

BAB 3

LANDASAN TEORI

3.1. Sistem Informasi

3.1.1. Sistem

Menurut Jerry Fitz Gerald, Arda F. Fitz Gerald dan Warren D Stalling, jr.(dalam Jogiyanto, 2002:24) dikemukakan bahwa sistem adalah suatu jaringan kerja dan prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.

3.1.2. Informasi

Menurut Jogiyanto (2005:8), Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (event) adalah sesuatu yang terjadi pada saat tertentu.

3.1.3. Sistem Informasi

Heitch dan David (dalam Jogiyanto, 2002:11) mendefinisikan, sistem informasi adalah suatu sistem dari dalam suatu organisasi yang dapat mempertemukan kebutuhan pengolahan transaksi harian serta mendukung operasi bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Keberhasilan suatu sistem Informasi yang diukur berdasarkan maksud pembuatannya tentu tergantung pada tiga faktor utama (Cook, 2001) yaitu:

1. Keserasian dan mutu data,
2. Pengorganisasian data,
3. Tatacara penggunaannya.

3.1.4. Komponen Sistem Informasi

Komponen-komponen pendukung sistem informasi antara lain adalah sebagai berikut:

1. Komponen *input* yaitu data yang masuk ke dalam sistem informasi.
2. Komponen model yaitu kombinasi prosedur, logika, dan model matematik yang memproses data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.
3. Komponen *output* yaitu hasil informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Komponen teknologi yaitu alat dalam sistem informasi yang digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan output, dan membantu pengendalian sistem.
5. Komponen basis data yaitu kumpulan data yang saling berhubungan yang tersimpan di dalam komputer dengan menggunakan *software database*.
6. Komponen kontrol yang dirancang untuk menanggulangi gangguan terhadap sistem informasi.

3.2. Sistem Informasi Berbasis Web

Sistem informasi dahulu dibuat secara konvensional (aplikasi berbasis desktop). Namun seiring dengan perkembangan teknologi internet, maka sistem informasi dibuat berbasis web karena sifatnya yang luas dan memungkinkan semua orang dapat mengakses informasi secara cepat dan mudah dari mana saja dan kapan saja selama didukung oleh internet sehingga pemasukkan data dapat

dilakukan darimana saja dan dapat dikontrol dari satu tempat sebagai sentral.

World Wide Web (WWW) merupakan salah satu bentuk layanan yang dapat diakses melalui internet. *Web* adalah fasilitas hypertext untuk menampilkan data berupa teks, gambar, suara, animasi, dan data multimedia lainnya.

Dalam dunia internet, terdapat dua sisi yang saling mendukung yaitu:

1. *Server* adalah penyedia berbagai layanan termasuk *web*. Layanan *web* ditangani oleh sebuah aplikasi bernama *web server*.
2. *Client* bertugas mengakses informasi yang disediakan oleh server. Pada layanan *web*, *client* dapat berupa *web browser*.

3.3. Basis Data

Basis data dapat diibaratkan sebagai sebuah pondasi bagi sistem untuk menyediakan informasi dan data yang dibutuhkan baik yang digunakan untuk menunjang sistem itu sendiri ataupun sebagai *output* untuk pengguna (Adriansyah, 2005).

Basis data merupakan himpunan kelompok data yang saling berkaitan. Data tersebut diorganisasikan sedemikian rupa agar tidak terjadi duplikasi yang tidak perlu, sehingga dapat diolah atau dieksplorasi secara cepat dan mudah untuk menghasilkan informasi (Sutedjo, 2002).

3.4. Replikasi Basis Data

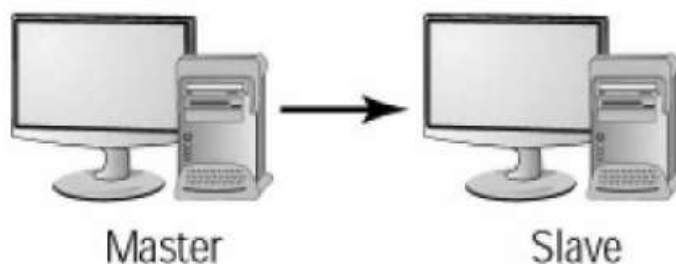
Replikasi basis data adalah seperangkat teknologi yang digunakan untuk menyalin dan mendistribusikan data dari satu *database* ke *database* yang lain. Dan selanjutnya, mensinkronisasikan antar basis data untuk menjaga konsistensi. Dengan replikasi, data dapat didistribusikan

ke lokasi yang berbeda dan pengguna yang jauh melalui LAN, WAN, *Dial-up Connection*, *wireless connections*, dan internet.

Dalam hal replikasi *database* pendistribusi data/pemilik data utama disebut sebagai *master* sedangkan yang menerima distribusi data/yang menjadi *database* duplikasi disebut sebagai *slave*. Tujuan replikasi adalah untuk mengurangi beban dari server dalam pengaksesan data untuk dibagi ke komputer-komputer lain dan juga untuk meningkatkan *performance database*. Salah satu kegunaan dari proses replikasi adalah bila ada satu *database* yang mengalami kerusakan maka data masih dapat dilihat melalui *database* yang lain.

4.4.1. Model-model replikasi *database*

- a. Replikasi *one master to one slave*.



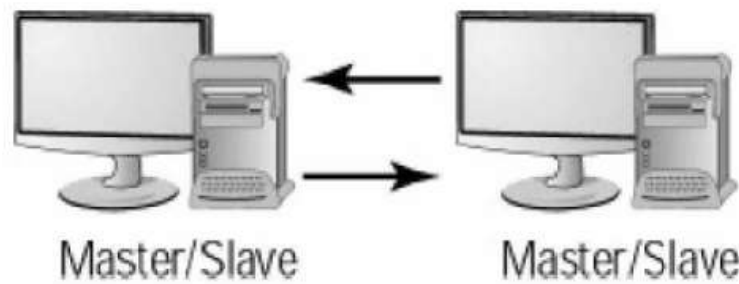
Gambar 3.1 Replikasi *one master to one slave*

- b. Replikasi *one master to many slaves*.



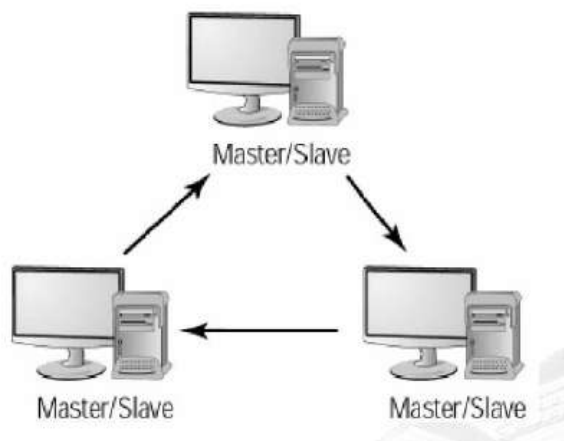
Gambar 3.2 Replikasi *one master to many slave*

- c. Replikasi *master/slave* to *master/slave* "circular relationship".



Gambar 3.3 Replikasi *master/slave* to *master/slave* "circular relationship"

- d. Replikasi *master/slave* "daisy chain" (terdiri banyak *master/slave* yang saling berhubungan).



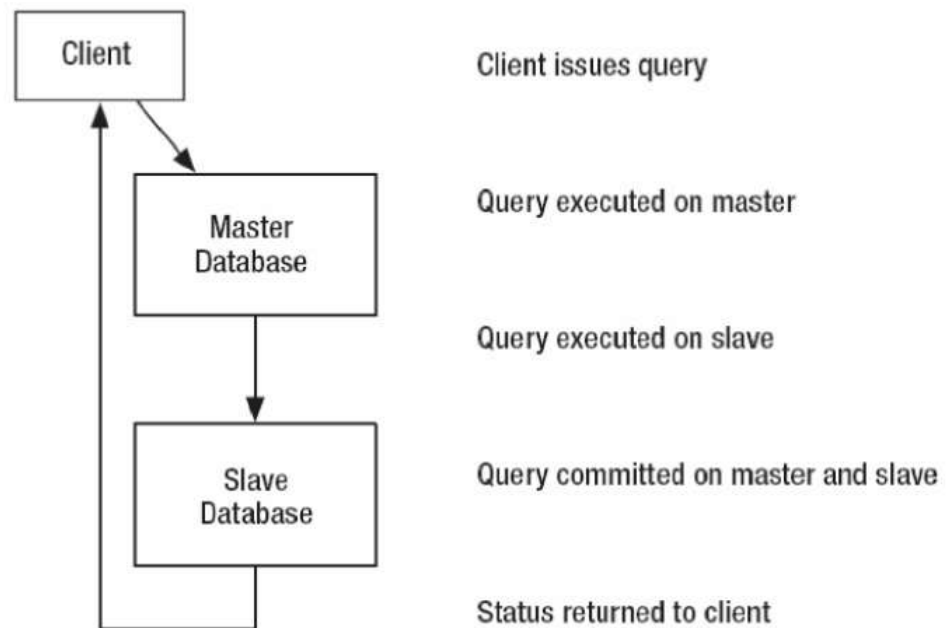
Gambar 3.4 Replikasi *master/slave* "daisy chain"

4.4.2. Dua Jenis Replikasi

- a. Replikasi *Synchronous*

Proses replikasi dilakukan secara *real-time* antara *master* dan *slave*. Keseluruhan proses penulisan pada disk *master* dan *slave* harus selesai dilakukan terlebih dahulu sebelum beranjak ke transaksi selanjutnya. Untuk jenis replikasi ini kebutuhan akan performansi sistem yang tinggi harus dipertimbangkan (kecepatan & jarak antar *site/node*). Keuntungan yang

dimiliki dari jenis ini adalah menyediakan *recovery* yang konsisten karena sinkronisasi data terjaga.

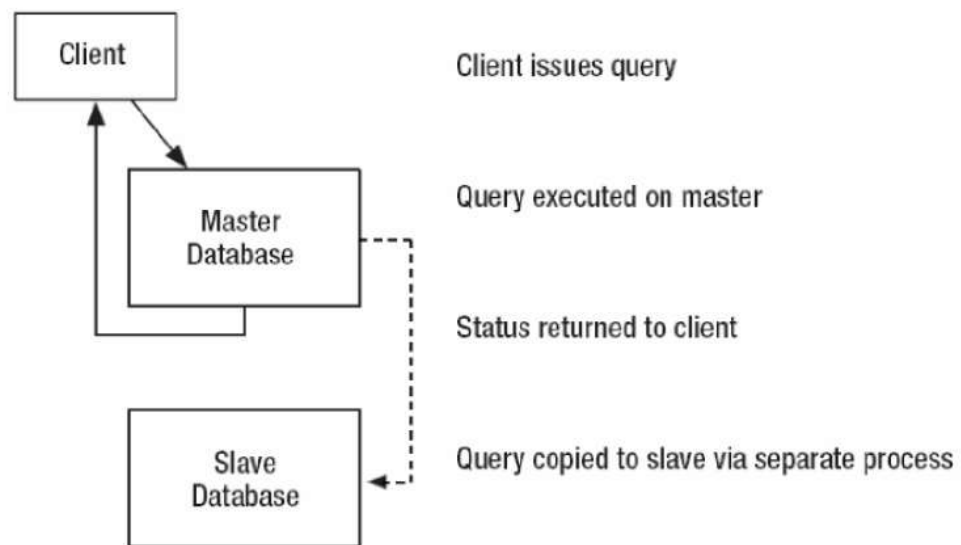


Proses: Client → Master Database → Slave Database → Client

Gambar 3.5 Replikasi Synchronous

b. Replikasi *Asynchronous*

Proses replikasi terjadi setelah transaksi di *master* selesai dilakukan. Pertukaran data dilakukan secara *buffering*, data akan diletakkan dalam sebuah *buffer* terlebih dahulu, kemudian pada jangka waktu tertentu akan direplikasi ke *disk slave*. Jenis replikasi ini tidak menjamin kesinkronan data apabila salah satu *site/node* mengalami *crash* saat replikasi belum selesai dilaksanakan. Keuntungan yang dimiliki dari jenis ini adalah efektifitas biaya proses transaksi.



Proses: *Client* → *Master Database* → *Client*

Gambar 3.6 Replikasi *Asynchronous*

4.4.3. Replikasi pada SQL Server

Untuk melakukan replikasi pada SQL Server buat *stored procedure* untuk menyimpan query yang melakukan pengambilan data dari *database* utama dan diberikan pada *database* replika. Pada *stored procedure* selain mengambil data yang dibutuhkan, juga dilakukan penyesuaian struktur data dari *database* utama ke *database* replika.

4.4.4. Replikasi pada MySQL

Mulai versi 5.0, MySQL sudah mendukung sistem replikasi yang mana sebuah *database server* yang berfungsi sebagai *master* dapat tereplikasi datanya ke dalam satu atau lebih *database server* yang difungsikan sebagai *slave*. Jenis replikasi pada MySQL adalah replikasi *Asynchronous*. Pada MySQL, replikasi dapat diberlakukan pada sebagian tabel atau pada keseluruhan *database*, tergantung pada kebutuhan.

Langkah-langkah umum pembuatan replikasi pada MySQL:

1. Koneksi jaringan (*master & slave*).

2. Instalasi MySQL versi 5.2 (*master & slave*).
3. Buat *database (master & slave)*.
4. Konfigurasi *master server*.
5. Konfigurasi *slave*.
6. *Testing*.

3.5. Tools

3.5.1. Framework .NET

Framework .NET merupakan suatu komponen Windows yang terintegrasi dan dibuat agar dapat menjalankan berbagai macam aplikasi .NET, termasuk pembangunan aplikasi *Web service*. Microsoft Framework .NET menyediakan semua tool dan teknologi yang diperlukan untuk membangun aplikasi terdistribusi.

Framework .NET terdiri atas dua elemen penting, yaitu CLR (*Common Language Runtime*) dan FCL (*Framework Class Library*). CLR adalah sebuah lingkungan yang digunakan untuk menjalankan program saat runtime, sedangkan untuk FCL diletakkan di atas CLR dan berfungsi untuk menyediakan layanan yang dibutuhkan oleh aplikasi .NET.

3.5.2. ASP .NET

ASP .NET adalah teknologi pembangunan aplikasi web yang dibangun oleh Microsoft. ASP .NET digunakan untuk membangun aplikasi web ataupun web XML yang dinamis. ASP .NET adalah bagian dari *platform .NET* milik Microsoft yang merupakan pengganti ASP (*Active Server Pages*).

ASP .NET sudah dibangun berdasarkan CLR yang dapat bekerjasama dengan semua aplikasi .NET Microsoft. ASP .NET memiliki beberapa kelebihan dibandingkan teknologi terdahulu, antara lain:

1. Kemudahan mengakses berbagai *library* .NET Framework secara konsisten dan *powerfull*, yang mempercepat pembangunan aplikasi.
2. Penggunaan berbagai bahasa pemrograman secara penuh, misalnya VB.NET, C#, J#, dan C++. Dalam ASP .NET bahasa-bahasa ini dapat digunakan secara penuh sebagaimana layaknya bekereja di *Windows Application*. Hal ini jauh berbeda dibandingkan ASP klasik yang menggunakan *scripting language*. Misalnya VBScript agak berbeda syntaxnya dnegan VB, dalam ASP .NET kita dapat menggunakan VB .NET dengan syntax yang 100% sama dengan ketika membangun *Windows Application*. Demikian juga dengan C#, tidak ada perbedaan *syntax* antara C# untuk membuat kode ASP .NET dengan *Windows Form*. Ini sangat memudahkan memprogram aplikasi Windows dan web secara konsisten.
3. Tersedia berbagai *web control* yang dapat digunakan membangun aplikasi secara cepat. *Programmer* dapat dengan mudah mengkaitkan data ke *web control* sebagaimana layaknya memprogram *windows application*. Hal ini sangat mempercepat pembuatan aplikasi dibandingkan harus menyusun kode-kode HTML secara manual.
4. *Code behind*, artinya kode-kode pemrograman yang menjadi *logic* aplikasi ditempatkan terpisah dengan kode *user interface* yang berbentuk HTML. Ini sangat memudahkan dalam *debugging*, karena kode untuk *presentation layer* tidak tercampur dengan kode *application logic*.

Siklus hidup halaman ASP .NET:

1. *Client* meminta *file* yang memiliki ekstensi *.aspx*.
2. ASP .NET membaca *file* dari *file sistem server*.

3. ASP .NET memeriksa tiap-tiap *tag* dalam *file* dalam memuat ke dalam memori
 - a. Jika *tag* berisi atribut *runat="server"*, ASP .NET memuat komponen *software* yang disebut kontrol *server*. Nama *tag* menentukan tipe dari kontrol *server*.
 - b. *Tag* tanpa atribut *runat="server"* adalah *tag* HTML biasa, ASP .NET akan memuatnya ke dalam komponen *software* dan kemudian *tag* aslinya akan dikirim kembali ke *browser client*.
 4. Setelah memuat semua *tag* ke dalam memori, ASP .NET menjalankan kode program dari tiap-tiap kontrol *server*. Kode ini akan mengakses sumber dari *server web*, kode dapat mengubah isi atau properti dari kontrol *server* di halaman dan dapat juga menambahkan atau menghilangkan kontrol *server*.
 5. Ketika semua kode dalam kontrol *server* selesai dijalankan, ASP .NET secara berurutan memberitahu tiap kontrol untuk *me-render* dirinya.
 - a. Jika *tag* asli adalah HTML biasa, ASP .NET akan mengirim tanpa memodifikasi.
 - b. Jika *tag* asli yang dibuat adalah kontrol *server*, ASP .NET akan memanggil method kontrol *render* untuk membuat HTML.
- Sesudah halaman meninggalkan *server*, ASP .NET melepas kontrol *server* dan sumber lainnya yang sudah dibuat sebelumnya.