BAB II

TINJAUAN PUSTAKA

2.1. Latar Belakang dan Sejarah Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata Kabupaten Indramayu

Rencana Pembangunan Jangka menengah Satuan Kerja Perangkat Daerah (RPJM) SKPD Dinas Pemuda Olahraga Kebudayaan dan Pariwisata Kabupaten Indramayu menguraikan kondisi saat ini Dinas Pemuda Olahraga Kebudayaan dan Pariwisata serta Visi dan Misi nya yang diharapkan memberikan arah yang tepat sehingga Dinas Pemuda Olahraga Kebudayaan dan Pariwisata mampu memberikan kontribusi nyata yang signifikan dalam rangka mewujudkan Visi Pemerintah Kabupaten Indramayu yaitu terwujudnya masyarakat Indramayu Remaja (Religius, Maju, Mandiri dan Sejahtera).

Dalam merealisasikan Rencana pembangunan Jangka Menengah Satuan Kerja Perangkat daerah dijabarkan kedalam Rencana Kerja Satuan Kerja Perangkat daerah (Renja SKPD) disusun setiap tahun yang memuat prioritas pembangunan/ kegiatan Satuan Kerja Perangkat daerah yang menuangkan Rencana Kegiatan dan Pendanaan.

Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata Kabupaten Indramayu adalah penggabungan dari kantor Pemuda dan Olahraga Kabupaten Indramayu dengan kantor Kebudayaan dan Pariwisata Kabupaten Indramayu, merujuk kepada PERDA No. 8 Tahun 2008 Tentang Dinas Daerah Kabupaten Indramayu, lanjutan Peraturan Bupati Indramayu Nomor 35 Tahun 2008

Tentang Organisasi Dan Tata Kerja Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata Kabupaten Indramayu, yang disahkan oleh PERDA pada tanggal 10 Agustus 2008 dan PERBUP pada tanggal 24 November 2008.

2.2. Visi Misi Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata Kabupaten Indramayu

2.2.1. Visi

Visi jangka menengah Dinas Pemuda Olahraga Kebudayaan dan Pariwisata Kabupaten Indramayu adalah "Terwujudnya Kabupaten Indramayu yang Religius, Maju, Mandiri dan Sejahtera melalui pemberdayaan sektor pemuda dan olahraga serta pengembangan budaya daerah dalam mendukung tercapainya Kabupaten Indramayu sebagai daerah tujuan Wisata di Jawa Barat".

2.2.2. Misi

Misi Jangka menengah Dinas Pemuda Olahraga Kebudayaan dan Pariwisata Kabupaten Indramayu adalah :

a. Meningkatkan kualitas produk dan terselenggaranya standarisasi pelayanan prima.
b. Meningkatkan Sadar Wisata dan Budaya serta sadar berolahraga diseluruh lapisan masyarakat.
c. Meningkatkan kegiatan promosi kebudayaan dan Pariwisata yang didukung oleh SDM dan informasi yang handal.

d. Menjadikan sektor Kebudayaan dan Pariwisata sebagai Wahana pemberdayaan ekonomi rakyat serta meningkatkan peran serta pemuda dalam kewirausahaan.
e. Mewujudkan sektor Kebudayaan dan Pariwisata dan sarana prasarana olahraga sebagai salah satu andalan yang mampu mendongkrak PAD.
f. Meningkatkan kualitas SDM melalui potensi olahraga dan peran serta pemuda.

2.3. Kedudukan, Tugas Pokok dan Fungsi Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata Kabupaten Indramayu

2.3.1. Kedudukan

Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata adalah unsur pelaksana Pemerintah Daerah dibidang Pemuda, Olahraga, Kebudayaan dan Pariwisata.

Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata dipimpin oleh seorang kepala yang berada dibawah dan bertanggung jawab kepada Bupati melalui Sekretaris Daerah.

2.3.2. Tugas Pokok dan Fungsi

Sesuai dengan Keputusan Bupati Nomor. 35 tahun 2008 Pasal 3 bahwa Dinas Pemuda Olahraga Kebudayaan dan Pariwisata mempunyai Tugas Pokok malaksanakan urusan Pemerintah Daerah dibidang Pemuda Olahraga Kebudayaan dan Pariwisata berdasarkan asas otonomi dan tugas pembantuan. Untuk menyelenggarakan tugas pokok tersebut sebagaimana dimaksud pada Peraturan

Bupati Indramayu No. 35 tahun 2008 Dinas Pemuda Olahraga Kebudayaan dan Pariwisata mempunyai fungsi :

a. Perumusan kebijaksanaan teknis dibidang Pemuda, Olahraga, Kebudayaan dan Pariwisata;
b. Penyelenggaraan urusan pemerintahan dan pelayanan umum Pemuda, Olahraga, Kebudayaan dan Pariwisata;
c. Pembinaan dan Pelaksanaan tugas dibidang Pemuda Olahraga Kebudayaan dan Pariwisata;
d. Pelaksanaan pelayanan teknis administrative ketaatausahaan;
e. Pelaksanaan penelolaan UPTD;
f. Pelaksanaan tugas lain yang diberikan oleh Bupati sesuai dengan tugas dan fungsinya.

2.4. Bentuk dan Makna Lambang Daerah Kabupaten Indramayu

2.4.1. Bentuk Lambang Daerah

Berdasarkan Peraturan Daerah Kabupaten Daerah Tingkat II Indramayu Nomor 8/1988, Lambang daerah berbentuk Perisai dengan latar belakang warna Biru dan warna pinggir Hitam, di dalamnya terdapat tulisan “DHARMA AYU” dan “MULIH HARJA”, serta gambar : Tali, Bulatan, Cakra, Bintang, Sungai Cimanuk, Perahu, Laut, Padi, Mangga dan Selendang.



Gambar II.1. Lambang Daerah Kabupaten Indramayu.

2.4.2. Makna Lambang Daerah

Dibawah ini adalah makna dari tiap-tiap bagian dari lambang daerah Kabupaten Indramayu.

1. Perisai sebagai senjata perang dengan latar belakang warna Biru dan warna pinggir Hitam melambangkan rasa aman, ulet dan penuh kesungguhan bagi rakyat dalam membangun daerahnya.
2. Tulisan “DARMA AYU” yang berwarna Merah dengan latar belakang Putih melambangkan :
   a. Tulisan “DARMA AYU” berasal dari nama Nyi Endang Darma yang ayu, yaitu orang kedua pendiri Indramayu dan nama Indramayu asalnya dari DARMA AYU.
   b. Warna Merah pada tulisan “DARMA AYU” dengan latar belakang Putih melambangkan bahwa Nyi Endang Darma seorang wanita yang berani dalam membela kesucian dan kebenaran.

3. Tali yang mengikat melingkari bulatan dan dua ujungnya melambangkan ikatan hubungan yang erat antara Pemerintah Daerah dengan Rakyatnya.
4. Ditengah-tengah perisai ada bulatan yang melambangkan tekat persatuan dan kesatuan dari segenap lapisan masyarakat.
5. Warna Hijau pada bulatan melambangkan kesuburan daerah yang memberikan kemakmuran penduduk.
6. Pada bulatan terdapat gambar-gambar :
    a. Cakra adalah senjata ampuh peninggalan Wiralodra, pendiri Indramayu yang melambangkan kewibawaan dan kesentausaan.
    b. Bintang bersudut lima berwarna Kuning Emas :
        1) Bintang melambangkan Ketuhanan Yang Maha Esa.
        2) Bersudut lima melambangkan Falsafah Negara.
        3) Warna Kuning emas melambangkan Kedaulatan.
    c. Padi, Mangga, Perahu, Laut dan Sungai Cimanuk melambangkan sumber kehidupan rakyat Indramayu.
    d. Garis gelombang Sungai Cimanuk berjumlah 7 (tujuh), melambangkan tanggal kelahiran Indramayu.
    e. Garis gelombang Laut berjumlah 10 (sepuluh), melambangkan bulan kelahiran Indramayu.
    f. Biji Padi setiap sisi berjumlah 15 (lima belas) di bawah dan 27 (dua puluh tujuh) di atas, melambangkan tahun kelahiran Indramayu yaitu tahun 1527.

g. Selendang warna Kuning emas Pusaka Nyi Endang Darma, melambangkan Pemerintah Daerah yang berwibawa dan demokratis yang senantiasa membela kepentingan rakyat, Daerah dan Negara.

h. Tulisan "MULIH HARJA" merupakan motto juang rakyat Indramayu, yang dipetik dari Prasasti Arya Wiralodra dan dituliskan di tengah selendang dengan warna Hitam yang berarti bahwa suatu saat nanti Indramayu akan kembali makmur.

2.5. Badan Hukum Dinas Pemuda Olahraga Kebudayaan dan Pariwisata Kabupaten Indramayu

Bentuk dan badan hukum Dinas Pemuda Olahraga Kebudayaan dan Pariwisata Kabupaten Indramayu adalah Pemerintahan.

2.6. Struktur Organisasi dan Pejabat Struktur Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata Kabupaten Indramayu

2.6.1. Susunan Organisasi

Sesuai Peraturan Bupati Nomor 35 Tahun 2008 Bab III Pasal 5 tentang Organisasi dan Tata Kerja Dinas Pemuda Olahraga Kebudayaan dan Pariwisata Kabupaten Indramayu Mempunyai Susunan sebagai berikut :

a. Kepala

b. Sekretariat, membawahkan :

   1. Sub Bagian Umum dan Kepegawaian
   2. Sub Bagian Keuangan

3. Sub Bagian Perencanaan dan Evaluasi

c. Bidang Pemuda Olahraga, membawahkan :

1. Seksi Pembinaan dan Pengembangan Pemuda
2. Seksi Pelatihan dan Olahraga
3. Seksi Kompetisi dan Perlombaan

d. Bidang Kebudayaan dan Pariwisata, Membawahkan :

1. Seksi Kebudayaan
2. Seksi Pariwisata
3. Seksi Promosi

e. Kelompok Jabatan Fungsional

Adapun Tugas Pokok masing-masing unsur Organisasi di Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata Kabupaten Indramayu, adalah sebagai berikut :

a. Kepala Dinas mempunyai tugas pokok memimpin, mengkoordinasikan dan mengendalikan Dinas dalam melaksanakan urusan Pemerintahan Daerah dibidang Pemuda, Olahraga, Kebudayaan dan Pariwisata.

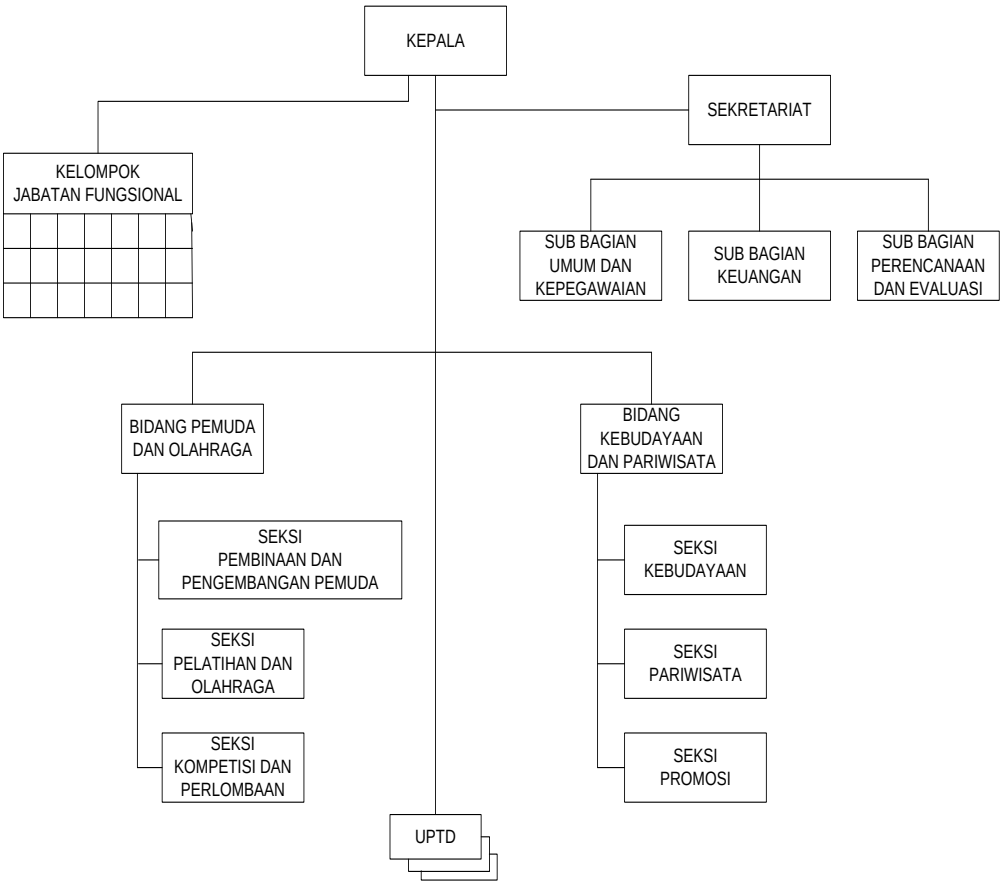
b. Sekretaris mempunyai tugas pokok memimpin, mengkoordinasikan dan mengendalikan tugas-tugas dibidang pengelolaan pelayanan kesekretariatan yang meliputi pengkoordinasian perencanaan program, pengelolaan urusan umum, perlengkapan, kepegawaian serta pengelolaan keuangan.

c. Kepala Bidang Pemuda dan Olahraga mempunyai tugas pokok melaksanakan kegiatan dibidang pembinaan dan pengembangan pemuda, pelatihan olahraga, kompetisi dan perlombaan.

d. Kepala Bidang Kebudayaan dan Pariwisata mempunyai tugas pokok melaksanakan kegiatan dibidang kebudayaan, pariwisata dan promosi.

2.6.2. Sturktur Organisasi

Berikut ini adalah struktur organisasi dari Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata Kabupaten Indramayu.



Gambar II.2. Struktur Organisasi Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata Kabupaten Indramayu.

2.7. Landasan Teori

2.7.1. Konsep Dasar Sistem

Sistem adalah suatu pengorganisasian yang saling berinteraksi, saling tergantung dan berintegrasi dalam kesatuan variabel atau komponen (Riyanto, Putra dan Indelarko, 2009). Terdapat dua kelompok pendekatan, yaitu menekankan pada prosedur dan komponen atau elemennya.

Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkelompok dan bekerjasama untuk melakukan kegiatan pencapaian sasaran tertentu. Makna dari prosedur sendiri, yaitu urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakan, kapan (*when*) dikerjakan dan bagaimana (*how*) mengerjakannya.

Sedangkan pendekatan yang menekankan pada komponen, mendefinisikan "sistem" sebagai berikut,

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

Beberapa penulis mendefinisikan "sistem" yang menekankan pada elemen atau komponennya.

Berkaitan dengan Sistem Informasi Geografis, dimana implementasinya memanfaatkan teknologi komputer, membawa "sistem" yang dimaksud adalah sistem berbasis komputer. Sistem berbasis komputer didefinisikan sebagai : Serangkaian atau tatanan elemen-elemen yang diatur untuk mencapai tujuan yang

ditentukan sebelumnya melalui pemrosesan informasi (Riyanto, Putra dan Indelarko, 2009).

2.7.2. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu di dalamnya, karakteristik sistem tersebut (Riyanto, Putra dan Indelarko, 2009) , yaitu :

1. Komponen sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk suatu kesatuan.

2. Batas sistem

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Lingkungan Luar Sistem

   Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

4. Penghubung Sistem

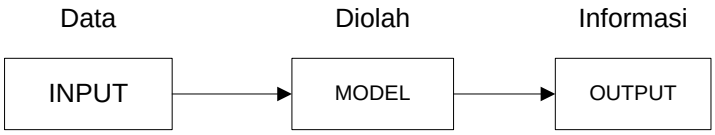
   Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Maka suatu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk suatu satu kesatuan.

2.7.3. Data

Data sering disebut sebagai bahan informasi. Data adalah fakta yang dikumpulkan dari pengukuran atau pengamatan. Definisi lain dari Data adalah

fakta mentah (dapat berupa angka, huruf, karakter khusus) yang menyampaikan sedikit arti. Agar data-data terkumpul menjadi berarti dan memberi manfaat, maka data-data tersebut harus diproses lebih lanjut (Riyanto, Putra dan Indelarko, 2009).

Data adalah sumber dari informasi. Data merupakan bentuk jamak dari bentuk tunggal datum atau data-item. Data adalah kenyataan yang menggambarkan sesuatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat yang tertentu. Kesatuan nyata (*fact* dan *entity*) adalah berupa suatu objek nyata seperti tempat, benda dan orang yang betul-betul ada dan terjadi. Agar menjadi informasi yang berguna, data perlu diolah melalui sebuah siklus. Siklus ini disebut siklus pengolahan data (*data processing life cycle*) (Riyanto, Putra dan Indelarko, 2009).



Gambar II.3. Siklus Pengolahan Data. (Sumber: Riyanto, Putra dan Indelarko, 2009).

2.7.4. Informasi

Informasi adalah arti dari hubungan dan penafsiran data yang mengijinkan seseorang membuat keputusan. Informasi dikatakan berharga jika informasi itu mempengaruhi proses pengambilan keputusan lebih baik. Sasaran utama dari sistem informasi adalah menyediakan informasi yang akurat dan penting. Informasi juga dapat berarti beberapa kesatuan yang tak terukur yang dapat mengurangi ketidakpastian tentang suatu peristiwa atau langkah.

Informasi merupakan pemrosesan data yang tampak dalam konteks untuk menyampaikan arti kepada orang lain. Definisi lain dari Informasi, yaitu (Riyanto, Putra dan Indelarko, 2009).

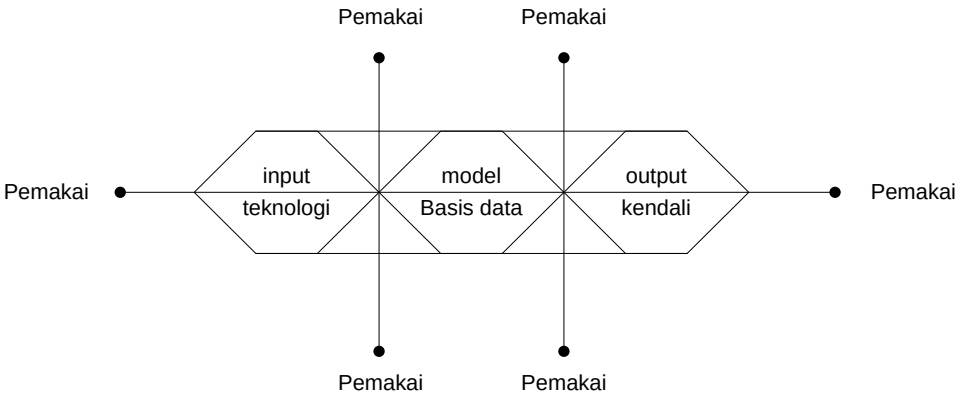
Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

2.7.5. Sistem Informasi

Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Riyanto, Putra dan Indelarko, 2009).

2.7.6. Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan, yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok basis data (*database block*) dan blok kendali (*controls block*). Sebagai suatu sistem keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya (Riyanto, Putra dan Indelarko, 2009).



Gambar II.4. Blok Sistem Informasi yang berinteraksi. (Sumber: Riyanto, Putra dan Indelarko, 2009).

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 matematika yang akan memanipulasi input dan data yang tersimpan di *databasei* dengan cara yang 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 *toll 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 tiga bagian utama, yaiut teknisi (*humanware* dan *brainware*), perangkat lunak (*software*) dan perangkat keras (hardware). Teknisi dapat berupa orang-orang yang mengetahui teknologi dan membuatnya dapat beroperasi.

5. Blok 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 di dalam *database* untuk keperluan penyediaan informasi lebih lanjut. Data di dalam *database* perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas.

   Organisasi *database* yang baik juga berguna untuk efisiensi kapasitas penyimpannya. *Database* diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management System*).

6. Blok Kendali

   Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sitem 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-kesalahan dapat langsung cepat diatasi.

2.7.7. Sistem Informasi Geografis

2.7.7.1. Sejarah Pengembangan Sistem Informasi Geografis

Saat 3500 tahun yang lalu, di dinding gua Lascaux, Perancis, para pemburu Cro-Magnon menggambah hewan mangsa mereka, juga garis yang dipercaya sebagai rute migrasi hewan-hewan tersebut. Catatan awal ini sejalan dua elemen struktur pada sistem informasi geografis modern sekarang ini. Pada tahun 1700-an teknik survei modern untuk pemetaan topografis diterapkan, termasuk juga versi awal pemetaan tematis.

Awal abad ke-20 memperlihatkan pengembangan "litografi foto", dimana peta dipisahkan menjadi beberapa lapisan. Perkembangan perangkat keras komputer yang dipacu oleh penelitian senjata nuklir membawa aplikasi pemetaan menjadi multifungsi pada awal tahun 1960-an.

Thun 1967 merupakan awal pengembangan SIG yang diterapkan di Ottawa, Ontario oleh Departemen Energi, Pertambangan Dan Sumber Daya. Dikembangkan oleh Roger Tomlinson, yang kemudian disebut CGIS (*Canadian GIS* - SIG Kanada), digunakan untuk menyimpan, menganalisis dan mengolah data yang dikumpulkan untuk Inventarisasi Tanah Kanada (CLI - *Canadian Land Inventory*), sebuah inisiatif untuk mengetahui kemampuan lahan di wilayah pedesaan Kanada dengan memetakan berbagai informasi pada tanah, pertanian, pariwisata, alam bebas, unggas dan penggunaan tanah pada skala 1:250000.

CGIS merupakan sistem pertama didunia dan hasil dari perbaikan aplikasi pemetaan yang memiliki kemampuan timpang susun (*overlay*), penghitungan, pendijitalan/pemindaian, mendukung sistem koordinat *national* yang membentang

di atas benua Amerika, memasukkan garis sebagai *arc* yang memiliki topologi dan menyimpan atribut dan informasi lokasional pada berkas terpisah.

Indonesia sudah mengadopsi sistem ini sejak Pelita ke-2 ketika LIPI mengundang UNESCO dalam menyusun "Kebijakan dan Program Pembangunan Lima tahun Tahap Kedua (1974-1979)" dalam penggunaan ilmu pengetahuan, teknologi dan riset.

2.7.7.2. Definisi Sistem Informasi Geografis (SIG)

Geografi adalah ilmu yang mempelajari permukaan bumi dengan menggunakan pendekatan keruangan, ekologi dan kompleks wilayah. Ketersediaan data yang bersifat geografi, dimana memiliki atribut utama keruangan, akan memudahkan banyak kepentingan. Sistem Informasi Geografis adalah sistem informasi khusus yang mengelolah data yang memiliki informasi spasial (bereferensi keruangan), atau dalam arti yang lebih sempit, adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelolah dan menampilkan informasi bereferensi geografis (Riyanto, Putra dan Indelarko, 2009).

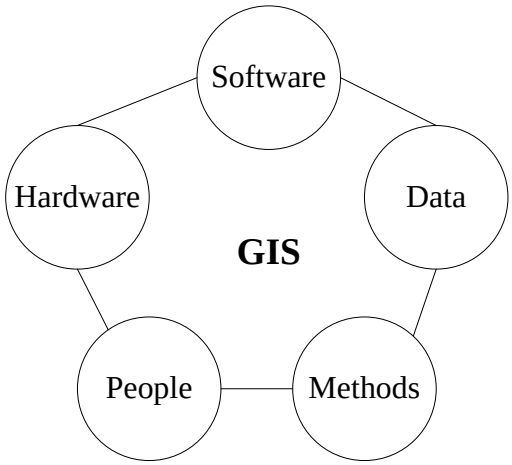
Aplikasi SIG terlibat dalam berbagai bidang disiplin ilmu, diantaranya yaitu :

1. Pemetaan Tanah dan Pemetaan Prasarana Kota.
2. Pemetaan Kartografi dan peta tematik.
3. Ukuran tanah dan fotogrametri.
4. Pengindraan jauh dan analisa citra.
5. Ilmu Komputer.

6. Perencanaan Wilayah.
7. Ilmu Tanah.
8. Geografi.

2.7.7.3. Komponen-komponen SIG

Komponen-komponen Sistem Informasi Geografis memiliki keterkaitan satu sama lainnya (Riyanto, Putra dan Indelarko, 2009).



Gambar II.5. Komponen SIG. (Sumber: Riyanto, Putra dan Indelarko, 2009).

1. Perangkat Keras (Hardware) Komputer. Terdiri dari beberapa komponen, yaitu :
   a. CPU (*Central Processing Unit*)
   b. *Memory* (Utama dan Tambahan)
   c. *Storage* (alat penyimpanan data dan informasi)
   d. Alat Masukkan (*Input Devices*)
   e. Alat Keluaran (*Output Devices*)

2. Perangkat Lunak (Software) Komputer

   Perangkat lunak yang dimaksud adalah yang mempunyai fungsi pemasukan data, manipulasi data, penyimpanan data, analisis data dan penayangan informasi geografis. Perangkat lunak SIG terdiri atas :

   a. Sistem Operasi, seperti Windows, Linux, UNIX, dan lain-lain.
   b. *Compiler*, menerjemahkan program yang ditulis dalam bahasa komputer pada kode mesin sehingga CPU mampu menjalankan program yang harus dieksekusi.
   c. Program Aplikasi pembangunan SIG, seperti MapInfo, ArcView, ArcInfo, ArcGIS, dan lain-lain.

3. Data dan Informasi Geografis

   Data yang dapat diolah dalam SIG merupakan fakta-fakta dipermukaan bumi yang memiliki referensi keruangan baik referensi secara relatif maupun referensi secara absolut dan disajikan dalam sebuah peta.

4. Sumberdaya Manusia

   Sumberdaya manusia yang terlatih merupakan sebagai komponen terakhir dari SIG. Perannya adalah sebagai pengoperasi perangkat keras dan perangkat lunak, serta menangani data geografis dengan kedua perangkat tersebut.

5. Methods (Prosedur)

   Model dan teknik pemrosesan yang perlu dibuat untuk berbagai aplikasi SIG.

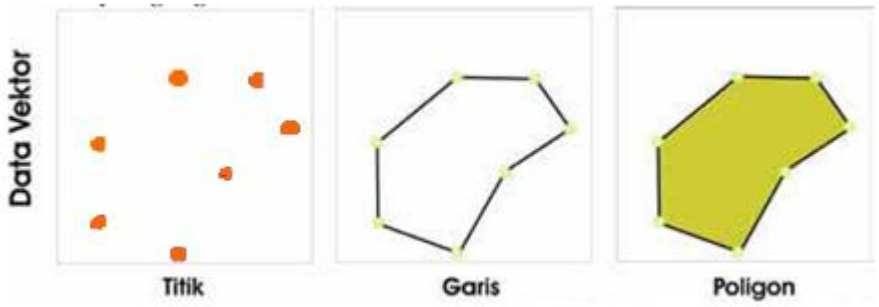
2.7.7.4. Model Data dalam SIG

Data geografis diorganisir menjadi dua bagian, yaitu Data Spasial dan Data Atribut/Tabular.

1. Data Spasial, yang menyimpan kenampakan-kenampakan permukaan bumi, seperti jalan, sungai, pemukiman, jenis penggunaan tanah, jenis tanah, dan lain-lain. Model data spasial dibedakan menjadi dua, yaitu : Model Data Vektor dan Model Data Raster.

   a. Model Data Vektor

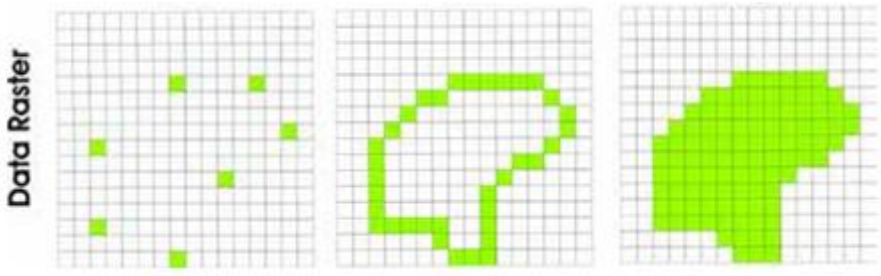
   Diwakili oleh simbol-simbol atau yang selanjutnya dalam SIG dekenal dengan *feature*, seperti *feature* titik (*point*), *feature* garis (*line*), dan *feature* area (*surface*). Data spasial ditampilkan, ditempatkan dan disimpan sebagai titk-titik, poligon, garis atau kurva beserta atributnya. Pada model data ini, poligon, garis atau kurva merupakan kumpulan titik-titik terurut yang dihubungkan. Pada poligon, titik awal dan titik akhir memilki nilai koordinat yang sama, sehingga bentuknya menjadi tertutup sempurna.



Gambar II.6. Data Vektor.

b. Model Data Raster

Merupakan data yang sangat sederhana, dimana setiap informasi disimpan dalam petak-petak bujursangkar (*grid*), yang membentuk sebuah bidang. Petak-petak bujur sangkar itu disebut dengan *pixel* (*picture element*). Posisi sebuah *pixel* dinyatakan dengan baris ke-m dan kolom ke-n. Data spasial ditampilkan, ditempatkan dan disimpan dalam struktur matriks atau kumpulan *pixel* yang membentuk *grid*. Setiap *pixel* memiliki koordinat dan resolusi (ukuran objek di permukaan bumi yang diwakili oleh satu *pixel*). Akurasi peta akan bergantung pada resolusi setiap *pixel*, semakin kecil resolusinya maka semakin akurat sebuah peta dan semakin detil objek peta yang digambarkan.



Gambar II.7. Data Raster.

2. Data Atribut/Tabular, yang menyimpan atribut dari kenampakan-kenampakan permukaan bumi tersebut. Misalnya, tanah yang memilki atribut tekstur, kedalaman, struktur, pH, dan lain-lain. Model data tersebut tersimpan ke dalam bentuk baris (*record*) dan kolom (*field*).

2.7.8. Sistem Informasi Geografis Berbasis Web (*Web GIS*)

Web GIS adalah sistem informasi geografis yang didistribusikan di seluruh lingkungan jaringan komputer untuk mengintegrasikan, menyebarkan, dan mengkomunikasikan informasi geografis secara visual di *World Wide Web* melalui *internet* (Riyanto, 2010).

2.7.9. Peta

2.7.9.1. Pengertian Peta

Peta merupakan penyajian grafis dari permukaan bumi dalam skala tertentu dan digambarkan pada bidang datar melalui sistem proyeksi peta dengan menggunakan simbol-simbol tertentu sebagai perwakilan dari objek-objek spasial di permukaan bumi (Riyanto, Putra dan Indelarko, 2009). Adapun syarat-syarat yang harus dimiliki pada peta, yaitu :

a. Peta tidak boleh membingungkan.
b. Peta harus dengan mudah dapat dimengerti atau ditangkap maknanya oleh si pemakai peta.
c. Peta harus memberikan gambaran yang sebenarnya.

2.7.9.2. Klasifikasi Peta

Berikut macam-macam peta, dimana peta dapat ditinjau dari 4 segi, yaitu sebagai berikut :

1. Macam peta ditinjau dari segi jenis.
    a. Peta Foto, peta yang dihasilkan dari mosaik foto udara atau ortofoto yang dilengkapi garis kontur, nama, dan legenda.

b. Peta Garis, peta yang menyajikan detail alam dan buatan manusia dalam bentuk titik, garis, dan luasan.

2. Macam peta ditinjau dari skala.
   a. Peta skala besar, yaitu peta dengan skala 1:50.000 atau lebih besar (1:25.000)
   b. Peta skala kecil, yaitu peta dengan skala 1:500.000 atau lebih kecil.
3. Macam peta ditinjau dari fungsinya.
   a. Peta umum (*general map*), merupakan peta yang berisi penampakan-penampakan umum, seperti jalan, bangunan, batas wilayah, garis pantai, elevasi, dan sebagainnya.
   b. Peta tematik, merupakan peta yang menunjukkan hubungan ruang dalam bentuk atribut tunggal atau hubungan atribut
   c. Kart, merupakan peta yang didesain untuk keperluan navigasi, nautical dan aeronautical. Peta kelautan yang ekuivalen dengan peta topografi disebut peta Batimetrik.
4. Macam peta ditinjau dari macam persoalan (maksud dan tujuan).

   Seperti peta kadaster, peta geologi, peta tanah, peta ekonomi, peta kependudukan, peta tanah guna tanah, dan sebagainya.

2.7.9.3. Proyeksi Peta

Proyeksi peta merupakan teknik-teknik yang digunakan untuk menggambarkan sebagaian atau keseluruhan permukaan tiga dimensi yang secara kasar berbentuk bola ke permukaan datar dua dimensi dengan distorsi seminimal

mungkin. Proyeksi-proyeksi yang sudah ada sekarang dapat dikelompokkan menurut bidang proyeksi yang digunakannya, yaitu :

a. Proyeksi azimuthal, menggunakan bidang datar sebagai bidang proyeksi (*planar*).
b. Proyeksi kerucut (*conical*), menggunakan kerucut (dapat didatarkan tanpa mengalami perubahan dan kerusakan) sebagai bidang proteksi.
c. Proyeksi silinder (*cylindrical*), menggunakan silinder (dapat didatarkan tanpa mengalami perubahan dan kerusakan) senagai bidang proyeksi.

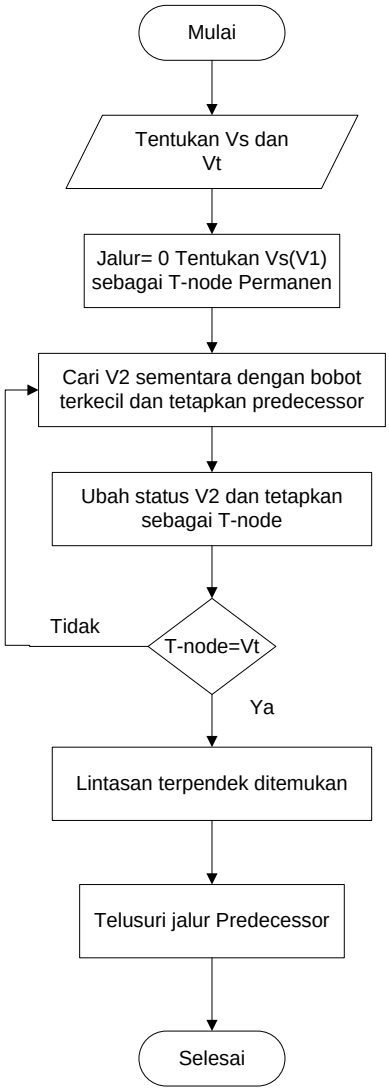
2.7.10. Algoritma Dijkstra

2.7.10.1. Sejarah Algoritma Dijkstra

Algoritma Dijkstra ditemukan oleh Edsger W. Dijkstra yang mrupakan salah satu varian bentuk algoritma popular dalam pemecahan persoalan yang terkait dengan masalah optimisasi dan bersifat sederhana. Algoritma ini menyelesaikan masalah dengan mencari sebuah lintasan terpendek (sebuah lintasan yang mempunyai panjang minimum) dari verteks a ke verteks z dalam graph berbobot, bobot tersebut adalah bilangan positif jadi tidak dapat dilalui oleh node negatif, namun jika terjadi demikian, maka penyelesaian yang diberikan adalah *infiniti*. Pada algoritma dijkstra node digunakan, karena algoritma dijkstra menggunakan diagram pohon (*tree*) untuk penentuan jalur lintasan terpendek dan menggunakan *graph* yang berarah.

2.7.10.2. Flowchart Algoritma Dijkstra

Berikut ini merupakan *flowchart* dari Algoritma Dijkstra dalam menentukan lintasan terpendek.



Gambar II.8. Flowchart Algoritma Dijkstra. (Sumber: Lubis, 2009).

2.7.10.3. Cara Kerja Algoritma Dijkstra

Algoritma ini mencari panjang lintasan terpendek dari verteks a ke verteks z dalam sebuah *graph* berbobot tersambung. Langkah-langkah dalam menentukan lintasan terpendek pada algoritma dijkstra (Lubis, 2009), yaitu :

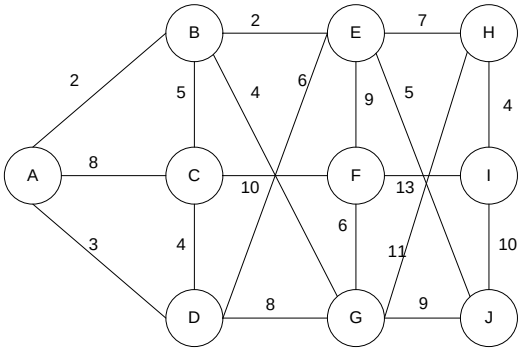
1. Bentuklah tabel yang terdiri dari node, status, bobot dan *predecessor*. Lengkapi kolom bobot yang diperoleh dari jarak node sumber ke semua node yang langsung terhubung dengan node sumber tersebut.
2. Tentukan node sumbernya dan tetapkan sebagai node terpilih, kemudian tentukan node yang langsung terhubung dengan node yang sudah terpilih.
3. Pilih node dengan bobot yang terendah dari node yang belum terpilih, diinisialisasikan dengan ‘0’ dan yang sudah terpilih diinisialisasikan dengan ‘1’.
4. Jika node dengan bobot terkecil ditemukan maka tetapkan sebagai node terpilih selanjutnya.
5. Tentukan node sementara yang terhubung pada node yang sudah terpilih sebelumnya dan merupakan bobot terkecil dilihat dari tabel dan tentukan sebagai node terpilih berikutnya.
6. Apakah node yang terpilih merupakan node tujuan?. Jika ya, maka kumpulkan node terpilih atau *predecessor* yang merupakan rangkaian yang menunjukkan lintasan terpendek.
7. Begitu seterusnya hingga mencapai node yang menjadi tujuan.

Algoritma dijkstra akan berhenti ketika node yang menjadi tujuan sudah ditemukan.

2.7.10.4. Implementasi *Graph* pada Algoritma Dijkstra

Pada Algoritma Dijkstra node digunakan karena algoritms dijkstra menggunakan diagram pohon untuk penentuan jalur lintasan terpendek dan

menggunakan *graph* yang berarah. Algoritma Dijkstra mencari jarak terpendek dari node asal ke node terdekat, kemudian ke node berikutnya, dan seterusnya. Berikut ini adalah contoh *graph* berarah dalam menencari jarak terpendek dari dari node asal ke node tujuan dengan melewati semua node yang ada (Lubis, 2009).



Gambar II.9. Graph untuk Algoritma Dijkstra.

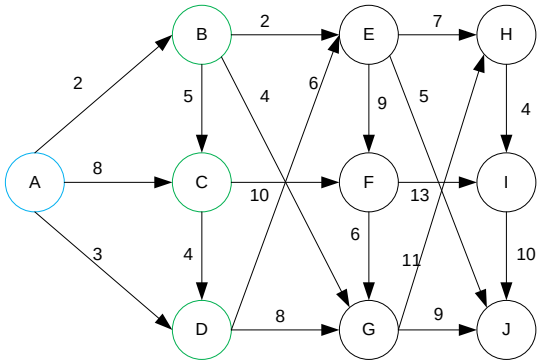
Dibawah ini adalah jarak beserta hubungan tetangga antara satu node dengan node lainnya.

Tabel II.1. Tabel Tetangga

	A	B	C	D	E	F	G	H	I	J
A	0	2	8	3	0	0	0	0	0	0
B	2	0	5	0	2	0	4	0	0	0
C	8	5	0	4	0	10	0	0	0	0
D	3	0	4	0	6	0	8	0	0	0
E	0	2	0	6	0	9	0	7	0	5
F	0	0	10	0	9	0	6	0	13	0
G	0	4	0	8	0	0	0	11	0	9
H	0	0	0	0	7	0	11	0	4	0
I	0	0	0	0	0	13	0	4	0	10
J	0	0	0	0	5	0	9	0	10	0

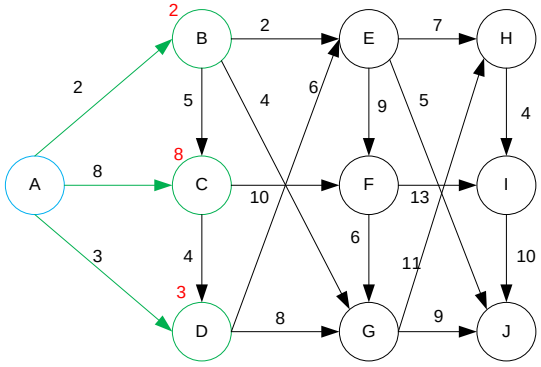
Adapun proses *graph* yang dilakukan dalam pencarian jalur terpendek dari A ke J dengan algoritma dijkstra sebagai berikut :

Pertama, tentukan node tetangga yang berhubungan dengan node awal atau node asal atau disebut juga dengan node sumber. Disini di inisialisasikan node awal tersebut dengan “A”.



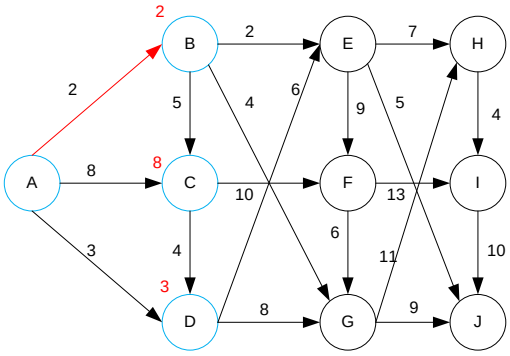
Gambar II.10. Proses Graph 1.

Dari gambar II.10, diketahui bahwa node tetangga yang terhubung dengan node A adalah B dengan jarak 2, C dengan jarak 8, dan D dengan jarak 3. Kemudian menentukan node yang terhubung dengan node A yang memiliki nilai bobot terkecil untuk dijadikan node selanjutnya,



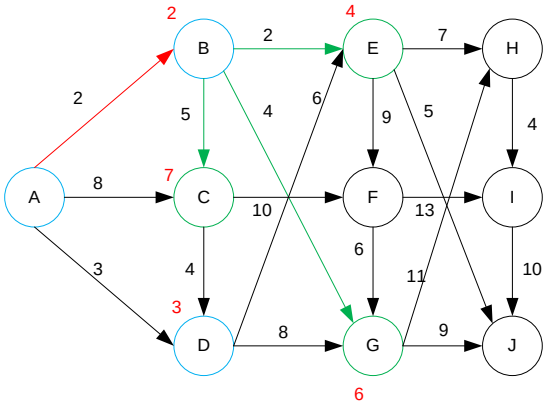
Gambar II.11. Proses Graph 2.

maka diperoleh node B sebagai node selanjutnya setelah node A, karena memiliki nilai bobot yang paling kecil yaitu 2 diantara bobot node C dan bobot node D.



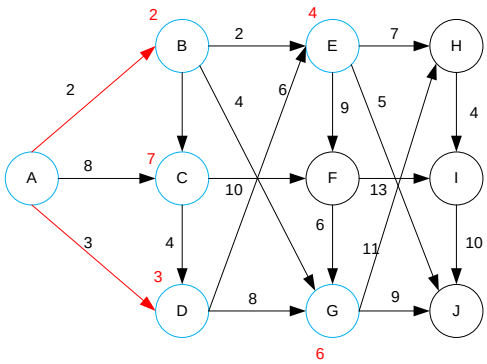
Gambar II.12. Proses Graph 3.

Dari node B ini, kita dapat melihat node tetangga yang terhubung dengan node B beserta bobot nilai yang dimiliki yaitu C dengan bobot nilai 7 (5+2), E dengan bobot 4 (2+2), dan G dengan bobot 6 (4+2).



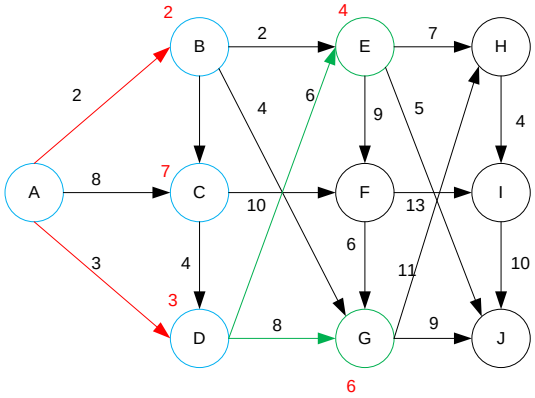
Gambar II.13. Proses Graph 4.

Kemudian, kita menentukan node selanjutnya dari node tetangga yang belum terpilih dan memiliki bobot nilai terkecil. Adapun node yang belum terpilih dari node yang sudah terpilih sebelumnya yaitu C, D, E, G. Untuk menentukan node selanjutnya, kita dapat memilih node belum terpilih tersebut yang memiliki bobot nilai terkecil, maka diperoleh node D dengan bobot 3.



Gambar II.14. Proses Graph 5.

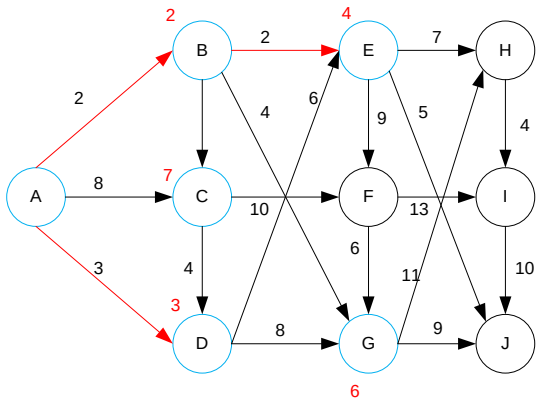
Dari node D ini, kita dapat melihat node tetangga yang terhubung dengan node D beserta bobot nilai yang dimiliki yaitu E dengan bobot nilai 4, E tetap memiliki bobot 4 karena untuk sampai ke node E dari node A lebih dekat dengan melewati node B dibandingkan harus melewati node D, lalu ada node G dengan bobot 6, G tetap memiliki bobot 6 karena untuk sampai ke node G dari node A lebih dekat dengan melewati node B dibandingkan harus melewati node D.



Gambar II.15. Proses Graph 6.

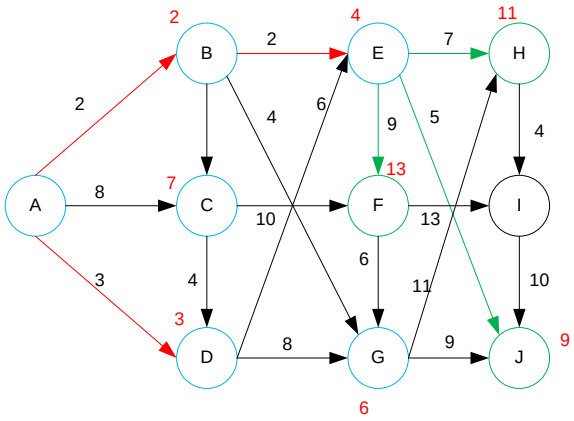
Kemudian, kita menentukan node selanjutnya dari node tetangga yang belum terpilih dan memiliki bobot nilai terkecil. Adapun node yang belum terpilih dari node yang sudah terpilih sebelumnya yaitu C, E, G. Untuk menentukan node

selanjutnya, kita dapat memilih node belum terpilih tersebut yang memiliki bobot nilai terkecil, maka diperoleh node E dengan bobot 4.



Gambar II.16. Proses Graph 7.

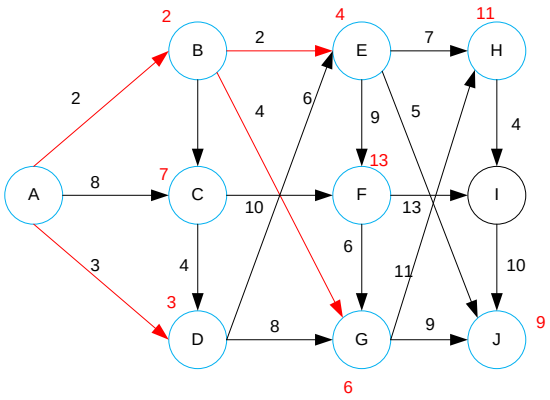
Dari node E ini, kita dapat melihat node tetangga yang terhubung dengan node E beserta bobot nilai yang dimiliki yaitu F dengan bobot nilai 13 (9+4), H dengan bobot 11 (7+4), dan J dengan bobot 9 (5+4)



Gambar II.17. Proses Graph 8.

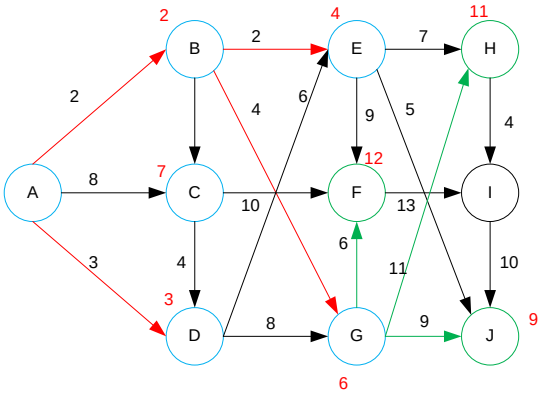
Kemudian, kita menentukan node selanjutnya dari node tetangga yang belum terpilih dan memiliki bobot nilai terkecil. Adapun node yang belum terpilih dari node yang sudah terpilih sebelumnya yaitu C, F, G, H, J. Untuk menentukan node

selanjutnya, kita dapat memilih node belum terpilih tersebut yang memiliki bobot nilai terkecil, maka diperoleh node G dengan bobot 6.



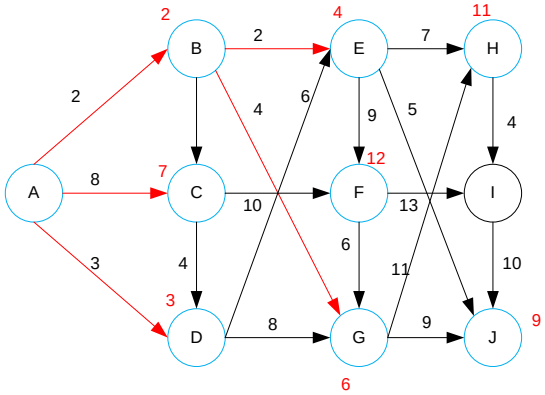
Gambar II.18. Proses Graph 9.

Dari node G ini, kita dapat melihat node tetangga yang terhubung dengan node G beserta bobot nilai yang dimiliki yaitu H dengan bobot nilai 11, H tetap memiliki bobot 11 karena untuk sampai ke node H dari node A lebih dekat dengan melewati node B dan node E dibandingkan harus melewati node B dan node G, lalu ada F denagn bobot nilai 12 (6+4), lalu ada J dengan bobot nilai 9, J tetap memiliki bobot 9 karena untuk sampai ke node J dari node A lebih dekat dengan melewati node B dan node E dibandingkan harus melewati node B dan node G.



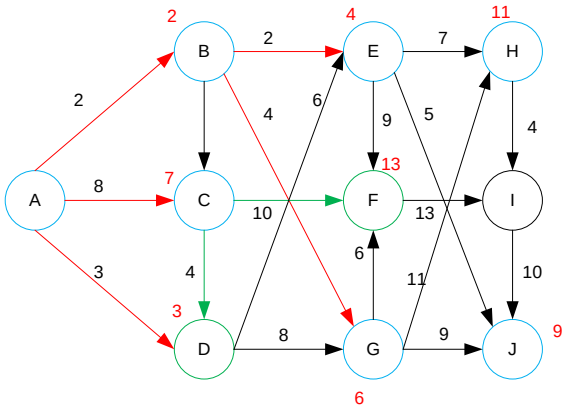
Gambar II.19. Proses Graph 10.

Kemudian, kita menentukan node selanjutnya dari node tetangga yang belum terpilih dan memiliki bobot nilai terkecil. Adapun node yang belum terpilih dari node yang sudah terpilih sebelumnya yaitu C, F, H, J. Untuk menentukan node selanjutnya, kita dapat memilih node belum terpilih tersebut yang memiliki bobot nilai terkecil, maka diperoleh node C dengan bobot 7.



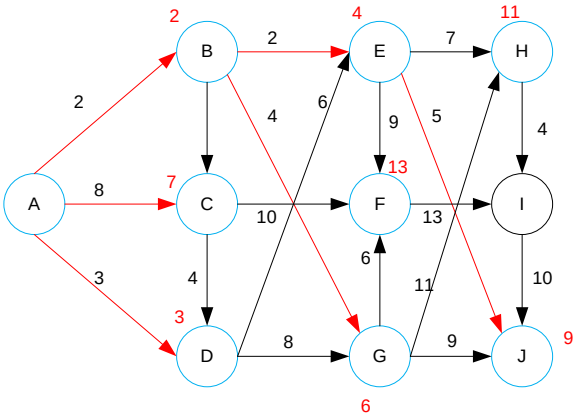
Gambar II.20. Proses Graph 11.

Dari node C ini, kita dapat melihat node tetangga yang terhubung dengan node C beserta bobot nilai yang dimiliki yaitu D dengan bobot nilai 3, D tetap memiliki bobot 3 karena untuk sampai ke node D lebih dekat dengan langsung dari node A ke node D dibandingkan harus melewati node B dan node C, lalu ada F dengan bobot nilai 12, F tetap memiliki bobot 12 karena untuk sampai ke node F dari node A lebih dekat dengan melewati node B dan node G dibandingkan harus melewati node C.



Gambar II.21. Proses Graph 12.

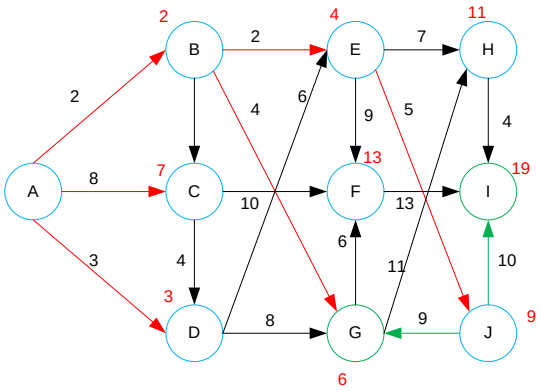
Kemudian, kita menentukan node selanjutnya dari node tetangga yang belum terpilih dan memiliki bobot nilai terkecil. Adapun node yang belum terpilih dari node yang sudah terpilih sebelumnya yaitu F, H, J. Untuk menentukan node selanjutnya, kita dapat memilih node belum terpilih tersebut yang memiliki bobot nilai terkecil, maka diperoleh node J dengan bobot 9.



Gambar II.22. Proses Graph 13.

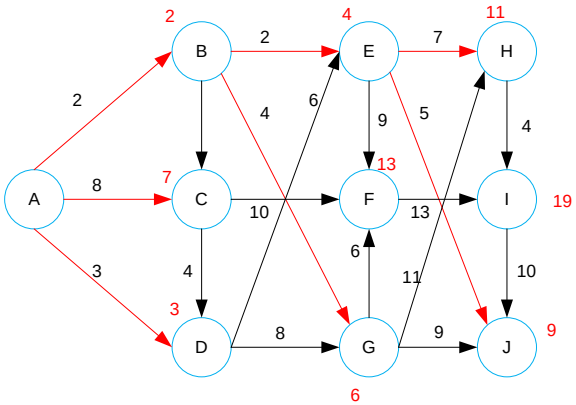
Dari node J ini, kita dapat melihat node tetangga yang terhubung dengan node J beserta bobot nilai yang dimiliki yaitu I dengan bobot nilai 19 (10+9), lalu ada G dengan bobot nilai 6, G tetap memiliki bobot 6 karena untuk sampai ke node G

dari node A lebih dekat dengan melewati node B dibandingkan harus melewati node J.



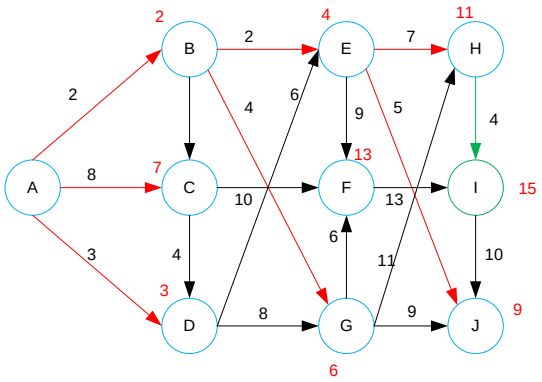
Gambar II.23. Proses Graph 14.

Kemudian, kita menentukan node selanjutnya dari node tetangga yang belum terpilih dan memiliki bobot nilai terkecil. Adapun node yang belum terpilih dari node yang sudah terpilih sebelumnya yaitu F, H, I. Untuk menentukan node selanjutnya, kita dapat memilih node belum terpilih tersebut yang memiliki bobot nilai terkecil, maka diperoleh node H dengan bobot 11.



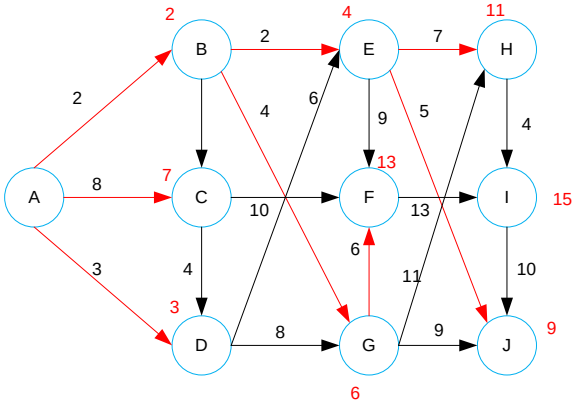
Gambar II.24. Proses Graph 15.

Dari node J ini, kita dapat melihat node tetangga yang terhubung dengan node H beserta bobot nilai yang dimiliki yaitu I dengan bobot nilai 15 (11+4).



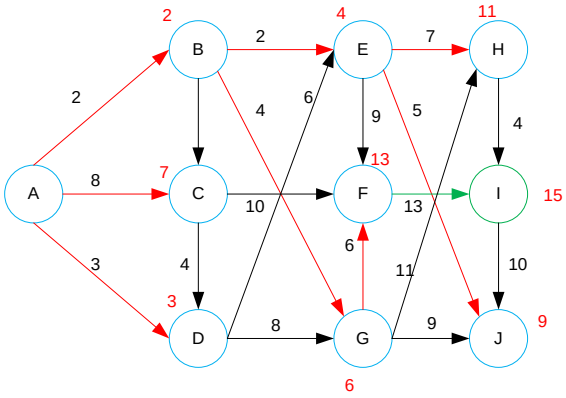
Gambar II.25. Proses Graph 16.

Kemudian, kita menentukan node selanjutnya dari node tetangga yang belum terpilih dan memiliki bobot nilai terkecil. Adapun node yang belum terpilih dari node yang sudah terpilih sebelumnya yaitu F dan I. Untuk menentukan node selanjutnya, kita dapat memilih node belum terpilih tersebut yang memiliki bobot nilai terkecil, maka diperoleh node F dengan bobot 13.



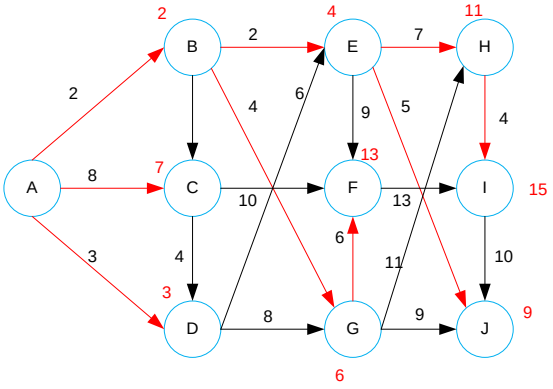
Gambar II.26. Proses Graph 17.

Dari node F ini, kita dapat melihat node tetangga yang terhubung dengan node F beserta bobot nilai yang dimiliki yaitu I dengan bobot nilai 15.

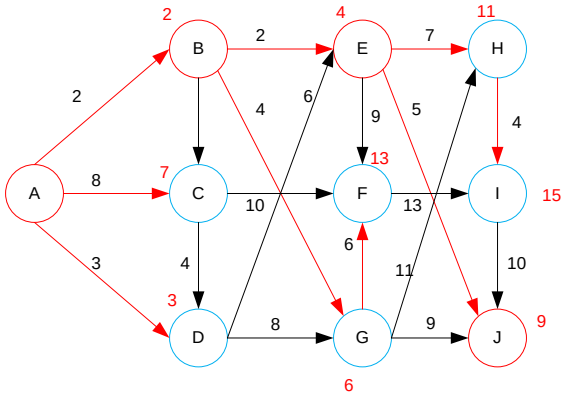


Gambar II.27. Proses Graph 18.

Selanjutnya hanya tinggal satu node tetangga lagi yang belum terpilih, maka I langsung menjadi node yang dipilih untuk node selanjutnya.



Gambar II.28. Proses Graph 19.



Gambar II.29. Proses akhir pencarian jalur terpendek.

Karena semua node sudah terpilih, maka pencarian node tetangga dihentikan. Dari gambar II.29, kita dapat mengetahui jalan mana yang dipilih dari A untuk sampai ke J, yaitu A – B – E – J, dengan total bobot nilainya adalah 9.

Selain dengan *graph* diatas, berikut ini adalah tabel iterasi berdasarkan *graph* diatas dalam menentukan jarak terpendek dari A ke J.

1. Pada awalnya status dari node yang belum terpilih diinisialisasikan dengan ‘0’ dan yang sudah terpilih diinisialisasikan dengan ‘1’ dimulai dari node A.
2. Tentukan bobot dari node yang berhubungan langsung dengan node sumber yaitu node A, seperti: dari node A ke node B=2, dari node A ke node C=8, dari node A ke node D=3, dan untuk node E,F,G,H,I,J diinisialisasikan dengan ‘-‘ karena tidak ada lintasan yang menghubungkan secara langsung.
3. *Predecessor* (node sumber) dari node A,B,C,D adalah A, karena jarak dihitung dari node A, sehingga node A disebut sebagai *predecessor* (node sumber), sedangkan untuk F,G,H,I,J diinisialisasikan dengan ‘-‘ dikarenakan tidak ada lintasan yang langsung menghubungkan dari node A, sehingga jaraknya tidak ada.

Tabel II.2. Iterasi Ke 1

Node	A	B	C	D	E	F	G	H	I	J
Status	1	0	0	0	0	0	0	0	0	0
Bobot	-	2	8	3	-	-	-	-	-	-
Predecessor	A	A	A	A	-	-	-	-	-	-

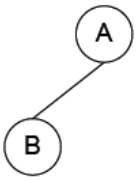


Gambar II.30. Node Terpilih Pada Iterasi 1

Selanjutnya pilih node yang memiliki bobot yang paling kecil dan status nya masih ‘0’, yaitu node B. Untuk itu status node B menjadi ‘1’ dan *predecessor*-nya masih tetap A dan node lain *Predecessornya*-nya masih sama. Jika node B sudah dipilih maka ada perubahan pada bobot node C, dimana awalnya bernialai 8 sekarang menjadi 7. Bobot 8 diperoleh dari node yang langsung bergerak dari A ke C, padahal terdapat jalur yang lebih pendek yaitu melalu B, dengan bobot 7, sehingga *predecessor* pada C menjadi B, karena node B sudah terpilih, selanjutnya diperolah node E, dimana untuk mencapai node E dan G dengan bobot 6, *predecessor* E dan G adalah B, di mana untuk mencapai node E dan G dari node A bisa melalui node B. Sehingga diperoleh :

Tabel II.3. Iterasi Ke 2

Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	0	0	0	0	0	0	0	0
Bobot	-	2	7	3	4	-	6	-	-	-
Predecessor	A	A	B	A	B	-	B	-	-	-

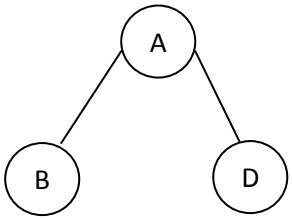


Gambar II.31. Node Terpilih Pada Iterasi 2

Dari data diatas didapatkan bahwa node D memiliki bobot yang paling kecil, sehingga akan berubah menjadi ‘1’ dan *predecessor*-nya masih tetap A. Sehingga diperolah :

Tabel II.4. Iterasi Ke 3

Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	0	1	0	0	0	0	0	0
Bobot	-	2	7	3	4	-	6	-	-	-
Predecessor	A	A	B	A	B	-	B	-	-	-

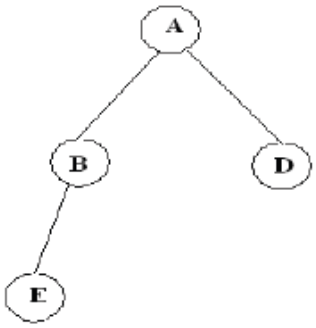


Gambar II.32. Node Terpilih Pada Iterasi 3.

Dari data diatas didapkan bahwa node E memiliki bobot yang paling kecil, sehingga statusnya beribah menjadi ‘1’. Jika node E sudah terpilih, maka node F mempunyai bobot 13, node H bobotnya 11 dan node J bobotnya 9. Untuk mencapai node F, H dan node J dari node A bisa melalui B, kemudian melalui node E dengan *predecessor*-nya berubah menjadi E. Sehingga diperoleh :

Tabel II.5. Iterasi Ke 4

Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	0	1	1	0	0	0	0	0
Bobot	-	2	7	3	4	13	6	11	-	9
Predecessor	A	A	B	A	B	E	B	E	-	E

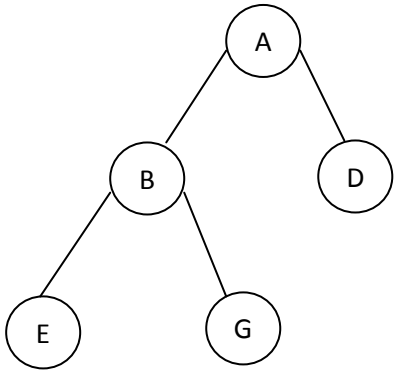


Gambar II.33. Node Terpilih Pada Iterasi 4.

Dari data tersebut didapatkan bahwa node G memiliki bobot yang paling kecil, sehingga statusnya akan berubah menjadi ‘1’, *predecessor*-nya masih tetap B. Jika node G sudah terpilih, maka ada perubahan bobot pada node F dengan bobot 13 berubah menjadi 12 dan *predecessor* E menjadi G. Sehingga diperoleh :

Tabel II.6. Iterasi Ke 5

Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	0	1	1	0	1	0	0	0
Bobot	-	2	7	3	4	12	6	11	-	9
Predecessor	A	A	B	A	B	G	B	E	-	E

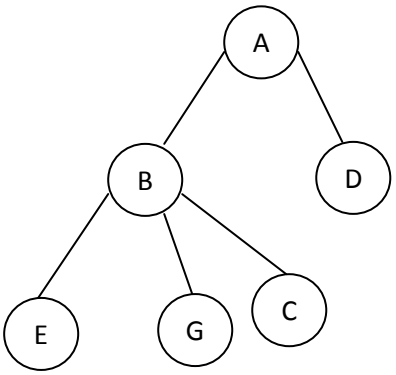


Gambar II.34. Node Terpilih Pada Iterasi 5.

Dari data tersebut didapkan bahwa node C memiliki bobot yang paling kecil, sehingga statusnya akan berubah menjadi ‘1’, *predecessor*-nya masih tetap B dengan bobot 7. Sehingga diperoleh :

Tabel II.7. Iterasi Ke 6

Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	1	1	1	0	1	0	0	0
Bobot	-	2	7	3	4	12	6	11	-	9
Predecessor	A	A	B	A	B	G	B	E	-	E

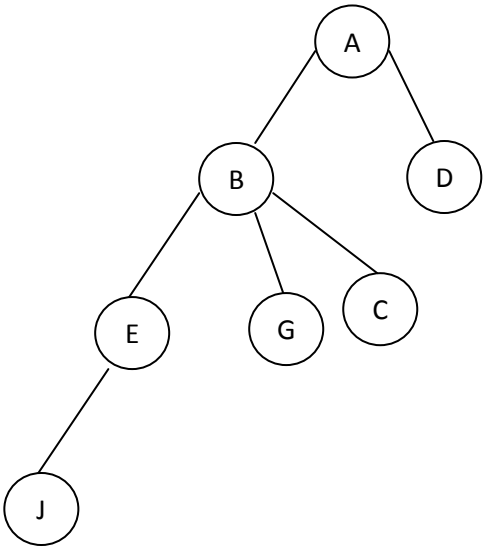


Gambar II.35. Node Terpilih Pada Iterasi 6.

Dari data diatas didapatkan bahwa node J memiliki bobot yang paling kecil, sehingga statusnya akan berubah menjadi ‘1’ dengan *predecessor*-nya E. Jika node J sudah terpilih, maka diperoleh node I dengan bobot 19 yang bersumber dari node A→ B → E →J →I. Sehingga diperoleh:

Tabel II.8. Iterasi Ke 7.

Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	1	1	1	0	1	0	0	1
Bobot	-	2	7	3	4	12	6	11	19	9
Predecessor	A	A	B	A	B	G	B	E	J	E

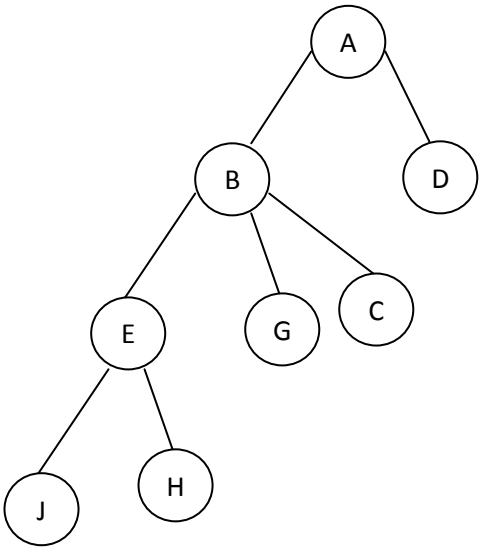


Gambar II.36. Node Terpilih Pada Iterasi 7.

Selanjutnya didapatkan bahwa node H memiliki bobot yang paling kecil, sehingga statusnya akan berubah menjadi ‘1’ dengan *predecessor*-nya E. Jika node J sudah terpilih, maka diperolah node I dengan bobot 19 berubah bobotnya menjadi 15 dengan *predecessor*-nya H yang bersumber dari node A→ B→ E→ H→ I. Sehingga diperoleh:

Tabel II.9. Iterasi Ke 8.

Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	1	1	1	0	1	1	0	1
Bobot	-	2	7	3	4	12	6	11	15	9
Predecessor	A	A	B	A	B	G	B	E	H	E

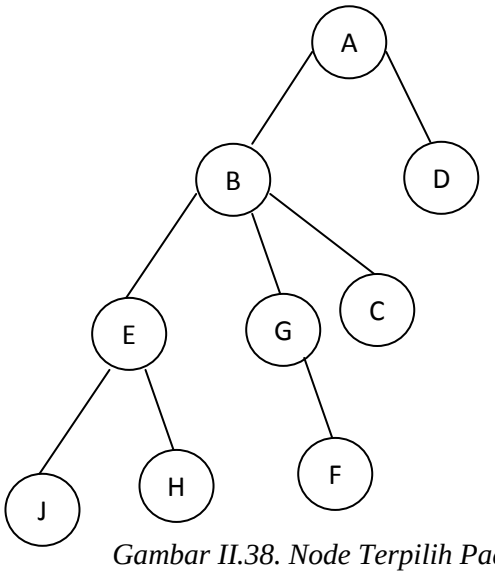


Gambar II.37. Node Terpilih Pada Iterasi 8.

Selanjutnya didapkan bahwa node F memiliki bobot yang paling kecil, sehingga statusnya akan berubah menjadi ‘1’, *predecessor*-nya masih tetap G dengan bobot 12. Sehingga diperoleh :

Tabel II.10. Iterasi Ke 9.

Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	1	1	1	1	1	1	0	1
Bobot	-	2	7	3	4	12	6	11	15	9
Predecessor	A	A	B	A	B	G	B	E	H	E

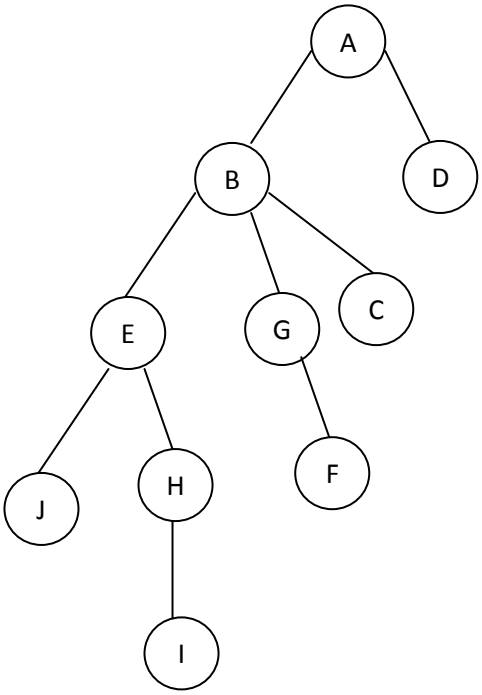


Gambar II.38. Node Terpilih Pada Iterasi 9.

Semua node telah terpilih dan hanya tinggal node I yang belum terpilih, selanjutnya status node I akan berubah menjadi ‘1’, *predecessor*-nya masih tetap H dengan bobot 15. Sehingga diperoleh:

Tabel II.11. Iterasi Ke 10.

Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	1	1	1	1	1	1	1	1
Bobot	-	2	7	3	4	12	6	11	15	9
Predecessor	A	A	B	A	B	G	B	E	H	E



Gambar II.39. Node Terpilih Pada Iterasi 10.

Hasil dari seluruh table adalah sebagai berikut :

Tabel II.12. Tabel Iterasi.

Iterasi 1										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	0	0	0	0	0	0	0	0	0
Bobot	-	2	8	3	-	-	-	-	-	-
Predecessor	A	A	A	A	-	-	-	-	-	-
Iterasi 2										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	0	0	0	0	0	0	0	0
Bobot	-	2	7	3	4	-	6	-	-	-
Predecessor	A	A	B	A	B	-	B	-	-	-
Iterasi 3										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	0	1	0	0	0	0	0	0
Bobot	-	2	7	3	4	-	6	-	-	-
Predecessor	A	A	B	A	B	-	B	-	-	-

bersambung,

Tabel II.12. Tabel Iterasi (Lanjutan).

Iterasi 4										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	0	1	1	0	0	0	0	0
Bobot	-	2	7	3	4	13	6	11	-	9
Predecessor	A	A	B	A	B	E	B	E	-	E
Iterasi 5										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	0	1	1	0	1	0	0	0
Bobot	-	2	7	3	4	12	6	11	-	9
Predecessor	A	A	B	A	B	G	B	E	-	E
Iterasi 6										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	1	1	1	0	1	0	0	0
Bobot	-	2	7	3	4	12	6	11	-	9
Predecessor	A	A	B	A	B	G	B	E	-	E
Iterasi 7										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	1	1	1	0	1	0	0	1
Bobot	-	2	7	3	4	12	6	11	19	9
Predecessor	A	A	B	A	B	G	B	E	J	E
Iterasi 8										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	1	1	1	0	1	1	0	1
Bobot	-	2	7	3	4	12	6	11	15	9
Predecessor	A	A	B	A	B	G	B	E	H	E
Iterasi 9										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	1	1	1	1	1	1	0	1
Bobot	-	2	7	3	4	12	6	11	15	9
Predecessor	A	A	B	A	B	G	B	E	H	E
Iterasi 10										
Node	A	B	C	D	E	F	G	H	I	J
Status	1	1	1	1	1	1	1	1	1	1
Bobot	-	2	7	3	4	12	6	11	15	9
Predecessor	A	A	B	A	B	G	B	E	H	E

Perhitungan akan berhenti karena semua node sudah terpilih. Sehingga akan menghasilkan jalur terpendek dari node A ke setiap node yang ada. Untuk melihat jalur mana yang terpilih dapat ditelusuri dari *predecessor*-nya, sehingga akan didapat :

A ⇨ B	: A – B	: 2
A ⇨ C	: A – B -	: 7
A ⇨ D	: A – D	: 3
A ⇨ E	: A – B – E	: 4
A ⇨ F	: A – B – G – F	: 12
A ⇨ G	: A – B – G	: 6
A ⇨ H	: A – B – E – H	: 11
A ⇨ I	: A – B – E – H – I	: 15
A ⇨ J	: A – B – E – J	: 9

2.7.11. Sistem Basis Data

2.7.11.1. Pengertian Sistem Basis Data

Sistem *database* didefinisikan sebagai suatu sistem yang terdiri atas kumpulan *file*/tabel yang saling berhubungan dan kumpulan program yang saling memungkinkan beberapa pemakai dan atau program lain untuk mengakses dan memanipulasi file/tabel tersebut (Fathansyah, 2002).

Database harus mempunya tiga fitur yang penting (Riyanto, Putra dan Indelarko, 2009), yaitu :

a. *Accessibility*

Mengacu pada kemampuan akses untuk menyimpan atau memperoleh kembali data dengan identitas tertentu.

b. *Generality*

Mengacu pada kemampuan dalam mengakses semua informasi untuk memperoleh kembali atau memodifikasi data.

c. *Flexibility*

Mengacu pada kemapuan dalam kemudahan penggunaan dan pengembangan *database*.

2.7.11.2. Komponen *Database*

Pada sistem *database* secara lengkap terdapat komponen-komponen utama, yaitu perangkat keras (*hardware*), sistem operasi (*operating system*), *database*, sistem pengolahan *database* (*Database Management System/DBMS*) dan pemakai (Riyanto, Putra dan Indelarko, 2009).

2.7.11.3. Model *Database* Relasional

Model *database* relasional menggunakan sekumpulan tabel berdimensi dua (biasa disebut relasi/tabel) yang merupakan tempat data disimpan. Masing-masing tabel terdiri dari baris dan kolom. Selain itu pada model *database* relasional terdapat istilah *key* (kunci), yaitu satu atau gabungan beberapa atribut yang dapat membedakan semua barisan data (*row*) dalam tabel secara unik (Riyanto, Putra dan Indelarko, 2009). Ada tiga macam *key* yang dapat diterapkan pada suatu tabel, yaitu :

a. *Superkey*, yaitu merupakan satu atau lebih atribut yang dapat membedakan setiap baris data dalam sebuah tabel secara unik.
b. *Candidat-key*, yaitu merupakan kumpulan atribut minimal yang dapat membedakan setiap baris data dalam sebuah tabel secara unik. Sebuah *candidat-key* tidak boleh berisi atribut atau kumpulan atribut yang telah menjadi *super-key*.
c. *Primary-key*, yaitu merupakan *candidat-key* yang unik yang digunakan sebagai acuan dan kunci utama.

2.7.11.4. Normalisasi Data

Normalisasi adalah proses langka-langkah yang fleksibel untuk menggantikan hubungan yang ditentukan oleh koleksi yang berurutan dimana hubungan mempunyai suatu struktur yang lebih reguler dan sederhana. Adapun tahap-tahap normalisasi (Riyanto, Putra dan Indelarko, 2009), yaitu :

1. Bentuk Normal Pertama (*First Normal Form/1NF*)

   Bentuk normal pertama mempunya ciri, yaitu data dibentuk dalam satu *record* dan nilai dari *field-field* yang menyatakan data tunggal dan bukan gabungan dari nilai-nilai. Tidak ada atribut bernilai ganda.

2. Bentuk Normal Kedua (*Second Normal Form/2NF*)

   Bentuk normal kedua mempunyai syarat, yaitu bentuk data memenuhi kriteria bentuk pertama. Atribut bukan kunci haruslah bergantung secara fungsional pada *primary key*, sehingga untuk membentuk normal kedua harus sudah ditentukan kunci-kunci *field*, kunci *field*

haruslah unik dan dapat mewakili atribut lain yang menjadi anggotanya.

3. Bentuk Normal Ketiga (*Third Normal Form/3NF*)

   Untuk menjadi bentuk ketiga maka relasi haruslah dalam bentuk normal kedua dan semua atribut bukan primer tidak mempunyai hubungan transitif. Dengan kata lain, setiap atribut bukan kunci haruslah bergantung hanya pada *primary key* dan *primary key* secara menyeluruh.

4. Bentuk Normal *Boyce-Codd* (*Boyce-Codd Normal Form/BCNF*)

   Boyce-Codd Normal Form mempunyai paksaan yang lebih kuat dari bentuk normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk normal pertama dan setiap atribut harus bergantung fungsi pada atribut *superkey*.

2.7.11.5. Tujuan *Database*

Tujuan awal dan utama dalam pengolahan data dalam *database* adalah agar dapat memperoleh atua menemukan kembali data yang dicari dengan mudah dan cepat. Hal yang sangat ditonjolkan dalam *database* adalah pengorganisasian data yang akan disimpan sesuai fungsi atau jenisnya.

2.7.12. Model Entity-Relationship

Pada Model Entity-Relationship, data yang ada diterjemahkan atau ditransformasikan dengan memanfaatkan sejumlah perangkat konseptual menjadi sebuah diagram data, yang umum disebut sebagai Diagram Entity-Relationship

(Diagram E-R). Ada dua komponen utama pembentuk Model Entity-Relationship, yaitu Entitas (*Entity*) dan Relasi (*Relation*) (Fathansyah, 2002).

2.7.12.1. Entitas (*Entity*) dan Himpunan Entitas (*Entitas Sets*)

Entitas merupakan individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Sekelompok entitas yang sejenis dan berada dalam lingkup yang sama membentuk sebuah Himpunan Entitas. Sederhananya, Entitas menunjuk pada individu suatu objek, sedangkan Himpunan Entitas menunjuk pada rumpun (*famili*) dari individu tersebut.

2.7.12.2. Atribut (*Attributes/Properties*)

Setiap entitas memiliki atribut yang mendeskripsikan karakteristik dari entitas tersebut. Atribut-atribut dapat dibedakan berdasarkan sejumlah pengelompokan. Ada atribut yang dijadikan sebagai *Key* dan yang lainnya disebut atribut deskriptif. Ada pula atribut yang tergolong atribut sederhana ataupun komposit, dan sebagainya.

2.7.12.2.1. *Key* dan Atribut Deskriptif

Key adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris (*row*) dalam tabel secara unik. Artinya, jika suatu atribut dijadikan sebagai *key*, maka tidak boleh ada dua atau lebih baris data dengan nilai yang sama untuk atribut tersebut.

2.7.12.2.2. Atribut Sederhana dan Atribut Komposit

Atribut Sederhana adalah atribut atomik yang tidak dapat dipilah lagi. Sementara, Atribut Komposit merupakan atribut yang masih dapat diuraikan lagi menjadi sub-sub atribut yang masing-masing memiliki makna.

2.7.12.2.3. Atribut Bernilai Tunggal dan Atribut Bernilai Banyak

Atribut bernilai tunggal ditujukan pada atribut-atribut yang memiliki paling banyak satu nilai untuk setiap baris data. Sedangkan, Atribut bernilai banyak ditujukan pada atribut-atribut yang dapat kita isi dengan lebih dari satu nilai, tetapi jenisnya sama.

2.7.12.2.4. Atribut Harus Bernilai dan Nilai Null

Ada sejumlah atribut pada sebuah tabel yang kita tetapkan harus berisi data, tidak boleh kosong. Atribut semacam itu disebut *Mandatory Attribute*. Adapula atribut-atribut lain pada suatu tabel yang nilainya boleh dikosongkan (*Non Mandatory*). Nilai Null digunakan untuk menyatakan/mengisi atribut-atribut demikian yang nilainya memang belum siap atau tidak ada.

2.7.12.2.5. Atribut Turunan

Atribut Turunan adalah atribut yang nilai-nilainya diperoleh dari pengolahan atau dapat diturunkan dari atribut atau tabel lain yang berhubungan.

2.7.12.3. Relasi (*Relationship*) dan Himpunan Relasi (*Relationship Sets*)

Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Kumpulan semua relasi diantara

entitas-entitas yang terdapat pada himpunan entitas tersebut membentuk Himpunan Relasi, yang pada umumnya lebih sering disingkat dengan istilah Relasi saja.

2.7.12.4. Kardinalitas/Derajat Relasi

Kardinalitas Relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dapat berupa :

a. Satu ke Satu (*One to One*)

   Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, dan begitu pula sebaliknya.



Gambar II.40. Relasi Satu ke Satu.

b. Satu ke Banyak (*One to Many*)

   Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setipa entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.



Gambar II.41. Relasi Satu ke Banyak.

c. Banyak ke Satu (*Many to One*)

Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B.



Gambar II.42. Relasi Banyak ke Satu.

d. Banyak ke Banyak (*Many to Many*)

Setian entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebaliknya, dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.



Gambar II.43. Relasi Banyak ke Banyak.

2.7.13. Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu system. Diagram konteks merupakan level tertinggi dari DFD (Data Flow Diagram) yang menggambarkan seluruh input ke system atau output dari system. Ia akan memberi gambaran tentang keseluruhan system. System dibatasi oleh *boundary* (dapat digambarkan dengan garis putus).

Dalam diagram konteks hanya ada satu proses dan tidak boleh ada store dalam diagram konteks (Al-Bahran, 2006).

2.7.14. Diagram Nol

Diagram nol adalah diagram yang menggambarkan proses dari data flow diagram. Diagram nol memberikan pandangan secara menyeluruh mengenai system yang ditangani, menunjukkan tentang fungsi-fungsi utama atau proses yang ada, aliaran data, dan eksternal entity. Pada level ini sudah dimungkinkan adanya atau digambarkannya data store yang digunakan. Untuk proses yang tidak dirinci lagi pada level selanjutnya, symbol '*' atai 'P' (*functional primitive*) dapat ditambahkan pada akhir nomor proses. Keseimbangan input dan output (*balancing*) antara diagram 0 dengan diagram konteks harus diperhatikan (Al-Bahran, 2006).

2.7.15. Diagram Rinci (Level Diagram)

Diagram rinci adalah diagram yang menguraikan proses apa yang ada dalam diagram 0 atau diagram level diatasnya (Al-Bahran, 2006).

2.7.16. Penomoran Level Pada DFD

Berikut ini adalah tabel penomoran yang digunakan untuk memberi nomor pada proses di dalam DFD (Al-Bahran, 2006).

Tabel II.13. Penomoran Level Pada DFD.

Nama Level	Nama Diagram	Nomor Process
0	Context	
1	Diagram 0	1.0, 2.0, 3.0, …
2	Diagram 1.0	1.1, 1.2, 1.3, …
2	Diagram 2.0	2.1, 2.2, 2.3, …
2	Diagram 3.0	3.1, 3.2, 3.3, …
3	Diagram 1.1	1.1.1, 1.1.2, …
3	Diagram 1.2	1.2.1, 1.2.2, …
3	Diagram 1.3	1.3.1, 1.3.2, …
dst		

Didalam satu level sebaiknya tidak terdapat lebih dari 7 buah proses dan maksimal 9 buah proses, bila terjadi maka harus dilakukan dekomposisi.

2.7.17. Elemen Dasar Dari Diagram Aliran Data

Pada Diagram Aliran Data (DAD) terdapat beberapa elemen dasar yang digunakan (Al-Bahran, 2006), yaitu :

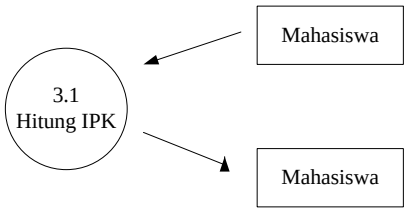
A. Kesatuan Luar (*External Entity*)

Sesuatu yang berada diluar system, tetapi ia memberikan data ke dalam system atau memberikan data dari system, disimbolkan dengan suatu kotak notasi. *Ekxternal entity* tidak termasuk bagian dari system. Bila sistem informasi dirancang untuk satu bagian (departemen) maka bagian lain masih terkait menjadi *eksternal entity*.

Pedoman pemberian nama kesatuan luar (*eksternal entity*), yaitu :

1. Nama terminal berupa kata benda.
2. Terminal tidak boleh memiliki nama yang sama kecuali memang objeknya sama (digambarkan dua kali, dimaksudkan untuk

membuat diagram lebih jelas). Bila demikian, maka terminal ini perlu diberi garis miring pojok kiri atas.



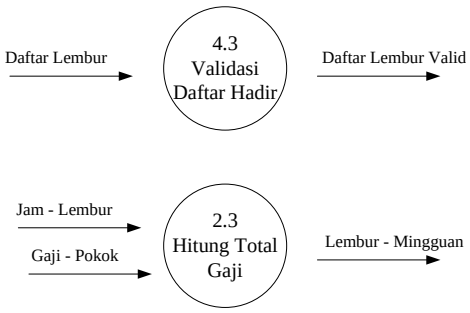
Gambar II.44. External Entity.

B. Arus Data (*Data Flow*)

Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari system. Arus data ditunjukkan dengan arah panah dan garis diberi nama atas arus data yang mengalir. Arus data ini mengalir di antara proses, data store dan menunjukkan arus data dari data yang berupa masukkan untuk sistem atau hasil proses sistem.

Pedoman pemberian nama aliran data, yaitu :

1. Nama aliran data yang terdiri dari beberapa aliran kata dihubungkan dengan garis sambung.
2. Tidak boleh ada aliran data yang namanya sama dan pemberian nama harus mencerminkan isinya.
3. Aliran data yang terdiri dari beberapa elemen dapat dinyatakan dengan grup elemen.
4. Hindari penggunaan kata ‘data’ dan ‘informasi’ untuk memberi nama pada aliran data.
5. Sedapat mungkin nama aliran data ditulis lengkap.



Gambar II.45. Arus Data.

Ketentuan lain :

1. Nama aliran data yang masuk kedalam suatu proses tidak boleh sama dengan nama aliran data yang keluar dari proses tersebut.
2. Data flow yang masuk ke atau keluar dari data store tidak perlu diberi nama bila :
   a. Aliran data sederhana dan mudah dipahami.
   b. Aliran data menggambarkan seluruh data item (satu *record* utuh).
3. Tidak boleh ada aliran data dari terminal ke data store atau sebaliknya, karena terminal bukan bagian dari sistem. Hubungan terminal dengan data store harus melalui proses.

C. Proses

Proses merupakan apa yang dikerjakan oleh sistem. Proses dapat mengelolah data atau aliran data masuk menjadi aliran data keluar. Proses berfungsi mentransformasikan satu atau beberapa data masukkan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan.

Pedoman pemberian nama proses, yairu :

1. Nama proses terdiri dari kata kerja dan kata benda yang mencerminkan fungsi proses tersebut, misalnya : Hitung Bonus, Pendataan Karyawan, Cetak Faktur, dll.
2. Jangan menggunakan kata proses sebagai bagian dari nama suatu proses.
3. Tidak boleh ada beberapa proses yang memiliki nama yang sama.
4. Proses harus diberi nomor. Urutan nomor sedapat mungkin mengikuti aliran atau urutan proses, namun demikian urutan nomor tidak berarti secara mutlak merupakan urutan proses secara kronologis.
5. Penomoran proses pada tingkat pertama (Diagram Nol) adalah 1.0, 2.0, 3.0, dst.
6. Penomoran proses pada tingkat kedua dari proses 1.0 (rincian dari proses 1.0) adalah 1.1, 1.2, 1.3, dst.
7. Context diagram tidak perlu diberi nomor.



Gambar II.46. Proses.

D. Simpanan Data (Data *Store*)

Simpanan data merupakan tempat penyimpanan data yang ada dalam sistem. Data Store dapat disimbolkan dengan sepasang dua garis sejajar atau dua garis dengan salah satu sisi samping terbuka. Proses dapat mengambil data dari atau memberikan data ke *database*.

Pedoman pemberian nama data *store*, yaitu :

1. Nama harus mencerminkan data store tersebut.
2. Bila namanya lebih dari satu kata maka harus diberi tanda sambung.

PELANGGAN	BARANG

Gambar II.47. Data Store.

2.7.18. Kamus Data

Kamus data berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasikan semua elemen data yang digunakan dalam sistem secara persis sehingga pemakai dan penganalisa sistem mempunyai dasar pengertian yang sama tentang masukkan, keluaran, penyimpanan, dan proses. Kamus data sering disebut juga dengan sistem data dictionary yaitu katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi.

Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir disistem dengan lengkap. Pada tahap analisis, kamus data digunakan sebagai alat komunikasi antara analisa sistem dengan pemakai sistem tentang data yang mengalir disistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. Kamus data memuat hal-hal sebagai berikut :

a. Nama arus data

   Nama arus data harus dicatat pada kamus data, sehingga mereka yang membaca DAD (Diagram Aliran Data) memerlukan penjelasan lebih

lanjut tentang suatu arus data tertentu dan dapat langsung mencarinya dengan mudah di kamus data.

b. Alias

Alias atau nama lain dari data dapat ditulis bila ada.

c. Bentuk data

Bentuk data perlu dicatat di kamus data, karena dapat dipergunakan untuk mengelompokkan kamus data ke dalam kegunaannya sewaktu perancangan sistem.

d. Arus data

Arus data menunjukkan dari mana data mengalir dan kemana data menuju. Keterangan arus data ini perlu dicatan di kamus data untuk memudahkan mencari arus data di DAD.

e. Penjelasan

Untuk memperjelas tentang makna dari arus data yang dicatat di kamus data, maka sebagian penjelasan dapat diisi dengan keterangan-keterangan tentang arus data tersebut.

2.7.19. WEB

Layanan WWW (*World-Wide Web*) atau biasa juga disebut Web merupakan jenis layanan yang paling popular dikalangan pengguna internet. WWW tidak hanya berfungsi sebagai media untuk mencari informasi, tetapi web sudah banyak digunakan secara komersial oleh hampir semua perusahaan-perusahaan diseluruh dunia untuk mengiklankan usaha mereka. Setiap dokumen

yang ditulis menggunakan suatu format standar yang disebut HTML (*HypertText Markup Language*).

2.7.20. HTML

HTML (*Hypertext Markup Language*) adalah bahasa program yang digunakan untuk menulis format dokumen yang dapat digunakan dalam *Web* (Sutisna, 2007). Dengan HTML, teks ASCII (file *.txt) dapat dirubah dengan kode-kode tertentu yang disebut tag untuk menjadi dokumen HTML, anda bisa menggunakan semua program text editor biasa, mulai dari Notepad hingga MS Word. HTML juga mendukung bagaimana suatu informasi ditampilkan. Sesuai dengan namanya, bahasa ini menggunakan *markup* untuk menandai perintah-perintahnya.

HTML terdiri atas elemen dan atribut dari elemen, dimana elemen dan atribut tersebut digunakan untuk mengatur tampilan *web*. Dokumen HTML yang paling sederhana adalah seperti berikut,

```
<HTML>
        <BODY>
                Isi dokumen HTML
        </BODY>
</HTML>
```

2.7.21. PHP

PHP (*HypertText Preprocessor*) adalah jenis *middleware open source*, sehingga dapat digunakan oleh siapa saja dengan cuma-cuma. Selain gratis, PHP memiliki berbagai kelebihan, diantaranya untuk membuat *website* yang dinamis. PHP akan dijalankan pada sisi server sehingga *client* tidak bisa melihat kode aslinya dan dengan PHP, halaman web dapat dibuat lebih interaktif.

2.7.22. MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi *user* serta menggunakan perintah standar SQL (*Structured Query Language*) (Nugroho, 2005). MySQL merupakan sebuah *database server* yang *free*, artinya kita bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya.

MySQL pertama kali dirintis oleh seorang programer database bernama Michael Widenius. Selain MySQL ada beberapa program *database server* lain yang menggunakan standar *query* berupa SQL, antara lain adalah :

1. Oracle
2. PostgreSQL
3. MySQL front
4. MsQL
5. SQL Server 97/..
6. Dll

MySQL adalah sebuah *database server*, dapat juga berperan sebagai *client* sehingga sering disebit *database client/server*, yang pen *source* dengan kemampuan dapat berjalan baik di OS (Operating Sistem) manapun, dengan *Platform* Windows maupun LINUX.

2.7.23. Dreamweaver

Dreamweaver merupakan software aplikasi yang digunakan sebagai HTML editor profesional untuk mendesain web secara visual. Aplikasi ini juga

dikenal dengan istilah WYSIWYG (*What You See Is What You Get*) yaitu bahwa anda tidak harus berusaha dengan tag-tag HTML untuk membuat sebuah situs. Selain itu, Dreamweaver juga memberikan keleluasaan kepada pengguna untuk menggunakannya sebagai media penulisan bahasa pemrograman web. Dreamweaver memiliki kemampuan untuk dapat berinteraksi dengan beberapa bahasa pemrograman seperti PHP, ASP, JavaScript, dan yang lainnya juga memberikan fasilitas maksimal kepada para desainer web yang menyertakan bahasa pemrograman web didalmnya.

2.7.24. Apache

Apache merupakan web server yang saat ini telah digunakan hampir dari 60% oleh server di dunia disebabkan karena sifat *software* ini sangat fleksibel dan dapat digunakan pada berbagai *platform* seperti Linux dan Windows

2.7.25. PhpMyAdmin

PhpMyAdmin merupakan *tools open source* yang dibuat menggunakan program PHP untuk mengakses *database* MySQL via web. Dengan menggunakan program ini kita dapat mengelolah *database* MySQL dengan sangat mudah.

Hampir semua operasi terhadap *database* dan perintah yang ada dapat dijalankan pada program tersebut, termasuk diantaranya menggunakan perintah *Database Definition Language* (DDL), *Database Manipulation Language* (DML) serta *Data Control Language* (DCL).

2.7.26. SVG Viewer

SVG (*Scalable Vector Graphic*) adalah salah satu standar pengkodean informasi geografis dalam sebuah bahasa XML yang didefinisikan oleh W3C dan telah menjadi bagian dari rekomendasi WMS (*Web Map Service*). SVG memberikan 3 tipe objek grafik, yaitu grafik vektor, gambar, dan text (Riyanto, 2010).

2.7.26.1. Konsep SVG

SVG memiliki beberapa konsep yang sering digunakan dalam membuat suatu objek, diantaranya adalah (Riyanto, 2010),

a. *Scalable*, yang berarti dapat diperbesar dan diperkecil secara seragam, tidak terbatas pada sebuah ukuran piksel tertentu.
b. *Vektor*, berisi objek geometris seperti garis-garis dan kurva. Hal ini memberikan fleksibilitas yang lebih tinggi dibandingkan dengan format gambar raster yang harus menyimpan informasi untuk tiap piksel dari grafik.
c. *Graphic*, pada SVG menyediakan sebuah deskripsi grafik vektor yang lengkap dan terstruktur, serta grafik vektor atau raster yang terolah, yang dapat dipakai secara *stand alone* atau sebagai *namespace* dari XML dengan tata bahasa lain.
d. XML, sebuah rekomendasi dari W3C untuk pertukaran informasi yang terstruktur. Dengan terbentuk pada spesifikasi yang sudah ada dan spesifikasinya diimplementasikan secara jelas, tata bahasa

berbasis XML sekarang terbuka untuk implementasi tanpa usaha rekayasa yang berat.

e. *Stylable*, keuntungan dari *style sheet* yaitu dalam hal kendali presentasional, fleksibilitas, cepatnya mengunduh dan peningkatan pengelolaan. SVG memberikan manipulasi berbasis skrip yang sama terhadap dokumen *tree* serta *style sheet*.

2.7.26.2. Komponen SVG

Terdapat beberapa komponen yang dimiliki oleh SVG, komponen-komponen tersebut diantaranya adalah (Riyanto, 2010),

1. Objek Grafik

   SVG menyediakan sebuah elemen path, yang dapat dipakai untuk membuat berbagai macam objek grafik dan juga menyediakan bentuk gambar dasar yang umum seperti kotak, elips, dan lain sebagainya.

2. Simbol

   SVG memberikan keleluasaan untuk membuat, menggunakan kembali dan melakukan *sharing* simbol tanpa membutuhkan sebuah registrasi terlebih dahulu. Komunitas pengguna dapat membuat dan memperbaiki simbol yang mereka butuhkan, tanpa harus meminta persetujuan komite.

3. Font

   Objek yang kaya grafik sering kali sangat bergantung pada *font* tertentu yang dipakai dan spasi yang tepat dari *glyph*. *Glyph* adalah sebuah bentuk perwakilan dari satu atau lebih karakter, tiap *glyph*

berisi atas beberapa *identifier* yang pendek (dalam beberapa kasus berupa *string* atau angka) bersama dengan struktur untuk me-*render glyph* tersebut.

4. Animasi

   SVG mengikutkan deklarasi elemen animasi yang dirancang secara kolaborasi antara SVG dan Group SYMM. Hal ini memungkinkan efek-efek animasi menjadi milik bersama dalam grafik *web* yang ada untuk diekspresikan ke dalam SVG.