BAB II

TINJAUAN PUSTAKA

1. Tinjauan Perusahaan

Pada tinjauan perusahaan ini akan dibahas mengenai sejarah berdirinya perusahaan, struktur organisasi serta uraian tugas dari masing-masing bagian yang terlibat.

2.1.1 Sejarah Berdirinya Dinas Tata Ruang dan Permukiman Provinsi Jawa Barat (Distarkim)

Dinas Tata Ruang Dan Permukiman Provinsi Jawa Barat merupakan salah satu unsur pemerintahan Provinsi Jawa Barat yang di bentuk berdasarkan peraturan Daerah no.15 tahun 2000. Dinas Tata Ruang Dan Permukiman Provinsi Jawa Barat itu sendiri mempunyai tugas pokok yaitu merumuskan kebijaksanaan operasional di bidang tata ruang dan permukiman di Jawa Barat serta melaksanakan sebagian kewenangan desentralisasi Provinsi Jawa Barat dan kewenangan lainnya yang dilimpahkan kepada gubernur.

Dalam menunjang dan mewujudkan visi dan misi Provinsi Jawa Barat untuk menjadi Provinsi termaju di Indonesia serta mitra terdepan ibu kota negara tahun 2010, Dinas Tata Ruang Dan Permukiman Provinsi Jawa Barat mengambil peran dalam pengembangan model pembangunan yang mengintegrasikan pertanian dan industri melalui :

1. Penciptaan Tata Ruang kawasan yang dinamis, ekonomis, antisipatif, transparan dan legitimate.

2. Mewujudkan permukiman yang sehat, aman, tertib, produktif, berwawasan lingkungan dan berkelanjutan.
3. Meningkatkan pelayanan prima di bidang Tata Ruang dan Permukiman, jasa konstruksi, dan uji mutu kepada instansi pemerintah, masyarakat, dan swasta.

Tujuan Dan Sasaran :

A. Visi

Dengan pelayanan prima, Dinas Tata Ruang Dan Permukiman Provinsi Jawa Barat menjadi andalan dalam mewujudkan Visi Jawa Barat pada tahun 2010.

B. Misi

1. Menciptakan aparataur yang beriman dan bertaqwa, profesional, bersih, dan beretos kerja tinggi.

   a) Tujuan :

      a.1. Mendukung peningkatan pelayanan masyarakat di bidang Tata Ruang dan Permukiman.

      a.2. Mewujudkan aparatur yang *professional/good governance*.

      a.3. Mewujudkan peran pelaku pembangunan yang profesional dalam penerapan hasil penelitian dan pengembangan teknologi arsitektur bangunan dan jati diri kawasan.

   b) Sasaran :

      b.1. Terselenggaranya pelayanan prima kepada masyarakat dibidang pengelolaan gedung negara di Jawa Barat.

      b.2. Tersedianya SDM Kabupaten/Kota yang memadai di bidang Tata

Ruang Dan Permukiman Di jawa Barat.

b.3. Terbinanya kemampuan aparatur antar unit kerja Dinas Tata Ruang dan Permukiman.

b.4. Terselenggaranya koordinasi pembinaan SDM dan kinerja pengguna jasa, penyedia jasa dan masyarakat jasa konstruksi di 22 Kabupaten/kota.

2. Menciptakan penataan ruang kawasan yang dinamis, akomodatif, antisipatif, transparan dan *legitimate*.

a) Tujuan :

a.1. Mewujudkan penataan ruang kawasan lintas kabupaten/kota yang dinamis, akomodatif, antisipatif, transparan dan *legitimate*.

a.2. Mewujudkan aparat kabupaten/kota yang berkualitas dalam pengendalian pembangunan bidang tata ruang dan permukiman.

a.3. Menciptakan kawasan yang menarik bagi penanam modal (*investor*).

b) Sasaran :

b.1 Tersedianya juklak/juknis/pedoman bidang tata ruang dan permukiman.

b.2 Tersedianya hasil perencanaan ruang kawasan untuk dijadikan acuan pelaksanaan pembangunan di 2 kawasan andalan.

b.3 Tersusunya tata ruang kawasan pada 7 kawasan andalan dan 6 *core* bisnis.

b.4 Tersusunnya rencana detail tata ruang kawasan yang dapat diaplikasikan.

b.5 Tersedianya petunjuk teknis pengendalian pemanfaatan bidang tata ruang dan permukiman.

b.6 Tersusunnya rencana tata bangunan dan lingkungan di kawasan tertentu dan strategis.

1. Mewujudkan permukiman yang sehat, aman, tertib, produktif, berwawasan lingkungan dan berkelanjutan.

c) Tujuan :

a.1 Mewujudkan lingkungan permukiman yang layak huni.

a.2 Mewujudkan pelayanan dan fasilitas standar pelayanan kebutuhan minimal prasarana dan sarana permukiman.

a.3 Mewujudkan peran serta masyarakat dalam pembangunan bidang permukiman.

d) Sasaran :

b.1 Tersedianya prasarana dan sarana dasar permukiman, melalui konsep tri bina di perkotaan dan perdesaan di 7 kawasan andalan.

b.2 Tersedianya pelaku pembangunan yang mempunyai kemampuan dalam penerapan hasil penelitian dan pengembangan teknologi, arsitektur bangunan dan jati diri kawasan di 7 kawasan andalan.

b.3 Tersedianya produk hasil penelitian dan pengembangan teknologi, arsitektur bangunan dan jati diri kawasan aplikatif.

b.4 Tersedianya N, S, P, M bidang tata ruang dan permukiman.

b.5 Terselenggaranya fasilitas peran serta masyarakat dalam bidang permukiman di Jawa Barat.

b.6 Tersedianya sistem informasi bidang perumahan dan permukiman.

b.7 Tersedianya Rencana Pembangunan dan Pengembangan Perumahan

Permukiman Daerah (RP4D).

1. Meningkatkan pelayanan prima di bidang Tata Ruang dan Permukiman, jasa konstruksi, dan uji mutu kepada instansi pemerintah, masyarakat, dan swasta.

e) Tujuan :

a.1 Menyempurnakan SOTK Dinas Tata Ruang dan Permukiman dalam rangka penyelenggaraan pelayanan prima.

a.2 Mewujudkan peran lembaga yang optimal sesuai dengan kewenangannya.

a.3 Mewujudkan pembinaan dalam rangka peningkataan kemitraan, SDM dan kinerja penyedia jasa, pengguna jasa dan masyarakat.

f) Sasaran :

b.1 Tersedianya sistem informasi jasa konstruksi dan uji mutu di Jawa Barat.

b.2 Terselenggaranya kemudahan pelayanan dan akses dalam memperoleh pendanaan dan jaminan pertanggungan resiko bidang usaha jasa konstruksi di Jawa Barat.

b.3 Terselenggaranya tata tertib usaha dan pelaksanaan serta pemanfaatan usaha jasa konstruksi di 22 kabupaten/kota.

2.2 Landasan Teori

Pada landasan teori ini diuraikan sejumlah teori untuk membantu dan memecahkan permasalahan yang ada. Beberapa landasan teori tersebut meliputi konsep dasar dan definisi-definisi yang berkaitan dengan perangkat lunak yang digunakan serta faktor-faktor pendukung dalam pelaksanaan perancangannya.

2.2.1 Definisi dan Konsep Dasar Sistem Informasi

Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen di dalam pengambilan keputusan. Pertanyaannya adalah dari mana informasi tersebut didapatkan? Informasi dapat diperoleh dari sistem informasi atau disebut juga dengan *processing system* atau *information processing system* atau *information-generating system*. Sistem informasi didefinisikan oleh Robert A. Leitchdan K. Roscoe Davis adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.[1]

2.2.1.1 Komponen Sistem Informasi

John Burch dan Gary Grudnitski mengemukakan bahwa sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran *(output block*), blok teknologi (*technology block*), blok basis data (*database block*), dan blok kendali (*controls block*). Sebagai suatu sistem, keenam blok tersebut masing-masing berinteraksi satu dengan yang lainnya membentuk dengan satu kesatuan

untuk mencapai sasarannya.

1. Blok Masukan

   Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model

   Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi

   Teknologi merupakan “kotak alat” (*tool-box*) dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data,menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem cara keseluruhan. Teknologi terdiri dari 3 bagian utama, yaitu teknisi (*humanware atau brainware*), perangkat lunak (*software*) dan perangakat keras (*hardware*). Teknisi dapat berupa orang-orang yang mengetahui teknologi dan membuatnya dapat beroperasi. Misalnya teknisi adalah operator komputer, pemrograman, operator pengolah kata, spesialis telekomunikasi, analis sistem, penyimpan data dan sebagainya.

5. Blok Basis Data

   Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu di simpan didalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data didalam basis data perlu di organisasikan sedeimikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management Systems*).

6. Blok Kendali

   Banyak hal yang dapat merusak informasi, seperti misalnya bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, kesalahan-kesalahan, ketidak efisienan, sabutase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.[1]

2.2.2 Model Proses Perangkat Lunak

Model-model proses perangkat lunak terdiri dari :

1. Linear Sequential Model

Model ini sering disebut model klasik atau *waterfall*. Model ini menyarankan pendekatan pengembangan secara sekuen dan sistematik untuk pengembangan

perangkat lunak dimulai di level sistem, berlanjut ke analisis, lalu perancangan, pemrograman, pengujian dan pemeliharaan. Model ini merupakan model yang tertua.[2]

Kelemahan model ini adalah:

1. Proyek-proyek nyata jarang mengikuti alur sekuen yang diusulkan model. Meskipun linear model dapat mengakomodasikan iterasi, namun model melakukan secara tidak langsung. Sebagai hasilnya, perubahan-perubahan dapat menyebabkan kebingungan saat tim pengembangan melakukannya.
2. Sering *customer* sulit menyatakan semua kebutuhannya secara eksplisit. Model ini memerlukan pernyataan eksplisit itu dan sulit mengakomodasi ketidakpastian yang terdapat di awal dari kebanyakan proyek.
3. *Customer* harus memiliki kesabarann. *Versi* yang dapat bekerja dari program tidak akan tersedia sampai akhir dari proyek. Kesalahan besar utama, jika tidak terdeteksi sampai pada program kerja dikaji ulang, maka kesalahan itu dapat mengakibatkan program sama sekali tidak dapat digunakan. Menyiapkan banyak sumber daya.

Alasan kelemahan model ini adalah:

1. Kebutuhan harus telah ditetapkan di awal di siklus hidup.
2. Kebutuhan divalidasi terlalu lambat.

Meskipun terdapat kelemahan, namun model ini memberikan daftar lengkap jumlah aktivitas minimal yang perlu terdapat di model pengembangan perangkat lunak. Umumnya, model-model pengembangan yang lain merupakan superset dari model pengembangan model *waterfall*, memuat aktivitas-aktivitas yang

lebih banyak daripada model *waterfall* ini.

2. *Prototyping Model*

Seringkali pemesan mendefinisikan kumpulan sasaran untuk diakomodasi oleh perangkat lunak tanpa dapat mengidentifikasi rincian-rincian masukan, pengolahan dan keluaran yang diharapkannya. Model *Prototype* cocok untuk situasi ini.

Pada model ini dimulai dengan pengumpulan informasi mengenai kebutuhan, dimana :

1. Pengembang dan pemesan bertemu dan mendefinisikan sasaran-sasaran umum
2. Mengidentifikasikan kebutuhan yang telah diketahui, dan
3. Mencari bidang-bidang yang masih memerlukan pendefinisian.

Setelah itu pengembang melakukan “perancangan kilat” terhadap kebutuhan yang telah teridentifikasi pada pertemuan. Perancangan berfokus pada representasi yang tampak oleh pengguna. Perancangan ini menuntun pembangunan *Prototype* perangkat lunak yang akan diberikan pembeli/pemakai. *Prototype* itu dievaluasi olah pemakai dan digunakan sebagai landasan untuk memperbaiki spesifikasi kebutuhan. Proses ini berulang sampai *Prototype* yang dikembangkan memenuhi kebutuhan pemakai, dan pengembang telah memahami permasalahan dengan lebih baik.[2]

Variasi model ini adalah :

1. *Throw away prototyping*

Prototype benar-benar hanya digunakan untuk pengumpulan kebutuhan-kebutuhan pemakai, dibuang kemudian dibuat perangkat lunak yang akan benar-benar digunakan di operasi sehari-hari.

2. *Dead prototyping*

Prototype yang *versi* terakhir merupakan produk perangkat lunak yang benar-benar dipasang dan digunakan di operasi sehari-hari.

3. *Rapid Application Development (RAD) Model*

RAD adalah proses pengembangan perangkat lunak yang semakin meningkat (*incremental*) yang menekankan pada siklus pengembangan yang cepat. Model RAD merupakan adaptasi "berkecepatan tinggi" dari *linear sequential model* dimana pembangunan berbasis pada komponen.

RAD harus didukung oleh perangkat dan lingkungan pengembangan yang memadai, biasanya dikembangkan berdasarkan orientasi komponen. Lingkungan pengembangan telah memiliki pustaka yang luar biasa besar lengkap seperti Java Development Kit beserta seluruh pustaka dari Sun dan vendor-vendor lain pendukung, *Visual Basic* beserta lingkungan *Microsoft* yang luar biasa dasyat, *Borland Delphi* berikut pustakanya.

Pemrograman biasanya berupa pembuatan antarmuka dan kode perekat untuk merekatkan beragam komponen-komponen yang telah terdapat di lingkungan pemrograman itu.

4. *Evolutionary Software Process Model*

Model evolusi adalah model perulangan. Model ini dicirikan dengan pengembang mengembangkan *versi-versi* sistem yang semakin lebih lengkap. Model telah mempertimbangkan unutk mengakomodasikan evolusi proyek secara lengkap.

Model ini terdiri dari

1. *Incremental Model*

   Model ini mengkombinasikan antara *linear sequential* model dengan filosofi iteratif pada *prototyping*. Pada masing-masing sekuen linear menghasilkan perangkat lunak yang semakin meningkat kompleksitasnya.

2. *Spiral Model*

   Model ini menggabungkan antara sifat alami iterasi dari *prototyping* dengan aspek sistematik dan terkendali dari *linear sequential* model. Model ini memberi peluang unutk pengembangan cepat.

3. *WINWIN Spiral Model*

   Model spiral menyarankan aktivitas komunikasi dengan pembeli. Sasaran aktivitas ini untuk memperoleh deskripsi kebutuhan pemakai. Sering pengembang dan pembeli terlibat negosiasi, pembeli harus menyeimbangkan antara fungsionalitas, kinerja dan karakteristik lain seperti ongkos dan waktu. Negosiasi terbaik adalah *WIN-WIN* dimana pembeli memperoleh kebutuhan mayoritasnya dan pengembang mengembangkan perangkat lunak yang realistik, di dalam anggaran yang tepat dan terpenuhi *deadline*.

4. *Concurrent Development Model*

   Model ini biasa digunakan untuk pengembangan sistem *client/server*.[2]

2.2.3 Basis Data

Istilah basis data banyak menimbulkan interpretasi yang berbeda. Anthoni J. Fabbri dan A. Robert Schwab, mendefinisikan basis data sebagai berikut

“Basis data adalah sistem berkas terpadu yang dirancang terutama untuk meminimalkan pengulangan data”

Menurut George Tsu-der Chou, basis data dapat didefinisikan sebagai berikut ini.

“Basis data sebagai kumpulan informasi bermanfaat yang diorganisasikan ke dalam tatacara yang khusus”.[3]

Basis data dimaksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas.

Sistem basis data adalah suatu sistem menyusun dan mengelola record-record menggunakan komputer untuk menyimpan atau merekam serta memelihara data opersional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil keputusan.

Untuk mengelola basis data diperlukan perangkat lunak yang disebut DBMS. DBMS adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol, dan mengakses basis data dengan cara yang praktis dan efisien.

2.2.3.1 Tahap Perancangan Basis Data

Perancangan basis data merupakan langkah untuk menentukan basis data yang diharapkan dapat mewakili seluruh kebutuhan pengguna. Perancangan basis data terdiri atas perancangan basis data secara konseptual, perancangan basis data secara logis, dan

perancangan basis data secara fisis.[3]

Beberapa komponen yanng terdapat pada perancangan basis data secara konseptual antara lain:

2. Entitas

   Entitas terkadang disebut tipe entitas atau kelas entitas. Entitas adalah objek yang dapat dibedakan dari objek-objek lainnya.

3. Atribut

   Atribut adalah item data yang menjadi bagian dari suatu entitas. Istilah lain dari attribut adalah properti.

4. Hubungan

   Hubungan adalah asosiasi atau kaitan antara dua entitas.

5. Kekangan

   Kekangan digunakan untuk melindungi integritas data (misalnya, melindungi kesalahan sewaktu pengisian data.

6. Domain

   Domain adalah himpunan yang berlaku bagi suatu atribut. Kekangan domain mendefinisikan nama, tipe, format, panjang, dan nilai masing-masing item data.

7. Integritas Referensial

   Integritas referensial adalah aturan-aturan yang mengatur hubungan antara kunci primer dengan kunci tamu milik tabel-tabel yang berbeda dalam suatu basis data relasional untuk menjaga konsistensi data.

8. Entitas

Entitas terkadang disebut tipe entitas atau kelas entitas. Entitas adalah objek yang dapat dibedakan dari objek-objek lainnya.

9. Atribut

Atribut adalah item data yang menjadi bagian dari suatu entitas. Istilah lain dari attribut adalah properti.

10. Hubungan

Hubungan adalah asosiasi atau kaitan antara dua entitas.

10. Kekangan

Kekangan digunakan untuk melindungi integritas data (misalnya, melindungi kesalahan sewaktu pengisian data.

11. Domain

Domain adalah himpunan yang berlaku bagi suatu atribut. Kekangan domain mendefinisikan nama, tipe, format, panjang, dan nilai masing-masing item data.

12. Integritas Referensial

Integritas referensial adalah aturan-aturan yang mengatur hubungan antara kunci primer dengan kunci tamu milik tabel-tabel yang berbeda dalam suatu basis data relasional untuk menjaga konsistensi data.

2.2.4 Metode Perancangan Sistem

Dalam tahap perancangan suatu sistem diperlukan adanya teknik-teknik penyusunan sistem untuk menganalisa dan mendokumentasikan data yang mengalir didalam sistem tersebut.

2.2.4.1 Diagram Konteks

Diagram konteks menggambarkan hubungan antara sistem dengan entitas luarnya. Diagram konteks berfungsi sebagai transformasi dari satu proses yang melakukan transformasi data *input* menjadi data *output*. Entitas yang dimaksud adalah entitas yang mempunyai hubungan langsung dengan sistem.

Suatu diagram konteks selalu mengandung satu dan hanya satu proses saja. Proses ini mewakili proses dari seluruh sistem. Diagram konteks ini menggambarkan hubungan *input* atau *output* antara sistem dengan dunia luarnya (kesatuan luar).

2.2.4.2 *Entity Relationship Diagram*

ERD hanya berfokus pada data, dengan menunjukkan “jaringan data” yang ada untuk suatu sistem yang diberikan. ERD sangat berguna bagi aplikasi di mana data dan hubungan yang mengatur data sangatlah kompleks. ERD pada mulanya diusulkan oleh Peter Chen untuk desain sistem *database* relasional dan telah dikembangkan oleh yang lainnya. Serangkaian komponen utama diidentifikasikan untuk ERD : objek data, atribut, hubungan dan berbagai tipe indikator. Tujuan utama dari ERD adalah untuk mewakili objek data dan hubungan mereka.

Kardinalitas model data harus dapat merepresentasikan jumlah peristiwa dari objek di dalam hubungan yang diberikan. Tillmann mendefinisikan kardinalitas dari *object-relationship pair* dengan cara sebagai berikut: kardinalitas merupakan spesifikasi dari sejumlah peristiwa dari satu [objek] yang dapat dihubungkan ke sejumlah peristiwa dari [objek] yang lain. Dengan mempertimbangkan semua kombinasi dari 'satu' dan 'banyak', dua [objek] dapat dihubungkan sebagai:

1.Satu-ke-satu (1:1)

2.Satu-ke-banyak (1:N)

3.Banyak-ke-satu (N:1)

4.Banyak-ke-banyak (M:N)

2.2.4.3 *Data Flow Diagram*

Data Flow Diagram (DFD –DAD/Diagram Alir Data) memperlihatkan hubungan fungsional dari nilai yang dihitung oleh sistem, termasuk nilai masukan, nilai keluaran, serta tempat penyimpanan internal. DAD adalah gambaran grafis yang memperlihatkan aliran data dari sumbernya dalam objek kemudian melewati proses yang mentransformasinya ke tujuan yang lain, yang ada pada objek lain. DAD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangan lingkungan fisik dimana data tersebut mengalir. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur (*structured analysis and design*). DFD merupakan alat yang cukup populer sekarang ini, karena dapat menggambarkan arus data di dalam sistem dengan terstruktur jelas.

Beberapa simbol yang digunakan dalam *Data Flow Diagram* (DFD) antara lain:

1. *External Entity* (kesatuan luar) atau *boundary* (batas sistem)

Setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima input dan menghasilkan output kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang,

organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.

2. *Data Flow* (arus data)

   Arus data (*data flow*) di DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (*process*), simpanan data (*data strore*) dan kesatuan luar (*external entity*). Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

3. *Process* (proses)

   Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang , mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Untuk *physical data flow diagram* (PDFD), proses dapat dilakukan oleh orang, mesin atua komputer, sedangkan untuk *logical data flow diagram* (LDFD), suatu proses hanya menunjukkan proses dari komputer. Setiap proses harus diberi penjelasan yang lengkap meliputu identifikasi proses, nama proses dan pemroses.

4. *Data Store* (simpanan luar)

   Simpanan data (*data store*) merupakan simpanan dari data yang dapat berupa, yaitu suatu file atau *database* di sistem komputer, suatu arsip atau catatan manual, suatu kotak tempat data di meja seseorang, suatu tabel acuan manual, dan suatu agenda atau buku.

2.2.4.4 Kamus Data

Kamus data merupakan sebuah daftar yang terorganisasi dari elemen data yang

berhubungan dengan system, dengan definisi yang tegar dan teliti sehingga pemakai dan analis system akan memiliki pemahaman yang umum mengenai input, output, komponen penyimpanan, dan bahkan kalkulasi inter-mediate.

Saat ini kamus data hampir selalu diimplementasikan sebagai bagian dari sebuah "peranti desain dan analisis terstruktur" *CASE*. Meskipun format kamus bervariasi dari peranti satu ke peranti yang lain, sebagian besar berisi informasi berikut ini :

1. *name* – nama sebenarnya dari data atau *item* kontrol, penyimpanan data, atau entitas eksternal.
2. aliasi – nama lain yang digunakan untuk entri pertama.
3. *where-used/how-used* – suatu daftar yang menggunakan data atau *item* kontrol dan bagaimana dia digunakan (misalnya, input ke proses, *output* dari proses, sebagai suatu penyimpanan, sebagai suatu entitas eksternal)
4. *content description* – suatu notasi untuk merepresentasikan isi.
5. *supplementary information* – informasi lain mengenai tipe data, harga preset (bila diketahui), batasan, dan lain-lain.

Sekali objek data atau kontrol *item name* dan aliasnya dimasukkan ke dalam kamus data, konsistensi dalam penamaan dapat ditingkatkan. Dimana, bila anggota tim analisis memutuskan untuk menamai *item* data yang baru saja ditarik sebagai **xyz**, tetapi **xyz** ternyata sudah ada di dalam kamus, peranti *CASE* yang mendukung kamus tersebut akan mengeluarkan peringatan dan menunjukkan bahwa nama itu telah ada. Hal ini meningkatkan konsistensi dari model analisis dan membantu kita untuk mengurangi kesalahan.

Informasi "*Where-used/how-used*" direkam secara otomatis dari model aliran.

Pada entri kamus dibuat, peranti men-*scan* DFD dan SFD untuk menentukan proses yang mana menggunakan data dan informasi kontrol dan bagaimana menggunakannya. Meskipun hal ini kelihatannya tidak penting, tetapi sebenarnya ini merupakan salah satu keuntungan penting dari kamus. Selama analisis selalu ada aliran perubahan yang terus menerus. Banyak perekayasa perangkat lunak bertanya, “Dimana objek data ini digunakan? Apa lagi yang harus berubah bila kita akan melakukan modifikasi terhadapnya? Pengaruh keseluruhan apakah yang akan terjadi sebagai akibat dari perubahan? Karena kamus data diperlakukan sebagai sebuah *database*, analis dapat bertanya *“where-used/how-used”*, dan memperoleh jawaban ke *query* yang ditulis di atas.

Notasi yang digunakan untuk mengembangkan diskripsi isi, yang diilustrasikan di dalam tabel 2.1, memungkinkan analis untuk mempresentasikan *data komposit* (misalnya, objek data) di dalam salah satu dari tiga cara fundamental yang dapat dikonstruksi olehnya:

1. sebagai sebuah urutan *item* data,
2. sebagai suatu pilihan dari antara serangkaian item data, atau
3. sebagai sebuah kelompok pengulangan *item* data.

Tabel 2.1 ***Notasi deskripsi isi untuk kamus data*** **(Roger S Presman, "Rekayasa Perangkat Lunak")**

Konstruk Data	Notasi	Arti
Berurutan	=	disusun atas
	+	dan
Pilihan	[|]	baik ini - atau
Pengulangan	{ }n	pengulangan ke-*n* dan
	( )	data opsional
	. .	komentar tidak dibatasi

Masing-masing entri *item* data dapat direpresentasikan sebagai bagian dari urutan, seleksi, dan pengulangan dapat menjadi objek data lain yang memerlukan penyaringan lebih jauh lagi di dalam kamus.

Untuk sistem berbasis komputer yang besar, kamus data berkembang dengan pesat dalam ukuran dan kompleksitasnya. Kenyataannya, sangat sulituntuk menjaga kamus secara manual. Karena alasan inilah mengapa peranti *CASE* harus digunakan.

2.2.5 *Intranet* dan Aplikasi *Web*

Intranet (*Internal Network*) mulai terdengar pada pertengahan tahun 1995 oleh beberapa penjual produk jaringan yang mengacu pada kebutuhan informasi dalam bentuk *web* dalam perusahaan. *Intranet* adalah jaringan komputer dalam perusahaan yang menggunakan komunikasi data standar seperti dalam *internet*. Artinya, kita dapat menggunakan semua fasilitas *internet* untuk kebutuhan dalam perusahaan. Dengan kata lain intranet dapat dikatakan berinternet dalam lingkungan perusahaan. Standar *internet* yang digunakan adalah standar protokol TCP/IP (*Transport Control Protocol*, ada yang

menyebutnya *Transmission Control Protocol/Internetworking Protocol*), TCP/IP memungkinkan protokol jaringan untuk berkomunikasi, menerima dan mengirimkan data ke terminal yang lain. Standar yang lain adalah FTP (*File Transfer Protocol*) yang merupakan pelayanan *resource sharing*, sebuah fasilitas untuk dapat mengambil file yang ada di internet. SMTP (*Simple Mail Transfer Protocol*) yang merupakan dasar dari *e-mail* untuk berkomunikasi serta MIME (*Multipurpose internet Mail Extensions*) yang merupakan standar untuk mendefinisikan format biner, grafik dan suara agar dapat ditransmisikan dengan *e-mail*. Protokol lainnya adalah NNTP (*Network News Transfer Protocol*) dan POP (*Post Office Protocol*).[4]

Pada awalnya aplikasi *Web* dibangun hanya dengan menggunakan bahasa yang disebut HTML (*HyperText Markup Language)*. Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML. Pada saat ini, banyak skrip seperti itu, antara lain yaitu PHP dan ASP, sedangkan contoh yang berupa objek adalah *applet*. Aplikasi *Web* itu dapat dibagi menjadi *Web* statis dan *Web* dinamis. *Web* statis dibentuk dengan menggunakan HTML saja. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus-menerus untuk mengikuti setiap perubahan yang terjadi. Kelemahan ini diatasi dengan model *Web* dinamis. Dengan menggunakan pendekatan *Web* dinamis, dimungkinkan untuk membentuk sistem informasi berbasis *web*. Dari sisi teknologi yang digunakan untuk membentuk *web* dinamis terdapat dua pengelompokan, yaitu teknologi pada sisi *client* dan teknologi pada sisi server.

Teknologi *Web* pada sisi *client* diimplementasikan dengan mengirimkan kode perluasan HTML atau program tersendiri dan HTML ke *client*. *Client*lah yang

bertanggung jawab dalam melakukan proses terhadap seluruh kode yang diterima. Kelemahan pendekatan seperi ini adalah terdapat kemungkinan bahwa *browser* pada *client* tidak mendukung fitur kode perluasan HTML. Kelebihan teknologi pada sisi *client*, yaitu memungkinkan penampilan yang bersifat dinamis. Contoh teknologi pada sisi *client*, yaitu Kontrol ActiveX, Java Applet, dan Skrip sisi-*client*. Teknologi *Web* pada sisi server memungkinkan pemrosesan kode di dalam server sehingga kode yang sampai pada pemakai berbeda dengan kode asli pada server. Contoh teknologi yang berjalan di server, yaitu CGI, ASP, JSP, PHP dan lain sebagainya. Keuntungan penggunaan teknologi pada sisi server adalah sebagai berikut:

1. Mengurangi lalu lintas jaringan dengan cara menghindari percakapan bolak-balik antara *client* dan *server*.
2. Mengurangi waktu pemuatan kode, menginat *client* hanya mengambil kode HTML saja.
3. Mencegah masalah ketidakkompatibelan *browser*.
4. *Client* dapat berinteraksi dengan data yang ada pada *server*.
5. Mencegah *client* mengetahui rahasia kode (mengingat kode yang diberikan ke *client* berbeda dengan kode asli pada *server*).[5]

1.. PHP dan MySQL

PHP merupakan bahasa pemrograman untuk *script web server-side*. Bahasa pemrograman PHP diciptakan pertama kali oleh Rasmus Lerdorf, seorang pemrogram C yang sangat handal. Semula PHP hanya digunakan untuk mencatat seberapa jumlah pengunjung pada *hompage*-nya. Rasmus adalah salah seorang pendukung *open source*.

Karen itulah ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis atau *freeware* pada tahun 1995. Setelah mempelajari YACC dan GNU Bison, Rasmus menambah kemampuan pada PHP 1.0 dan menertbitkan PHP 2.0 sebagai pengembangan dari PHP 1.0 yang telah ada sebelumnya.

PHP 2.0 mampu berhubungan dengan *database* dan dapat diintegrasikan dengan HTML. Pada tahun 1996, PHP telah digunkana oleh banyak *website* di dunia. Sebuah kelompok pengembang *software* yang terdiri dari Rasmus, Zeew Suraski, Andi Gutman, Stig Bakken, Shane Carveo dan Jim Winstead bekerja selama tujuh bulan untuk melakukan penyempurnan terhadap PHP 2.0, akhirnya pada tanggal 6 juni 1998, PHP 3.0 resmi dikeluarkan ke dunia pemrograman. Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai. Versi ini banyak dipakai sebab versi ini mampu dipakai untuk membangun aplikasi *web* kompleks tetapi tetap memiliki kecepatan proses dan stabilitas yang tinggi. PHP versi 4.2 telah diterbitkan pada tanggal 22 April 2002 dengan log kelompok fungsi, sampai dengan versi 4.3.7 tercatat 125 kelompok fungsi yang dimiliki oleh PHP. Pada juni 2004, Zend merilis PHP 5.0. Versi ini adalah versi yang paling mutakhir dari PHP.

Kelebihan PHP

PHP memiliki beberapa kelebihan yang tidak dimiliki oleh bahasa-bahasa sejenisnya, yaitu:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaanya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana - mana dari mulai IIS

sampai dengan apache, dengan configurasi yang relatif mudah.

3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahamanan, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*linux*, *unix*, *windows*) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

MySQL (*My Structure Query Language*) adalah sebuah program pembuat *database* yang bersifat *open source*. MySQL sebenarnya produk yang berjalan pada *platform Linux*. Karen sifatnya *open source*, maka MySQL dapat dijalankan pada semua *platform* baik *Windows* maupun *Linux*. Selain itu, MySQL juga merupakan program pengakses *database* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi *Multi User* (Banyak Pengguna). Saat ini *database* MySQL telah digunakan hampir oleh semua *programer database*, apalagi dalm pemrograman *web*.

Kelebihan lain dari MySQL adalah menggunakan bahasa *Query* standar yang dimiliki SQL (*Structure Query Language*). SQL adalah suatu bahasa permintaan yang terstruktur yang telah distandarkan untuk semua program pengakses *database* seperti Oracle, Posgres SQL, SQL Server, dan lain-lain.

Sebagai sebuah program penghasil *database*, MySQL tidak dapat berjalan sendiri tanpa adanya sebuah aplikasi lain (*interface*). MySQL dapat didukung oleh hampir semua program aplikasi baik yang *open source* seperti PHP maupun yang tidak, yang ada pada *platform Windows* seperti Visual Basic, Delphi, dan lainnya.[6]

2.2.7 Pengelolaan Data Spasial (peta)

Data spasial merupakan sekumpulan data yang saling berkaitan dengan masalah keruangan, baik diatas permukaan bumi ataupun dibawah pemukaan bumi, seperti area, lokasi, detail, dan lain-lain. Data ini merupakan hasil pengamatan dan pengidentifikasian dilapangan yang hasilnya digambarkan dalam bentuk grafik dan dutunjukan oleh peta-peta dengan skala dan proyeksi tertentu. Pengolahan data spasial ini yang akan mengolah data atribut dari data spasial (peta) sendiri, seperti ID, kelas, tema, judul, skala, wilayah administrasi, pemilik, dan lain-lain.

Metadata dapat diartikan sebagai data tentang data (spasial), berisikan informasi mengenai karakteristik data dan memegang peran penting di dalam mekanisme pertukaran data. Mekanisme pertukaran data ini diatur oleh koordinator IDSD Provinsi Jawa Barat dalam hal ini Badan Perencanaan Daerah (BAPEDA) Provinsi Jawa Barat. Metadata dipergunakan untuk melakukan dokumentasi data spasial yang berhubungan tentang siapa, apa, kapan, dimana, dan bagaimana data spasial dipersiapkan.

Tujuan standar metadata ini adalah untuk memberikan istilah dan definisi yang sama untuk konsep-konsep yang berhubungan dengan metadata -metadata ini. Selain itu standar ini juga dibuat untuk dokumentasi data spasial digital. Standar memberikan nama-nama unsur dan kelompok unsur-unsur data yang digunakan untuk tujuan-tujuan tertentu, definisi-definisi kelompok unsur dan unsur-unsur data, dan informasi mengenai nilai-nilai yang diberikan untuk unsur-unsur data.

Dalam pekerjaan Pengembangan IDSD yang ada dilingkungan Dinas TARKIM, diharapkan data spasial yang telah terkumpul dan telah dilakukan analisis serta

dilakukan penyeragaman data juga dalam metadata IDSD, dapat memperkaya data spasial yang ada dilingkungan Provinsi Jawa Barat khususnya yang sesuai dengan Tugas Pokok dan Fungsi Dinas Tata Ruang dan Permukiman Provinsi Jawa Barat.