# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang Masalah

Desa sebagai bagian terkecil dari sistem pemerintahan administratif di Indonesia masih melakukan pekerjaan konvensional dalam hal pengolahan serta penyampaian data dan informasi yang berhubungan dengan desa, serta komunikasi dengan penduduknya. Hal ini terjadi karena kurangnya teknologi informasi komunikasi (TIK) di kantor desa, kurangnya sumber daya manusia (SDM), besarnya anggaran yang dibutuhkan untuk mengimplementasikan TIK sampai dengan pembiayaan *maintenance*. Oleh karena itu dibutuhkan suatu sistem untuk mempermudah proses penyampaian informasi dan komunikasi yang efisien di sisi biaya dan pengoperasiannya.

*Cloud computing* adalah suatu paradigma yang memungkinkan pengguna untuk melakukan akses secara *online* terhadap informasi. *Cloud computing* menawarkan kolaborasi arsitektur *Software as a Service* (SaaS), layanan *on demand*, serta kemampuan akses kapan pun dan di mana pun. Dengan adanya *cloud computing* maka dapat dibangun suatu arsitektur layanan pengolahan dan akses informasi desa yang bernama E-Desa. E-Desa merupakan *public service* yang menerapkan *cloud computing,* yaitu berupa layanan SaaS dalam bentuk *Content Management System* (CMS) dan layanan *Infrastruktur as a Service* (IaaS) sebagai media penyimpanan data. CMS E-Desa menyediakan fitur-fitur sesuai kebutuhan desa dan dapat digunakan oleh banyak desa.

Di dalam arsitektur layanannya, *cloud provider* akan menyediakan infrastruktur dan penyimpanan data secara terpusat di *cloud server*. Setiap desa yang menyewa paket layanan E-Desa mempunyai media penyimpanan datanya sendiri di dalam *cloud server*. Pengguna tidak perlu tahu bagaimana dan darimana sumber daya komputasinya dipenuhi oleh *cloud provider*. Selain itu, pemerintah desa tidak perlu direpotkan dengan penyediaan *server, storage, software, maintenance* sistem, serta tenaga IT. Pemerintah desa cukup berlangganan layanan dengan memilih paket yang disediakan sesuai kebutuhan serta membayar biaya sewa rutin, kemudian mengisi dan mengolah konten didalamnya.

Harapan dengan adanya E-Desa adalah agar terciptanya keseragaman dan kemudahan dalam hal pengolahan informasi desa sehingga akses informasi desa menjadi lebih cepat, mudah, murah, dapat dilakukan kapan saja dan di mana saja, serta terciptanya jaringan desa yang kuat yang mampu berbagi informasi. Berdasarkan latar belakang yang telah dijelaskan, maka penelitian tugas akhir ini diberi judul yaitu **Rancang Bangun Arsitektur *Cloud Computing* pada E-Desa sebagai Layanan Akses Informasi Desa**.

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan, permasalahan yang akan diselesaikan adalah:

1. Lambatnya pengolahan dan penyampaian data dan informasi desa, serta komunikasi dengan penduduknya, karena kurangnya TIK, SDM, dan

anggaran di kantor desa, sehingga baik pihak pemerintah ditingkat atas desa maupun penduduk sulit mengakses informasi desa yang dibutuhkan.

2. Belum adanya implementasi arsitektur *cloud computing* untuk layanan di desa yang menyediakan aplikasi-aplikasi, *maintenance system,* infrastruktur dan penyimpanan data secara terpusat di *cloud server.*
3. Belum adanya sebuah CMS E-Desa berbasis *cloud computing* sebagai *public service* yang seragam, siap pakai, *on demand,* mampu mengolah informasi desa serta bisa digunakan oleh banyak desa.

## 1.3 Maksud dan Tujuan

Berdasarkan permasalahan diatas, maka maksud dari ditulisnya skripsi ini adalah untuk merancang dan membangun sebuah arsitektur *cloud computing* pada E-Desa sebagai layanan akses informasi desa secara *online.*

Adapun tujuan yang akan dicapai dari penulisan skripsi ini adalah :

1. Mempermudah pemerintah desa dalam mengolah data dan informasi desa yang siap disampaikan baik kepada Pemerintah ditingkat atas desa maupun penduduk, karena layanan yang berbasis *web* sehingga penyampaian informasi lebih cepat, mudah, murah, serta dapat dilakukan kapan saja dan di mana saja.
2. Mengimplementasikan sebuah arsitektur *cloud computing* untuk layanan di desa yang menyediakan layanan, aplikasi-aplikasi, *maintenance system,* infrastruktur dan penyimpanan data secara terpusat di *cloud server.*

3. Menghasilkan sebuah CMS E-Desa berbasis *cloud computing* sebagai *public service* dengan fitur-fitur serta modul-modul siap pakai, *on demand*, mampu mengolah informasi desa, menciptakan keseragaman dalam hal pengolahan informasi desa, serta bisa digunakan oleh banyak desa.

## 1.4 Batasan Masalah/Ruang Lingkup Kajian

Adapun batasan masalah dalam pembuatan sistem ini adalah :

1. Studi kasus saat ini yaitu di beberapa desa yang ada pada Kecamatan Cimenyan Kabupaten Bandung, yaitu Desa Ciburial dan Desa Cikadut.
2. Sistem :
    a. Menerapkan *Software as a Service* dan *Infrastructure as a Service* dalam merancang dan membangun arsiektur *cloud computing* pada E-Desa.
    b. *Software as a Service* (SaaS) yang diberikan adalah CMS (*Content Management System*) dengan fitur-fitur di dalamnya.
    c. Memanfaatkan layanan *Infrastruktur as a Service* (IaaS) dalam penyimpanan data, sumber daya pemroses dan control terhadap SaaS.
    d. Menyediakan layanan yang *customize* dengan pembagian paket-paket berdasarkan layanan yang dibutuhkan, harga sewa dan *storage*.
    e. Layanan E-Desa yaitu sebuah CMS dengan fitur-fitur serta modul-modul siap pakai yang disediakan untuk pemerintah desa meliputi manajemen informasi desa (seperti inforrmasi umum, letak geografis, aparatur desa, data monografi, potensi, dan berita terbaru desa),

manajemen konten dan tema web, penyimpanan arsip desa, dan forum komunikasi.

f. Layanan di dalam situs E-Desa dari sebuah desa yang dapat diakses oleh masyarakat umum meliputi akses informasi, *download* arsip desa, kirim email ke desa, dan forum komunikasi.

g. Sistem yang dibangun akan berintegrasi dengan *google map* dan situs jejaring sosial seperti *facebook* dan *twitter*.

3. Pemodelan sistem yang digunakan yaitu menggunakan pemodelan terstruktur dimana untuk menggambarkan hubungan antara objek data menggunakan ERD (*Entity Relationship Diagram*) dan untuk menggambarkan fungsi-fungsi (dan sub-fungsi) yang mentransformasi aliran data menggunakan DFD (*Data Flow Diagram*).

4. Perangkat lunak penunjang :

   a. Sistem operasi Windows XP SP 3.

   b. Bahasa pemograman untuk membangun CMS menggunakan XHTML (*Extensible Hypertext Markup Language*), CSS (*Cascading Style Sheet*), PHP (*Hypertext Preprocessing*), Javascript, dan AJAX.

   c. Tools pembangun aplikasi:

      1) Macromedia Dreamweaver 8 sebagai editor teks untuk pemrograman berbasis *web*.

      2) WampServer 2.01 yang berisi Apache sebagai web server, MySQL sebagai database server, dan PhpMyadmin sebagai pengelolaan *database*.

3) *Browser* seperti Mozilla Firefox 10, Google Chrome 5, dan Opera 10 untuk mengakses aplikasinya.

d. Menggunakan VPS (*Virtual Private Server*) dengan sistem operasi Debian 64 bit sebagai hosting serta menggunakan Filezilla untuk mengaksesnya.

## 1.5 Metedologi Penelitian

Metodologi penelitian yang akan digunakan adalah sebagai berikut :

**1. Tahap pengumpulan data**

a. Studi Literatur

Pengumpulan data dengan cara mengumpulkan literatur, jurnal, paper dan bacaan-bacaan yang ada kaitannya dengan judul penelitian.

b. Studi Lapangan

1) Observasi

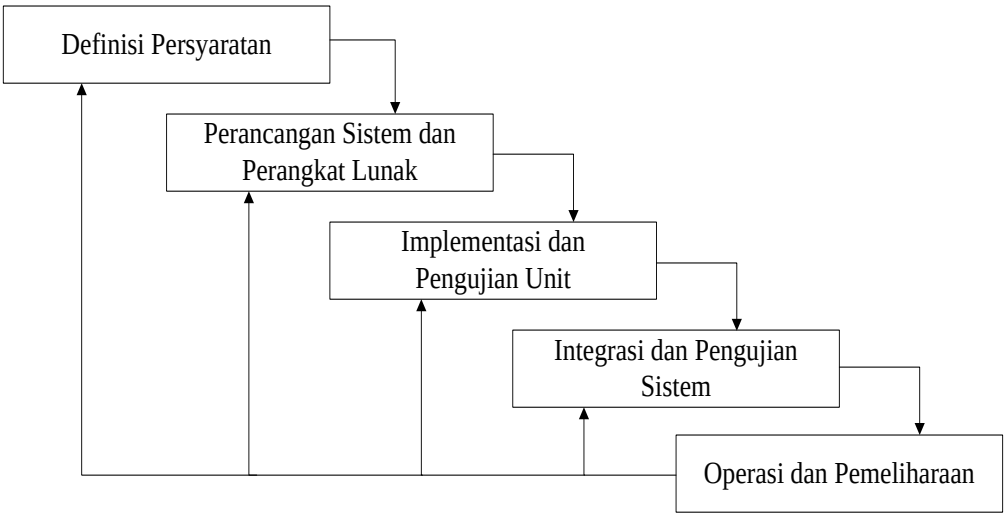
Teknik pengumpulan data dengan mengadakan penelitian dan tinjauan langsung terhadap permasalahan yang diambil.

2) Wawancara

Teknik pengumpulan data dengan tanya jawab secara langsung yang ada kaitannya dengan tema yang diambil.

**2. Metodologi perkembangan perangkat lunak**

Model yang digunakan untuk membangun dan mengembangkan aplikasi arsitektur *cloud computing* pada E-Desa mengadopsi dari model *waterfall*. Model ini melingkupi aktivitas-aktivitas yang dapat dilihat pada gambar 1.1.




**Gambar 1.1 Metode *Waterfall* [1]**

a. Definisi Persyaratan

   Pelayanan, batasan, dan tujuan sistem ditentukan melalui konsultasi dengan *user* sistem.

b. Perancangan Sistem dan Perancangan Perangkat Lunak

   Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungan-hubungannya.

c. Implementasi dan Pengujian Unit

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian unit melibatkan verifikasi bahwa setiap unit telah memenuhi spesifikasinya.

d. Integrasi dan Pengujian Sistem

Unit program/program individual diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa persyaratan sistem telah dipenuhi. Setelah pengujian sistem, perangkat lunak dikirim kepada pelanggan.

e. Operasi dan Pemeliharaan

Ini merupakan fase siklus hidup yang paling lama. Sistem di-*install* dan dipakai. Pemeliharaan mencakup koreksi dari berbagai *error* yang tidak ditemukan pada tahap-tahap terdahulu, perbaikan atas implementasi unit sistem dan pengembangan pelayanan sistem, sementara persyaratan-persyaratan baru ditambahkan.

## 1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penulisan laporan skripsi ini adalah sebagai berikut :

**BAB I PENDAHULUAN**

Bab ini berisi latar belakang, identifikasi masalah, maksud dan tujuan, batasan masalah, metodologi penelitian dan sistematika penulisan.

**BAB II LANDASAN TEORI**

Bab ini berisi data-data tempat penelitian berlangsung serta teori-teori pendukung yang menunjang dalam perancangan dan pembangunan sistem.

**BAB III ANALISIS DAN PERANCANGAN SISTEM**

Bab ini berisi analisis pada sistem yang sedang berjalan, deskripsi sistem yang akan dibangun, analisis kebutuhan dalam pembagunan sistem serta perancangan sistem yang dikembangkan.

**BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM**

Bab ini berisi implementasi sistem yang dibangun, ujicoba dan hasil pengujian sistem.

**BAB V KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan yang diperoleh dari hasil pengujian sistem, serta saran perbaikan dan pengembangan sistem ke depan.

# BAB II

# LANDASAN TEORI

## 2.1 Definisi Desa [2]

Desa, berdasarkan Undang-undang Nomor 32 Tahun 2004 tentang Pemerintahan Daerah, adalah kesatuan masyarakat hukum yang memiliki batas-batas wilayah yurisdiksi, berwenang untuk mengatur dan mengurus kepentingan masyarakat setempat berdasarkan asal-usul dan adat istiadat setempat yang diakui dan/atau dibentuk dalam sistem Pemerintahan Nasional dan berada di kabupaten/kota, sebagaimana dimaksud dalam Undang-Undang Dasar Negara Republik Indonesia Tahun 1945. Landasan pemikiran dalam pengaturan mengenai desa adalah keanekaragaman, partisipasi, otonomi asli, demokratisasi dan pemberdayaan masyarakat.

Kewenangan desa adalah:

1. Menyelenggarakan urusan pemerintahan yang sudah ada berdasarkan hak asal usul desa

2. Menyelenggarakan urusan pemerintahan yang menjadi kewenangan kabupaten/kota yang diserahkan pengaturannya kepada desa, yakni urusan pemerintahan yang secara langsung dapat meningkatkan pelayanan masyarakat.
3. Tugas pembantuan dari Pemerintah, Pemerintah Provinsi, dan Pemerintah Kabupaten/Kota
4. Urusan pemerintahan lainnya yang diserahkan kepada desa.

Undang-undang Nomor 32 Tahun 2004 tentang Pemerintahan Daerah mengakui otonomi yang dimiliki oleh desa ataupun dengan sebutan lainnya dan kepada desa melalui pemerintah desa dapat diberikan penugasan ataupun pendelegasian dari Pemerintah ataupun pemerintah daerah untuk melaksanakan urusan pemerintah tertentu. Sedang terhadap desa di luar desa geneologis yaitu desa yang bersifat administratif seperti desa yang dibentuk karena pemekaran desa ataupun karena transmigrasi ataupun karena alasan lain yang warganya pluralistis, majemuk, ataupun heterogen, maka otonomi desa akan diberikan kesempatan untuk tumbuh dan berkembang mengikuti perkembangan dari desa itu sendiri.

Kepala Desa pada dasarnya bertanggung jawab kepada rakyat Desa yang dalam tata cara dan prosedur pertanggungjawabannya disampaikan kepada Bupati atau Walikota melalui Camat. Kepada Badan Permusyawaratan Desa, Kepala Desa wajib memberikan keterangan laporan pertanggungjawabannya dan kepada rakyat menyampaikan informasi pokok-pokok pertanggungjawabannya namun tetap harus memberi peluang kepada masyarakat melalui Badan Permusyawaratan Desa untuk menanyakan dan/atau meminta keterangan lebih lanjut terhadap hal-hal yang bertalian dengan pertanggungjawaban dimaksud.

Pemerintahan Desa adalah penyelenggaran urusan pemerintahan oleh Pemerintah Desa dan Badan Permusyawaratan Desa dalam mengatur dan mengurus kepentingan masyarakat setempat berdasarkan asal-usul dan adat istiadat setempat yang diakui dan dihormati dalam sistem Pemerintahan Negara Kesatuan Republik Indonesia.

Profil Desa/Kelurahan adalah gambaran menyeluruh tentang karakter Desa dan Kelurahan yang meliputi data dasar keluarga, potensi sumber daya alam, sumber daya manusia, kelembagaan, prasarana dan sarana serta perkembangan kemajuan dan permasalahan yang dihadapi desa dan kelurahan.

Potensi Desa/Kelurahan adalah keseluruhan sumber daya yang dimiliki atau digunakan oleh Desa dan Kelurahan baik sumber daya alam, sumber daya manusia dan kelembagaan maupun prasarana dan sarana untuk mendukung percepatan

kesejahteraan masyarakat. Ruang lingkup data potensi Desa/Kelurahan terdiri dari 4 (empat) variabel yaitu [3]:

1. Potensi Sumber Daya Alam;
2. Potensi Sumber Daya Manusia;
3. Potensi Kelembagaan;
4. Potensi Prasarana dan Sarana.

### 2.1.1 Monografi Desa [4]

Dalam rangka mewujudkan penyelenggaraan Pemerintahan Desa yang efektif dan efisien perlu penyajian data administrasi pemerintahan desa secara menyeluruh, terpadu, akurat dan dapat dipertanggungjawabkan kebenarannya yang disusun dalam monografi desa dan kelurahan. Monografi desa adalah himpunan data yang dilaksanakan oleh pemerintah desa yang tersusun secara sistematis, lengkap, akurat, dan terpadu dalam penyelenggaraan pemerintahan, pelaksanaan pembangunan dan pembinaan kemasyarakatan.

Monografi Desa bertujuan untuk:

1. Memperoleh data dan informasi mengenai penyelenggaraan pemerintahan, pelaksanaan pembangunan, dan pembinaan masyarakat secara sistematis, lengkap, akurat, dan terpadu.
2. Memudahkan pengolahan data dan informasi yang siap disampaikan kepada Pemerintah ditingkat atas desa.
3. Memfungsikan Desa dalam kedudukannya sebagai sumber data dan informasi dalam penetapan berbagai kebijaksanaan pemerintah.

Informasi yang dimuat dan perlu didata dalam suatu Data Monografi Desa diantaranya:

A. Bidang Pemerintahan

Bidang pemerintahan berisi data umum desa, data keuangan desa, dan data kewenangan yang dilaksanakan oleh pemerintahan desa serta penyelenggara pemerintahan desa. Bidang pemerintahan memuat poin-

poin berikut:

1. Umum
2. Pertanahan
3. Kependudukan
4. Jumlah Perangkat Desa/Kelurahan
5. Pembinan RT/RW
6. Jumlah Pelayanan Masyarakat
7. Pajak/Retribusi
8. Badan Perwakilan Desa (BPD)
9. Peraturan Desa
10. Keputusan Kepala Desa
11. Keuangan dan Sumber-Sumber Pendapatan Desa
12. Kemanan Desa
13. Survey dan Penelitian

B. Bidang Pembangunan

Bidang pembangunan berisi data sarana dan prasarana, transportasi, komunikasi, inventarisasi serta potensi desa yang menunjang pembangunan suatu desa. Bidang pembangunan memuat poin-poin berikut:

1. Agama
2. Kesehatan
3. Pendidikan
4. Sarana Olah Raga/Kesenian/Kebudayaan dan Sosial
5. Prsarana Perhubungan
6. Komunikasi

7. Alat Transportasi
8. Industri
9. Pariwisata
10. Pengairan
11. Pertanian
12. Perkebunan
13. Pertanaman dan Lingkungan Hidup
14. Perikanan
15. Peternakan
16. Kehutanan
17. Pertambangan Bahan Galian
18. Perdagangan/Jasa
19. Perkoperasian
20. Perumahan dan Jenis Komplek Pemukiman
21. Jumlah Proyek Desa yang Dibiayai
22. Kejuaraan Lomba Desa yang Pernah Didapat
23. Typologi Desa
24. Kelembagaan Desa
25. Inventrisasi Bidang Teknologi Tepat Guna yang Telah dimanfaatkan

C. Bidang Kemasyarakatan

Bidang kemasyarakatan berisi data-data yang berhubungan dengan penduduk desa dan sumber daya manusia yang ada di suatu desa. Bidang kemasyarakatan memuat poin-poin berikut:

1. Keagamaan
2. Kesehatan
3. Olahraga

4. Kesenian
5. Organisasi Sosial
6. Tenaga Kerja
7. Transmigrasi

### 2.1.2 Desa Ciburial

Desa Ciburial Kecamatan Cimenyan sebagai salah bagian dari Kabupaten Bandung mempunyai luas wilayah 599,216 Ha. Terletak pada ketinggian antara 750 m dampai dengan 1200 m di atas permukaan laut. Secara administratif wilayah Desa Ciburial berbatasan di:

1. Sebelah Utara : Kabupaten Bandung Barat
2. Sebelah Selatan : Kotamadya Bandung
3. Sebelah Barat : Kabupaten Bandung Barat
4. Sebelah Timur : Desa Mekarsaluyu Kecamatan

Secara topografis Desa Ciburial merupakan kawasan yang berbukit dengan variasi kemiringan antara 0,8 %, 8,15 % hingga 40 %. Wilayah ini merupakan salah satu wilayah tangkapan air yang berfungsi menjaga keseimbangan hidrolis cekungan Bandung.

Data wilayah administrasi Desa Ciburial:

| | | |
|---|---|---|
| Kabupaten (kode) | : | Bandung (32) |
| Kecamatan (kode) | : | Cimenyan (06) |
| Desa (kode) | : | Ciburial (009) |
| Jumlah RT | : | 48 (empat puluh delapan) * |
| Jumlah RW | : | 12 (dua belas) * |
| Jumlah Dusun | : | III (Tiga) * |
| Jumlah Penduduk (Jiwa) | : | 11.525 Jiwa * |
| Jumlah Kepala Keluarga (KK) | : | 3.307 KK * |

*) = data per Oktober 2009

#### *2.1.2.1 Visi dan Misi*

Visi Desa Ciburial adalah **"**Terwujudnya Masyarakat Desa Ciburial yang Sejahtera dan Dinamis dalam Nuansa Religius dan Berwawasan Lingkungan sebagai Desa Pendidikan dan Wisata**"**. Adapun makna dari Visi tersebut adalah:

1. Terwujudnya Masyarakat Desa Ciburial yang Sejahtera dan Dinamis, mengandung arti kondisi Desa Ciburial yang masyarakatnya memiliki keberdayaan secara sosial dan ekonomi sehingga mampu melangsungkan kehidupan individu maupun kemasyarakatan secara layak serta senantiasa penuh tenaga dan semangat untuk cepat bergerak dan mudah menyesuaikan diri dengan keadaan untuk terus menuju perubahan ke arah yang lebih baik.
2. Dalam Nuansa Religius dan Berwawasan Lingkungan, mengandung arti bahwa segala aktivitas kehidupan di Desa Ciburial senantiasa dijiwai oleh nilai-nilai budaya, norma, dan agama dan kepedulian terhadap lingkungan.
3. Sebagai Desa Pendidikan dan Wisata, Desa Ciburial berdasarkan potensi kewilayahan merupakan desa tempat diselenggarakannya berbagai kegiatan pendidikan dan kegiatan pariwisata yang berkualitas.

Untuk terwujudnya visi tersebut empat upaya/cara atau misi yang akan mendukung pencapaiannya yaitu:

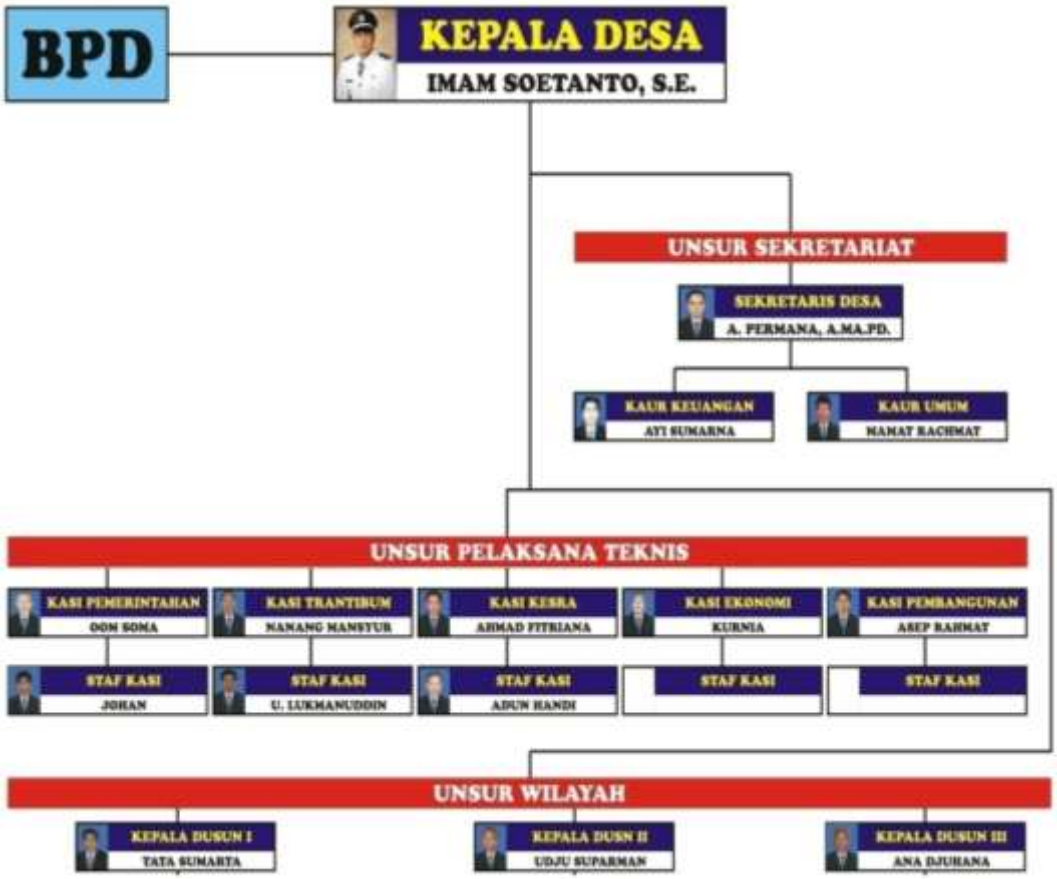
1. Mewujudkan Pemerintah Desa yang Bersih, Amanah, dan Transparan serta Berorientasi Pelayanan kepada Masyarakat.
2. Meningkatkan Kesejahteraan Sosial dan Ekonomi Masyarakat.
3. Meningkatkan Kualitas Sumber Daya Manusia yang Berbasis Iman dan Takwa.

4. Mewujudkan Lingkungan Masyarakat yang Bersih, Aman, dan Tertib.

### *2.1.2.2 Struktur Organisasi Perangkat Desa*

Perangkat Desa adalah unsur-unsur pembantu Kepala Desa yang terdiri dari unsure staf, pelaksana teknis dan kewilayahan dalam melaksanakan tugas dan wewenang kepala desa. Berikut diterangkan data perangkat Desa Ciburial tahun 2011 menurut keputusan Kepala Desa Ciburial No. 01/Kep-Des/2011.

1. Kepala Desa : Imam Soetanto, S. E.
2. Sekretaris Desa : Asep Permana, A.MA.PD.
3. Kepala Seksi Pemerintahan : Oom Soma
4. Kepala Seksi Perekonomian : Kurnia
5. Kepala Seksi Trantib : Nanang Mansyur
6. Kepala Seksi Kesra : Adun Handi
7. Kepala Seksi Pembangunan : Asep Rahmat
8. Kepala Usuran Umum : Ahmad Fitriana
9. Kepala Urusan Keuangan : Ayi Sumarna
10. Kepala Dusun I : Tata Sumarta
11. Kepala Dusun II : Udju Suparman
12. Kepala Dusun III : Ana Djuhana
13. Staf Seksi Pemerintahan : Johan
14. Staf Seksi Trantib : Ujang Lukmanuddin
15. Staf Seksi Kesra : Mamat Rachmat




**Gambar 2.1 Struktur Organisasi Desa Ciburial**

### 2.1.3 Desa Cikadut

Desa Cikadut Kecamatan Cimenyan sebagai salah bagian dari Kabupaten Bandung mempunyai luas wilayah 295 Ha. Terletak pada ketinggian antara 700 meter dampai dengan 1200 m di atas permukaan laut. Secara administratif wilayah Desa Ciburial berbatasan di:

1. Sebelah Utara : Desa Mekar Manik
2. Sebelah Selatan : Kel. Pasir Impun
3. Sebelah Barat : Desa Mandala Mekar
4. Sebelah Timur : Desa Sindanglaya

Data wilayah administrasi Desa Cikadut:

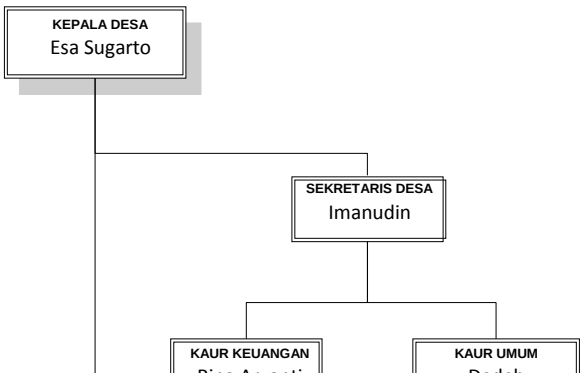
| | | |
|---|---|---|
| Kabupaten (kode) | : | Bandung (32) |
| Kecamatan (kode) | : | Cimenyan (06) |
| Jumlah RT | : | 56 (lima puluh enam) * |
| Jumlah RW | : | 14 (empat belas) * |
| Jumlah Dusun | : | III (Tiga) * |
| Jumlah Penduduk (Jiwa) | : | 8.986 Jiwa * |
| Jumlah Kepala Keluarga (KK) | : | 2567 KK * |

*) = data tahun 2011

### *2.1.3.1 Struktur Organisasi Perangkat Desa*

Berikut diterangkan data perangkat Desa Cikadut Tahun 2011 berdasarkan Perda No. 11 th. 2007.

1. Kepala Desa : Esa Sugarto
2. Sekretaris Desa : Imanudin
3. Kepala Seksi Pemerintahan : Komar S.Pd.I
4. Kepala Seksi Perekonomian : Andriani Budi H.
5. Kepala Seksi Trantib : Tini A.
6. Kepala Seksi Kesra : Lilis Herlina
7. Kepala Seksi Pembangunan : Lili
8. Kepala Usuran Umum : Dedeh Rosmawati
9. Kepala Urusan Keuangan : Rina Aryanti
10. Kepala Dusun I : Asep Rahmat S.Pd.
11. Kepala Dusun II : Uas Usnan
12. Kepala Dusun III : Tarya Suzeta



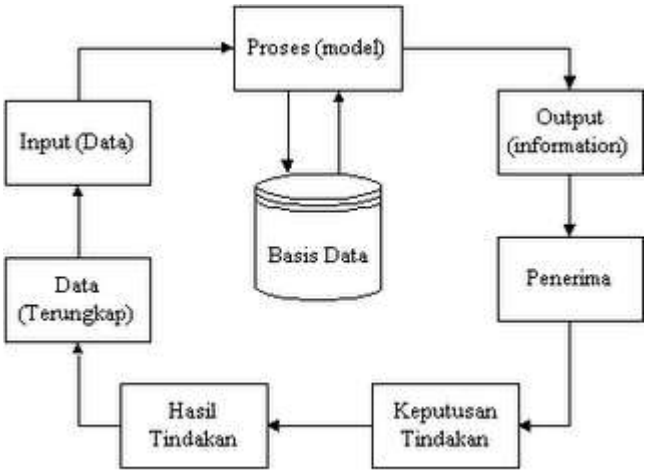
**Gambar 2.2 Struktur Organisasi Desa Cikadut**

## 2.2 Informasi [5]

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal *data-item.* Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu. Kejadian-kejadian nyata yang sering terjadi perubahan dari suatu nilai yang disebut dengan transaksi. Misalnya penjualan adalah transaksi perubahan nilai barang menjadi nilai uang. Kesatuan nyata adalah suatu objek nyata seperti tempat, benda, dan orang yang betul-betul ada dan terjadi.

### *2.2.1 Siklus Informasi*

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi. Data yang diolah untuk menghasilkan informasi menggunakan suatu model proses yang tertentu. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus ini oleh John Burch disebut dengan siklus informasi (*information cycle*). Siklus ini juga disebut sebagai siklus pengolahan data (*data processing cycle*).




**Gambar 2.3 Siklus Informasi**

### *2.2.2 Kualitas Informasi*

Kualitas dari suatu informasi (*quality of information*) tergantung dari tiga hal, yaitu informasi harus akurat (*accurate*), tepat pada waktunya (*timeliness*) dan relevan (*relevance*).

Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Tepat pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

### *2.2.3 Nilai Informasi*

Nilai dari informasi (*value of information*) ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan di dalam suatu sistem informasi umumnya digunakan untuk beberapa kegunaan.sehingga tidak memungkinkan dan sulit untuk menghubungkan suatu bagian informasi pada suatu masalah yang tertentu dengan biaya untuk memperolehnya, Karena sebagian besar informasi dinikmati tidak hanya oleh satu pihak di dalam perusahaan.

### 2.2.4 Sistem Informasi

Sistem informasi dalam sebuah sistem meliputi pemasukan data ( input ) kemudian diolah melalui suatu model dalam pemrosesan data, dan hasil informasi akan ditangkap kembali sebagai suatu input dan seterusnya sehingga membentuk siklus informasi yang dapat diperoleh dari sistem informasi sebagai sistem khusus dalam organisasi untuk mengolah informasi tersebut Sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk

mencapai suatu tujuan yaitu menyajikan informasi. Pengertian lain dari sistem informasi adalah sekumpulan prosedur organisasi yang pada saat dilaksanakan dan memberikan informasi bagi pengambil keputusan dan/atau untuk mengendalikan informasi.

### *2.2.5 Komponen Sistem Informasi*

Komponen dari sistem informasi terdiri dari blok masukan (*input blok*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok basis data (*database block*) dan blok kendali (*control block*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapainya suatu sasaran.

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 *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 secara keseluruhan. Teknologi terdiri dari 3 bagian utama yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*). Teknisi dapat berupa orang-orang yang mengetahui teknologi dan membuatnya dapat beroperasi. Misalnya teknisi adalah operator komputer, *programmer*, operator pengolah kata, spesialis telekomunikasi, analis siste, penyimpan data dan lain 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 disimpan dalam basis data untuk keperluan penyedia informasi lebih lanjut. Basis data diakses atau dimanipulasi dengan perangkat lunak paket yang disebut DBMS (*Database Management Systems*).

6. Blok Kendali

Banyak hal yang dapat merusak sistem inforamsi, seperti misalnya bencana alam, api, temperature, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, kesalahan-kesalahan, ketidakefisienan, sabotase 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 dapat langsung cepat diatasi.

## 2.3 Basis Data [6]

Basis data terdiri dari 2 kata, yaitu Basis dan Data. **Basis** kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan **Data** adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, symbol, teks, gambar, bunyi, atau kombinasinya.

Basis data dapat didefinisikan dalam sejumlah sudut pandang seperti:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Prinsip utama dari basis data adalah pengaturan data/arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Basis data juga bukan hanya sekedar disimpan dalam media penyimpanan elektronis. Artinya tidak semua bentuk penyimpanan data secara elektronis bisa disebut basis data. Media penyimpanan elektronis juga dapat berisi *file* teks (yang diolah oleh program pengolah kata), *file spreadsheet*, dan lain-lain, tetapi tidak disebut sebagai basis data karena didalamnya tidak terdapat pemilahan dan pengelompokan data sesuai jenis atau

fungsi data, sehingga akan menyulitkan pada saat pencarian data. Yang sangat diutamakan pada basis data adalah pengaturan/pemilahan/pengorganisasian/pengelompokan data yang akan kita simpan sesuai fungsi atau jenisnya. Pemilahan/pengorganisasian/pengelompokan ini dapat berupa sejumlah file atau tabel yang terpisah atau dalam bentuk pendefinisian kolom-kolom (*field*) data dalam setiap *file/*tabel.

### *2.3.1 Operasi Dasar Basis Data*

Di dalam sebuah disk, basis data dapat diciptakan dan dapat ditiadakan. Selain itu dapat pula ditempatkan beberapa (lebih dari satu) basis data. Sementara dalam sebuah basis data, kita dpat menempatkan satu/lebih file/tabel. Pada tabel inilah sesungguhnya data disimpan/ditempatkan. Setiap basis data umumnya dibuat untuk mewakili sebuah semesta data yang spesifik.

Operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data meliputi:

1. Pembuatan basis data baru (*create database*).
2. Penghapusan basis data (*drop database*).
3. Pembuatan *file*/tabel baru ke suatu basis data (*create table*).
4. Penghapusan *file*/tabel dari suatu basis data (*drop table*).
5. Penambahan/pengisian data baru ke sebuah *file*/tabel di suatu basis data (*insert*).
6. Pengambilan data dari sebuah *file*/tabel (*retrieve/search*).
7. Pengubahan data dari sebuah *file*/tabel (*update*).
8. Penghapusan data dari sebuah *file*/tabel (*delete*).

Operasi yang berkenaan dengan pembuatan objek (basis data dan tabel) merupakan operasi awal yang hanya dilakukan sekali dan berlaku seterusnya. Sedang operasi-operasi yang berkaitan dengan isi table (data) merupakan operasi rutin yang akan berlangsung berulang-ulang dan karena itu operasi-operasi inilah yang lebih tepat mewakili aktivitas pengelolaan (*management*) dan pengolahan (*processing* data dalam bisnis data.

### *2.3.2 Sifat-Sifat Basis Data*

Istilah basis data pada umunya juga menyiratkan serangkaian sifat yang terkait yaitu [7]:

1. Berbagi Data

Data yang disimpan dalam basis data tidak secara umum dipegang semata-mata untuk digunakan oleh seseorang. Suatu basis data secara normal diharapkan bisa diakses oleh lebih dari satu orang, dan mungkin dapat diakses di waktu yang sama.

2. Integrasi Data

   Salah satu bentuk tanggung jawab pemakaian basis data yang utama adalah memastikan bahwa data terintegrasi. Hal tersebut menyiratkan bahwa suatu basis data harus menjadi koleksi data yang mana setidaknya secara ideal tidak menjadi rendudansi data.

3. Integritas Data

   Tanggung jawab lain yang timbul sebagai konsekuensi dari data bersama adalah bahwa basis data perlu menunjukan integritas. Artinya, jika hubungan antar objek yang ada pada dunia nyata diwakili oleh data di dalam basis data, maka perubahan menjadi salah satu dari mitra pada suatu hubungan harus secara akurat menggambarkan perubahan yang terjadi di dalam hubungan tersebut.

4. Keamanan Data

   Salah satu cara yang utama untuk memastikan integritas basis data adalah dengan melakukan pembatasan akses, dengan kata lain pengamanan basis data. Cara yang utama tersebut telah dilakukan pada sistem basis data dengan penjelasan atas beberapa detail pada sekumpulan pengguna yang diberi hak, atau pada umumnya lebih pada bagian basis data.

5. Abstraksi Data

   Suatu basis data dipandang sebagai model nyata. Informasi yang disimpan di dalam basis data pada umumnya merupakan sebuah usaha untuk menyajikan sifat dari beberapa objek sesungguhnya. Oleh karena itu, sebuah basis dimaksudkan untuk merekam rincian aktivitas yang relevan. Dikatakan relevan karena tidak ada basis data yang bisa menyimpan semua sifat objek sesungguhnya.

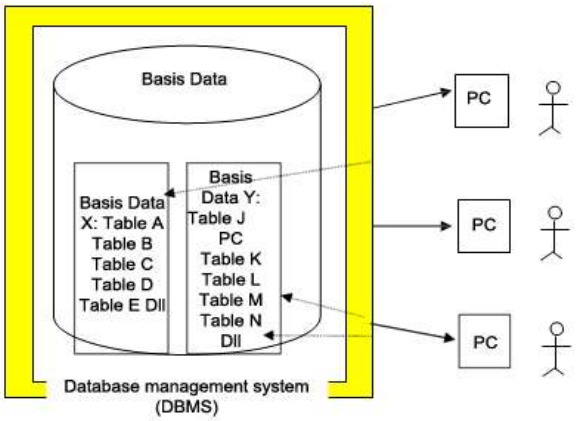
6. Independensi Data

   Salah satu konsekuensi dari abstraksi data adalah gagasan untuk *buffering data* dari proses yang menggunakan data. Sebagai contoh, tidak ada program aplikasi yang menggunakan data yang perlu diubah. Dengan demikian jika suatu perubahan dibuat ke dalam beberapa bagian dari suatu sistem aplikasi, maka semestinya perubahan itu tidak mempengaruhi struktur data dasar yang digunakan oleh aplikasi.

### 2.3.3 Sistem Basis Data

Basis data hanyalah sebuah objek yang pasif/mati. Ia tidak akan pernah berguna jika tidak ada pengelola atau pembuatnya. Yang menjadi pengelola atau penggeraknya

secara langsung adalah program atau aplikasi (*software*). Gabungan keduanya yaitu basis data dan pengelolanya akan menghasilkan suatu sistem. Karena itu, secara umum sebuah sistem basis data merupakan sistem yang terdiri dari kumpulan tabel (*file*) yang saling berhubungan (dalam sebuah basis data pada sebuah sistem komputer) dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai dan/atau program lain untuk mengakses dan memanipulasi tabel (*file*) tersebut.




**Gambar 2.4 Sistem Basis Data [6]**

### *2.3.3.1 Komponen Sistem Basis Data*

Dalam sebuah sistem basis data terdapat beberapa komponen utama yaitu:

1. Perangkat Keras (*Hardware*)
2. Sistem Operasi (*Operating System*)
3. Basis Data (*Database*)
4. DBMS (*Database Management System*)
5. Pemakai (*User*)
6. Perangkat lunak lain yang sifatnya *optional*

#### 2.3.3.1.1 Perangkat Keras (Hardware)

Perangkat keras yang biasanya ada dalam sebuah basis data adalah:

1. Komputer (satu untuk sistem yang *stand-alone* atau lebih dari satu untuk sistem jaringan)
2. Memori sekunder yang *on-line* (*Harddisk*)

3. Memori sekunder *off-line* (*Tape* atau *Removable Disk*) untuk keperluan *Backup* data
4. Media atau perangkat komunikasi (untuk sistem jaringan)

#### 2.3.3.1.2 Sistem Operasi (*Operating System*)

Secara sederhana sistem operasi merupakan program yang mengaktifkan atau menjalankan sistem komputer, mengendalikan seluruh sumber daya (*resource*) dalam komputer dan melakukan operasi-operasi dasar dalam komputer (operasi *I/O*, pengelolaan file, dan lain-lain) sejumlah sistem operasi yang banyak digunakan seperti MS-DOS, MS-Windows, UNIX, Linux, Mac, Sun-Solaris, dan lain-lain. Program basis data hanya akan aktif (*running*) jika sistem operasinya telah aktif.

#### 2.3.3.1.3 Basis Data (Database)

Sebuah sistem basis data dapat memiliki beberapa basis data. Setiap basis data dapat berisi sejumlah objek basis data (seperti tabel, indeks dan lain-lain). Di samping menyimpan data, setiap basis data juga menyimpan definisi struktur basis data dan objek – objeknya secara detail).

#### 2.3.3.1.4 DBMS (*Database Management System)*

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakaian secara langsung, tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus. Perangkat lunak ini disebut sebagai DBMS, yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali. DBMS juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, keakuratan dan konsistensi data.

Perangkat lunak yang termasuk DBMS adalah dBase III+, DbaseIV, FoxBase, RBase, MS-Access dan Borland Paradox, Borland Interbase, MS-SQLServer, MySQL, CA-Open Ingres, Oracle, Informix dan Sybase.

#### 2.3.3.1.5 Pemakai (User)

Ada beberapa tipe pemakai terhadap suatu sistem basis data yang dibedakan berdasarkan cara mereka berinteraksi terhadap sistem, diantaranya:

1. *Programmer* Aplikasi

   Pemakai yang berinteraksi dengan basis data melalui *Data Manipulation Language* (DML), yang disertakan dalam program yang ditulis menggunakan bahasa pemrograman induk (C, Pascal, Cobol dan lain-lain)
2. *User* Mahir (*Casual User*)

Pemakai yang berinteraksi dengan sistem tanpa menulis program. Mereka menyatakan *query* dengan bahasa yang sudah disediakan oleh DBMS.

3. *User* Umum (*End User*)

   Pemakai yang berinteraksi dengan sistem basis data melalui pemanggilan suatu program aplikasi permanen (*executable program*) yang telah disediakan sebelumnya

4. *User* Khusus (*Specialized User*)

   Pemakai yang menulis aplikasi basis data non konvensional, tetapi untuk keperluan-keperluan khusus, seperti untuk aplikasi *AI*, Sistem Pakar, Pengelolaan Citra, dan lain-lain, yang bisa saja mengakses basis data dengan/tanpa DBMS yang bersangkutan.

#### 2.3.3.1.6 Perangkat Lunak Lain

Aplikasi (perangkat lunak) ini bersifat *optional*. Artinya, ada atau tidak adanya tergantung pada kebutuhan DBMS yang digunakan dan lebih berperan dalam pengorganisasian data dalam basis data, sementara bagi pemakai basis data khususnya *end user* dapat disediakan program khusus untuk melakukan pengisian, perubahan, dan pengambilan data. Program ini ada yang sudah disediakan bersama DBMS-nya ada juga yang harus dibuatkan sendiri dengan aplikasi lain yang khusus.

### 2.3.4 Basis Data Relasional

Model basis data relasional menunjukkan suatu mekanisme yang digunakan untuk mengelola data secara fisik dalam memori sekunder yanga akan berdampak pula pada bagaimana kit mengelompokkan dan membentuk keseluruhan data yang terkait dalam sistem yang kita tinjau.

Pada model relasional basis data akan disebar kedalam table dua dimensi. Setiap table terdiri atas lajur mendatar yang disebut dengan baris data (Row/Record) dan lajur vertical yang biasa disebut dengan kolom (Coloum/Field). Untuk menerepakan sebuah basis data yang terdiri atas sejumlah table yang saling berhubungan maka kita membutuhkan sebuah perangkat lunak yaitu *Database Manegement System* (DBMS).

#### 2.3.4.1 Operasi dan Bahasa Basis Data

Disamping melakukan operasi pengisian data ke masing-masing table yang strukturnya telah disiapkan, operasi-operasi lain yang dapat kita lakukan adalah:

1. Menampilkan data (query).

2. Pengubahan data (update/edit).

3. Pengapusan data (delete).

Teknis penerapan operasi-operasi basis data tersebut sangat tergantung dari DBMS yang digunakan. Tetepi secara umum, ada dua macam cara untuk menjlankan operasi-operasi basis data, yaitu:

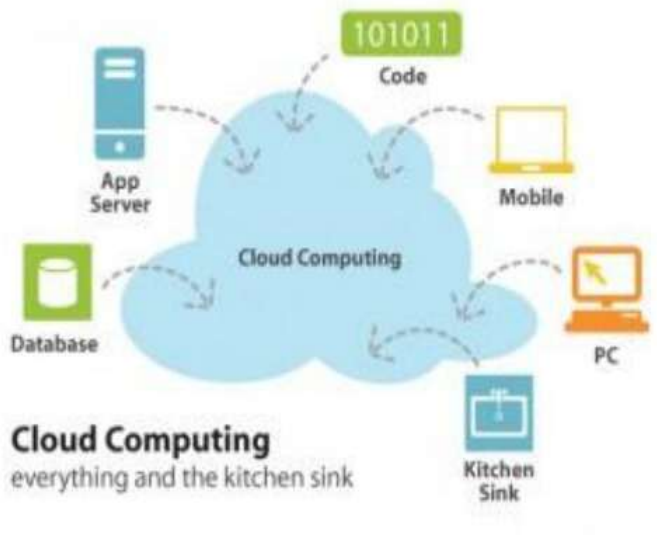
a. Menjalankan program khusus unktuk melakukan opeerasi basis data secara interaktif. Program semacam ini umumnya disediakan oleh setiap DBMS untuk mempermudah pemakai (user) dalam melakukan operasi-operasi basis data.

b. Menggunakan bahasa basis data (*Data Manipulation Language*). Dengan bahasa basis data user harus menuliskan perintah-perintk khusus untuk melakukan operasi-operasi basis data. Bahasa basis data tersebut ada yang memang berlaku khhusus unktuk suatu DBMS (seperti bahasa dBase 3+, dan Foxbase, Quel untuk CA-Open Ingres), tetapi ada pula bahasa basis data yang sudah menjadi standar (seperti SQL unktuk berbagai MS-SQL Server, Oracle dan CA-Open Ingres).

## 2.4 *Cloud Computing* [8]

*Cloud computing* terdiri dari dua kata yaitu "*cloud*" yang berarti awan dan "*computing*" yang berarti komputasi. Kunci dari *cloud computing* sebenarnya terletak pada kata "*cloud*" atau "awan", ini dapat dikatakan juga sebagai internet yang merupakan suatu lingkungan yang cukup besar yang didalamnya terdapat komputer yang saling berhubungan. Komputer tersebut bisa berupa *PC Personal* atau suatu jaringan komputer.

Sebelumnya pada komputer *desktop* biasa, perangkat lunak dijalankan pada komputer tersebut. Semua dokumen (*file*) yang dibuat, diolah dan disimpan pada komputer tersebut, ini dikenal dengan istilah PC Sentris. Sedangkan *cloud computing* menyediakan *storage* dan perangkat lunak pada server yang ada di internet, ini dikenal dengan istilah Dokumen Sentris.




**Gambar 2.5 Ilustrasi *Cloud Computing***

### 2.4.1 Definisi *Cloud Computing*

Terdapat beberapa sudut pandang terhadap pengertian atau definisi tentang *cloud computing*, diantarnya:

1. *Cloud computing* adalah teknologi komputasi komputer dengan memanfaatkan internet sebagai terminal utamanya guna mengelola piranti lunak hingga infrastruktur sebagai bentuk layanannya.
2. *Cloud computing* adalah evolusi selanjutnya dari internet ("awan") mulai dari tenaga komputasi hingga infrastruktur komputasi, aplikasi-aplikasi, proses bisnis, hingga kolaborasi yang muncul sebagai layanan yang dapat diakses pada saat dibutuhkan kapanpun dan dimanapun.
3. *Cloud computing* adalah sebuah gaya komputasi yang besar, elastis dan terukur, yang berhubungan dengan kemampuan IT sebagai penyedia layanan (*as a service*) kepada pelanggan eksternal menggunakan teknologi internet.

*Cloud computing* biasanya dipakai pada suatu perusahaan, organisasi, ataupun komunitas. Aplikasi dan data berada pada komputer *server* yang bersifat "*cloud*" untuk digunakan oleh kelompok tertentu pada suatu perusahaan tersebut, baik lintas perusahaan maupun lintas *platform* dan diakses via internet. Sehingga para pengguna yang berhak mengakses *file* atau aplikasinya dapat menggunakan komputer manapun

yang terhubung dengan koneksi internet. *Services* yang ada pada teknologi *cloud* berbasis HTTP, HTML, XML, JavaScript, teknologi sejenis yang lainnya.

### 2.4.2 Sejarah *Cloud Computing*

Pada waktu itu, para *developers* dan arsitekur teknologi menggunakan sebuah gambar awan untuk mengilustrasikan sumber daya yang dapat diakses via web. Sejak saat itulah mereka mendefinisikan bahwa awan sangat logis untuk dijadikan suatu penghubung antara media lokal dengan sumber daya pada suatu jaringan internet. Kebanyakan *developers* kebingungan untuk memastikan terminologi dari suatu *cloud computing.* Sebenarnya para *developers* tersebut sudah menyediakan suatu *host* dan ini bisa dikatakan sebagai "*Cloud Services*". Hanya saja dari kebenaran argument tersebut, deskripsi dari *cloud computing* belum cukup akurat, mereka beranggapan bahwa syarat *cloud computing* adalah:

1. *Web Services*

   Konsep sederhana ini berfungsi untuk menerima *request* dari pengguna dan mengrimkan respon kepada pengguna tersebut, *request* ini bisa berupa kumpulan kode.
2. *Cloud OS (Operating Systems)*

   *Cloud OS* bertindak sebagai *interface* antara *web service* dengan CPU atau perangkat keras lainnya agar dapat memanggil layanan yang ada pada *cloud* lalu memproses layanan tersebut.
3. *Cloud FX*

   Ini berfungsi untuk para *developer* yang ingin mengembangkan layanan atau aplikasi *cloud* dan berupa bahasa pemrograman.
4. *Cloud Application*

   Ini merupakan aplikasi yang mampu berjalan pada *cloud*, dan bisa diakses oleh para pengguna.
5. *Infrastructure as a Service*
6. *Platform as a Service*
7. *Software as a Service*
8. *IT Professionals* dan *System Administrator*

   Karakterisitik ini sangat berpengaruh untuk kelangsungan dari *cloud computing*, ini bertugas untuk me*monitoring*, mengatur, dan bertanggung jawab atas layanan yang telah dibangun.
9. *Developers* dan Arsitektur Teknologi

Dibutuhkan *developer* atau arsitektur teknologi untuk membangun dan mengembangkan aplikasi *cloud computing.*

10. Pengguna

    Tanpa adanya pengguna yang menggunakan aplikasi *cloud* yang telah dibuat oleh para pembangun aplikasi, maka *cloud computing* ini akan terasa mubazir.

### 2.4.3 Model Layanan *Cloud Computing*

Konsep dari pengembangan layanan sistem *cloud computing* dibagi menjadi beberapa tipe layanan, diantaranya adalah:

1. *Software as a Service (SaaS)*
2. *Platform as a Services (PaaS)*
3. *Infrastructure as a Service (Iaas)*

#### *2.4.3.1 Software as a Service (SaaS)*

Model layanan *Software as a Service* (*SaaS*) adalah model layanan yang paling banyak dikembangkan untuk sistem *cloud computing*. Model layanan ini berupa aplikasi atau *software* berbasis web, yang diberikan kepada berbagai pengguna oleh *vendor* atau pemilik sistem tersebut. Pengguna pun tidak perlu memiliki aplikasi tersebut untuk menggunakannya, melainkan pengguna memerlukan koneksi internet untuk dapat mengakses ataupun menggunakannya. Contoh perusahaan yang mengembangkan layanan ini adalah Google dengan contoh aplikasinya yaitu Google Docs, yang berfungsi sebagai *word processor, spreadsheet, presentation creator*.




**Gambar 2.6 Arsitektur Model Layanan SaaS**

#### *2.4.3.2 Platform as a Service (PaaS)*

Model layanan *platform as a service* merupakan layanan yang menyediakan *platform* seperti bahasa pemrograman, *tools*, ataupun sistem operasi yang nantinya mampu mengembangkan aplikasi berbasis konsumen. Keuntungan dari PaaS adalah bahwa perusahaan yang menggunakan layanan ini tidak perlu khawatir untuk instalasi,

perawatan, dan keamanan pada server karena *provider* PaaS akan menangani itu semua. Contoh implementasi dari PaaS adalah Microsoft Windows Azure dan Google App Engine.




**Gambar 2.7 Arsitektur Model Layanan PaaS**

### *2.4.3.3 Infrasturcture as a Service (IaaS)*

Model layanan disediakan dari IaaS adalah sumber daya pemroses, storage, kapasitas jaringan, dan sumber daya komputasi lainnya. Dimana konsumen dapat mengembangkan dan menjalankan aplikasi khusus. IaaS memungkinkan pengguna untuk menjalankan beberapa tugas komputer pada mesin *virtual* yang tak terbatas. Artinya pengguna diberikan kebebasan untuk merakit PC *virtual* sesuai dengan keinginan istilah dari konsep ini biasa disebut *virtualization*, salah satu implementasi dari model layanan ini adalah Amazon EC2.




**Gambar 2.8 Arsitektur Model Layanan IaaS**

### 2.4.4 Model Pengembangan *Cloud Computing*

Ada beberapa model *cloud computing* yang sudah dikembangkan saat ini dengan berbagai fungsi dari masing-masing model pengembangannya, yaitu:

1. *Public Cloud*

   Model pengembangan ini adalah model pengembangan yang paling popular. Model ini diimplementasikan kepada berbagai server yang berjalan pada

berbagai *data center* agar dapat diakses dibelahan bumi manapun. Model pengembangan ini telah dilakukan oleh salah satunya adalah Google.

2. *Private Cloud*

   Model pengembangan ini biasanya dilakukan oleh suatu perusahaan atau organisasi dengan memegang atribut dari *cloud computing* yaitu, elastisitas, *Self Service, Pay-by-Use,* dan *programmability*. Ini ada karena mampu mengefisiensikan biaya seperti mengurangi konsumsi listrik dan pendingin untuk server mereka sebelumnya. Vendor yang menyediakan model ini diantaranya adalah vmWare, Microsoft, IBM, Sun, dan lain-lain.

3. *Hybrid Cloud*

   Model ini merupakan kombinasi atau gabungan dari model *public* dan *private* yang masih memiliki entitas unik namun terikat bersama-sama oleh standar teknologi atau kepemilikan data dan portabilitas aplikasi.

4. *Community Cloud*

   Model ini diimplementasikan ketika beberapa perusahaan atau organisasi memiliki kesamaan konteks data.

### 2.4.5 Keuntungan *Cloud Computing*

Terdapat berbagai keuntungan yang dapat diperoleh dalam penggunaan teknologi c*loud computing*, diantaranya adalah:

1. Mengurangi biaya untuk merakit suatu komputer atau PC

   Pengguna tidak perlu mengeluarkan banyak biaya untuk merakit komputer untuk menjalankan aplikasi yang mereka butuhkan, karena segala bentuk proses, media penyimpanan akan dilakukan pada *cloud*.

2. Improvisasi *Perfomance*

   *Cloud computing* berusaha untuk menyesuaikan kinerja aplikasinya, jika kebutuhan tugas yang dilakukan semakin banyak maka *cloud computing* akan berusaha untuk menyesuaikan bebean kerja tersebut.

3. Mengurangi biaya infrastruktur teknologi

   Ini akan terasa pada suatu perusahaan atau organisasi, karena investasi perusahaan untuk menggunakan infrastruktur teknologinya akan lebih hemat.

4. *Maintenance*

   Segala bentuk *maintenance* akan dilakukan di *cloud*, sehingga suatu perusahaan atau perorangan tidak perlu repot untuk *maintenance* komputasi, atau aplikasi.

5. Mengurangi biaya *software*

   Biasanya *software desktop* akan selalu ada yang lebih baru, dan itu harus dibeli kembali untuk menginstallnya, maka akan dibutuhkan lagi suatu biaya untuk

membeli *software* tersebut. Jika ada *cloud* software akan merubah versinya tanpa perlu menginstalnya kembali.

6. *Update software* otomatis

   *Software* yang dipakai pengguna pada *cloud* biasanya akan terus melakukan *update* secara otomatis.

7. Meningkatkan kinerja komputer

8. *Storage* yang besar/tak terbatas

   Karena sudah tersimpan pada *cloud*, maka media penyimpanan yang kita miliki sangat besar bahkan tak terbatas dibandingkan dengan media penyimpanan yang ada pada *PC desktop* biasa.

9. Data lebih *safe*

   *Safe* ini berarti jika PC yang biasa dipakai mengalami suatu bencana, *crash hardware* maupun *software*, maka data yang ada pada PC tersebut tidak akan terselamatkan, sedangkan pada *cloud* dijamin keamanannya, karena data yang disimpan tidak hanya disimpan pada satu *server* saja.

10. Improvisasi kompatibilitas Sistem Operasi

    Sistem komputasi yang ada pada *cloud* tidak tergantung pada sistem operasi. Misalnya pengguna lebih sering menggunakan aplikasi berbasis Windows, sedangkan perusahaan di tempat ia bekerja menggunakan sistem operasi Linux. Hal ini mampu diatasi oleh *cloud computing*.

11. Improvisasi kompatibilitas format *file*

    Pada aplikasi berbasis web, *file* dokumen apapun bisa diakses, sebagai contoh, jika pengguna lebih sering menggunakan Microsoft Office 2007, sedangkan pada PC yang dipakai masih menggunakan Microsoft 2003, maka pengguna tidak bisa mengakses *file* tersebut. Maka aplikasi berbasis *web* akan mampu untuk menangani masalah tersebut.

12. Kolaborasi lebih mudah

    *Cloud computing* akan memudahkan dalam men-*share* atau membagi data-data sesuai keinginan pengguna

13. Akses *file* dimanapun dan kapanpun

    Pengguna tidak akan tergantung pada tempat dan waktu untuk mengakses *file*, karena aplikasi berbasis web tidak akan pernah “mati”.

14. *Latest Version Availability*

    Jika pengguna mengakses dan melakukan *update* disuatu tempat yang jauh dari biasanya, maka pada saat pengguna berada ditempat yang berbeda, *file* yang baru di*update* yang akan tampil.

### 2.4.6 Kerugian *Cloud Computing*

Beberapa kerugian ataupun kendala yang masih mungkin terjadi pada *cloud computing* yaitu:

1. Membutuhkan koneksi internet
2. Tidak berjalan dengan baik jika koneksinya lambat

   Karena biasanya aplikasi yang dipakai butuh *bandwidth* yang sesuai untuk men*download* aplikasi yang dipakai.
3. Fitur yang terbatas

   Sekarang ini fitur-fitur pada *cloud computing* masih terbatas dibandingkan dengan aplikasi pada PC *desktop*. Contohnya adalah Google Docs, yang memiliki fitur yang umum atau sederhana dibandingkan dengan Microsoft Office yang sudah *full-featured*.
4. Penyimpanan data yang tidak aman

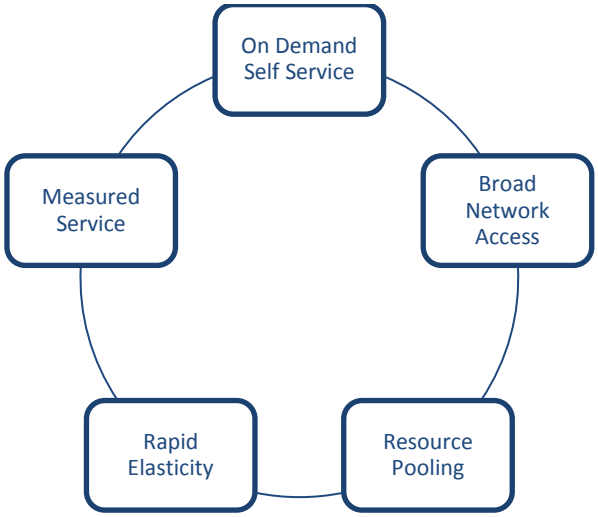
   Data yang disimpan pada *cloud*, tidak menjamin bahwa pengguna lain yang tidak berkewenangan dapat mengakses data yang telah disimpan.
5. *Service Level*

   *Cloud Provider* mungkin tidak akan konsisten dengan *performance* dari aplikasi atau transaksi, sehingga pengguna harus paham mengenai *service level* untuk *transaction response time, data protection*, kecepatan data *recovery*.
6. *Privacy*

   Karena pihak lain pun kemungkinan menggunakan *cloud provider* yang sama, maka ada kemungkinan juga data yang disimpan bisa dibaca oleh pihak lain tersebut.

### 2.4.7 Karakteristik *Cloud Computing* [9]

Terdapat beberapa karakteristik dari *Aplikasi cloud computing*, yaitu:

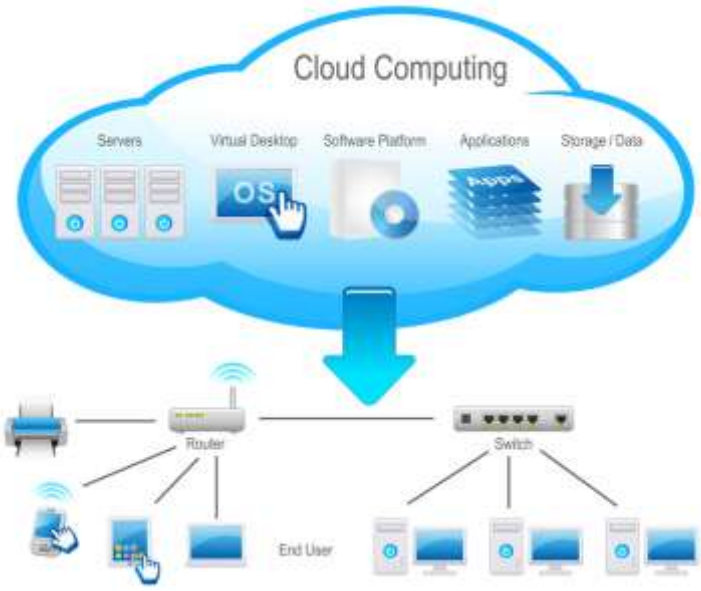



**Gambar 2.9 Karakteristik *Cloud Computing***

1. *On Demand Self Service,* layanan yang diberikan sesuai dengan kebutuhan dari pengguna sesuai dengan yang tersedia pada *provider*. Sehingga pengguna dapat memilih layanan yang sudah disediakan.
2. *Rapid Elasticity*, pengguna dapat *upgrade* atau menambah kapasitas *storage* paket sesuai dengan kebutuhan di masa yang akan datang.
3. *Broad Network Access,* layanan yang diberikan diakses melalui jaringan internet, sehingga dapat diakses dimana saja dan kapan saja.
4. *Resource Pooling*, data yang diakses disediakan secara terpusat, sehingga pengguna tidak perlu tahu bagaimana dan darimana permintaan akan sumber daya komputasinya dipenuhi oleh *provider*, karena dapat digunakan secara massal/bersamaan.
5. *Measured Service*, dimana layanan harus disediakan secara terukur, sehingga kebutuhan dari pengguna dapat terlihat karena nantinya akan digunakan dalam proses pembayaran/penyewaan.

### 2.4.8 Arsitektur *Cloud Computing* [10]

Kata kunci dari komputasi awan adalah kala cloud atau awan. Yang dimaksud awan disini adalah jaringan komputer masif atau bahkan satu buah PC yang saling terhubung di sebuah grid. Komputer-komputer ini berjalan secara paralel dan mengombinasikan sumber daya untuk menghasilkan kekuatan komputasi secara super computer.




**Gambar 2.10 Arsitektur *Cloud Computing***

Secara fisik, *cloud* adalah kumpulan komputer dan server yang bisa diakses secara publik melalui internet. Biasanya hardwere yang terlibat di *cloud computing* ini milik pihak ketiga yang diletakan di lokasi data center tertentu. Komputer *cloud* bisa menjalankan berbagai jenis sistem operasi.

Satu hal yang penting penting dalam *cloud computing* adalah kemampuan pemrosesan dari sitem, bukan tampilan dekstop-nya. Dan semua pengguna nantinya akan mengakses komputasi yang disediakan oleh *cloud server* menggunakan jaringan internet dari berbagai macam media. Namun jenis hardwere ataupun sistem operasiyang ada dibelakang layar tetap tidak akan terlihat.

Arsitektur *cloud* ini sebenarnya sangat sederhana, tidak dibutuhkan manajemen khusus untuk menghubungkan semua komputer dan mengalokasikan pemrosesan task ke pengguna. Pengguna tinggal mengakses menggunkan antarmuka di komputernya, kemudian sudah langsung bisa mengakses task atau service di dokumen. Request atau permintaandari pengguna ini kemudian bisa dilewatkan ke manajemen sistem. Layanan ini akan mengambil sumber daya yang ada di *cloud*, lalu menjalankan aplikasi web tertentu, kemudian bisa menjalankan aplikasi web lainnya. Setelah aplikasi web dijalankan, sistem kemudian memonitor dan mengatur fungsi penggunaan *cloud* sehingga sumber daya dibagi berdasarkan kebutuhan yang diperlukan.

Ada beberapa sifat dari *cloud computing*, yaitu sebagai berikut:

1. infrastruktur di-sharing, beberapa klien bisa berbagi platform teknologi dan bahkan satu instance aplikasi.
2. Service atau layanan diaskes *on demand*, saat diperlukan saja. Jadi layanannya bervariasi dari segi pengguna, kapasitas, transaksi, atau kombinasinya.
3. Servis bisa diatur skalanya, tidak ada batas untuk pengembangannya.
4. Jika ada biaya, biasanya biaya berdasarkan konsumsi servisnya. Jadi bukan biaya fix yang tidak berubah. Ada yang menggunakan variabel waktu, data, dan sebagainya. Servis bisa diakses dari area manapun di dunia dengan menggunakan berbagai perangkat yang berbeda.

### 2.4.9 Implementasi *Cloud Computing*

*Cloud computing* saat ini sedang ada di masa awal perkembangannya. Di masa depan mungkin akan semakin banyak layanan aplikasi menarik yang bisa digunakan. Beberapa contoh layanan *cloud computing* yang telah diterapkan saat ini diantaranya:

1. *Cloud computing* untuk keluarga
    a. Berkomunikasi dengan email
    b. Berkolaborasi dalam penjadwalan
    c. Berkolaborasi pada hal yang harus dilakukan
    d. Kolaborasi dalam pengaturan keuangan keluarga
    e. Berkolaborasi dalam daftar contact
    f. Berbagi foto
2. *Cloud computing* untuk bisnis
    a. Pengaturan jadwal
    b. Manajemen contcat list
    c. Manajemen proyek
    d. Kolaborasi report

e. Kolaborasi laporan keuangan

f. Perusahaan virtual

3. *Cloud computing* untuk sekolah (*e-learning*)
4. *Cloud computing* untuk jejaring sosial.

## 2.5 *Content Management System* (CMS)

*Content Management System atau sering disebut dengan CMS adalah* aplikasi web yang berisikan *template* untuk mengelola isi halaman web secara mudah. Pengguna **CMS** tidak memerlukan pengetahuan pemrograman web yang handal karena proses instalasi dan cara penggunaannya sudah user friendly. CMS ada yang dibuat khusus untuk menyesuaikan kasus yang ada dan biasanya berbayar, ada juga yang berupa template instan yang fungsionalitasnya dibuat dengan menyeuaikan pada beberapa proses bisnis yang ada didunia nyata yang dapat digunakan secara gratis.




**Gambar 2.11 Pengolahan *Content Management System***

Aplikasi *CMS* instan yang banyak terdapat di internet saat ini dibuat menggunakan *scripting language PHP* dan database-nya adalah MySQL. Saat ini perkembangan Content Management System cukup pesat, banyak vendor yang membuat CMS instant yang didistribusikan secara gratis. Perkembangan CMS instan ini dipicu oleh perkembangan web 2.0 yang memungkinkan interaksi dalam arti yang cukup luas antara pengelola web dan pengunjung web.

Umumnya, sebuah CMS terdiri dari dua elemen:

1. Aplikasi manajemen isi (Content Management Application, [CMA])

   Elemen CMA memperbolehkan si manajer isi yang mungkin tidak memiliki pengetahuan mengenai HTML (HyperText Markup Language), untuk mengatur pembuatan, modifikasi, dan penghapusan isi dari suatu situs w*eb* tanpa perlu memiliki keahlian sebagai seorang Webmaster.
2. Aplikasi pengiriman isi (Content Delivery Application [CDA]).

   Elemen CDA menggunakan dan menghimpun informasi-informasi yang sebelumnya telah ditambah, dikurangi atau diubah oleh si empunya situs web untuk meng-*update* atau memperbaharui situs *web* tersebut. Kemampuan atau fitur dari sebuah sistem CMS berbeda-beda, walaupun begitu, kebanyakan dari software ini memiliki fitur publikasi berbasis *web*, manajemen format, kontrol revisi, pembuatan index, pencarian, dan pengarsipan.

Selain perkembangan teknologi web dan infrastruktur internet, perkembangan pesat *Content Management System* juga dipicu oleh kebutuhan masyarakat dan pelaku bisnis yang menginginkan web dapat mendukung kegiatan bisnis mereka secara mudah dalam hal pengelolaan content, cepat dalam pembuatan web, serta murah dalam pengadaannya.

Bahkan saat ini, pengembangan aplikasi web CMS sudah banyak yang menggunakan konsep freamework. Dengan konsep ini, sangat dimungkinkan untuk melakukan modifikasi terhadap beberapa fitur **CMS** instan tersebut karena semuanya dibuat lebih modular. Berikut adalah beberapa contoh web CMS instant yang dibuat oleh vendor dan dapat digunakan secara gratis dan scripting language yang digunakan adalah PHP dengan database server MySQL.

1. CMS untuk membuat personal blog, contohnya WordPress.

2. CMS untuk membuat web e-Commerce, contohnya PrestaShop, OsCommerce, Opencart, Drupal.
3. CMS untuk membuat web e-Learning, contohnya Moodle.
4. CMS untuk membuat personal web, contoh joomla!, Mambo.
5. CMS untuk membuat web e-Forum, contoh phpBB, MYBB,Vbulletin.
6. CMS freamework berbasis scripting language PHP, yaitu Code Igniter (CI), RubyOnRail (RoR), CakePHP, FuseBox.

## 2.6 Teknologi Internet

*Internet* adalah suatu jaringan komputer global yang terbentuk dari jaringan-jaringan komputer lokal dan regional yang memungkinkan komunikasi data antar komputer yang terhubung ke jaringan tersebut. Internet awalnya merupakan rencana dari Departemen Pertahanan Amerika Serikat (*US Departement Of Defense*) pada sekitar tahun 1960. Dimulai dari suatu proyek yang dinamakan ARPANET atau *Advanced Research Project Agency Network.* Beberapa universitas di Amerika Serikat diantaranya UCLA, Stanford, UC Santa Barbara dan University of Utah, diminta bantuan dalam mengerjakan proyek ini dan awalnya telah berhasil menghubungkan empat komputer di lokasi universitas yang berbeda tersebut. Perkembangan ARPANET ini cukup pesat jika dilihat dari perkembangan komputer pada saat itu. Karena perkembanganya sangat pesat, jaringan komputer ini tidak dapat lagi disebut sebagai APRANET karena semakin banyak komputer dan jaringan-jaringan regional yang terhubung. Konsep ini kemudian berkembang dan dikenal sebagai konsep *Internetworking* (*Jaringan antar jaringan*). Oleh karena itu istilah internet menjadi semakin popular, dan orang menyebutnya jaringan besar komputer tersebut dengan istilah internet.

## 2.7 Aplikasi *Web*

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. 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:

1. Mengurangi lalu lintas jaringan dengan cara menghindari percakapan bolak-balik antara *client* dan server.
2. Mengurangi waktu pemuatan kode, mengingat *client* hanya mengambil kode HTML saja.
3. Mencegah masalah ketidakkompatibelan *browser*.
4. *Client* dapat berinteraksi dengan data yang ada pada *server*.

## 2.8 Hosting

Hosting merupakan jasa layanan ruang (space) untuk menyimpan data text, gambar, audio dan video yang selanjutnya file-file tersebut bisa diakses dari mana saja (online). Hosting juga dilengkapi dengan fasilitas protokol seperti FTP dan HTTP untuk memudahkan dalam mengakses data, selain itu hosting juga bisa digunakan untuk mengelola email.

Biasanya penyedia layanan hosting (hosting provider) menyediakan paketpaket tertentu yang disesuaikan dengan kebutuhan pelanggan, seperti quota space harddisk, quota bandwidth, bahasa pemograman, dan database. Hosting terdiri dari *shared hosting*, *dedicate hosting dan virtual private server*.

### 2.8.1 *Shared Hosting*

Merupakan layanan hosting yang digunakan bersama-sama oleh pengguna lainnya. Para pengguna layanan ini menggunakan sumber daya baik software maupun hardware atau IP Address secara bersama-sama. Hosting seperti ini dipengaruhi oleh setiap proses yang dilakukan oleh pengguna, jika ada pengguna hosting yang melakukan proses yang berlebihan dan membebani server, maka pengguna lainnya akan terkena

imbasnya, seperti server menjadi lambat bahkan ada kemungkinan website menjadi tidak bisa diakses.

Sebagai contoh jika server *Shared hosting* terkena hack, atau hardware failure maka seluruh pengguna *Shared hosting* akan terkena dampaknya seperti website tidak bisa diakses. Pemilik server tentunya akan melakukan pemantauan secara bertahap dan rutin terhadap proses yang terjadi didalam server, biasanya admin akan melakukan suspend account terhadap pengguna yang melakukan pelanggaran atau mengganggu terhadap kinerja server.

*Shared hosting* cocok bagi Anda yang mempunyai blog, web presence, toko online. Dari sisi harga jenis hosting ini lebih murah jika dibandingkan dengan 23 jenis hosting lainnya, kembali kepada paket yang disediakan oleh hosting provider.

Kekurangan dari *Shared hosting* ini, pengguna tidak bisa leluasa dalam menggunakan sumber daya server, atau menggunakan software sesuai dengan kebutuhan karena hak penuh terhadap administrasi server dipegang sepenuhnya oleh admin.

### 2.8.2 *Dedicated Hosting*

Jenis hosting ini biasa digunakan untuk solusi hosting terhadap kebutuhan yang berskala besar atau kompleks. Pengguna bisa memilih atau menyediakan spesifikasi server yang diinginkan, yang kemudian ditempatkan pada sebuah data center. Pengguna bisa menggunakan sumber daya server secara leluasa karena administrasi server dikelola atau dimiliki oleh pemilik server.

Dedicated server biasa digunakan oleh penyedia layanan hosting (*Shared hosting* vps), *application service provider*, atau website yang memiliki trafik dan atau membutuhkan resource yang tinggi. Jika dilihat dari harga, pengelolaan dedicated server cukup mahal jika dibandingkan dengan *shared-hosting* atau VPS.

### 2.8.3 *Virtual Private Server*

*Virtual Private Server* (VPS) merupakan teknologi perangkat lunak yang memungkinkan pembagian sumber daya ke dalam beberapa virtual mesin (server). Tiap-tiap *virtual server* akan mempunyai sistem operasi yang berjalan secara mandiri. VPS menyediakan akses penuh terhadap user root, setiap VPS mempunyai konfigurasi yang bisa diatur layaknya pada server fisik (dedicated server).

Kelebihan dari penggunaan VPS antara lain:

1. Performa yang baik seperti layaknya dedicated server
2. Kebebasan dalam menggunakan sistem operasi atau software yang dibutuhkan
3. Menghemat biaya, jika dibandingkan dengan menggunakan dedicated server

VPS biasanya digunakan pada website atau aplikasi yang memerlukan resource medium diatas shared hosting. Layanan VPS memungkinkan jika pengguna memiliki website atau aplikasi yang memerlukan resource cukup tinggi dengan budget rendah.

## 2.9 FileZilla

FileZilla atau juga dikenal dengan sebutan FileZilla Client, adalah salah satu software FTP gratis, open source, cross-platform. FileZilla tersedia untuk Windows, Linux, dan Mac OS X. Software ini mendukung FTP, SFTP, dan FTPS (FTP diSSL/TLS). FileZilla Server adalah produk lain dari FileZilla Client. Ini adalah server FTP yang didukung oleh proyek yang sama dan fitur-fitur dukungan untuk FTP dan FTP melalui SSL / TLS.

Fitur utama dari Filezilla adalah :

1. Site manager (Manajer situr) Mengizinkan pengguna untuk membuat daftar situs FTP beserta data koneksinya, seperti nomor port yang akan digunakan, protokol yang digunakan, dan apakah akan menggunakan log anonim atau normal. Untuk log normal, nama pengguna dan kata sandinya tersebut akan disimpan. Penyimpanan kata sandi adalah opsional.
2. Message log (Log pesan) Ditampilkan di bagian atas jendela. Fitur ini menampilkan output berjenis konsol (*console-type*) yang menunjukkan perintah yang dikirim oleh FileZilla dan respons yang diterima dari server.
3. File dan folder view ditampilkan di bawah pesan log (Message log), yang menyediakan sebuah tampilan grafis antarmuka untuk FTP. Pengguna dapat menavigasi folder dan melihat serta mengubah isinya pada komputer lokal dan server dengan menggunakan tampilan antarmuka gaya Explorer. Pengguna dapat men-*drag* dan *drop* file antara komputer lokal dan server.
4. Transfer queue (transfer antrian) Ditampilkan di sepanjang bagian bawah jendela, menunjukkan status *real-time*setiap antrian atau transfer file yang aktif.

## 2.10 PHP [11]

PHP singkatan dari PHP *Hypertext Preprocessor* yang digunakan sebagai bahasa script server-side dalam pengembangan web yang disisipkan pada dokumen HTML.

Penggunaan PHP memungkinkan web dapat dibuat dinamis sehingga maintenance situs web tersebut menjadi lebih mudah dan efesien. PHP merupakan softwere Open-Source yang disebarkan dan dilisensikan secara gratis serta dapat di-download secara bebas dari situs resminya http://www.php.net. PHP ditulis menggunakan bahasa C.

Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), Cold Fusion, ataupun Perl. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai di *command line*. Artinya PHP dapat berjalan tanpa melibatkan *web server* maupun *browser*. Untuk membuat suatu file PHP, program yang diperlukan adalah sebuah editor teks seperti Notepad, Ultra Edit, PHP Coder, PHP Editor atau Macromedia Dreamweaver.

### *2.10.1 Sejarah PHP*

Kelahiran PHP bermula pada saat Rasmus Lerdorf membuat sejumlah skrip Perl yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya. Yakni pada tahun 1994. Skrip-skrip ini selanjutnya dikemas menjadi *tool* yang disebut "Personal Home Page". Paket inilah yang menjadi cikal bakal PHP. Pada tahun 1995, Rasmus menciptakan PHP/FI versi 2. Pada versi inilah pemrogram dapat menempelkan kode terstruktur di dalam tag HTML. Yang menarik, kode PHP juga bisa berkomunikasi dengan *database* dan melakukan perhitungan-perhitungan yang kompleks.

### *2.10.2 Kelebihan-kelebihan PHP*

PHP memiliki banyak kelebihan yang tidak dimiliki oleh bahasa script sejenis. PHP difokuskan pada pembuatan script server-side, yang bisa melakukan apa saja yang dapat dilakukan CGI, seperti mengumpulkan data dari form, menghasilkan isi halaman web dinamis, dan kemampuan mengirim serta menerima cookies, bahkan lebih dari kemampuan CGI.

Pada saat ini PHP cukup popular sebagai peranti pemrograman web. PHP dapat berfungsi pada *server-server* yang berbasis UNIX, Windows, Macintosh, dan RISC OS. PHP juga mendukung banyak server seperti Apache, Microsoft Internet Information Server (MIIS), Personal Web Server (PWS), Netscape and iPlanet servers, Oreilly Website Pro server, audium, Xitami, dan OmniHTTPd.

PHP tidak terbatas pada hasil keluaran HTML. PHP juga memiliki kemampuan untuk mengolah keluaran gambar, file PDF, dan Movies Flash. PHP juga dapat menghasilkan teks seperti XHTML dan file XML lainnya.

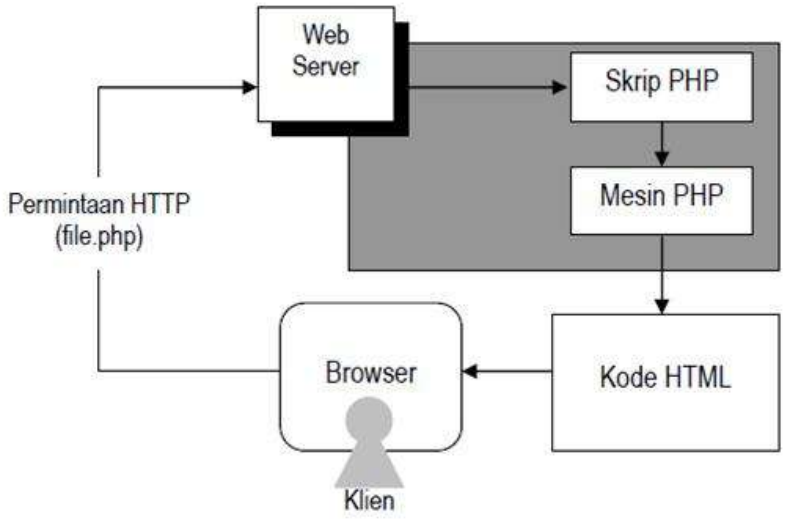
### *2.10.3 Konsep Kerja PHP*

Adapun alur kerja PHP adalah [12]:

1. Client me-request halaman web yang berisi script PHP.
2. Jika file yang di-request ditemukan, maka server akan meneruskannya ke PHP Interpreter (Penterjemah PHP) yang akan bekerja menghasilkan dokumen/halaman

HTML berdasarkan script PHP. Jika dalam script tersebut terdapat permintaan terhadap database, maka akan terjadi proses query data ke database server.

3. Dokumen HTML hasil interpretasi oleh PHP Interpreter dikembalikan ke web server.
4. Web Server mengirim dokumen HTML (Response) ke Client.
5. Web Browser akan menterjemahkan dokumen HTML ke Display/Monitor.




**Gambar 2.12 Konsep Kerja PHP [12]**

Dari gambar 2.12 dapat diambil kesimpulan bahwa untuk membangun sebuah situs yang dinamis dan interaktif, maka ada beberapa hal yang diperlukan yaitu Web Server, Database Server dan sebuah Script Interpreter (dalam hal ini PHP Interpreter).

### *2.10.4 PHP dan Database*

Salah satu kelebihan dari PHP adalah mampu berkomunikasi dengan berbagai *database* yang terkenal. Dengan demikian, menampilkan data yang bersifat dinamis, yang diambil dari *database*, merupakan hal yang mudah untuk diimplementasikan. Itulah sebabnya sering dikatakan bahwa PHP sangat cocok untuk membangun halaman-halaman web dinamis. Pada saat ini PHP sudah dapat berkomunikasi dengan berbagai *database* meskipun dengan kelengkapan yang berbeda. Beberapa diantaranya adalah:

1. DBM
2. FilePro (Personix, Inc)
3. Informix

4. Ingres
5. Interbase
6. Microsoft Access
7. MSQL
8. MySQL
9. Oracle
10. PostgreSQL
11. Sybase.

## 2.11 XHTML (*eXtensible Hypertext Markup Language*)

XHTML adalah sebuah kombinasi dari HTML dan XML (Extensible Markup Language) dan merupakan bentuk sempurna (dalam arti, lebih ketat dan bersih dibandingkan HTML). Tujuan XHTML adalah menggantikan HTML. XHTML adalah bentuk HTML yang didefinisikan sebagai aplikasi XML yang berarti XHTML berisi semua elemen dalam HTML 4.01 dikombinasikan dengan sintaks XML. XHTML adalah bahasa markup sebagaimana HTML, tetapi dengan gaya bahasa lebih baik. XHTML merupakan versi HTML yang memenuhi persyaratan XML dan merupakan sebuah dokumen HTML, dan XHTML menjadi standar internasional dengan spesifikasi yang ditetapkan oleh W3C (World Wide Web Consortium). Beberapa ciri dari XHTML:

1. Elemen XHTML harus bersarang dengan sempurna.
2. Elemen XHTML harus ditulis dalam huruf kecil.
3. Elemen XHTML harus selalu diakhiri dengan tag penutup.
4. Dokumen XHTML harus memiliki satu root elemen.

## 2.12 CSS (*Cascading Style Sheet*) [13]

CSS (*Cascading Style Sheet*) merupakan sebuah perintah yang mampu berjalan di antara tag HTML. Dengan menggunakan CSS dapat diciptakan sebuah format template pada setiap halaman web dengan mudah dan dinamis. Di dalam halaman web terdapat beberapa elemen HTML yang kesemuanya itu dapat dikendalikan dengan menggunakan sebuah bahasa script CSS. Dalam pengaturannya biasanya style disimpan didalam sebuah file yang bernama style.sheet.css. Style sheet ini dapat berupa file maupun embedded bersama tag HTML itu sendiri.

Keuntungan menggunakan CSS yaitu, seorang *designer web* atau *programmer* dapat menghemat pengaturan web, karena HTML hanya mengijinkan untuk pengaturan halaman per elemen saja. Jika membuat website yang besar tentunya akan dipersulit dengan pengaturan halaman baik itu warna *background,* jenis teks, maupun ukuran teks. Dengan adanya kesulitan tersebut maka CSS telah memberi solusi. Jadi jika menggunakan CSS cukup sekali saja dalam menentukan sebuah bentuk layout halaman web yang kemudian semua halaman akan dihubungkan dengan file CSS yang telah dibuat. Selain itu, CSS memberikan kemudahan dalam mengelola web, jadi kapanpun tampilan halaman dapat diganti dengan hanya merubah file dokumen style sheet yang telah dibuat.

## 2.13 JavaScript

JavaScript adalah bahasa skrip yang populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode JavaScript dapat disisipkan dalam halaman web menggunakan tag SCRIPT.

JavaScript pertama kali dikembangkan oleh Brendan Eich dari Netscape. *Navigator* sebelumnya telah mendukung Java untuk lebih bisa dimanfaatkan para programmer yang non-Java. Maka dikembangkanlah bahasa pemrograman bernama *LiveScript* untuk mengakomodasi hal tersebut. Bahasa pemrograman inilah yang akhirnya berkembang dan diberi nama JavaScript, walaupun tidak ada hubungan bahasa antara Java dengan JavaScript.

JavaScript bisa digunakan untuk banyak tujuan, misalnya untuk membuat efek *rollover* baik di gambar maupun teks, dan untuk membuat Ajax. JQuery merupakan Javascript Library, yaitu kumpulan kode/fungsi Javscript siap pakai, sehingga mempermudah dan mempercepat dalam membuat kode Javascript, termasuk dalam membuat kode Ajax.

## 2.14 Ajax (*Asynchronous Javascripst and XML*) [14]

Ajax merupakan suatu teknik/metode pengambilan data dari server menggunakan sebuah fungsi dalam Javascript, yaitu XMLHttpRequest (XHR). XHR merupakan objek yang dirancang untuk memungkinkan permintaan layanan ke server HTTP secara asinkron. Asinkron berarti berarti bahwa klien dapat meminta layanan dari server dan tidak perlu menunggu server melayaninya. Dengan dmikian, meskipun informasi dari server belum didapat, layanan kepada pengguna tetap bisa dilakukan. Dengan Ajax, memungkinkan aplikasi web dapat berinteraksi dengan server di belakang layar, sehingga tidak mempengaruhi halaman web secara keseluruhan. Artinya, halaman web tidak perlu dipanggil ulang secara keseluruhan hanya untuk mengganti sebagian dari isi halaman.

### 2.14.1 Kemampuan AJAX

Manfaat Ajax yang biasanya digunakan dalam suatu website, diantaranya:

1. *Real time validation*

   Selain untuk validasi terhadap data berupa email, pencocokan password, umur (berupa angka), contoh lainnya adalah pengecekan username. Hal ini sangat berguna, terutama pada kondisi yang tidak memungkinkan pengiriman keseluruhan data ke client yang dibutuhkan untuk melakukan validasi ketika halaman di-load di awal mula.
2. *Auto Complete* atau *Auto Suggest*

   *Auto Complete* yaitu dimana ketika user menuliskan satu kalimat yang belum utuh, maka secara otomatis akan tersedia daftar data-data yang menyambungkan kalimat tersebut secara utuh, sehingga user tidak perlu lagi mengetikkan kata yang dicari.
3. *Load and demand*

   Berdasarkan event tertentu, sebuah halaman HTML dapat mengambil tambahan data di belakang layar, sehingga memungkinkan browser menampilkan halaman web secara lebih cepat dan dapat diatur waktu penampilannya.
4. *Refresh data* dan *server push*

   Halaman web dapat mengambil data dari server tertentu untuk menampilkan informasi terkini, misalnya nilai tukar, dolar, stok, dan ramalan cuaca. Kemampuan ini memungkinkan halaman web melihat informasi terkini tanpa harus me-refresh/reload halaman web.
5. *Partial submit*

   Halaman web bisa melakukan submit (pengiriman) sebagian form terlebih dahulu, tanpa meminta halaman web untuk me-refresh-nya terlebih dahulu.
6. *Mashup* (mencampur data)

   Halaman web dapat menampilkan data menggunakan berbagai media seperti *proxy server side* ataupun menggunakan skrip eksternal. Mislnya menggabungkan nilai tukar dolar ke rupiah dengan halaman web e-commerce.

## 2.15 WampServer

WAMP merupakan kependekan dari Windows, Apache, MySQL dan PHP. Wamp adalah sebuah aplikasi yang dapat menjadikan sebuah komputer menjadi sebuah server. WAMP ini merupakan program yang opensource. Kegunaannya untuk membuat jaringan lokal sendiri dalam artian dapat membuat aplikasi berbasis web secara offline untuk masa coba-coba di komputer. Seorang user tidak perlu direpotkan dengan menginstall satu persatu dari komponen web server seperti Apache, MySQL dan PHP karena Wamp merupakan satu paket dari komponen-komponen tersebut.

Apache adalah Web server, MySQL adalah database (DBMS), PHP adalah bahasa scripting yang dapat memanipulasi informasi yang dibuat di database dan menghasilkan halaman web dinamis setiap dijalankan oleh browser. Program lain juga dapat dimasukkan dalam paket, seperti phpMyAdmin yang menyediakan antarmuka pengguna grafis untuk manajer database MySQL, atau bahasa scripting Python alternatif atau Perl.

## 2.16 MySQL

MySQL merupakan *software* yang tergolong sebagai DBMS yang bersifat *Open Source*. Open Source menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat MySQL), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara men*download* di internet secara gratis.

MySQL awalnya dibuat oleh perusahaan konsultan bernama TcXyang berlokasi di Swedia. Saat ini pengembangan MySQL berada di bawah naungan perusahaan Oracle. Adapun *software* yang dapat diunduh di situs www.mysql.com.

Sebagai *software* DBMS, MySQL memiliki sejumlah fitur seperti:

1. *Multiplatform*

   MySQL tersedia pada beberapa *platform*, seperti Windows, Linux, UNIX, dan lain-lain.
2. Andal, cepat, dan mudah digunakan.
3. MySQL tergolong sebagai *database server* (*server* yang melayani permintaan *database*) yang andal, dapat menangani *database* yang besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses *database*, sekaligus mudah untuk digunakan. Berbagai *tool* pendukung juga tersedia yang mungkin dibuat oleh pihak lain. MySQL juga mampu menangani sebuah tabel yang berukuran dalam *terabyte (*1024 *gigabyte*) namun ukuran sesungguhnya sangat bergantung pada batasan sistem operasi. Sebagai contoh, pada sistem operasi Solaris 9/10, batasan ukuran *file* sebesar 16 TB.
4. Jaminan keamanan akses

   MySQL mendukung pengamanan *database* dengan berbagai kriteria pengaksesan. Sebagai contoh, dimungkinkan untuk mengatur *user* tertentu agar bisa mengakses data yang bersifat rahasia, sedangakan *user* lain tidak boleh mengakses. MySQL juga mendukung konektivitas dengan menggunakan ODBC (*Open Database Connectivity*), *database* yang ditangani MySQL dapat diakses

melalui program yang dibuat oleh Visual Basic. MySQL juga mendukung program klien yang berbasis Java untuk berkomunikasi dengan *database* MySQL melalu JDBC (*Java Database Connectivity*). MySQL juga bisa diakses melalui aplikasi berbasis web.

5. Dukungan SQL

   Seperti tersirat dalam namanya, MySQL mendukung perintah SQL (*Standard Query Language*). Sebagaimana diketahui, SQL merupakan standar dalam pengaksesan *database relational.* Pengetahuan SQL akan memudahkan untuk menggunakan MySQL.

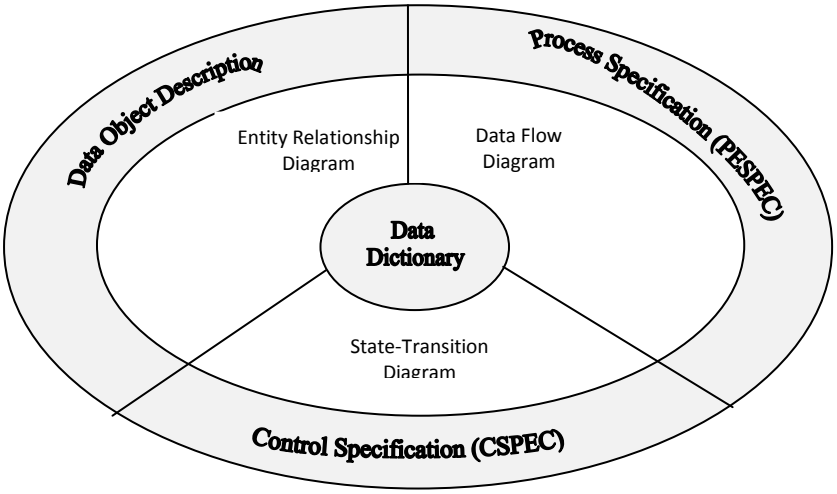
## 2.17 Pemodelan Analis Terstruktur [15]

Pada tingkat teknik, rekayasa perangkat lunak dimulai dengan serangkaian tugas pemodelan yang membawa kepada suatu spesifikasi lengkap dari persyaratan representasi dan representasi desan yang komprehensif bagi perangkat lunak yang akan dibangun. Model analis, yang sebenarnya merupakan serangkaian model, merupakan representasi teknis yang pertama dari sistem.

Analis terstruktur adalah aktivitas pembangunan model. Dengan menggunakan notasi yang sesuai dengan prinsip analisis operasional, menciptakan model yang menggambarkan muatan dan aliran informasi (data dan kontrol), membagi sistem secara fungsional dan secara *behavioral*, dan menggambarkan esensi dari apa yang harus dibangun. Analisis terstruktur bukan merupakan metode tunggal yang diaplikasikan secara konsisten oleh semua yang menggunakannya. Analisis terstruktur lebih merupakan campuran yang berkembang selama lebih 20 tahun.

### 2.17.1 Elemen Model Analis

Model analis harus mencapai tiga sasaran utama: (1) untuk menggambarkan apa yang dibutuhkan oleh pelanggan, (2) untuk membangun dasar bagi pembuatan desain perangkat lunak, (3) untuk mmbatasi serangkaian persyaratan yang dapat divalidasi begitu perangkat lunak dibangun. Untuk mencapai sasaran tersebut, model analisis yang ditarik selama analisis terstruktur berlangsung. Model analisis meliputi masing-masing diagram, spesifikasi, dan deskripsi, serta kamus yang ditunjukkan pada Gambar 2.13



**Gambar 2.13 Struktur Model Analis [15]**

Pada inti model ada kamus data (*data dictionary*), yaitu penyimpanan yang berisi deskripsi dari semua objek data yang dikonsumsi atau diproduksi oleh perangkat lunak. Disitu ada tiga diagram yang mengelilingi inti.

*Entity Relationship Diagram* (ERD) menggambarkan hubungan antara objek data. ERD adalah notasi yang digunakan untuk melakukan aktivitas pemodelan data. Atribut dari masing-masing objek data yang ditulis pada ERD dapat digambarkan dengan menggunakan *deskripsi objek data*.

*Data Flow Diagram* (DFD) melayani dua tujuan: (1) untuk memberikan mengenai bagaimana data ditransformasi pada saat data bergerak melalui system, dan (2) untuk mengambarkan fungsi-fungsi (dan sub-fungsi) yang mentransformasi aliran data. DFD memberikan informasi tambahan yang digunakan selama analisis domain informasi dan berfungsi sebagai dasar bagi pemodelan fungsi. Deskripsi setiap fungsi disajikan pada DFD diisikan dalam sebuah *spesifikasi proses/process specification (PSPEC).*

*State-transition diagram (STD)* menunjukkan bagaimana sistem bertingkah laku sebagai akibat dari kejadian eksternal. STD berfungsi sebagai dasar bagi pemodelan tingkah laku. Informasi tambahan mengenai aspek control dari perangkat lunak diisikan dalam *spesifikasi control/control specification (CSPEC).*

### 2.17.2 Data Flow Diagram (DFD)

Ide dari suatu bagan untuk mewakili arus data dalam suatu sistem bukanlah hal yang baru. Pada tahun 1967, Martin dan Estrin memperkenalkan suatu algoritma program dengan menggunakan symbol lingkaran dan panah untuk mewakili arus data. E. Yourdan dan L.L. Constantine juga menggunakan notasi simbol ini untuk menggambarkan arus data dalam perancangan program. G.E Whitehouse tahun 1973 juga menggunakan notasi semacam ini untuk membuat model-model sistem matematika. Penggunaan notasi arus data ini sangat membantu sekali untuk memahami suatus sistem pada semua tingkat kompleksitasnya seperti diungkapkan oleh Chris Gane dan Trish Sarson. Pada tahap analisis, penggunaan notasi ini sangat membantu sekali dalam komunikasi dengan pemakai sistem untuk memahami sistem secaa logika. Diagram yang menggunakan notasi-notasi ini untuk menggambarkan arus dari data sistem (DFD).

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (misalnya lewat telepon, surat, dan sebagainya) atau lingkungan fisik dimana data tersebut akan disimpan (misalnya *harddisk, tape* dan lain sebagainya). DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. DFD merupakan alat yang cukup popular sekarang ini, karena dapat menggambarkan arus data di dalam sistem dengan terstruktur dan jelas. Lebih lanjut DFD merupakan dokumentasi dari sistem yang baik [5].

### 2.17.3 Pemodelan Data

Pemodelan data menjawab serangkaian pertanyaan spesifik yang relevan dengan berbagai aplikasi pemrosesan data. Pertanyaan-pertanyaan tersebut meliputi "Apakah objek utama yang akan diproses oleh sistem?", "Bagaimana komposisi dari masing-masing objek data dab atribut apa yang menggambarkan objek tersebut?", "Di mana objek saat ini berada?", "Bagaimana hubungan antara masing-masing objek data dan bjek yang lainnya?", dan "Bagaimana hubungan antara objek dengan proses yang mentransformasikannya?".

Untuk menjawab pertanyaan-pertanyaan tersebut, metode pemodelan data menggunakan ERD. ERD memungkinkan perekayasa perangkat lunak mengidentifikasi objek data dan hubungannya dengan menggnakan notasi grafis. Pada konteks analisis terstruktur, ERD menetapkan semua data yang dimasukkan, disimpan, ditransformasi, dan diproduksi pada suatu aplikasi [15].

#### 2.17.3.1 Komponen Model Data [16]

Model data memperoleh masukan dari langkah perencanaan dan analisis. Di sini, pembuat model bersama dengan analisis mengumpulkan informasi tentang kebutuhan basis data dengan meninjau ulang dokumentasi yang ada dan mewawancarai pengguna akhir (*end user*).

Model data memiliki dua keluaran. Pertama, sebuah diagram *entity-relationship* yang menyajikan struktur data dalam bentuk bergambar. Diagram menjadi alat yang berharga untuk mengkomunikasikan model tersebut kepada *end user*. Komponen yang kedua adalah suatu dokumen data. Dokumen tersebut menguraikan objek data, relasi, dan aturan yang diperlukan oleh basis data secara detail. Kamus menyediakan detail yang diperlukan oleh pengembang basis data untuk membangun basis data fisik.

#### 2.17.3.2 Entity Relationship Diagram (ERD)

Model *Entity-Relationship* (ER) mula-mula diusulkan oleh Peter pada tahun 1976 sebagai cara untuk mempersatukan pandangan basis data jaringan dan relasional. Langkah sederhana dari model ER adalah model data konseptual yang memandang dunia nyata sebagai kesatuan (*entities*) dan hubungan (*relationships*). Komponen dasar model merupakan diagram *entity-relationship* yang digunakan untuk menyajikan objek data secara visual. Chen menulis bahwa model telah diperluas dan kini bisa digunakan untuk merancang basis data. Kegunaan model ER dalam perancangan basis data adalah [16]:

1. Mampu memetakan model relasional dengan baik. Pembangunan yang digunakan di dalam model ER dengan mudah diubah ke dalam tabel relasional.

2. Sederhana dan mudah dipahami hanya dengan sedikit pelatihan. Oleh Karena itu, model bisa digunakan oleh perancang basis data untuk mengkomunikasikan perancangan kepada pengguna akhirnya.
3. Model ini juga bisa digunakan sebagai suatu rencana perancangan oleh pengembang basis data untuk menerapkan suatu model data dalam perangkat lunak manajemen basis data spesifik.

#### 2.17.3.3 Objek Data, Atribut, dan Hubungan

Objek data adalah representasi dari hampir semua informasi gabungan yang harus dipahami oleh perangkat lunak. Dengan informasi gabungan kita mengartikan sesuatu yang memiliki sejumlah sifat atau atribut yang berbeda. Dengan demikian “luas” (sebuaah harga tunggal) tidak akan menjadi objek data yang valid, tetapi **dimensi** (gabungan antara ketinggian, luas, dan kedalaman) dapat ditentukan sebuah objek.

Atribut menentukan properti suatu objek data dan mengambil salah satu dari tiga karakteristik yang berbeda. Atribut dapat digunakan untuk (1) menamai sebuah contoh dari objek data, (2) mnggambarkan contoh, (30 membuat referensi ke contoh yang lain pada tabel yang lain. Sebagai tambahan, satu atribut atau lebih harus didefinisikan sebagai sebuah *pengidentifikasi* - di mana atribut pengidentifikasi akan menjadi sebuah “kunci” ketika kita ingin menemukan sebuah contoh dari objek data. Hubungan objek data disambungkan satu dengan yang lainnya dengan bebagai macam cara [15].

#### 2.17.3.4 Kardinalitas dan Modalitas

Kardinalitas merupakan spesifikasi dari sejumlah peristiwa dari satu objek yang dapat dihubungkan ke sejumlah peristiwa dari objek lain. Kardinalitas biasanya diekspresikan sebagai secara sederhana ‘satu’ atau ‘banyak’. Dengan mempertimbangkan semua kombinasi dari ‘satu’ dan ‘banyak’, dua objek dapat dihubungkan sebagai:

1. Satu-ke-satu (1:1), suatu peristiwa dari objek ‘A’ dapat berhubungan dengan satu dan hanya satu kejadian dari objek ‘B’, begitu pula sebaliknya.
2. Satu-ke-banyak (1:N), suatu peristiwa dari objek ‘A’ dapat berhubungan dengan satu atau lebih kejadian dari objek ‘B’, tetapi sebuah kejadian ‘B’ dapat berhubungan dengan hanya satu kejadian dari ‘A’.

3. Banyak-ke-banyak (M:N), suatu peristiwa dari objek ‘A’ dapat berhubungan dengan satu atau lebih kejadian dari objek ‘B’, begitu pula sebaliknya.

Kardinalitas mendefinisikan jumlah maksimum dari hubungan objek ang ikut serta di dalam sebuah hubungan, tetapi tidak memberikan sebuah indikasi apakah objek data tertentu harus berpartisipasi di dalam hubungan atau tidak. Untuk menspesifikasikan informasi ini, model data akan menambahkan modalitas ke *object-relationship pair*.

Modalitas dari suatu hubungan adalah nol bila tidak ada kebutuhan eksplisit untuk hubungan yang terjadi atau hubungan itu bersifat opsional. Modalitas bernilai satu jika suatu kejadian dari hubungan merupakan perintah [15].

## 2.18 Teknik Pengumpulan Data

Dalam penelitian dan pengujian dilakukan teknik pengumpulan data menggunakan kuesioner. Dari hasil kuesioner tersebut akan dilakukan perhitungan untuk mengambil kesimpulan terhadap penilaian penerapan sistem yang baru dan mengukur sikap dari pengguna aplikasi yang telah dibangun dengan menggunakan skala likert.

### 2.18.1 Kuesioner atau Angket

Sebagian besar penelitian umumnya menggunakan kuesioner sebagai metode yang dipilih untuk mengumpulkan data. Kuesioner atau angket merupakan suatu teknik pengumpulan data secara tidak langsung (peneliti tidak langsung bertanya jawab dengan responden). Kuesioner adalah suatu daftar yang berisi pertanyaan-pertanyaan untuk tujuan khusus yang memungkinkan penganalisis untuk mengumpulkan data mengenai sikap, keyakinan, perilaku, dan karakteristik dari orang yang berada di dalam organisasi serta pendapat dari responden yang dipilih. Responden mempunyai kebebasan untuk memberikan jawaban atau respon sesuai dengan persepsinya. Dalam membuat kuesioner, harus dilalui prosedur sebagai berikut:

1. Merumuskan tujuan yang akan dicapai dengan kuesioner.
2. Mengidentifikasikan variabel yang akan dijadikan sasaran kuesioner.

3. Menjabarkan setiap variabel menjadi sub-varibel yang lebih spesifik dan tunggal.
4. Menentukan jenis data yang akan dikumpulkan, sekaligusn untuk menentukan teknik analisisnya.

Ada dua jenis format kuesioner, yaitu

1. Format bebas (free format) berisi dengan pertanyaan-pertanyaan yang harus diisi oleh responden di tempat yang sudah disediakan.
2. Kuesioner format pasti (fixed format), kuesioner tipe ini mempunyai beberapa bentuk pertanyaan, yaitu:
   a. Check-off Question, jenis dari pertanyaan-pertanyaan ini dibuat sehingga responden dapat memeriksa jawaban-jawaban yang sesuai.
   b. Yes/No Question, jenis dari pertanyaan ini memungkinkan responden untuk menjawab “YA” atau “TIDAK”.
   c. Opinion/ Choice Question, jenis dari pertanyaan-pertanyaan ini memungkinkan responden untuk memberikan pendapatnya.

Kelebihan teknik kuesioner, yaitu

1. Kuesioner baik untuk sumber data yang banyak dan tersebar.
2. Responden tidak merasa terganggu, karena dapat mengisi kuesioner dengan memilih waktunya sendiri.
3. Kuesioner secara relatif lebih efisien untuk sumber data yang banyak.
4. Karena kuesioner baisanya tidak mencatumkan identitas responden, maka hasilnya dapat lebih objektif.

Kekurangan teknik kuesioner, yaitu

1. Kuesioner tidak menggaransi responden untuk menjawab pertanyaan dengan sepenuh hati.
2. Kuesioner cenderung tidak fleksibel, artinyaa pertanyaan yang harus dijawab terbatas yang dicantumkan di kuesioner saja, tidak dapat dikembangkan lagi sesuai dengan situasinya.
3. Pengumpulan sampel tidak dapat dilakukan secara bersama-sama dengan daftar pertanyaan.
4. Kuesioner yang lengkap sulit untuk dibuat.

## 2.19 Skala *Likert* [17]

Skala *Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dengan skala *Likert*, maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun item–item instrumen yang dapat berupa pernyataan atau pertanyaan. Jawaban setiap item instrumen yang menggunakan skala likert mempunyai gradasi dari sangat positif sampai sangat negatif, yang dapat berupa kata–kata antara lain: sangat setuju dengan skor 5, setuju dengan skor 4, ragu–ragu dengan skor 3, tidak setuju dengan skor 2, sangat tidak setuju dengan skor1.

Instrument peneletian yang menggunakan skla likert dapat dibuat dalam bentuk *checklist* ataupun pilihan ganda.

Contoh :

Berilah jawaban pertanyaan berikut sesuai dengan pernyataan anda, dengan cara memberi tanda (√) pada kolom yang tersedia.

**Tabel 2.1 Tabel pertanyaan skala likert**

| No. | Pertanyaan | Jawaban | | | | |
|---|---|---|---|---|---|---|
| | | SS | ST | RG | TS | STS |
| 1. | Prosedur kerja yang baru itu akan segera diterapkan di perusahaan anda. | | √ | | | |
| 2. | ..... | | | | | |

SS = Sangat Setuju diberi skor 5
ST = Setuju diberi skor 4
RG = Ragu-ragu diberi skor 3

| | | |
|---|---|---|
| TS = Tidak Setuju | diberi skor | 2 |
| STS = Sangat Tidak Setuju | diberi skor | 1 |

Kemudian dengan teknik pengumpulan data angket, maka instrument tersebut misalnya diberikan kepada 100 orang karyawan yang diambil secara random. Dari 100 orang pegawai setelah dilakukan analis misalnya:

| | | |
|---|---|---|
| 25 | Orang menjawab | SS |
| 40 | Orang menjawab | ST |
| 5 | Orang menjawab | RG |
| 20 | Orang menjawab | TS |
| 10 | Orang menjawab | STS |

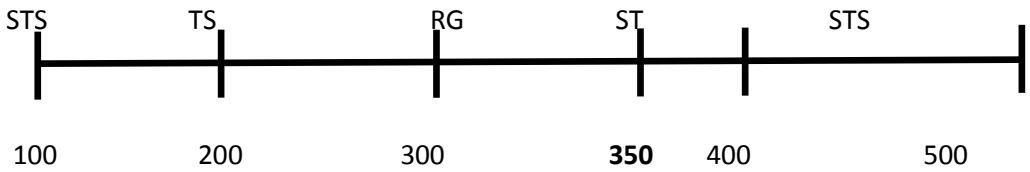
Berdasarkan data tersebut 65 orang (40+25) atau 65% karyawan menjawab setuju dan sangat setuju. Jadi kesimpulannya mayoritas karyawan setuju adanya metode kerja baru.

Data interval tersebut juga dapat dianalisis dengan menghitung rata-rata jawaban berdasarkan skoring setiap jawaban dari responden. Berdasarkan skor yang telah ditetapkan dapat dihitung sebagai berikut :

| | | |
|---|---|---|
| Jumlah skor untuk 25 orang yang menjawab SS | = 25 x 5 | = 125 |
| Jumlah skor untuk 40 orang yang menjawab ST | = 40 x 4 | = 160 |
| Jumlah skor untuk 5 orang yang menjawab RG | = 5 x 3 | = 15 |
| Jumlah skor untuk 20 orang yang menjawab TS | = 20 x 2 | = 40 |
| Jumlah skor untuk 10 orang yang menjawab STS | = 10 x 1 | = 10 |
| Jumlah total | | = 350 |

Jumlah skor ideal (kriterium) untuk seluruh item = 5 x 100 = 500 (seandainya semua menjawab SS). Jumlah skor yang diperoleh dari penelitian 350. Jadi berdasarkan data tu maka tingkat persetujuan terhadap metode kerja baru itu = (350 : 500) x 100% = 70% dari yang diharapkan (100%).

Secara kontinum dapat digambarkan seperti berikut :




Jadi berdasarkan data yang diperoleh dari 100 responden maka rata-rata 350 terletak pada daerah setuju.

Data yang diperoleh adalah data interval, dan biasanya skala ini digunakan untuk mengukur sikap atau karakteristik tertentu yang dipunyai oleh seseorang.