

2. DASAR TEORI

2.1. Sistem Informasi Geografis

2.1.1. Pengenalan Sistem Informasi Geografis

Sistem Informasi Geografis (Geographic Information System) adalah sebuah sistem berbasis komputer yang digunakan untuk menyimpan, memanipulasi, dan menganalisis informasi geografis dan data yang menyertainya.

Peta merupakan komponen utama dalam pembuatan Sistem Informasi Geografis. Tujuan umum dari peta adalah menggambarkan topografi dari tanah, batas wilayah administratif beserta unit administratif (jalan, sungai, gunung). Dengan berkembangnya penelitian tentang sumber daya alam, maka peta tematik digunakan sebagai media untuk penyampaian informasi tentang *spatial distribution* seperti geologi, geomorfologi, lapisan tanah, vegetasi.

Sebelum komputer tersedia secara luas, data geografi dilambangkan dengan titik, garis, dan area pada sebuah kertas atau film. Ketiga komponen (titik, garis, dan area) tersebut dilambangkan dengan menggunakan simbol, tekstur, dan warna yang dijelaskan dalam legenda peta. Untuk memuat informasi dalam skala kecil, peta konvensional relatif mudah dibaca, tetapi jika informasi yang ditampilkan dalam skala besar, maka pengguna akan kesulitan dalam menterjemahkan informasi yang disampaikan.

Dikarenakan adanya banyak keterbatasan yang ada dengan menggunakan peta konvensional, maka pada tahun 1970an para ilmuwan mulai menciptakan Sistem Informasi geografis. Hal ini dapat dilakukan dikarenakan komputer sudah menjadi produk massal yang mudah didapat. Dengan menggunakan Sistem Informasi Geografis maka biaya untuk melakukan pencetakan, *update* dapat diminimalisir.

4 kemampuan yang dimiliki oleh Sistem Informasi Geografis yang berdasarkan komputer adalah *Input, Data Management, Manipulation* dan *Analysis, Output*. Dengan menggunakan 4 kemampuan tersebut, SIG dapat mengintegrasikan *data layer* untuk menampilkan *real data* dalam cara dan perpestif yang berbeda.

Sistem Informasi Geografis digunakan karena SIG merupakan alat yang ampuh untuk menangani *spatial data*. Dalam Sistem Informasi Geografis data di kelola dalam format digital. Sehingga data menjadi lebih kecil tempat penyimpanannya daripada dalam bentuk peta kertas, tabel, atau bentuk konvensional lainnya. Data dalam jumlah besar juga bisa di kelola dan di manipulasi dengan kecepatan yang tinggi dan biaya yang murah untuk per unitnya.

2.1.2. Data

2.1.2.1. Input

Data Input yang digunakan pada SIG yang terpenting adalah peta digital. Peta digital ini nantinya akan digabungkan dengan informasi dari data yang lain dan akan dilakukan proses analisa untuk memberikan suatu output atau informasi.

Data input pada SIG ini secara garis besar dibagi menjadi dua jenis, yaitu :

a. *Spatial Data*

Data yang menggambarkan fitur-fitur dari lokasi geografis. Misalnya, titik, garis, polyogon biasanya digunakan untuk menggambarkan fitur geografis seperti jalan, sungai, danau, dan lain-lain.

b. *Non Spatial Attribute Data*

Data yang menggambarkan informasi seperti nama jalan, salinitas dari sungai, komposisi tanah suatu dataran tertentu.

Peta pada umumnya menggambarkan topografi suatu daerah ataupun batas-batas (administratif) suatu wilayah atau negara, juga dapat digunakan untuk menggambarkan distribusi keruangan (*spatial distribution*) yang disebut sebagai peta tematik (*thematic*) yaitu yang menampilkan keadaan geologi, geomorfologi, tanah, vegetasi atau sumber daya alam. Namun demikian, suatu peta juga dapat menggambarkan distribusi sosial ekonomi suatu masyarakat, seperti peta desa tertinggal, peta kependudukan, dan sebagainya.

Peta yang digunakan dalam SIG adalah peta digital, peta digital dibuat dari peta konvensional yang kemudian diubah dengan alat bantu untuk dimasukkan kedalam komputer dalam bentuk *vector file*. Alat bantu yang sering digunakan adalah digitizer, atau untuk lebih sederhananya menggunakan mouse. Hasil dari proses digitasi tersebut dibagi menjadi 3 kelompok yaitu, titik (*point*), garis (*line*),

kurva tertutup (*polygon*) yang ketiganya akan dihubungkan dengan *database* untuk menampung informasi tentang obyek tersebut.

Atribut dari peta tersebut akan diberi kode untuk mengidentifikasi atribut tersebut. Misalnya untuk Kota A diberi ID 10, Kota B diberi ID 11. Untuk atribut anggota dari Kota A seperti Jalan, Museum diberi kode tersendiri, misalnya Jalan A1 di Kota A diberi ID 1001.

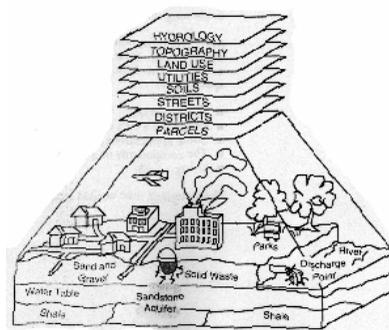
2.1.2.2. Output

Hasil *output* dari SIG ini adalah sebuah peta konvensional yang berisi informasi dari hasil analisis yang telah dilakukan. *Output* yang dihasilkan dapat berupa *hardcopy* maupun *softcopy*.

Yang dimaksud dengan *hardcopy*, hasil dari analisa SIG dicetak ke dalam selembar kertas. Sedangkan untuk *softcopy* hasil dari analisa ditransfer ke dalam *image file*.

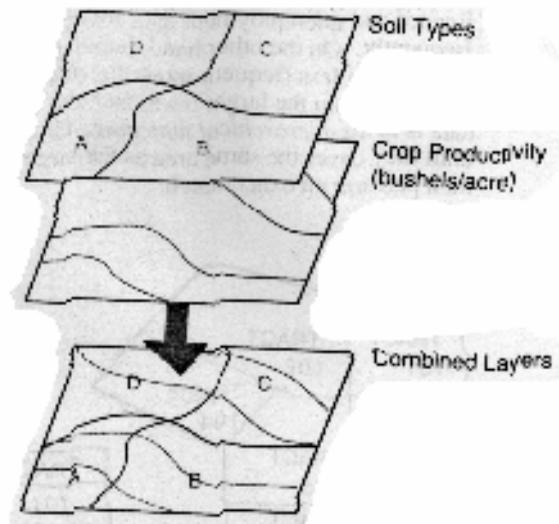
2.1.2.3. Teknik Penyimpanan

Dalam SIG teknik penyimpanan data geografis dalam dilakukan dengan sistem *layer*. Misalnya untuk menyimpan informasi tentang keadaan tanah disimpan di *layer* tersendiri, kemudian untuk produktivitas lahan disimpan pada *layer* tersendiri. Kemudian data dari tiap *layer* tersebut dapat digabungkan untuk memperoleh informasi yang lebih jelas tentang kondisi keadaan tanah yang bagaimana yang dapat meningkatkan produktivitas lahan.



Gambar 2.1. Sistem Layer Pada SIG

Sumber : Understanding GIS The Arc/Info Method



Gambar 2.2. Penggabungan Layer Pada SIG

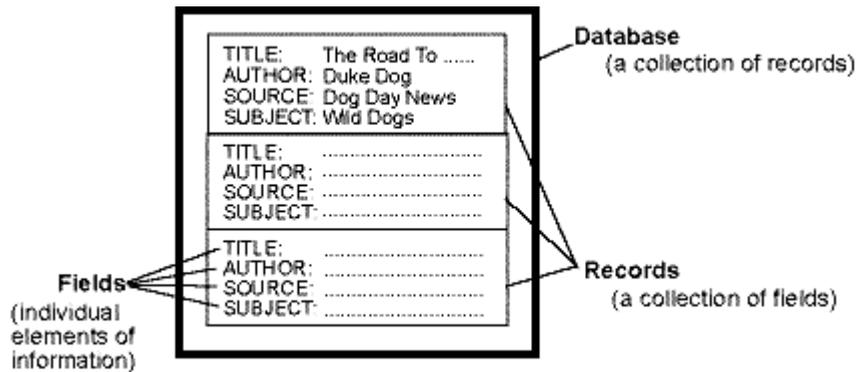
Sumber : Understanding GIS The Arc/Info Method

Dengan adanya sistem *layer* pada SIG, penggabungan peta-peta yang memuat informasi yang berbeda akan membuat *user* menjadi lebih mudah dalam menerima informasi yang diberikan. Seperti pada Gambar 2.1.2 dengan menggabungkan *layer Soil Types* dan *layer Crop Productivity*, maka *user* akan lebih mudah menerima informasi bahwa tipe tanah yang bagaimana yang akan membuat hasil panen menjadi banyak

2.2. Database

2.2.1. Definisi Database

Definisi dari *database* ada beberapa macam, secara garis besar dapat dirumuskan bahwa definisi dari *database* adalah sekumpulan data terorganisasi yang disimpan secara terstruktur untuk tujuan tertentu.



Gambar 2.3. Diagram Struktur Konseptual Database

Sumber : Humboldt State University Library

2.2.2. Penyimpanan Data

Database pada dasarnya terdiri dari beberapa *datasheet*. *Datasheet* sekilas terlihat seperti *spreadsheet* yang terdiri dari kolom dan baris yang disebut *field* dan *record*. Setiap *record* terdiri dari elemen penting informasi untuk item tertentu. Sedangkan *field* adalah bagian dari elemen penting yang membentuk sebuah *record*.

2.2.3. Relational Database

Relational Database adalah sebuah *database* yang terdiri dari data yang tersimpan di dalam tabel dan mempunyai hubungan antara satu tabel dengan yang lain melalui sebuah *field*.

Di tahun 1970 Dr E.F. Codd mengemukakan sebuah model tentang *relational database* pada tulisannya yang berjudul “A Relational Model of Data for Large Shared Data Banks”. Pada tulisan tersebut Dr. Codd menyebutkan 12 aturan untuk *relational database* yang diberi nama *Codd’s Rules*.

Kedua belas aturan tersebut adalah

- a. Data disimpan dalam bentuk tabel.
- b. Data dapat diakses secara logik.
- c. *Null* diinterpretasikan sebagai nilai yang tidak diketahui (*unknown value*).
- d. Database menjelaskan tentang dirinya sendiri.
- e. Hanya satu bahasa yang digunakan untuk berkomunikasi dengan DBMS.
- f. Menyediakan alternatif untuk melihat data.
- g. Mendukung operasi relasional.

- h. Data fisik berdiri sendiri.
- i. Data logik berdiri sendiri.
- j. Integritas data merupakan fungsi dari DBMS.
- k. Mendukung operasi terdistribusi.
- l. Integritas data tidak dapat ditembus.

2.2.4. Data Integrity

Data integrity meliputi *consistency*, *accuracy*, dan *correctness* dari data yang tersimpan di *database*. *Data integrity* tidak meliputi keamanan fisik, toleransi kesalahan, dan perlindungan data (*backups*). *Data integrity* tugasnya adalah memastikan tidak memasukkan sampah kedalam, atau dalam bahasa inggris ***data integrity is about keeping the garbage out.***

Ada 4 tipe utama *data integrity* mereka adalah *entity*, *domain*, *referential*, dan *user-defined*. Secara umum *entity integrity* dilakukan pada level baris (*row level*), *domain integrity* dilakukan pada level kolom (*column level*), *referential integrity* dilakukan pada level tabel (*table level*).

Data integrity dapat dijalankan pada *database server* atau pada aplikasi luar (*external application*). Misalnya pada pemrograman javascript digunakan untuk melakukan pengecekan di HTML sebelum data di masukkan pada *database server*. Ini adalah salah satu contoh untuk pelaksanaan *data integrity* pada aplikasi. Untuk *data integrity* yang dilaksanakan pada *database server* contohnya adalah pada SQL Server 7, disini SQL Server mengizinkan untuk melakukan pengecekan secara tersentralisir pada *database server* sehingga aplikasi yang memakainya tidak perlu memikirkan lagi masalah *data integrity*.

2.2.4.1. Entity Integrity

Dilakukan untuk meyakinkan bahwa setiap baris (*record*) di dalam tabel bersifat unik, atau dengan kata lain untuk memastikan bahwa tidak ada data yang kembar. Contohnya untuk *customer* harus memiliki kode ID yang berbeda.

Pada *entity integrity* ini sering kali digunakan *primary key* sebagai *unique value* pada kolom tertentu. Jadi setiap tabel harus mempunyai PK untuk memastikan tidak ada data yang kembar.

Primary key ini tidak mengizinkan adanya *NULL* pada saat mengisi data. Dalam satu tabel hanya dapat digunakan satu *primary key*. Tetapi cara kerja *primary key* dapat dilakukan pada kolom yang lain. Misalnya untuk memastikan bahwa tidak ada ID yang kembar dan nama yang kembar. Penggunaan metode *primary key* ini pada kolom yang lain disebut juga dengan *Alternative key*.

2.2.4.2. Domain Integrity

Kumpulan data yang berada pada ruang tertentu (*domain*) untuk bisa dikatakan *valid*. Atau dengan kata lain *domain integrity* mendefinisikan nilai yang boleh dimasukkan untuk kolom tertentu dengan batasan dari tipe data, format, *range* nilai yang boleh dimasukkan.

Domain dalam *database* mengacu pada nilai yang diizinkan pada kolom, bukan seperti pada Internet atau Windows NT. Contohnya untuk kolom umur, nilai yang diizinkan antara 0 sampai 100, tidak boleh *null*. *Domain* juga bisa diartikan sebagai nilai *default* untuk kolom atau *rule* untuk nilai yang dapat dimasukkan pada kolom.

2.2.4.3. Referential Integrity

Tugasnya untuk menjaga sinkronisasi relasi antar tabel yang berhubungan. Untuk melakukan tugasnya tersebut *Referential integrity* membutuhkan *Primary key* dan *Foreign key* sebagai pedoman. *Foreign key* adalah turunan *Primary key* dari tabel lain yang mempunyai hubungan dengan tabel tersebut.

Misalnya ada dua buah tabel “order” dan “customer”, masing-masing tabel mempunyai PK. Untuk “order” PK-nya “Order_ID”, dan untuk “customer” PK-nya “Customer_ID”. Pada tabel “order” terdapat FK yang bernama “Order_ID” yang merupakan salinan dari PK tabel “customer”. Untuk memelihara relasi keduanya, seharusnya FK pada “order” hanya dapat menerima nilai yang terdapat pada PK “customer”.

Tujuan utama dari adanya *referential integrity* adalah untuk menjaga agar data yang terdapat pada *child table* dapat selalu diakses oleh *parent table*. *Referential integrity* ini dijalankan terutama pada saat *INSERT*, *UPDATE*, *DELETE* pada

database server. Karena pada ketiga *command* diatas terjadi perubahan pada data di dalam tabel.

Meskipun PK-FK *constraint* sering digunakan dalam *referential integrity* tetapi ada cara lain yang dapat digunakan untuk menjaga relasi antar tabel, yaitu dengan memasukkan perintah *trigger* dan *stored procedure* pada tabel.

2.2.4.4. User-Defined Integrity

User-defined integrity dilakukan untuk aturan khusus lainnya yang tidak terdapat pada *integrity rule* yang telah ada. Biasanya aturan ini dilakukan melalui penggunaan *trigger* dan *stored procedure*.

2.2.4.5. Implementing Data Integrity

Dalam menggunakan *data integrity* dalam dilakukan dengan dua pendekatan. Yang pertama *declarative data integrity* dan berikutnya *procedural data integrity*.

Pada *declarative data integrity* menggunakan *constraint*, yang dideklarasikan pada kolom tertentu. *Constraint* dapat dimasukkan pada kolom atau sekumpulan kolom pada tabel yang meliputi *Primary Key*, *unique*, *Foreign Key*, *Check*, dan *default*.

Declarative data integrity mendukung proses *domain*, *entity*, dan *referential integrity*. Dan dideklarasikan ketika *create* atau *alter* tabel atau kolom, mempunyai nilai lebih untuk overhead maupun *error* dibandingkan dengan *procedural data integrity*. DDI secara otomatis dilakukan ketika *update*, *insert* maupun *delete* dengan cara menolak memodifikasi data jika dirasa akan menimbulkan *error* pada tabel atau *database*.

Procedural data integrity pada umumnya menggunakan *view*, *trigger*, dan *stored procedure* untuk menjalankannya. PDI mendukung *domain* dan *referential integrity* tetapi tidak *entity integrity*. Lebih fleksibel dalam menjalankan *data integrity* tetapi lebih kompleks dan lebih besar resiko kesalahan yang ditimbulkan dibandingkan dengan *declarative data integrity*.

2.3. Entity Relationship Diagram

2.3.1. Definisi Dan Kegunaan

Secara umum definisi dari *Entity Relationship Diagram* (ERD) dapat dikatakan sebagai representasi secara grafis dari hubungan antar tabel di dalam suatu *database*. Kata representasi disini dapat dimaksudkan untuk *Data Model*, *Database Schema*, *Database Diagram* atau nama yang lainnya.

ERD menggambarkan hubungan logikal antar tabel, dan tidak ada hubungan dengan cara penyimpanan data secara fisik di dalam *database*. Tidak ada standarisasi dalam menggambarkan ERD, tetapi secara umum ada kesamaan antara cara satu dengan yang lainnya.

2.3.2. Aturan

Dalam membuat *design* sebuah ERD setiap *entity* harus diberi nama yang unik dan berada pada kotak tersendiri. Untuk menghubungkan dua buah *entity* digambarkan dengan sebuah garis (*line*) dan diberi nama serta informasi yang berkaitan dengan hubungan yang terjadi antara dua *entity* tersebut.

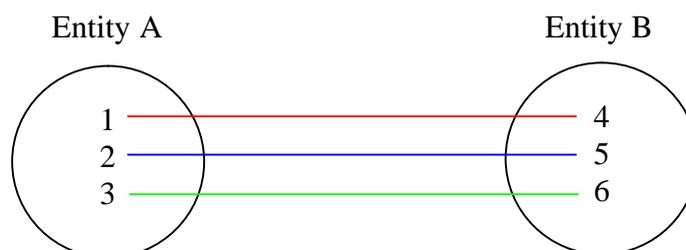
Untuk menggambarkan apakah sebuah *entity* tersebut *mandatory* terhadap *entity* yang lain dengan '1'. Pada ujung dari garis merupakan tipe dari relasi yang terjadi, misalnya *One to One*, *One to Many*, *Many to Many*.

2.3.3. Relasi

Ada tiga jenis relasi yang umum digunakan pada saat melakukan *design* pada ERD. Ketiga jenis relasi tersebut adalah :

a. One to One

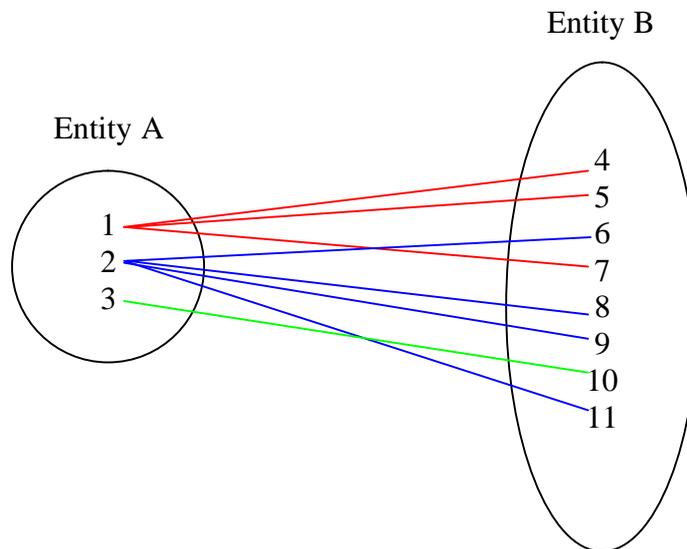
Item pada *Entity A* tepat mempunyai satu hubungan dengan *Entity B*.



Gambar 2.4 Diagram One to One

b. One to Many

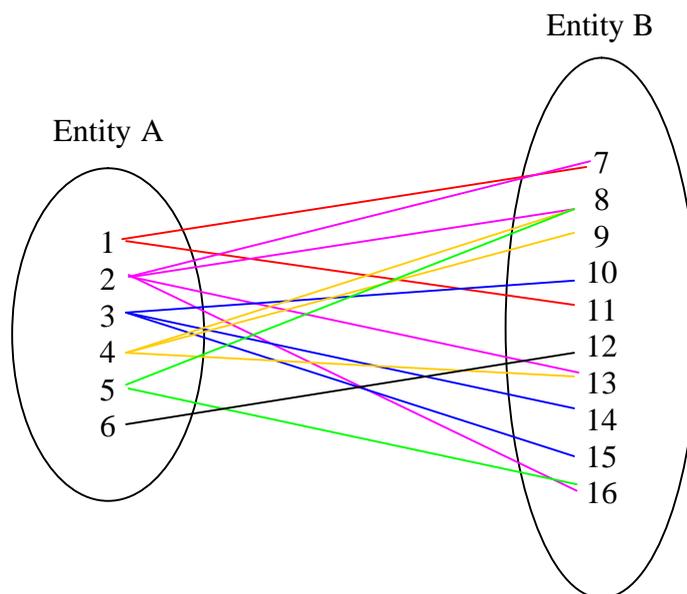
Item pada *Entity A* boleh mempunyai hubungan dengan 2 atau lebih item pada *Entity B*.



Gambar 2.5 Diagram One to Many

c. Many to Many

Item pada *Entity A* dapat mempunyai hubungan dengan lebih dari satu item pada *Entity B*, demikian juga item pada *Entity B* dapat mempunyai hubungan lebih dari satu item pada *Entity A*.



Gambar 2.6 Diagram Many to Many

Tipe garis untuk masing masing jenis relasi dapat dilihat pada gambar berikut :



Gambar 2.7 Tipe garis berdasarkan tipe relasi

Untuk fungsi *mandatory* yang melambangkan bahwa *Entity* tersebut harus mempunyai item terlebih dahulu baru item pada *entity* yang lain bisa di isi item dan dihubungkan. Tipe garisnya hampir sama, hanya saja tanda ‘o’ diganti dengan ‘I’. Untuk relasi *Many to Many* tidak bisa diberi *mandatory*..

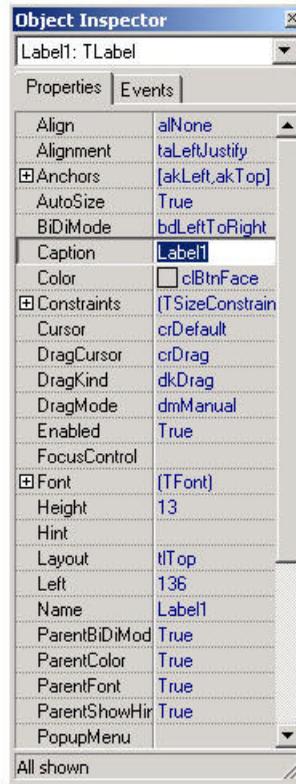


Gambar 2.8 Tipe garis dengan mandatory

2.4. Compiler

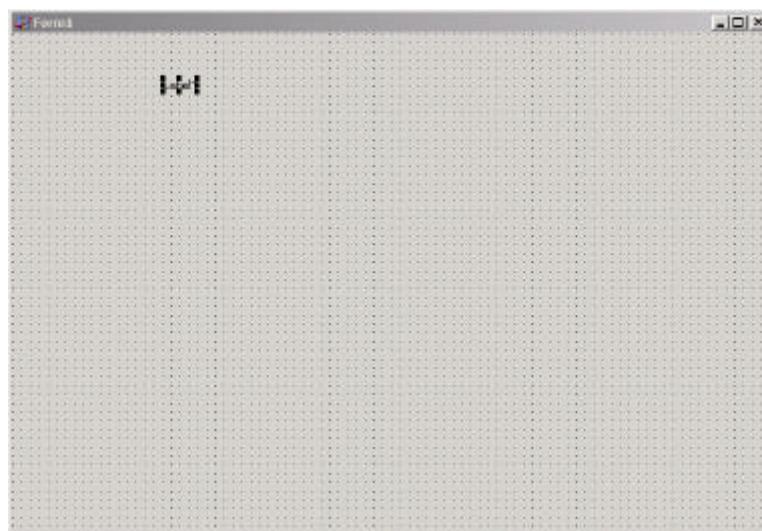
Salah satu kompiler visual yang ada dipasaran, dan cukup handal guna membantu *programmer* membuat sistem. Dengan bahasa pemrograman Pascal, *syntax* yang digunakan cukup mudah dimengerti. Banyaknya komponen yang tersedia dan kemudahan dalam membuat komponen baru sesuai dengan kebutuhan adalah salah satu nilai lebih dari Delphi 5 ini.

Untuk merubah spesifikasi komponen dalam *design time* dapat memanfaatkan fasilitas *Object Inspector*. *Object Inspector* ini berfungsi sebagai fasilitas untuk menampilkan *properties* dan *event handler* dari komponen yang ditempatkan pada *form*.



Gambar 2.9. Object Inspector

Form adalah media untuk menampilkan sistem yang dibuat. Di dalam *form* inilah dapat ditempatkan komponen yang terdapat di Delphi. Baik itu komponen yang dapat dilihat maupun komponen yang secara *default* akan disembunyikan bila sistem dijalankan.



Gambar 2.10. Form

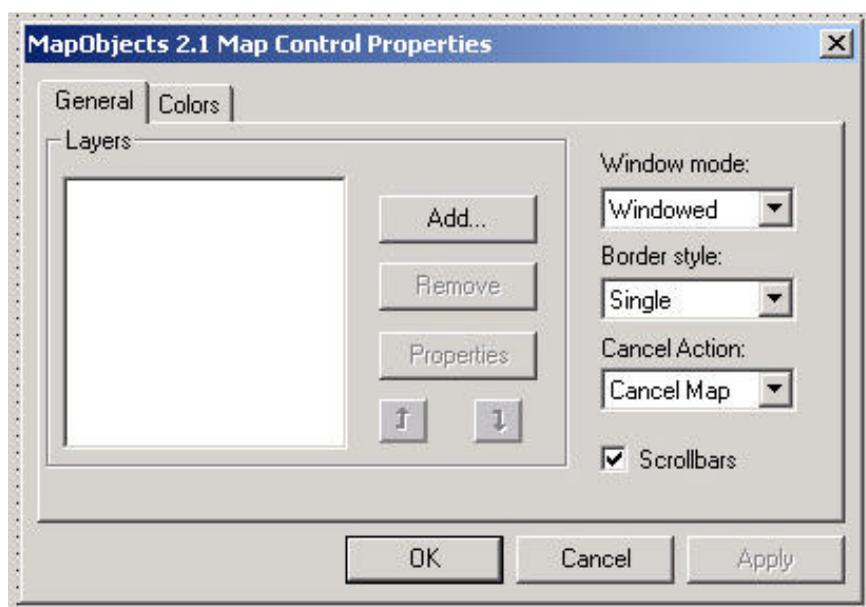
ADO komponen yang ada di dalam Delphi dapat digunakan untuk sarana komunikasi dengan *database server* yang ada. Di dalam ADO ini terdapat banyak jenis yang membuat penulis leluasa dalam memilih jenis komponen yang dipilih untuk melakukan koneksi dengan *database server*.

2.5. Komponen Yang Digunakan

Untuk menampilkan peta pada *compiler* yang dipilih, maka digunakan Map Object 2.1 sebagai komponen pendukung. Komponen ini didapatkan dari ESRI dan versi yang digunakan versi *Evaluation Copy 90 days*.

Cara kerja komponen ini sama dengan cara kerja SIG, yaitu menggunakan sistem layer untuk menampilkan peta. Komponen ini memiliki fasilitas untuk mendukung analisa yang akan dilakukan. Misalnya dapat mengidentifikasi daerah *polygon, line, point*.

Pada *design time* isi dari Map Object dapat diganti melalui Map Object *property*. Untuk menampilkan *property* dari Map Object dengan melakukan *double click* pada komponen Map Object pada *form*. Dari *property* ini dapat dipilih warna dari peta yang ditampilkan, tipe tampilan *border* dari komponen Map Object., menampilkan *scrollbar* atau tidak, menghapus dan menambah *layer*, menggantu *layer property*.



Gambar 2.11 Map Object Property

Pengisian *layer* dapat dilakukan juga pada saat *running time*. Untuk melakukannya diperlukan beberapa *command*. Penggantian *property* juga dapat dilakukan. Contoh *command* untuk menambah *layer*.

```
layer := IMoMapLayer(CreateOleObject('MapObjects2.MapLayer'));
layer.GeoDataset := IMoGeoDataset(dc.FindGeoDataset('States'));
sym := layer.Symbol;
sym.Color := clYellow;
lys := Map1.Layers;
lys.Add(layer);
```

2.6. Normalisasi

Langkah pertama dalam membangun database adalah membangun tabel dengan benar. Tabel-tabel dengan desain yang baik akan mengurangi kompleksitas aplikasi juga menambah fleksibilitas.

Struktur data yang baik merupakan faktor utama dalam kesuksesan aplikasi. Delphi berlandaskan pada database model relasional. Dengan mengikuti aturan dari model ini, data dapat dimanipulasi dengan mudah, sehingga data mentah atau data dalam dunia nyata dapat mengikutinya. Aturan semacam ini dinamakan normalisasi.

Normalisasi adalah reduksi bertahap yang dilakukan pada sekumpulan tabel. Prosedur normalisasi bersifat *reversible*, artinya hasil normalisasi selalu dapat dikembalikan ke keadaan awal. Hal ini juga berarti bahwa tidak ada informasi yang hilang.

Database relational selalu menggunakan field kunci untuk mendefinisikan relasi antar tabel. Semakin banyak tabel maka semakin banyak relasi yang diperlukan untuk menghubungkan semua tabel. Sebuah tabel tidak langsung harus berhubungan dengan setiap tabel lain tetapi setiap tabel dalam database tidak ada yang berdiri sendiri (berhubungan secara tidak langsung).

2.6.1 Normalisasi Bentuk Pertama

Definisi dari bentuk normal pertama (First Normal Form / 1NF) adalah tabel yang tidak mengandung pengulangan group data, semua *keys attribute* sudah

didefinisikan, semua *attribute dependent* pada *primary key*. Misalnya ada sebuah tabel sebagai berikut :

Tabel 2.1 Kabupaten Mojokerto

Kode Kecamatan	Nama Kecamatan	Kode Desa	Nama Desa
1023	Dawarblandong	10231	Dawarblandong
1023	Dawarblandong	10232	Japan
1023	Dawarblandong	10233	Modongan
1056	Sooko	10561	Sooko
1056	Sooko	10562	Mojoranu

Dari data diatas dapat disimpulkan bahwa struktur data yang diperoleh adalah :

Kode Kecamatan

Nama Kecamatan

1-n Kode Desa

1-n Nama Desa

Untuk memulai menjadikan tabel diatas bentuk normal pertama, maka perlu dibetuk *primary key* dan menghilangkan group data yang berulang. Hasilnya adalah sebagai berikut :

Kode Kecamatan - *Primary key*

Nama Kecamatan

Kode Desa - *Primary key*

Nama Desa

2.6.2 Normalisasi Bentuk Kedua

Definisi dari bentuk normalisasi kedua (Second Normal Form / 2NF) adalah tabel sudah dalam bentuk normalisasi pertama, setiap field bukan kunci bergantung sepenuhnya pada kunci (*primary key*). Dengan kata lain, tidak ada non kunci yang bergantung pada sebagian komponen-komponen kunci. Pada normalisasi bentuk kedua ini Nama Kecamatan bergantung (*dependent*) pada Kode Kecamatan, Nama Desa bergantung pada Kode Desa. Oleh sebab itu dipisahkan tabel yang dalam bentuk normalisasi pertama menjadi beberapa tabel seperti pada contoh berikut :

Tabel DesaKode Desa - *Primary key*

Nama Desa

Tabel KecamatanKode Kecamatan - *Primary key*

Nama Kecamatan

2.6.3 Normalisasi Bentuk Ketiga

Bentuk Normal Ketiga (Third Normal Form / 3NF) adalah tabel sudah dalam kondisi 2 NF dan tidak ada non kunci yang tergantung pada elemen bukan kunci.

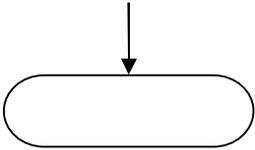
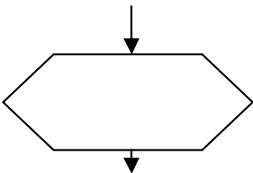
Pada contoh diatas nampaknya sudah memenuhi kriteria dari bentuk normal ketiga. Maka tabel tidak perlu lagi diubah.

2.7. Flowchart

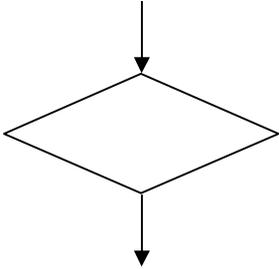
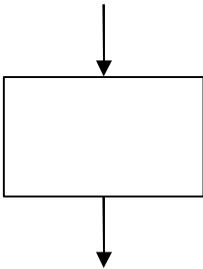
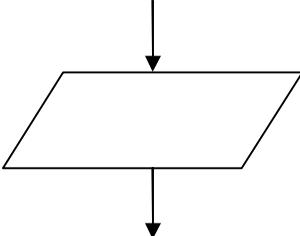
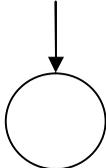
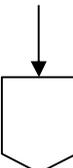
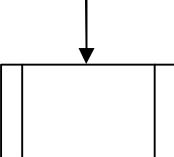
Flowchart merupakan bagan yang menggambarkan urutan instruksi dan hubungan antar proses dalam komputer dengan menggunakan simbol simbol tertentu. Atau dengan kata lain, flowchart merupakan representasi grafis dari urutan proses yang dijalankan sepenuhnya oleh komputer.

Beberapa simbol dalam flowchart, antara lain :

Tabel 2.2 Contoh Lambah Flowchart

Lambang	Keterangan
	Terminal : <ul style="list-style-type: none"> ○ Start / Mulai ○ Stop / Selesai
	Preparation : <ul style="list-style-type: none"> ○ Harga Awal ○ Assignment

Tabel 2.2 Contoh Lambah Flowchart (Sambungan)

Lambang	Keterangan
	Decision : <ul style="list-style-type: none"> ○ Condition ○ Operator Logika dan Boolean ○ Hasil : YA atau TIDAK
	Process : <ul style="list-style-type: none"> ○ Operasi ○ Assignment
	Input / Output : <ul style="list-style-type: none"> ○ Input Data ○ Cetak Data
	Flow Line
	Connector : <ul style="list-style-type: none"> ○ Konektor untuk halaman yang sama
	Connector : <ul style="list-style-type: none"> ○ Konektor pada halaman yang berbeda
	Modularity : <ul style="list-style-type: none"> ○ Prosedur / Function

Komponen Penting dalam Flowchart :

- a. Variable
 - Diawali dengan huruf kemudian diikuti dengan huruf atau angka
 - Tidak boleh ada spasi, operator aritmatika dan logika
- b. Konstanta
 - Alphanumerik, misal : 'PETRA', '10', 'ID4'
 - Number, misal : 1000, -10, 0.5, 2e5
 - Boolean, misal True, False
- c. Proses Assignment
 - Pemberian nilai pada suatu variabel, misal :
 - $A \leftarrow 1$, $A \leftarrow B$, $A \leftarrow \text{'PETRA'}$, $A \leftarrow \text{True}$
- d. Operator
 - Aritmatika, misal : $()$, $^$, $*$, $/$, $+$, $-$, mod
 - Logika, misal : $>$, $=$, $<$, $=$, $\langle \rangle$
 - Boolean, misal : AND, OR, NOT
 - Fungsi Aritmatik, misal : Sin, Cos, Ln, Exp, Abs
- e. Operasi
 - $A \leftarrow A + 5$
 - $A \leftarrow 1 - 2 * \text{Sin}(X) ^ 2$
 - $A \leftarrow (4 > 5) \text{ OR } (3 < 5)$

2.8. Pemetaan Pendidikan SLTA

Dalam membuat pemetaan pendidikan SLTA ini ada yang pertama kali dilakukan adalah tahap perencanaan. Tahap perencanaan ini bertujuan untuk dua kepentingan pokok yaitu sekolah dan guru.

- a. Sekolah
 - i. Perluasan kesempatan belajar
 - ii. Rehabilitasi sekolah
- b. Guru
 - i. Kebutuhan guru seluruhnya
 - ii. Kebutuhan guru menurut mata pelajaran

Tahap-tahap perencanaan disusun berdasarkan 4 kerangka utama, yaitu :

- a. Tahap pertama adalah pengumpulan dan pengolahan data serta penggambaran peta dasar.
- b. Tahap kedua melakukan diagnosa berdasarkan hasil pengolahan/indikator yang dihasilkan dari tahap pertama untuk merumuskan informasi sebagai bahan perumusan kebijakan.
- c. Tahap ketiga adalah penyusunan proyeksi dan perkiraan serta kebutuhan ruang kelas seluruhnya atau unit gedung baru dan tambahan ruang kelas baru serta kebutuhan guru. Pada tahap ini diperhitungkan juga gedung yang perlu direhabilitasi.

Untuk penjelasan masing-masing kerangka sebagai berikut :

- a. Tahap pertama

Data yang dikumpulkan meliputi beberapa variabel sebagai berikut :

- Jumlah penduduk usia sekolah
- Jumlah sekolah
- Jumlah siswa menurut kelompok usia, siswa baru tingkat I
- Jumlah guru,
- Jumlah kelas dan ruang kelas menurut kondisi

Setelah data terkumpul, dilakukan penggambaran peta yang meliputi :

- Batas administrasi pemerintahan.
- Lokasi sekolah
- Jalan, sungai.

Pengolahan data dilakukan setelah proses penggambaran peta selesai.

Pengolahan data meliputi beberapa indikator yang meliputi indikator pemerataan dan indikator mutu.

Indikator pemerataan :

- Angka Partisipasi Murni (APK)

Perbandingan antara jumlah siswa usia sekolah dengan jumlah penduduk usia sekolah. APK ideal adalah 100%, tetapi hal itu sulit karena masih diperbolehkannya siswa mengulang bahkan sampai dua tahun.

$$APK = \frac{\sum SISWA}{\sum PENDUDUKUSIA SEKOLAH} \quad (2.1)$$

- Tingkat Pelayanan Sekolah (TPS)

Menunjukkan besarnya kesempatan belajar atau banyaknya anak yang dapat dilayani oleh suatu sekolah. Bila angka TPS tinggi berarti kesempatan belajarnya rendah, sebaliknya bila angkanya rendah berarti kesempatan belajarnya tinggi. Kesempatan belajar rendah juga berarti daerah tersebut kekurangan sekolah dan disebut tinggi berarti daerah tersebut sudah cukup sekolah. TPS digunakan untuk mengetahui daerah tersebut termasuk dalam wilayah tertentu dalam pembangunan pendidikan.

$$TPS = \frac{\sum LULUSANSLTP}{\sum LEMBAGASLTA \times 5} \quad (2.2)$$

- Angka Melanjutkan (AM)

Banyaknya lulusan tingkat SLTP yang melanjutkan ke tingkat SLTA. AM ideal adalah 100 %, berarti semua lulusan tingkat SLTP melanjutkan ke SLTA. AM digunakan untuk beberapa hal yaitu menentukan daerah tersebut termasuk wilayah tertentu dalam pembangunan pendidikan, menghitung proyeksi siswa dan proyeksi kelas dalam perencanaan pendidikan.

$$AM = \frac{\sum SISWABARUSLTA}{\sum LULUSANSLTP} \quad (2.3)$$

- Rasio Siswa per Sekolah (S/Sek)

Perbandingan antara jumlah siswa dengan jumlah sekolah. Rasio ini diperoleh dengan membagi jumlah siswa SLTA dengan jumlah lembaga SLTA. Semakin besar angkanya berarti merupakan sekolah besar bila dilihat dari rata-rata siswa per sekolahnya.

$$S / SEK = \frac{\sum SISWA}{\sum SEKOLAH} \quad (2.4)$$

- Rasio Siswa per Kelas (S/K)

Perbandingan antara jumlah siswa dengan jumlah kelas. Rasio ini diperoleh dengan membagi jumlah siswa SLTA dengan jumlah kelas SLTA. Rasio yang ideal adalah 40 yaitu sesuai dengan standar sekolah yang dibangun.

$$S / K = \frac{\sum \text{SISWA}}{\sum \text{KELAS}} \quad (2.5)$$

- Rasio Kelas per Ruang Kelas (K/RK)

Perbandingan antara jumlah kelas dengan jumlah ruang kelas. Rasio ini diperoleh dengan membagi jumlah kelas dengan jumlah ruang kelas. Rasio yang ideal adalah 1, berarti setiap ruang kelas hanya dipakai sekali. Nilai < 1 berarti ruang kelas yang digunakan tidak efisien karena terdapat ruang kelas kosong, nilai > 1 berarti ruang kelas yang digunakan lebih dari sekali.

$$R / RK = \frac{\sum \text{KELAS}}{\sum \text{RUANGKELAS}} \quad (2.6)$$

Indikator mutu :

- Persentase Ruang Kelas Baik (RKK)

Perbandingan antara jumlah ruang kelas dalam kondisi baik dengan jumlah ruang kelas seluruhnya. Idealnya persentase ruang kelas yang baik ini 100%, berarti semua ruang kelas yang ada, dalam kondisi yang baik.

$$RKK = \frac{\sum \text{RUANGKELASBAIK}}{\sum \text{RUANGKELAS}} \quad (2.7)$$

- Persentase Guru Layak (GL)

Perbandingan antara guru yang layak mengajar dengan jumlah guru seluruhnya. Idealnya persentase guru layak ini adalah 100 % yang berarti semua guru yang ada layak mengajar. Guru yang layak mengajar jika memiliki ijazah minimal Diploma 3 / Sarjana Muda Keguruan.

$$GL = \frac{\sum \text{GURULAYAK}}{\sum \text{GURU}} \times 100\% \quad (2.8)$$

- Persentase Tambahan Kebutuhan Guru (TKG)

Perbandingan antara tambahan kebutuhan guru yang ada dibandingkan dengan jumlah kebutuhan guru seluruhnya. Kebutuhan guru sama dengan jumlah kelas dikalikan 42 dibagi 24. Angka 42 adalah jumlah jam pelajaran per minggu, dan angka 24 adalah beban mengajar guru per minggu. Tambahan kebutuhan guru adalah jumlah kebutuhan guru dikurangi jumlah guru yang ada. Persentase tambahan kebutuhan guru adalah tambahan kebutuhan guru dibagi dengan kebutuhan guru seluruhnya dikalikan 100.

$$TKG = \frac{\left(\frac{\sum KELAS \times 42}{24} \right) - \sum GURUSLTA}{\frac{\sum KELAS \times 42}{24}} \times 100\% \quad (2.9)$$

b. Tahap kedua

Tahap ini dinamakan tahap diagnosa, karena pada tahap ini hasil perhitungan indikator pada tahap satu digunakan untuk proses analisa. Analisa ini diperlukan untuk mendapatkan wilayah yang sesuai dengan kondisi SLTA.

Indikator yang digunakan untuk penentuan wilayah pendidikan ini adalah AM dan TPS. Dari keempat wilayah yang ada, wilayah 2 adalah wilayah yang paling baik, sedangkan wilayah 3 adalah wilayah yang paling jelek.

Keempat wilayah pendidikan tersebut adalah :

- 1) Wilayah 1 adalah wilayah di mana AM tinggi dan angka TPS tinggi atau kesempatan belajar rendah (T,R).
- 2) Wilayah 2 adalah wilayah di mana AM tinggi dan angka TPS rendah atau kesempatan belajar tinggi (T,T).
- 3) Wilayah 3 adalah wilayah di mana AM rendah dan angka TPS rendah atau kesempatan belajar tinggi (R,T).
- 4) Wilayah 4 adalah wilayah di mana AM rendah dan angka TPS rendah atau kesempatan belajar rendah (R,R).

Untuk wilayah 1, berarti bahwa AM sudah tinggi walaupun angka TPS tinggi yang berarti kesempatan belajar rendah. Hal itu juga berarti walaupun sekolah kurang ternyata banyak anak yang bersekolah. Untuk itu, saran yang diusulkan

adalah pembangunan gedung baru atau tambahkan ruang kelas. Dengan dilakukannya pembangunan ini diharapkan AM yang tinggi akan tetap tinggi dan akan diperbaiki kesempatan belajarnya. Dengan dilakukannya pembangunan gedung baru dan tambahan ruang kelas baru diharapkan daerah ini akan meningkat menjadi wilayah 2.

Untuk wilayah 2, berarti bahwa wilayah ini mampu mencapai AM yang tinggi. Hal ini wajar karena angka TPS rendah yang berarti kesempatan belajar juga cukup tinggi. Hal itu berarti karena sekolah banyak sehingga anak yang bersekolah juga banyak. Untuk itu, saran yang diusulkan adalah agar kondisi seperti ini tetap dipertahankan karena sudah cukup baik. Daerah ini adalah daerah yang terbaik jika dibandingkan dengan ketiga wilayah lainnya.

Untuk wilayah 3, berarti bahwa AM rendah walaupun kesempatan belajar tinggi dilihat dari angka TPS yang rendah. Hal itu berarti jumlah sekolah banyak tetapi banyak anak yang tidak bersekolah. Untuk itu, saran yang diusulkan adalah penyuluhan tentang pentingnya pendidikan, pemberian beasiswa atau JPS, orang tua asuh, teman asuh, dan pelaksanaan berbagai pola belajar. Dengan dilakukan penyuluhan atau pemberian beasiswa diharapkan AM akan meningkat. Daerah ini merupakan daerah yang terburuk jika dibandingkan dengan ketiga wilayah lainnya.

Untuk wilayah 4, berarti bahwa AM rendah walaupun kesempatan belajar rendah dilihat dari angka TPS yang tinggi. Hal itu berarti jumlah sekolah sedikit sehingga yang bersekolah juga sedikit. Untuk itu, saran yang diusulkan adalah pemabngunan gedung baru atau tambahn ruang kelas atau pelaksanaan program belajar lainnya dan disesuaikan dengan kondisi daerah. Dengan dilakukannya pembangunan sekolah diharapkan AM yang rtendah akan meningkat dengna ditambahkan kesempatan belajarnya.

c. Tahap Ketiga

Pada tahap ketiga ini dilakukan perhitungan proyeksi terhadap masalah pendidikan. Untuk perhitungan proyeksi ini digunakan data 2 tahun berurutan, hal ini sesuai dengan persyaratan untuk perhitungan proyeksi yang dikeluarkan oleh Departemen Pendidikan Nasional, Direktorat Jenderal Pendidikan Daar dan Menengah pada bulan Juni tahun 2002.

Perhitungan proyeksi meliputi : angka pertumbuhan penduduk usia sekolah, proyeksi penduduk usia sekolah, proyeksi siswa, proyeksi kelas, tambahan ruang kelas, kebutuhan guru.

Rumus yang digunakan dalam perhitungan proyeksi dapat dilihat pada bagian berikut ini,

1) Proyeksi penduduk usia sekolah

Untuk menyusun proyeksi penduduk usia sekolah menggunakan angka pertumbuhan. Untuk itu, perlu diketahui terlebih dulu angka pertumbuhannya.

Rumus menghitung angka pertumbuhan:

$$APPUS_n = \frac{(PUS_{n-1} - PUS_{n-2})}{PUS_{n-2}} \times 100 \quad (2.10)$$

Keterangan :

$APPUS_n$: angka pertumbuhan penduduk usia sekolah dari tahun n-1 ke n-2

PUS_{n-1} : penduduk usia sekolah tahun n-1

PUS_{n-2} : penduduk usia sekolah tahun n-2

Angka pertumbuhan yang dihasilkan dapat negatif atau positif, bila hasilnya negatif maka proyeksi akan menurun dan bila positif berarti proyeksi akan naik atau lebih besar dari tahun sebelumnya. Setelah diketahui angka pertumbuhan, langkah berikutnya dapat disusun proyeksi penduduk usia sekolah dengan menggunakan rumus berikut :

$$PUS_n = APPUS_n \times PUS_{n-1} \quad (2.11)$$

Keterangan

PUS_n : proyeksi penduduk usia sekolah tahun n

$APPUS_n$: angka pertumbuhan PUS dari tahun n-1 dan n-2

PUS_{n-1} : penduduk usia sekolah tahun n-1

2) Proyeksi siswa

Untuk menyusun proyeksi siswa tahun ke-n dapat digunakan rumus sebagai berikut :

$$PS_n = APK_{n-1} \times PUS_n \quad (2.12)$$

Keterangan

PS_n : proyeksi siswa tahun ke-n

APK_{n-1} : APK tahun ke n-1

PUS_n : proyeksi penduduk usia sekolah tahun ke-n

3) Proyeksi kelas

Untuk menyusun proyeksi kelas digunakan angka parameter rasio siswa per kelas, seperti pada rumus berikut :

$$PK_n = \frac{PS_n}{S / K_{n-1}} \quad (2.13)$$

Keterangan

PK_n : proyeksi kelas tahun n

PS_n : proyeksi siswa tahun n

S/K_{n-1} : rasio siswa per kelas tahun n-1

4) Proyeksi kebutuhan tambahan ruang kelas

Setelah diketahui proyeksi siswa dan kelas, langkah berikutnya adalah menghitung kebutuhan tambahan ruang kelas

$$TRK_n = \frac{PS_n}{(S / K_n) \times (K / RK_n)} - (RKA + RKS) \quad (2.14)$$

Keterangan

TRK_n : proyeksi tambahan ruang kelas tahun n

PS_n : proyeksi siswa tahun n

S/K_n : perkiraan rasio siswa per kelas tahun n

K/RK_n : perkiraan rasio kelas per ruang kelas tahun n

RKa : ruang kelas yang ada tahun n-1

RKs : ruang kelas yang sedang dibangun tahun n-1 (bila tersedia datanya)

5) Kebutuhan guru

Untuk menghitung kebutuhan guru dapat digunakan rumus sebagai berikut :

$$KB_n = \frac{K_n \times 42}{24} \quad (2.15)$$

Keterangan

KB_n : proyeksi kebutuhan guru tahun n

K_n : proyeksi jumlah kelas tahun n

42 : jumlah jam belajar seluruh mata pelajaran

24 : jumlah beban mengajar guru per minggu