanada pada tahun 1967 dengan nama CGIS (*Canadian GIS*). Perkembangan perangkat keras komputer mikro memacu ESRI, CARIS, MapInfo, dan vendor lainnya menggabungkan pendekatan pemisahan data spasial dan atribut pada aplikasi GIS generasi pertama dengan pendekatan baru yaitu organisasi data atribut menjadi struktur database. Pada tahun 1980-an hingga 1990-an, perkembangan industri memungkinkan GIS diterapkan pada *workstation* UNIX dan komputer pribadi, dan sejak tahun 1990-an aplikasi SIG terus dikembangkan untuk berbagai bidang dan objek yang beragam (Sutanta, 2010). Akhir abad ke-20, berbagai sistem GIS distandarisasi menjadi lebih sedikit *platform*, dan pengguna juga mengeksplor dan menampilkan data GIS via internet sehingga perlu standarisasi format data dan transfer. Aplikasi GIS saat ini juga semakin banyak yang dikembangkan dengan memanfaatkan google map karena relatif mudah digunakan, berbasis *web*, dan menyediakan pilihan tampilan *terain* (kontur permukaan bumi), citra satelit, dan penunjuk jalan (Sutanta, 2010).

Seiring perkembangan aplikasi, definisi GIS juga berkembang karena merupakan bidang kajian ilmu dan teknologi yang relatif baru, sehingga muncul beragam definisi tentang GIS. Menurut ESRI (1990) dalam Purnomo (1990) GIS adalah sistem komputer yang menangani serta memakai data yang menggambarkan tempat-tempat dipermukaan bumi. Menurut Wiradisastra (2000) GIS adalah sistem informasi yang dirancang untuk bekerja dengan data bereferensi spasial atau berkoordinat geografi. Purwadhi (2001) mendefinisikan GIS sebagai sistem yang mengorganisir perangkat keras, perangkat lunak, dan data, serta dapat mendayagunakan sistem penyimpanan, pengolahan, maupun analisis data secara simultan, sehingga dapat diperoleh informasi berkaitan dengan aspek keruangan. Setiawan (2011) mendefinisikan GIS sebagai sistem yang berorientasi kepada letak geografis di atas permukaan bumi, berbasis komputer yang mempunyai kemampuan mengolah, memanipulasi serta menampilkan data spasial maupun atribut.

Aplikasi berbasis berbasis GIS telah banyak dikembangkan perseorangan dengan objek dan alat bantu yang beragam. Aplikasi GIS untuk Perguruan Tinggi misalnya, telah dikembangkan oleh Sutanta (2010) dan Suratmo et.al. (2014). Aplikasi GIS objek wisata, museum, cagar budaya, atau candi ditemukan pada penelitian Prabowo et.al. (2013), Haryanto dan

Sholeh (2014), Zuliyanto et.al (2014), Arkiang et. Al. (2014), Sholeh et.al (2015), Setiadi dan Yuliansyah (2015), Suraya dan Sholeh (2015), dan Kusumaningsih et.al. (2016). Aplikasi GIS untuk Ruang Terbuka Hijau Kota misalnya dikembangkan oleh Mildawani dan Susilowati (2012) dan Andika et.al. (2015). Aplikasi GIS industri pernah dibuat oleh Ariandono et.al. (2014) dan Wibowo et.al. (2014). Aplikasi GIS untuk objek layanan kesehatan dapat ditemukan misalnya pada Hege et.al. (2014) dan Nurnawati (2014). Sedangkan aplikasi GIS untuk beberapa objek lain, seperti rumah ibadah, rumah kos, agen tiki, atau sawah pernah dikembangkan pada Sari et.al (2013), Pratikto et.al. (2014), Suharto et.al (2014), dan Ardianto et.al. (2016). Aplikasi-aplikasi GIS tersebut sebagian besar dikembangkan berbasis *WebGIS* dengan melibatkan bahasa pemrograman, dan sebagian yang lain berbasis perangkat bergerak, sebagian besar menggunakan perangkat lunak yang bisa diperoleh secara gratis dan sebagian kecil lainnya berbayar. Aplikasi GIS juga telah banyak dikembangkan dan disediakan oleh lembaga pemerintah, beberapa contohnya bisa diakses pada laman <http://gis.kominfo.go.id/>, <http://gis.depkes.go.id/>, <http://gis.pusairpu.go.id/>, <http://potensiwisata.bantulkab.go.id/>, <http://kewilayahan.bantulkab.go.id/>.

Pengembangan aplikasi GIS bisa memanfaatkan perangkat lunak berbayar (misal ArcView, ArcGIS, Global Mapper, MapInfo, ErMapper, Geomedia, Autocadmap) maupun yang gratis (misal Grass, QuantumGIS, MapWindow) ((Suyono, 2006) (Manik, 2009) (<https://www.technogis.co.id/>, 10 Sept 2017)). Terlepas dari perangkat lunak yang digunakan, inti dari sebuah perangkat lunak SIG adalah harus mampu menyediakan fungsi-fungsi untuk pemasukan, pembetulan, penyimpanan, dan transformasi data; pengaturan; *link*; analisa data geografis, serta juga interaksi dengan pengguna (input *query*) (<http://www.geologinesia.com/>, 10 Sept 2017).

Dalam pendekatan baru sistem GIS, organisasi data atribut dikembangkan menjadi struktur *database*. Dengan demikian, informasi yang ditampilkan dalam GIS adalah bersumber dari data yang tersimpan di dalam *database*. Berdasarkan studi pendahuluan yang dilakukan

dalam penelitian ini, diketahui aplikasi-aplikasi GIS yang ada saat ini umumnya dikembangkan untuk tujuan dan/atau objek yang bersifat spesifik, dan menggunakan desain *database* yang spesifik pula. Dalam kondisi ini, maka aplikasi GIS yang telah dikembangkan akan relatif sulit disesuaikan jika terjadi perkembangan data. Perkembangan data tersebut hanya bisa dilakukan dengan cara “membongkar” kembali desain *database* dan kemudian mengubahnya sesuai perkembangan data yang dibutuhkan. Inisiatif untuk mengatasi masalah ini telah dilakukan pada Nurnawati et.al. (2016) dengan mengintegrasikan desain *database*. Berdasarkan hasil penelitian tersebut, makalah ini memberikan sebuah kerangka desain *database* yang skalabel untuk mengantisipasi perkembangan data pada berbagai aplikasi berbasis GIS.

## 2. METODE PENELITIAN

Penelitian ini dilaksanakan dalam 3 tahapan, yaitu studi pendahuluan untuk memperoleh gambaran tentang konsep, penerapan, dan desain *database* pada berbagai aplikasi GIS yang selama ini digunakan. Selanjutnya, diidentifikasi potensi permasalahan pada desain *database* yang ada, dan kemudian diusulkan alternatif solusi desain *database* yang bersifat skalabel, dapat mengakomodasi perkembangan data dalam *database* pada GIS, dan dapat diterapkan pada berbagai aplikasi berbasis GIS. Desain *database* dilakukan dengan perangkat lunak MySQL.

## 3. HASIL DAN PEMBAHASAN

### Kebutuhan Data Dasar pada Aplikasi GIS

Pengembangan aplikasi berbasis GIS pada dasarnya memerlukan data yang dapat dikelompokkan sebagai data pengguna, data objek yang akan ditampilkan dan lokasi dimana objek tersebut berada. Data pengguna umumnya menyimpan data pengelola aplikasi GIS, di dalamnya memuat item data *username* dan *password*, ada juga yang dilengkapi dengan *email*, *no\_hp*, *status*, *level*, dan atribut-atribut pendukung lainnya. Data objek umumnya menyimpan data objek-objek yang akan ditampilkan kepada dalam aplikasi GIS, di dalamnya memuat item data kode, nama,

alamat, kontak, keterangan, dan lokasi objek yang terdiri atas posisi garis lintang (*latitude*) dan bujur (*longitude*), ada juga yang dilengkapi dengan foto, video, dan atribut-atribut lainnya sebagai pelengkap informasi pada GIS. Kebutuhan data dasar tersebut masih bisa ditambahkan dengan atribut lain yang relevan dengan tujuan aplikasi GIS yang dikembangkan.

Pada data pengguna *username* merupakan atribut yang dipilih sebagai kunci utama (*primary key*), sedangkan pada data bisa menggunakan *kode\_objek*. Kedua atribut tersebut lazimnya bersifat unik (*unique*) dan tidak boleh kosong (*not null*) pada setiap kemunculan *record* dalam tabel di *database*. Selanjutnya, untuk menghubungkan antara data pengguna dan objek, bisa dicatat data pengguna yang bertanggungjawab atau mengisi data objek. Untuk keperluan ini bisa digunakan atribut *username* sebagai kunci penghubung (*foreign key*) yang ditempatkan pada data objek.

### Desain Database yang Umum Digunakan pada Aplikasi GIS

Berdasarkan kebutuhan data dasar pada aplikasi GIS, pengembang aplikasi menyusun *database* menggunakan *software* DBMS yang mendukung untuk pengembangan aplikasi GIS. Mengacu pada penjelasan sebelumnya, dapat disusun desain minimal *database* untuk sebuah aplikasi GIS, misal diberi nama *db\_gis1* dan ditampilkan dalam format *sql*, yaitu sebagai berikut:

```
-----
-- Database: db_gis1
-----
-- Table structure for table objek
CREATE TABLE IF NOT EXISTS objek (
  kode char(10) NOT NULL DEFAULT,
  nama varchar(50) NOT NULL,
  alamat varchar(50) NOT NULL,
  kontak varchar(50) NOT NULL,
  keterangan varchar(100) NOT NULL,
  latitude int(11) NOT NULL,
  longitude int(11) NOT NULL,
  username varchar(10) NOT NULL,
  PRIMARY KEY (kode)),
```

```
FOREIGN KEY (username) REFERENCES pengguna
(username),
```

```
-----
-- Table structure for table pengguna
CREATE TABLE IF NOT EXISTS pengguna (
  username varchar(10) NOT NULL DEFAULT,
  password varchar(10) NOT,
  email varchar(50) NOT NULL,
  no_hp varchar(12) NOT NULL,
  status char(1) NOT NULL,
  level char(1) NOT NULL,
  PRIMARY KEY (username));
-----
```

Sesuai kebutuhan riil item data dalam aplikasi GIS, pemilihan jenis tipe data, ukuran, dan properti lainnya bisa berbeda dengan desain *database* di atas, tergantung pada DBMS yang digunakan atau detail kebutuhan penyimpanan data. Desain *database* tersebut jika ditampilkan dalam bentuk diagram kerelasian antar relasi tampak seperti Gambar 1.



Gambar 1. Diagram kerelasian antar relasi desain *db\_gis1*

### Permasalahan pada Desain Database

Desain *db\_gis1*, jika dicermati lebih lanjut akan ditemukan beberapa masalah yang masih mungkin terjadi, baik pada data pengguna maupun pada data objek. Potensi masalah dimaksud antara lain sebagai berikut:

1. Data pengguna
  - a. Data pengguna hanya dapat bertambah pada jumlah *record*
  - b. Data status pengguna tidak bisa berkembang kecuali diubah secara manual pada *status* dalam *database*
  - c. Data level pengguna tidak bisa berkembang kecuali diubah secara manual pada *level* dalam *database*.

2. Data objek
  - a. Data objek hanya dapat bertambah pada jumlah *record*
  - b. Data kode\_objek merepresentasikan satu jenis objek, ketika terjadi perubahan dan/atau penambahan objek baru sulit ditambahkan.
  - c. Data alamat objek sulit dikelola lebih lanjut karena disimpan sebagai item, ketika terjadi perubahan alamat maka hanya bisa diubah secara manual.
  - d. Data objek mungkin dienti oleh operator pada level tertentu dan masih perlu proses verifikasi oleh operator atau administrator, sehingga diperlukan adanya tambahan status, misal “aktif” atau “tidak aktif”, dan “belum aktif”.

### Solusi Desain *Database* pada Aplikasi GIS

Alternatif solusi desain *database* yang diusulkan di atas, secara umum dapat dilakukan dengan cara berikut;

1. meningkatkan level pandangan perancang *database* ke posisi *conceptual view* sehingga hasil desain *database* tidak hanya spesifik untuk satu aplikasi GIS, tetapi bisa digunakan oleh beberapa/lebih banyak aplikasi, dan
2. memperluas pandangan perancang *database* sehingga hasil desain *database* tidak hanya spesifik untuk satu aplikasi GIS, tetapi bisa digunakan untuk skala yang lebih luas dalam aspek pengguna, objek data, dan wilayah.

Contoh-contoh permasalahan riil yang dapat terjadi dan alternatif solusi yang dapat diterapkan pada desain *database* untuk berbagai aplikasi GIS adalah sebagai berikut:

1. Jika data status pengguna hanya berisi “aktif” atau “tidak aktif”, dan kemudian dibutuhkan perubahan atau penambahan status baru “belum aktif” sebagai status pengguna baru yang belum diverifikasi, maka perubahan tersebut hanya bisa dilakukan secara manual dan relatif sulit dilakukan.

Agar perubahan ini bisa dilakukan secara lebih mudah, maka diperlukan sebuah tabel referensi “status\_pengguna”, di dalamnya dapat memuat “kode\_status” dan “status”,

sehingga di dalam tabel pengguna hanya perlu dicatat “kode\_status” sedangkan status pengguna bisa diperoleh dengan cara menemukan “status” di dalam tabel “status\_pengguna” dengan kunci penghubung “kode\_status”

2. Jika data level pengguna hanya berisi “administrator” atau “operator”, dan kemudian dibutuhkan perubahan atau penambahan level baru “operator\_propinsi” untuk operator yang bertanggungjawab mengelola data pada tingkat propinsi; “operator\_kabupaten” untuk operator yang bertanggungjawab mengelola data pada tingkat kabupaten, maka perubahan tersebut hanya bisa dilakukan secara manual dan relatif sulit dilakukan.

Agar perubahan ini bisa dilakukan secara lebih mudah, maka diperlukan sebuah tabel referensi “level\_pengguna”, di dalamnya dapat memuat “kode\_level” dan “level\_pengguna”, sehingga di dalam tabel pengguna hanya perlu dicatat “kode\_level” sedangkan data level pengguna bisa diperoleh dengan cara menemukan “level\_pengguna” di dalam tabel “level\_pengguna” dengan kunci penghubung “kode\_level”.

3. Data kode\_objek merepresentasikan satu jenis objek, ketika terjadi perubahan dan/atau penambahan objek baru sulit ditambahkan.

Jika data kode\_objek hanya berisi nilai-nilai kode setiap objek tanpa ada pengelompokan tertentu, maka apabila dibutuhkan perubahan berupa pengelompokan objek berdasarkan kategori objek dan/atau sub kategori objek, maka perubahan tersebut hanya bisa dilakukan secara manual dan relatif sulit dilakukan.

Agar perubahan ini bisa dilakukan secara lebih mudah, maka diperlukan tabel referensi “kategori” di dalamnya dapat memuat “kode\_kategori” dan “kategori”, sehingga di dalam tabel objek perlu ditambahkan “kode\_kategori” sedangkan data kategori objek bisa diperoleh dengan cara menemukan “kategori\_objek” di dalam tabel “kategori” dengan kunci penghubung “kode\_kategori”.

Jika objek pada kategori tertentu juga perlu dikelompokkan ke dalam sub kategori tertentu, maka bisa ditambahkan tabel referensi "sub\_kategori", di dalamnya dapat memuat "kode\_sub\_kategori" dan "sub\_kategori", sehingga di dalam tabel kategori hanya perlu dicatat "kode\_sub\_kategori" sedangkan data sub kategori objek bisa diperoleh dengan cara menemukan "kategori\_objek" di dalam tabel "sub\_kategori" dengan kunci penghubung "kode\_sub\_kategori".

4. Jika data alamat hanya berisi item data alamat setiap objek tanpa ada pengkodean propinsi, kabupaten, kecamatan, dan desa, maka perubahan alamat objek hanya bisa dilakukan secara manual dan relatif sulit dilakukan.

Agar perubahan ini bisa dilakukan secara lebih mudah, maka diperlukan tabel referensi "propinsi" di dalamnya dapat memuat "kode\_propinsi" dan "propinsi", sehingga di dalam tabel objek perlu ditambahkan "kode\_propinsi" sedangkan alamat propinsi objek bisa diperoleh dengan cara menemukan "propinsi" di dalam tabel "propinsi" dengan kunci penghubung "kode\_propinsi".

Dengan cara yang sama, maka diperlukan tambahan tabel referensi "kabupaten", "kecamatan". Selanjutnya alamat dalam tabel objek cukup mencatat alamat (bisa nama jalan dan nomor, atau nama dusun, berserta RT dan RW) saja, sedangkan alamat propinsi, kabupaten, kecamatan, dan desa objek bisa dilacak ke dalam tabel "propinsi", "kabupaten", "kecamatan", serta "desa".

Dalam hal ini tabel "objek" berrelasi dengan tabel "propinsi" menggunakan "kode\_propinsi", tabel "propinsi" berrelasi dengan tabel "kabupaten" menggunakan "kode\_kabupaten", tabel "kabupaten" berrelasi dengan tabel "kecamatan" menggunakan "kode\_kabupaten", tabel "kecamatan" berrelasi dengan tabel "desa" menggunakan "kode\_kecamatan". Dan akhirnya, di dalam tabel "objek" perlu ditambahkan kolom "kode\_desa" untuk mengabungkan dengan tabel "desa".

Penting dicatat, bahwa kode-kode data wilayah yang digunakan pada kode\_propinsi, kode\_kabupaten, kode\_kecamatan, dan kode\_desa tersebut seharusnya mengacu pada kode standar yang sudah ada dan ditetapkan secara formal oleh Pemerintah Republik Indonesia. Aturan pengkodean dimaksud terdapat pada Permendagri No. 66 Tahun 2011, tentang kode dan data wilayah dalam Sistem Informasi Administrasi Kependudukan (SIAK) yang dikembangkan oleh Kemendagri dan berlaku secara nasional (Kemendagri, 2011). Di dalam peraturan tersebut, kode data wilayah di Indonesia didefinisikan sebagai berikut:

- kode propinsi char[2]
- kode kabupaten/kota char[4]
- kode kecamatan char[6]
- Kode desa char[10]

Contoh item kode wilayah dalam SIAK yang dapat diterapkan dalam *database* untuk aplikasi-aplikasi GIS adalah sebagai berikut:

- "34" = DIY
- "3401" = Kabupaten Kulon Progo
- "340106" = Kecamatan Sentolo
- "34010620008" = Desa Banguncipto

Saat ini, terdapat 34 item data nama propinsi, 423 item data kabupaten, dan 2.879 item data kecamatan, serta 82.030 desa yang bisa diterapkan pada berbagai aplikasi yang memerlukan/melibatkan informasi lokasi/wilayah/alamat di Indonesia.

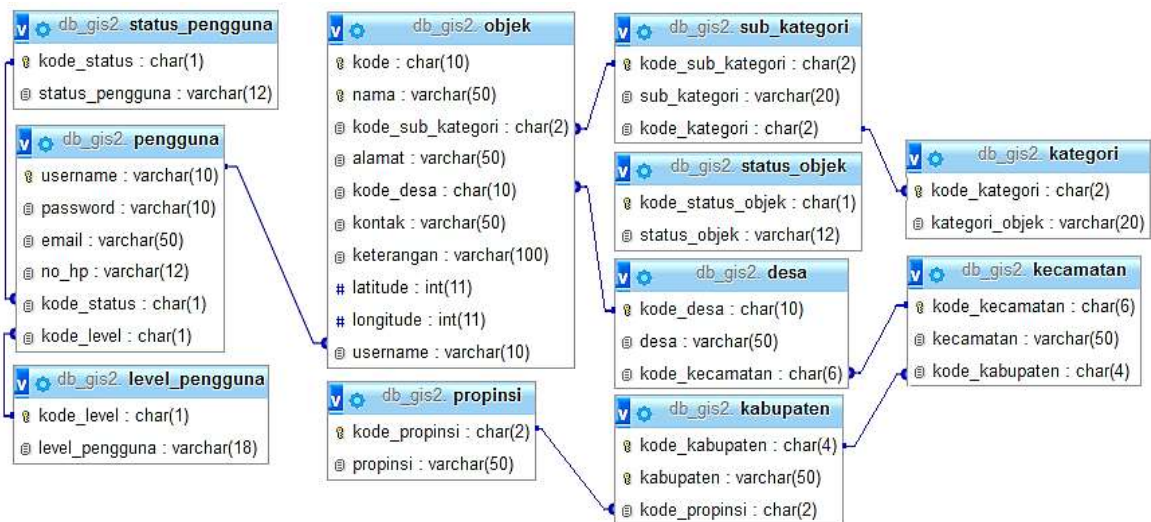
5. Data objek mungkin dienti oleh operator pada level tertentu dan masih perlu proses verifikasi oleh operator level di atasnya atau administrator, sehingga diperlukan adanya tambahan status, misal "aktif" atau "tidak aktif", dan "belum aktif". Solusi untuk kebutuhan ini dapat diberikan tambahan atribut status pada objek.

Mengacu pada deskripsi desain datanase yang diusulkan di atas, dapat disusun desain minimal *database* untuk aplikasi GIS, misal diberi nama *db\_gis2*, jika ditampilkan dalam bentuk diagram kerelasian antar relasi tampak seperti Gambar 2, sedangkan dalam format `sql`, yaitu sebagai berikut:

```

-----
-- Database: db_gis2
-----
-- Table structure for table desa
CREATE TABLE IF NOT EXISTS desa (
  kode_desa char(10) NOT NULL,
  desa varchar(50) NOT NULL,
  kode_kecamatan char(6) NOT NULL,
  PRIMARY KEY (kode), FOREIGN KEY
(kode_kecamatan) REFERENCES kecamatan
(kode_kecamatan)),
-----
-- Table structure for table kecamatan
CREATE TABLE IF NOT EXISTS kecamatan (
  kode_kecamatan char(6) NOT NULL,
  kecamatan varchar(50) NOT NULL,
  kode_kabupaten char(4) NOT NULL,
  PRIMARY KEY (kode_kecamatan), FOREIGN KEY
(kode_kabupaten) REFERENCES kabupaten (kode_
kabupaten)),
-----
-- Table structure for table kabupaten
CREATE TABLE IF NOT EXISTS kabupaten (
  kode_kabupaten char(4) NOT NULL,
  kabupaten varchar(50) NOT NULL,
  kode_propinsi char(2) NOT NULL,
  PRIMARY KEY (kode_kabupaten), FOREIGN KEY
(kode_propinsi) REFERENCES propinsi
(kode_propinsi)),
-----
-- Table structure for table propinsi
CREATE TABLE IF NOT EXISTS propinsi (
  kode_propinsi char(2) NOT NULL,
  propinsi varchar(50) NOT NULL,
  PRIMARY KEY (kode_propinsi)),
-----
-- Table structure for table sub_kategori
CREATE TABLE IF NOT EXISTS sub_kategori (
  kode_sub_kategori char(2) NOT NULL,
  sub_kategori varchar(20) NOT NULL,
  kode_kategori char(2) NOT NULL
  PRIMARY KEY (kode_sub_kategori), FOREIGN KEY
(kode_kategori) REFERENCES kategori (kode_
kategori)),
-----
-- Table structure for table kategori
CREATE TABLE IF NOT EXISTS kategori (
  kode_kategori char(2) NOT NULL,
  kategori_objek varchar(20) NOT NULL,
  PRIMARY KEY (kode_kategori)),
-----
-- Table structure for table pengguna
CREATE TABLE IF NOT EXISTS pengguna (
  username varchar(10) NOT NULL DEFAULT ''
  COMMENT 'nama user',
  password varchar(10) NOT NULL COMMENT
'password',
  email varchar(50) NOT NULL,
  no_hp varchar(12) NOT NULL COMMENT 'no hp',
  kode_status char(1) NOT NULL COMMENT 'status
aktif-tidak aktif',
  kode_level char(1) NOT NULL,
  PRIMARY KEY (username), FOREIGN KEY
(kode_status) REFERENCES status_pengguna
(kode_status), FOREIGN KEY (kode_level)
REFERENCES level_pengguna (kode_level)),
-----
-- Table structure for table level_pengguna
CREATE TABLE IF NOT EXISTS level_pengguna (
  kode_level char(1) NOT NULL,
  level_pengguna varchar(18) NOT NULL,
  PRIMARY KEY (kode_level)),
-----
-- Table structure for table status_pengguna
CREATE TABLE IF NOT EXISTS status_pengguna (
  kode_status char(1) NOT NULL,
  status_pengguna varchar(12) NOT NULL,
  PRIMARY KEY (kode_status)),
-----
-- Table structure for table objek
CREATE TABLE IF NOT EXISTS objek (
  Kode_objek char(10) NOT NULL DEFAULT '',
  nama varchar(50) NOT NULL,
  kode_sub_kategori char(4) NOT NULL,
  alamat varchar(50) NOT NULL,
  kode_desa char(10) NOT NULL,
  kontak varchar(50) NOT NULL,
  keterangan varchar(100) NOT NULL,
  latitude int(11) NOT NULL,
  longitude int(11) NOT NULL,
  username varchar(10) NOT NULL,
  PRIMARY KEY (kode_objek), FOREIGN KEY
(kode_sub_kategori) REFERENCES sub_kategori
(kode_sub_kategori), FOREIGN KEY (kode_desa)
REFERENCES desa (kode_desa), FOREIGN KEY
(username) REFERENCES pengguna (username)),
-----
-- Table structure for table status_objek
CREATE TABLE IF NOT EXISTS status_objek (
  kode_status_objek char(1) NOT NULL,
  status_objek varchar(12) NOT NULL,
  PRIMARY KEY (kode_status_objek)),
-----

```



Gambar 2. Diagram kerelasiaan antar relasi desain db\_gis2

Desain *database* pada *db\_gis2* memungkinkan untuk mengakomodasi pertumbuhan dan perkembangan data dalam *database* dalam aspek pengguna, objek data, dan wilayah. Data pengguna dapat tumbuh dan berkembang dari awalnya hanya satu kategori level yang sudah ditentukan (pada *db\_gis1*) menjadi bisa diperluas dengan adanya tabel referensi “status” dan “level” (pada *db\_gis2*) sehingga bisa diterapkan pada status dan level pengguna yang lebih luas. Data objek dapat tumbuh dan berkembang dari awalnya hanya pada satu kategori objek (pada *db\_gis1*) menjadi bisa diperluas dengan adanya tabel referensi “kategori”, “sub\_kategori”, “propinsi”, “kabupaten”, “kecamatan”, “desa”, serta “status” (pada *db\_gis2*) sehingga bisa diterapkan pada skala aplikasi yang lebih luas. Penambahan tabel-tabel referensi tersebut memungkinkan desain *database* pada *db\_gis2* digunakan untuk berbagai aplikasi berbasis GIS pada kategori dan sub kategori objek yang berbeda-beda. Sebagai contoh, kategori objek bisa terdiri atas “pariwisata”, “pendidikan”, “kantor pemerintah”, “kesehatan”, “budaya”, perumahan”, dan/atau lainnya. Pada kategori tertentu bisa terdiri atas beberapa sub kategori, misal “pariwisata” dapat memuat sub kategori, misal “agrowisata”, “alam”, “budaya”, “museum”, “ziarah”, dan/atau lainnya. Jika diperlukan, pada sub kategori objek masih bisa ditambahkan sub dari

sub kategori objek, misal kategori objek “wisata” pada sub kategori “alam” bisa dikelompokkan lagi menjadi sub sub kategori “pegunungan”, “pantai”, “hutan”, dan/atau lainnya. Dengan demikian, kategori, sub kategori, dan sub sub kategori objek masih bisa ditambahkan/disesuaikan dengan kebutuhan aplikasi GIS yang dibangun.

#### 4. KESIMPULAN

Desain *database* yang skalabel untuk pengembangan berbagai aplikasi berbasis GIS dapat diperoleh dengan cara mengubah pendekatan desain *database* dari yang bersifat spesifik untuk objek di wilayah tertentu, menjadi desain yang lebih umum yaitu dapat diterapkan untuk berbagai macam objek di wilayah yang lebih luas. Hal ini bisa dicapai dengan dua cara sekaligus. Pertama adalah meningkatkan level pandangan perancang *database* ke posisi *conceptual view* sehingga hasil desain *database* bisa digunakan oleh beberapa aplikasi berbeda, dan kedua memperluas pandangan perancang *database* sehingga *database* didesain agar bisa digunakan untuk skala yang lebih luas dalam aspek pengguna, objek/jenis data, dan wilayah.

#### 5. REFERENSI

[1]. Andika, L. A. A. dan Ariandi, M. 2015. Sistem Informasi Geografis Ruang Terbuka Hijau Kawasan Perkotaan (RTHKP) Palembang. *Student Colloquium*

- Sistem Informasi & Teknik Informatika (SCSITI)*. Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Bina Darma Palembang. 21-22 Agustus 2015.
- [2]. Ariandono, Y. Hamzah, A. dan Nurnawati, E.K. 2014. Sistem Informasi Lokasi Industri di Kabupaten Serang Berbasis Geographic Information System (GIS), *Jurnal Script*. ISSN: 2338-6304. 1(2). Juni 2014.
- [3]. Arkiang, W.P. Sutanta, E. dan Nurnawati, E.K. 2014. Aplikasi Sistem Informasi Lokasi Hotel Berbintang di Yogyakarta Berbasis Webgis. *Jurnal Script*. Jurusan Teknik Informatika, IST AKPRIND Yogyakarta. ISSN: 2338-6304. 2(1). Desember 2014.
- [4]. Suratmo, D. Suraya, dan Widyastuti, N. 2014. Membangun Sistem Informasi Geografis Perguruan Tinggi Daerah Istimewa Yogyakarta Berbasis Web, *Jurnal Script*. Jurusan Teknik Informatika IST AKPRIND Yogyakarta. ISSN: 2338-6304. 2(1). Desember 2014.
- [5]. Prabowo, E.T. Sholeh, M. Dan Iswahyudi, C. 2013. Sistem Informasi Geografis Dalam Pencarian Lokasi Museum di Daerah Istimewa Yogyakarta, *Jurnal Script*. Jurusan Teknik Informatika IST AKPRIND Yogyakarta. ISSN: 2338-6304. 1(1). Desember 2013.
- [6]. Pratikto, H. S. Suraya, dan Sutanta, E. 2014. Sistem Pencarian dan Pemesanan Rumah Kos Menggunakan Sistem Informasi Geografi (SIG). *Jurnal Script*. Jurusan Teknik Informatika IST AKPRIND Yogyakarta. ISSN: 2338-6304. 1(2). Juni 2014.
- [7]. Wibowo, H. Lestari, U. Triyono, J. 2014. Sistem Informasi Potensi Industri di Kabupaten Bantul Berbasis Geographic Information System dan Location Based Service. *Jurnal Script*. Jurusan Teknik Informatika IST AKPRIND Yogyakarta. ISSN: 2338-6304. 1(2). Juni 2014.
- [8]. Haryanto, R. dan Sholeh, M. 2014. Sistem Informasi Berbasis Geografi untuk Pendukung Promosi Wisata Budaya di Kota Yogyakarta. *Jurnal Jarlit*, Bappeda Kota Yogyakarta. ISSN: 1978-0052. Vol 10: 64-75. Juli 2014
- [9]. Hege, Y.B.L. Lestari, U., dan Nurnawati, E.K.. 2014. Sistem Informasi Geografis (SIG) Pelayanan Kesehatan di Kotamadya Yogyakarta Berbasis Web. *Jurnal Script*. ISSN: 2338-6304. 1(2). Juni 2014.
- [10]. Ardianto, J. Lestari, U. Dan Triyono, J. 2016. Sistem Informasi Geografis Pemetaan Area Persawahan Desa Gantung Kabupaten Belitung Timur. *Jurnal Script*. Jurusan Teknik Informatika IST AKPRIND Yogyakarta. ISSN: 2338-6304, 3(2). Juni 2016.
- [11]. Manik, L. 2009. *Seputar Sistem Informasi Geografis, Educational Geographic*. <http://www.sig.depdiknas.go.id/>, diakses: 01 September 2009.
- [12]. Mildawani, I. dan Susilowati, D. 2012. Aplikasi Sistem Informasi Geografis (SIG) dalam Analisis Pemanfaatan dan Pengelolaan Ruang Terbuka Hijau Kota (RTHK) Studi Kasus: Kota Depok.
- [13]. Nurnawati, E.K. 2014. Aplikasi Mobile Berbasis Lokasi untuk Penyedia Lokasi Layanan Kesehatan di Yogyakarta. *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST) 2014*. IST AKPRIND Yogyakarta. ISSN: 1979-91x. eISSN: 2541-528x. 15 November 2014, Hal. 369-376.
- [14]. Nurnawati, E.K. Ermawati, dan Ardyrusmaryya, D. 2016. Pemanfaatan Basis Data Terintegrasi pada Sistem Informasi Perangkat Bergerak, *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST) 2016*. IST AKPRIND Yogyakarta. ISSN: 1979-91x, eISSN: 2541-528x. 26 November 2016. Hal. 369-376.
- [15]. Kusumaningsih, R. Y. R. Kurniawan, T. dan Pradhityo, S. 2016. Aplikasi Pengembangan Sistem Informasi Bangunan Cagar Budaya di Kota Yogyakarta Berbasis Web Mobile dan Location Based Service. *Jurnal Jarlit*. Bappeda Kota Yogyakarta. ISSN: 1978-0052. Vol. 12. Hal. 53-63. Juli 2016.
- [16]. Sholeh, M. Iswahyudi, C. dan Prabowo, E.T. 2014. E-Museum: Informasi Museum



- di Yogyakarta Berbasis Location Based Service, *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST) 2014*. IST AKPRIND Yogyakarta. Hal. A-51-A-58.
- [17].Suraya dan Sholeh, M. 2015. e-Museum Sebagai Media Memperkenalkan Cagar Budaya di Kalangan Masyarakat. *Jurnal Jarlit*. Bappeda Kota Yogyakarta. ISSN: 1978-0052. Vol. 11. Hal. 24-32. Juli 2015.
- [18].Sutanta, E. 2010. Membangun SIG Perguruan Tinggi DIY dengan CMS Joomla dan Google Map. *Jurnal Information Management and Technology (IMATEC)*. STMIK Denpasar. ISSN: 2087-5312. 1(1): 37-41.
- [19].Suyono, S. 2006. *Geokomputasi dan SIG*, IST AKPRIND Yogyakarta. Yogyakarta.
- [20].Setiadi, T. dan Yuliansyah, H. 2015. Kampung Wisata Online Berbasis SIG sebagai Upaya Meningkatkan Partisipasi Warga dalam Mengelola dan Mempromosikan Pariwisata Kota Yogyakarta. *Jurnal Jarlit*. Bappeda Kota Yogyakarta. ISSN: 1978-0052. Vol. 11. Hal. 85-91. Juli 2015.
- [21].Suharto, U. H. Triyono, J. dan Sutanta. E. 2014. Aplikasi Pencarian Lokasi Agen TIKI (Titipan Kilat) di Yogyakarta untuk Android. *Skripsi*. Teknik Informatika, IST AKPRIND, Yogyakarta.
- [22].Sari, W.E. Sholeh, M. dan Hamzah, A. 2013. Penerapan JQuery Mobile dan PHP Data Object pada Aplikasi Pencarian Lokasi Tempat Ibadah di Yogyakarta. *Jurnal Script*. Jurusan Teknik Informatika IST AKPRIND Yogyakarta. ISSN: 2338-6304. 1(1). Desember 2013.
- [23].Zuliyanto, Hamzah, A. dan Suraya. 2014. Sistem Informasi Lokasi Wisata Candi di DIY Berbasis Geographic Information System (GIS). *Jurnal Script*. Jurusan Teknik Informatika IST AKPRIND Yogyakarta. ISSN: 2338-6304. 1(2). Juni 2014.
- [24].Kemendagri RI. 2011. *Permendagri No. 66 Tahun 2011 tentang Kode dan Data Wilayah dalam Sistem Informasi Administrasi Kependudukan (SIK)*. Jakarta.
- Laman Internet:  
 [25].<http://gis.kominfo.go.id/>, 15 Sept 2017.  
 [26].<http://gis.depkes.go.id/>, 15 Sept 2017.  
 [27].<http://gis.pusair-pu.go.id/>, 15 Sept 2017.