BAB 1

PENGAMBILAN KEPUTUSAN

1.1 Pengertian Pengambilan Keputusan

Menurut Prajudi Atmosudirjo keputusan merupakan suatu pengakhiran daripada proses pemikiran tentang suatu masalah atau problema untuk menjawab pertanyaan apa yang harus diperbuat guna mengatasi masalah tersebut, dengan mejatuhkan pilihan pada suatu alternatif. Untuk mendapatkan pilihan tersebut perlu dilakukan suatu kegiatan mengumpulkan data dan informasi yang diperlukan serta menentukan metode pengambilan keputusan yang akan digunakan sebagai dasar untuk mengambil keputusan [**?**].

Menurut James A. F. Stoner pengambilan keputusan merupakan pemilihan di antara alternatif-alternatif antara lain :

1. Ada pilihan atas dasar logika atau pertimbangan.

2. Ada beberapa alternatif yang harus dan dipilih salah satu yang terbaik.

3. Ada tujuan yang dicapai dan keputusan itu makin mendeketkan pada tujuan tersebut.

By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

Pengambilan keputusan adalah proses membuat pilihan dengan mengidentifikasi keputusan, mengumpulkan informasi, dan menilai resolusi alternatif. Menggunakan proses pengambilan keputusan langkah demi langkah dapat membantu Anda membuat keputusan yang lebih disengaja dan bijaksana dengan mengorganisir informasi yang relevan dan menentukan alternatif. Pendekatan ini meningkatkan kemungkinan Anda akan memilih alternatif yang paling memuaskan.

Menurut *Oxford Advanced Learner's Dictionary* istilah pengambilan keputusan berarti proses memutuskan tentang sesuatu yang penting, terutama dalam sekelompok orang atau dalam suatu organisasi. Trewatha Newport mendefinisikan proses pengambilan keputusan melibatkan pemilihan tindakan dari antara dua atau lebih alternatif yang mungkin untuk sampai pada solusi untuk masalah yang diberikan.

1.2 Fungsi dan Tujuan Pengambilan Keputusan

1.2.1 Fungsi Pengambilan Keputusan

Fungsi pengambilan keputusan bekerja ketika pengambilan keputusan yang dilakukan seorang individu maupun organisasi dihadapkan pada alterntif-alternatif pilihan pemecah masalah yang dapat digunakan. Pengambilan keputusan sebagai suatu kelanjutan dari cara pemecahan masalah memiliki fungsi antara lain:

1. Awal dari semua aktivitas manusia yg sadar dan terarah, baik secara individual maupun secara kelompok, baik secara institusional maupun secara organisasional.

2. Suatu yang bersifat futuristik, artinya bersangkut paut dengan hari depan, masa yg akan datang, dimana efeknya atau pengaruhnya berlangsung cukup lama.

1.2.2 Tujuan Pengambilan Keputusan

Secara umum, tujuan pengambilan keputusan guna memperoleh pilihan terbaik diantara alternatif-alternatif yang ada. Kegiatan dalam organisasi ditujukan untuk mencapai tujuan organisasinya. Yang diinginkan dari tujuan tersbut adalah dapat berjalan lancar dan mencapai tujuan. Namun, kerap kali terjadi hambatan dalam pelaksanaannya, misalnya hanya manyakut satu masalah saja dan jika dipecahkan tidak menimbulkan akibat lain. Namun, ada kemungkinan terjadi masalah yang pemecahannya menghendakin dua hal kontadiksi terpecah sekaligus. Oleh karena itu, tujuan dalam pengambilan keputusan dapat dibedakan menjadi dua, yaitu :

1. Tujuan yang bersifat tunggal
   Tujuan pengambilan keputusan yang bersifat tunggal yaitu apabila terjadi dalam keputusan yang dihasilkan hanya menyangkut satu masalah saja, dengan artian bahwa sekali diputuskan maka tidak ada kaitannya dengan masalah lain.

2. Tujuan yang bersifat ganda
   Tujuan pengambilan keputusan yang bersifat ganda yaitu terjadi jika keputusan

yang dihasilkan itu menyangkut lebih dari satu masalah, yang artinya keputusan yang diambil itu sekaligus memecahkan dua (atau lebih) masalah yang sifatnya kontradiktif atau yang sifatnya tidak kontradiktif.

1.3 Dasar-Dasar Pengambilan Keputusan

Menurut George R. Terry, dasar-dasar pengambilan keputusan adalah sebagai berikut :

1. Intuisi
   Pengambilan keputusan yang berdasarkan intuisi atau perasaan bersifat subjektif, sehingga mudah terkena pengaruh.

2. Pengalaman
   Pengambilan keputusan berdasarkan pengalaman memiliki manfaat bagi pengetahuan praktis. Karena pengalaman seseorang dapat memperkirakan keadaan sesuatu, dapat memperhitungkauntung ruginya, baik buruknya keputusan yang akan dihasilkan.

3. Fakta
   Pengambilan keputusan berdasarkan fakta dapat memberikan keputusan yang sehat, solid, dan baik. Dengan fakta, maka tingkat kepercayaan terhadap pengambilan keputusan dapat lebih tinggi,sehingga orang dapat menerima keputusan-keputusan yang dibuat itu dengan rela dan lapang dada.

4. Wewenang
   Biasanya dilakukan oleh pimpinan terhadap bawahannya atau orang yang lebih tinggi kedudukannya kepada orang yang lebih rendah kedudukannya.

5. Rasional
   Keputusan yang dihasilkan lebih objektif, logis, lebih transparan, konsisten untuk memaksimumkan hasil atau nilai dalam batas kendala tertentu, sehingga dapat dikatakan mendekati kebenaran atau sesuai dengan apa yang diinginkan.

1.4 Macam-macam Pengambilan Keputusan

Berikut adalah macam-macam pengambilan keputusan :

1. Keputusan *Auto Generated*
   Pada Keputusan yang satu ini keputusannya diambil dengan cepat dan kurang memperhatikan, mempertimbangkan data, informasi, fakta, dan pada lapangan keputusan nya.

2. Keputusan *Induced*
   Dan Keputusan induced ini diambil dengan berdasarkan *scientific management*

atau manajemen ilmiah, yang sehingga keputusan itu logis, ideal, rasional untuk dilaksanakan dan resikonya relatif kecil, dalam proses pengambilan keputusan lebih lambat.

1.5 Proses Pengambilan Keputusan

Sebagaimana dibuktikan oleh definisi sebelumnya, proses pengambilan keputusan adalah urusan konsultatif yang dilakukan oleh sekelompok profesional untuk mendorong berfungsinya organisasi dengan lebih baik. Dengan demikian, ini adalah kegiatan yang berkelanjutan dan dinamis yang meliputi semua kegiatan lain yang berkaitan dengan organisasi. Oleh sebab itu, proses pengambilan keputusan memainkan peran penting dalam fungsi organisasi. Karena pikiran intelektual terlibat dalam proses pengambilan keputusan, itu membutuhkan pengetahuan ilmiah yang kuat ditambah dengan keterampilan dan pengalaman di samping kematangan mental.

Selanjutnya, proses pengambilan keputusan dapat dianggap sebagai sistem *check* and *balance* yang membuat organisasi tumbuh baik dalam arah vertikal dan linier. Ini berarti bahwa proses pengambilan keputusan mencari tujuan. Sasarannya adalah sasaran bisnis yang telah ditetapkan, misi perusahaan, dan visinya. Untuk mencapai tujuan-tujuan ini, perusahaan mungkin menghadapi banyak kendala dalam administrasi, operasional, sayap pemasaran dan domain operasional. Masalah-masalah seperti itu diselesaikan melalui proses pengambilan keputusan yang komprehensif. Tidak ada keputusan datang sebagai tujuan itu sendiri, karena di dapat berkembang masalah baru untuk dipecahkan. Ketika satu masalah diselesaikan, yang lain muncul dan seterusnya, sehingga proses pengambilan keputusan, seperti yang dikatakan sebelumnya, bersifat kontinu dan dinamis.

Proses pengambilan keputusan dapat dipandang sebagai suatu sistem. Komponen sistem terdiri dari masukan, proses, dan keluaran[**?**].

- Masukan (*input*)
  Masukan dalam proses pengambilan keputusan adalah data dan informasi. Data dapat berupa suatu keadaan, gambar, suara, huruf, angka atau bahasa yang dapat digunakan sebagai bahan untuk melihat lingkungan, objek, kejadian ataupun suatu konsep. Data ini masih memerlukan pengolahan data agar menjadi informasi yang lebih berdaya guna dan hasil pengolahan data dinamakan informasi. Pada pengambilan keputusan, informasi dapat dijadikan sebagai masukan dalam proses pengambilan keputusan. Informasi merupakan data yang telah diolah, dapat disajikan dalam bentuk laporan bulanan, laporan triwulan, laporan tahunan ataupun laporan rekapitulasi. Semakin lengkap data dan informasi yang disajikan, semakin cepat proses pengambilan keputusan dapat diambil. Namun, terkadang informasi yang disajikan tidak cukup banyak, dan terkadang terkendala waktu dan biaya untuk mengumpulkan informasi tersebut.

- Proses
  Proses pengambilan keputusan merupakan langkah-langkah yang diambil oleh

seorang pengambil keputusan untuk mendapatkan keputusan terbaik. Seorang pengambil keputusan tentu saja akan memikirkan secara seksama dan detail tentang keputusan yang akan diambil. Proses ini biasanya ada di dalam pemiliran, sehingga seorang bawahan terkadang melihat keputusan yang diambil seperti ”tiba-tiba”, walaupun pasti ada proses pemikiran yang dilakukan. Disamping itu, terkadang ada pemikiran yang dilakukan juga terdapat unsur subjektifitas sehingga keputusan yang diambil juga tidak objektif. Untuk mengatasi hal tersebut, maka dibangunlah sistem pendukung keputusan. Sitem ini befungsi untuk memberikan dukungan kepada pengambil keputusan bukan untuk menggantikannya.

- Keluaran (*output*)
  Keluaran dari proses pengambilan keputusan adalah keputusan yang dipilih oleh seorang pengambil keputusan, dimana keputusan ini tentunya merupakan keputusan terbaik. Seorang pengambil keputusan akan memerlukan kecakapan dan penguasaan materi yang baik untuk dapat mengambil keputusan terbaik. Pada sistem pendukung keputusan, keluaran dari proses pengambilan keputusan adalah informasi tentang alternati-alternatif atau pilihan-pilihan terbaik dari berbagai alternatif yang ditawarkan, namun keputusan tetap diambil oleh seorang pengambil keputusan.

Kebutuhan akan data dan informasi untuk mengkasilkan kebutuhan yang berkualitas memerlukan dukungan data dan informasi. Perkembangan teknologi komputer pada masa kini mengakibatkan teknologi menjadi bagian penting bagi dunia bisnis dan berbagai bidang lain. Berbagai dukungan teknologi komputer bagi sistem pendukung manajemen dikembangkan dalam bentuk:

1. *Decission Support System* (DSS)
2. *Executive Information System* (EIS)
3. *Expert Systems* (ES)
4. *Group Decision Support System* (GDSS)
5. *Artificial Neural Network* (ANN)
6. *Hybrid Support System* (HSS)

1.6 Teknik Pengambilan Keputusan

Terdapat lima teknik dalam pengambilan keputusan, yaitu :

1. *Operation Research*
   merupakan dengan menggunakan suatu metode-metode scientific (yang terdiri dari teknik-teknik matematis) dalam analisis dan pemecahan suatu maslah tertentu, penerapan dalam teknik ini yaitu usaha inventarisasi.

2. *Linear Programming*
   merupakan dengan memakai rumus-rumus matematik yang disebut juga dengan vector analysis.

3. *Gaming War Games*
   merupakan dengan teori yang biasa dipakai dalam menentukan strategi.

4. *Probability*
   merupakan dengan sebuah teori kemungkinan yang bisa diterapkan pada kalkulasi rasionalitas hal-hal yang tidak normal, dalam mengenai sebuah keputusan yang dipertimbangkan dan diperhitungkan.

5. *Rangking and statistical weighting*
   Yaitu dengan cara:

   (a) Menempatkan berbagai faktor yang akan mempengaruhi suatu keputusan akhir

   (b) menimbang suatu faktor-faktor yang bisa dibandingkan dan yang tercakup didalam setiap alternatif.

1.7 Klasifikasi Pengambilan Keputusan

Secara garis besar pengambilan keputusan dapat diklasifikasi menjadi dua model, yaitu :

1. Pengambilan keputusan banyak atribut (*Multi Attribute Decision Making*)
   Model pengambilan keputusan ini digunakan untuk menentukan alternatif terbaik terhadap beberapa alternatif yang ada berdasarkan pada kriteria tertentu. Kriteria biasanya berupa ukuran-ukuran, aturan-aturan atau standar yang digunakan dalam pengambilan keputusan.

2. Pengambilan keputusan banyak tujuan (*Multi Objective Decision Making*)
   Model ini tidak tidak memilih alternatif terbaik, namun menentukan jumlah produksi untuk masing-masing alternatif yang memaksimalkan sejumlah tujuan yang ada.

1.8 Tahapan Pengambilan Keputusan

Pengambilan keputusan merupakan proses pemilihan alternatif tindakan untuk mencapai tujuan atau sasaran tertentu. Pengambilan keputusan dilakukan dengan pendekatan sistematis terhadap permasalahan melalui proses pengumpulan data menjadi informasi serta ditambah dengan faktor faktor yang perlu dipertimbangkan dalam pengambilan keputusan. Keputusan diambil berdasarkan pada keadaan lingkungan atau kondisi yang ada, seperti kondisi pasti, kondisi beresiko, kondisi tikak pasti,

dan kondisi konflik. Secara umum, tahapan pengambilan keputusan adalah sebagai berikut :

1. Mengidentifikasi masalah
   Suatu organisasi ketika menghadapi suatu permasalahan perlu mengetahui apakah itu masalah (*problem*) atau hanya sekedar isu *issue* belaka. Yang dimaksud masalah adalah persoalan yang harus dipecahkan atau diselesaikan, sedangkan isu adalah persoalan yang perlu dibicarakan (tidak harus diselesaikan).

2. Menganalisis masalah
   Untuk mengetahui timbulnya masalah, lebih dulu mengetahui data dan informasinya. Pada tahapan ini, fungsi unit pengolahan data sangat penting sebab memungkinkan terdapat informasi yang tidak dapat dipertanggungjawabkan.

3. Membuat beberapa alternatif pemecahan masalah
   Dalam tahap ini, perlu diketahui penyebab tibulnya masalah. Lalu, dibuatlah beberapa alternatif penyelesaian masalah. Dalam pembuatan alternatif, masing-masing alternatif perlu ditunjukkan kelebihan dan kekurangannya.

4. Penilaian dan pemilihan alternatif
   Setelah berbagai alternatif diidentifikasikan, kemudian dilakukan evaluasi terhadap masing-masing alternatif dan dipilih sebuah alternatif yang berbaik.

5. Melaksanakan keputusan
   Ketika alternatif terbaik sudah dipilih, keputusan tersebut kemudian diterapkan. Namun, sering kali keputusan mengalami kegagalan karena tidak diterapkan dengan benar.

6. Evaluasi dan pengendalian
   Mekanisme sistem evaluasi dan pengendalian perlu dilakukan agar tujuan yang diharapkan dari sebuah keputusan dapat terealisasi.

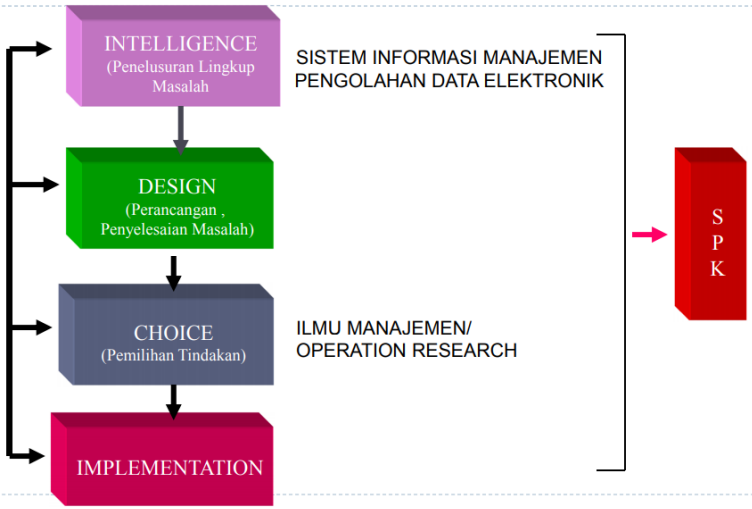
Menurut Herbert A. Simon, ada 4 tahap proses pengambilan keputusan:

1. Tahap Penelusuran (*Intelligence*)
   Tahap pendefinisian masalah dan identifikasi informasi yang berkaitan dengan persoalan dan keputusan yg akan diambil.

2. Tahap Perancangan (*Design*)
   Tahap analisa dalam mencari alternatif-alternatif pemecahan masalah.

3. Tahap Pemilihan (*Choice*)
   Manajemen memilih alternatif solusi yang diperkirakan paling sesuai.

4. Implementasi
   Tahap pelaksanaan dari keputusan yang telah diambil. Menyusun serangkaian tindakan yg terencana.



Gambar 1.1 Proses Pengambilan Keputusan Menurut Simon

1.9 Model Pemecahan Masalah

Dalam tahapan pengambilan keputusan, memilih model pemecahan masalah yang cocok untuk menyelesaikan masalah yang ada adalah hal yang penting. Model pemecahan masalah merupakan representasi dari realita yang ada di lapangan yang memiliki tujuan untuk penyederhanaan dalam pengambilan keputusan. Terdapat 4 model umun pengambilan keputusan, yaitu :

1. Model *Iconic* (*Scale*)
   penyederhanaan dari model abstrak; replika fisik dari sebuah sistem, biasanya berdasarkan perbedaan skala dibandingkan aslinya.

2. Model Analog
   berlawanan dengan model iconic, tidak mirip dengan sistem yang riel tetapi mempunyai perilaku yang mirip.

3. Model *Mathematical* (*Quantitative*)
   hubungan yang kompleks dari banyak sistem umumnya tidak dapat sepenuhnya terwakili. Untuk dapat mengabstraksikannya diperlukan pemanfaatan model-model matematis. (Probability, B/C ratio, others)

4. Model Mental
   memberikan gambaran subyektif bagaimana seseorang memikirkan tentang suatu situasi.

BAB 2

SISTEM PENDUKUNG KEPUTUSAN

2.1 Sejarah Perkembanan Sistem Pendukung Keputusan

Sejarah evolusi sistem pendukung keputusandimulai pada tahun 1965, yang dibutuhkan oleh industri untuk menyimpan data dan menggabungkan ide, orang, sistem dan teknologi. Pada masa itu dimulai pembangunan *main frame IBM System* 360 untuk mendukung terciptanya *Management Information System (MIS)* yang menitik beratkan pada fasilitas kepada manajer dalam bentuk laporan yang terstruktur dan periodik seperti laporan keuangandan laporan transaksi. Sebuah penelitian tentang implementasi sistem pendukung keputusan *model driven decision support system* dipublikasikan dalam jurnal bisnis pada tahun 1970-an. Penelitian ini menjadi pionir bagaimana komputer dan model analisis dapat membantu manajer sebagai pengambil keputusan. Pengguna model dalam sistem pendukung keputusan terus dikembangkan pada tahun 1980-an dan diikuti oleh perkembangan *knowledge oriented Decision Support Systems(DSS)*. Awal 1990-an, sistem pendukung keputusan dibangun menggunakan teknologi basis data relasional. Sejak dikenalkannya teknologi *web* sistem pendukung keputusan ini berkembang menjadi *Web-based DSS* [**?**].

By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

2.2 Definisi Sistem Pendukung Keputusan

Dengan kemajuan teknologi komputer yang semakin meningkat, lahirlah cara-cara baru dan sarana pengambilan keputusan berbantuan komputer. Sebagai akibatnya, seiring berlalunya waktu, berbagai definisi DSS muncul:

- Little (1970) mendefinisikan DSS sebagai serangkaian prosedur berbasis model untuk memproses data dan penilaian untuk membantu seorang manajer dalam pengambilan keputusannya.

- Keen dan Scott Morton (1978), menyatakan bahwa Sistem Pendukung Keputusan menggabungkan sumber daya intelektual individu dengan kemampuan komputer untuk meningkatkan kualitas keputusan. Ini adalah sistem pendukung berbasis komputer untuk pengambil keputusan manajemen yang menangani masalah semi-terstruktur. ”

- Mann dan Watson (1984) menyatakan bahwa ”sistem pendukung keputusan adalah sistem interaktif yang memberikan pengguna akses mudah ke model keputusan dan data untuk mendukung tugas pengambilan keputusan yang semi-terstruktur dan tidak terstruktur.”

- Bidgoli (1989) mendefinisikan DSS sebagai sistem informasi berbasis komputer yang terdiri dari perangkat keras / lunak dan elemen manusia yang dirancang untuk membantu pembuat keputusan di tingkat mana pun. Namun, penekanannya adalah pada tugas-tugas semi-terstruktur dan tidak terstruktur. ”

- Sprague dan Watson (1996) mendefinisikan DSS sebagai sistem berbasis komputer yang membantu pengambil keputusan menghadapi masalah yang tidak terstruktur melalui interaksi langsung dengan data dan model analisis.

- Sauter (1997) mencatat bahwa DSS adalah sistem berbasis komputer yang menyatukan informasi dari berbagai sumber, membantu dalam organisasi dan menganalisis informasi dan memfasilitasi evaluasi asumsi yang mendasari penggunaan model tertentu.

- Turban, Rainer, dan Potter (2005) secara luas mendefinisikan DSS sebagai ”sistem informasi berbasis komputer yang menggabungkan model dan data dalam upaya untuk memecahkan masalah semi-terstruktur dan beberapa tidak terstruktur dengan keterlibatan pengguna yang luas.”

Berdasarkan pengertian yang telah dikemukakan oleh beberapa ahli, maka dapat disimpulkan bahwa Sistem pendukung keputusan (DSS) adalah program terkomputerisasi yang digunakan untuk mendukung penentuan, penilaian, dan tindakan yang diambil dalam suatu organisasi atau bisnis. A DSS menyaring dan menganalisis sejumlah besar data, mengumpulkan informasi komprehensif yang dapat digunakan untuk memecahkan masalah dan dalam pengambilan keputusan.

2.3 Tujuan Sistem Pendukung Keputusan

Tujuan implementasi sistem pendukung keputusan antara lain :

1. Sistem pendukung keputusan berbasis komputer dapat memungkinkan mengambil keputusn dengan waktu yang cepat karena dengan dukungan sistem yang dapat memproses data secara cepat dengan jumlah banyak.

2. Sistem pendukung keputusan ditujukan untuk membantu dalam mengambil keputusan, bukan untuk manggantikan tugas seorang pengambil keputusan. Sehingga dengan data dan informasi yang yang akurat diharapkan dapat membantu membuat keputusan yang berkualitas.

3. Mampu menghasilkan keputusan yang efektif (sesuai tujuan) dan efisien.

4. Meningkatkan kemampuan untuk mendeteksi adanya kesalahan pada suatu sistem sehingga dapat dilakukan antisipasi.

2.4 Karakteristik Sistem Pendukung Keputusan

Karakteristika dari sistem pendukung keputusan, yaitu[**?**] :

1. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan.

2. Aadanya *interface* manusia atau mesin dimana manusia *(user)* tetap memegang kontrol proses pengambilan keputusan.

3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur serta mendukung beberapa keputusan yang saling berinteraksi.

4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan.

5. Memiliki subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.

6. Memiliki dua komponen utama, yaitu data dan model.

Menurut Turban karakteristik yang membedakan adalah :

1. Sistem pendukung keputusan dirancang untuk membantu mengambil keputusan dalam memecahkan masalah yang sifatnya semiterstruktur maupun tidak terstruktur.

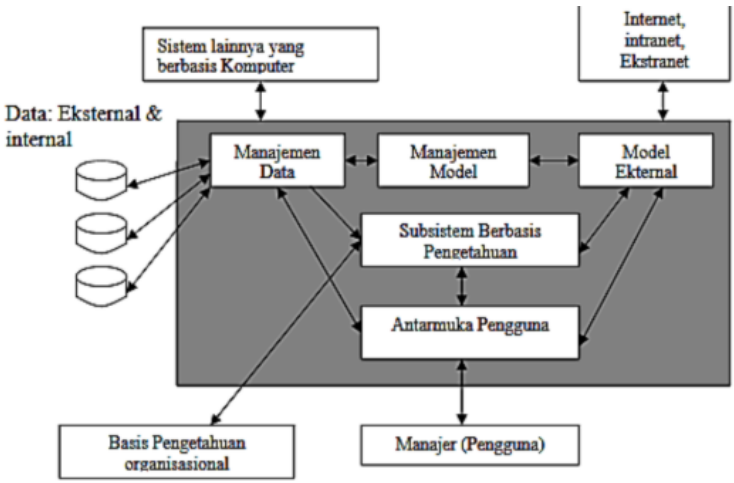
2. Dalam proses pengolahannya, sistem pendukung keputusan mengkombinasikan penggunaan model-model atau teknis-teknis analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari informasi.

3. Sistem pendukung keputusan, dirancang sedemikian rupa sehingga dapat digunakan dan dioperasikan dengan mudah oleh orang-orang yang memiliki dasar pengoperasian komputer tingkat tinggi. Oleh karena itu, pendekatan yang dilakukan biasanya model interaktif.

4. Sistem pendukung keputusan dirancang dengan menekankan pada aspek fleksibilitas serta kemampuan adaptasi yang tinggi.

2.5 Komponen Sistem Pendukung Keputusan

Sistem pendukung keputusan dibangun oleh lima komponen yaitu subsistem manajemen data, subsistem manajemen model, subsistem antar muka pengguna, dan subsistem manajemen berbasis pengetahuan. Berdasarkan definisi, sistem pendukung keputusan harus mencakup tiga komponen utama yaitu subsistem manajemen model, subsistem manajemen model, dan antarmuka pengguna.



Gambar 2.1 Komponen Sistem Pendukung Keputusan

Subsistem yang membangun sistem pendukung keputusan pada 6.12 dijelaskan sebagai berikut :

1. Manajemen Data
   Subsistem manajemen data berisi data yang relevan untuk suatu situasi dan dikelola oleh *Database Management System (DBMS)*. Subsisem ini dapat diinterkoneksikan dengan data *warehouse* perusahaan yang relevan untuk pengambilan keputusan.

2. Manajemen Model
   Subsistem manajemen model merupakan paket perangkat lunak yang menyimpan model keuangan, statistik, ilmu manajemen, atau model kuantitatif lainnya

yang memberikan kemampuan analitik yang tepat. Perangkat lunak ini sering disebut *Model Based Management System*(MBMS) dan dapat diimplementasikan pada sistem pengembangan *web* untuk berjalan pada *server* aplikasi.

3. Antar Muka Pengguna
   Subsistem antarmuka pengguna merupakan dukungan komunikasi antara sistem dengan pengguna. *Web browse* rmenjadi salah satu antar muka yang menampilkan dalam bentuk grafis dan interaktif dengan pengguna.

4. Manajemen Berbasis Pengetahuan
   Subsistem manajemen berbasis pengetahuan bertindak sebagai komponen independen yang memberikan kemampuan intelegensi untuk memperbesar pengetahuan pengambil keputusan. Perusahaan memiliki sistem manajemen pengetahuan (*knowledge management*). Keterhubungan subsistem ini dengan sistem pendukung keputusan dapat melalui inter koneksi dengan *web server*. Sistem pendukung keputusan yang melibatkan manajemen pengetahuan merupan sistem sistem pendukung keputusan yang cerdas (*Intelligent Decision Support System*) atau *Knowledge Based Decision Support System* (KB-DSS).

2.6 Tahapan Pengembangan Sistem Pendukung Keputusan

Proses pengembangan sistem pendukung keputusan meliputi enam tahapan, yaitu:

1. Tahap pra desain
   Pada tahap pra desain terdapat tiga subtahapan, yaitu :

   (a) Perencanaan. Ditentukannya kebutuhan sistem, diagnosa masalah, dan menentukan tujuan pengembangan sistem pendukung keputusan.

   (b) Penelitian. Melihat kebutuhan pengguna dan sumber daya yang tersedia dalam lingkungan sistem pendukung keputusan.

   (c) Analisis. Menentukan pendekatan pengembangan terbaik, menentukan sumber daya yang dibutuhkan, dan menentukan model normatif.

2. Tahap desain
   Desain antar muka, dialog, basis data, model, dan komponen pengetahuan.

3. Tahap konstruksi
   Implementasikan desain pada tahap desain ke dalam program sistem pendukung keputusan.

4. Tahap implementasi
   Melakukan pemeliharaan dan dokumentasi.

5. Tahap pemeliharaan dan dokumentasi
   Melakukan proses secara berulang-ulang untuk meningkatkan kualitas sistem.

2.7 Kemampuan dan Keterbatasan Sitem Pendukung Keputusan

Sistem Pendukung Keputusan memiliki kemampuan dan keterbatasan, diantaranya adalah sebagai berikut :

2.7.1 Kemampuan Sistem Pendukung Keputusan

Berikut adalah beberapa kemampuan pada sistem pendukung keputusan[**?**] :

1. Menunjang pembuatan keputusan manajemen dalam menangani masalah semi terstruktur dan tidak terstruktur
2. Membantu manajer pada berbagai tingkatan manajemen, mulai dari manajemen tingkat atas sampai manajemen tingkat bawah
3. Menunjang pembuatan keputusan secara kelompok maupun perorangan
4. Menunjang pembuatan keputusan yang saling bergantung dan berurutan
5. Menunjang tahap-tahap pembuatan keputusan antara lain intelligensi, desain, *choice*, dan *implementation*
6. Kemampuan untuk melakukan adaptasi setiap saat dan bersifat fleksibel
7. Kemudahan melakukan interaksi sistem
8. Meningkatkan efektivitas dalam pembuatan keputusan daripada efisiensi
9. Mudah dikembangkan oleh pemakai akhir
10. Kemampuan pemodelan dan analisis pembuatan keputusan
11. Kemudahan melakukan pengaksesan berbagai sumber dan format data

2.7.2 Keterbatasan Sitem Pendukung Keputusan

Meskipun sistem pendukung keputusan memiliki banyak kemampuan, namun masih memiliki keterbatasan seperti[**?**] :

1. Ada beberapa kemampuan manajemen dan bakat manusia yang tidak dapat dimodelkan, sehingga model yang ada dalam sistem tidak semuanya mencerminkan persoalan sebenarnya.
2. Kemampuan suatu SPK terbatas pada pembendaharaan pengetahuan yang dimilikinya (pengetahuan dasar serta model dasar).
3. Proses-proses yang dapat dilakukan oleh Sistem Pendukung Keputusan biasanya tergantung juga pada kemampuan perangkat lunak yang digunakannya.

Sistem Pendukung Keputusan tidak memiliki kemampuan intuisi seperti yang dimiliki oleh manusia. Bagaimana pun canggihnya suatu sistem pendukung keputusan, hanyalah sautu kumpulan perangkat keras, perangakat lunak dan sistem operasi yang tidak dilengkapi dengan kemampuan berpikir.

2.8 Penerapan Sistem Pendukung Keputusan

Sistem pendukung keputusan dapat diterapkan pada :

1. *Customer Relationship Management*
   (a) Meningkatkan *memory* pelanggan
   (b) Memaksimalkan laba
   (c) Memaksimalkan nilai pelanggan *(up-selling)*
   (d) Mengidentifikasi dan memperlakukan pelanggan yang paling berharga
2. Perbankan
   (a) Mengoptimalkan proses aplikasi pinjaman
   (b) Mendeteksi transaksi penituan
   (c) Memaksimalkan nilai pelanggan
   (d) Mengoptimalkan cadangan kas dengan *forecasting*
3. Ritel dan logistik
   (a) Mengoptimalkan tingkat ketersediaan di lokasi berbeda
   (b) Meningkatkan tata letak toko dan promosi penjualan
   (c) Mengoptimalkan logistik dengan memprediksi efek musiman
   (d) Meminimalisir kerugian
4. Manufaktur dan *maintenance*
   (a) Memprediksi kerusakan mesin
   (b) Mengidentifikasi *anomaly* dalam sistem produksi
   (c) Mengoptimalkan kapasitas prosuksi
   (d) Menemukan cara-cara untuk mengingkatkan kualitas produk
5. Pedagangan dan *Securities Trading*
   (a) Memprediksi perubahan harga obligasi tertentu
   (b) Perkiraan flutuasi saham
   (c) Menilai dampak peristiwa pada pergerakan pasar
   (d) Mengidentifikasi dan mencegah kegiatan penipuan dalam perdagangan
6. Asuransi
   (a) Perkiraan klaim biaya
   (b) Optimalisasi rencana
   (c) Optimalisasi pemasaran
   (d) Identifikasi dan mencegah klaim palsu
7. Komputer *hardware* dan *software*, sains dan rekayasa, pemerintah dan pertahanan, keamanan dan penegakan hukum, industri hiburan, dan olahraga.

BAB 3

MULTI CRITERIA DECISION MAKING

3.1 Pengertian MCDM

Multiple Criteria Decision Making (MCDM) adalah suatu metode pengambilan keputusan untuk menetapkan alternatif terbaik dari sejumlah alternatif berdasarkan beberapa kriteria tertentu [**?**].). Kriteria biasanya berupa ukuran-ukuran atau aturan-aturan atau standar yang digunakan dalam pengambilan keputusan. Secara umum dapat dikatakan bahwa MCDM menyeleksi alternatif terbaik dari sejumlah alternatif. Beberapa fitur umum yang akan digunakan dalam MCDM [**?**], yaitu:

1. Alternatif, alternatif adalah obyek-obyek yang berbeda dan memiliki kesempatan yang sama untuk dipilih oleh pengambil keputusan.

2. Atribut, atribut sering juga disebut sebagai karakteristik, komponen, atau kriteria keputusan. Meskipun ada kebanyakan kriteria bersifat satu level, namun tidak menutup kemungkinan adanya sub kriteria yang berhubungan dengan kriteria yang telah diberikan.

By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

3. Konflik antar kriteria, beberapa kriteria biasanya mempunyai konflik antara satu dengan yang lainnya, misalnya kriteria keuntungan akan mengalami konflik dengan kriteria biaya.

4. Bobot keputusan, bobot keputusan menunjukkan kepentingan relatif dari setiap kriteria. Pada MCDM akan dicari bobot kepentingan dari setiap kriteria.

5. Matriks keputusan, suatu matriks keputusan x yang berukuran m x n, berisi elemen-elemen xij, yang merepresentasikan rating dari alternatif Ai (I = 1,2,, m) terhadap kriteria Cj (j = 1,2,, n).

MCDM memiliki dua kategori yakni *Multiple Objective Decision Making* (MODM) dan *Multiple Attribute Decision Making* (MADM).

3.2 *Multiple Objective Decision Making* (MODM)

Multiple Objective Decision Making (MODM) adalah suatu metode dengan mengambil banyak kriteria sebagai dasar dari pengambilan keputusan yang didalamnya mencakup masalah perancangan (*design*), dimana teknik-teknik matematik untuk optimasi digunakan dan untuk jumlah alternatif yang sangat besar (sampai dengan tak terhingga).

3.3 *Multiple Attribute Decision Making* (MADM)

Multiple Attribute Decision Making (MADM) adalah suatu metode dengan mengambil banyak kriteria sebagai dasar pengambilan keputusan, dengan penilaian yang subjektif menyangkut masalah pemilihan, dimana analisis matematis tidak terlalu banyak dan digunakan untuk pemilihan alternatif dalam jumlah sedikit.

3.3.1 Konsep Dasar MADM

Pada umumnya model MADM dilakukan melalui tiga tahapan antara lain penyusunan komponen-komponen situasi, analisis, dan sintesis informasi. Pada tahap penyusunan komponen-komponen situasi, dibentuk tabel taksiran yang berisikan identifikasi alternatif dan spesifikasi tujuan, kriteria dan atribut. Salah satu cara untuk menspesifikasikan tujuan situasi — Oi, i=1,,t — adalah dengan cara mendaftar konsekuensi-konsekuensi yang mungkin dari alternatif yang telah teridentifikasi — Ai, i=1,,n—. Selain itu juga disusun atribut-atribut yang akan digunakan — ak, k=1,,m —.

3.3.2 Metode-Metode Penyelesaian Masalah MADM

Ada bebebrapa metode yang dapat digunakan untuk menyelesaikan masalah MADM , antara lain:

1. *Simple Additive Weighting Method* (SAW)

2. *Weighted Product* (WP)
3. ELECTRE
4. *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS)
5. *Analytic hierarchy Process* (AHP)

BAB 4

SIMPLE ADDITIVE WEIGHTING

4.1 Pendahuluan Metode SAW

Simple additive weighting (SAW) merupakan metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua kriteria. Metode SAW membutuhkan proses normalisasi matrik keputusan, ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Metode ini mengharuskan pembuat keputusan menentukan bobot bagi setiap atribut. Nilai total untuk sebuah alternatif diperoleh dengan menjumlahkan seluruh hasil perkalian antar rating dan bobot tiap atribut. Rating tiap atribut haruslah bebas dimensi dalam arti telah melewati proses normalisasi sebelumnya. Metode SAW mengenal adanya dua atribut yaitu kriteria keuntungan (*benefit criteria*) dan kriteria biaya (*cost criteria*). Perbedaan mendasar dari kedua kriteria ini adalah dalam pemilihan kriteria ketika mengambil keputusan.

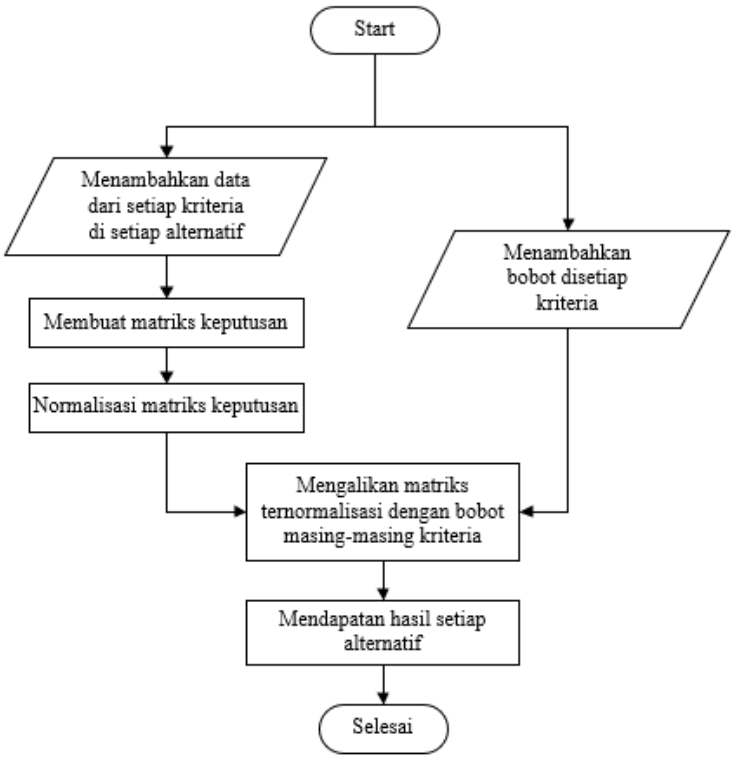
4.2 Prinsip Dasar SAW

Menurut Fishburn dan Mac Crimmon prinsip dasar SAW adalah :

By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

- Konsep dasar SAW adalah mencari penjumlahan terbobot dari rating kinerja ternormalisasi (R) pada setiap alternatif pada semua bobot atribut (W).
- Metode SAW membutuhkan proses normalisasi matriks ke suatu skala yang dapat diperbandingkan dengan semua rating alterntif yang ada.

4.3 Langkah-langkah Metode SAW



Gambar 4.1 Algoritma *Simple Additive Weighting*

Langkah-langkah dalam metode *simple additive weighting* (SAW) ialah sebagai berikut:

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu Ci.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria (Ci) kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut

(atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R.

4. Hasil akhir diperoleh dari proses pengurutan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (Ai) sebagai solusi.

Rumus untuk melakukan normalisasi tersebut ialah :

- Untuk *benefit criteria*

$$R_{ij} = \frac{x_{ij}}{max_i x_{ij}}, \quad \text{jika j adalah benefit criteria} \tag{4.1}$$

- Untuk *cost criteria*

$$R_{ij} = \frac{min_i x_{ij}}{x_{ij}}, \quad \text{jika j adalah cost criteria} \tag{4.2}$$

Dengan :
R_{ij} = nilai rating kinerja normalisasi
x_{ij} = nilai atribut yang dimiliki dari setiap kriteria
$max_i \quad x_{ij}$ = nilai terbesar dari setiap kriteria i
$min_i \quad x_{ij}$ = nilai terkecil dari setiap kriteria i
Dimana r_{ij} adalah rating kinerja ternormalisasi dari alternatif Ai pada atribut C_j, $i = 1, 2, ...m$ dan $j = 1, 2, ...n$

Nilai preferensi untuk setiap alternatif (V_i), ialah sebagai berikut :

$$V_i = \sum_{j=1}^{n} w_j r_{ij} \tag{4.3}$$

Dengan :
V_i = urutan untuk setiap alternatif
w_j = nilai bobot dari setiap kriteria
r_{ij} = nilai rating kinerja ternormalisasi
Nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i lebih dipilih.

4.4 Kelebihan dan Kekurangan Metode SAW

Metode *Simple Additive Weighting* (SAW) memiliki kelebihan dan kekurangan seperti berikut [**?**] :

4.4.1 Kelebihan

Kelebihan dari model *Simple Additive Weighting* (SAW) dibandingkan dengan model pengambilan keputusan yang lain adalah :

1. Kemampuan melakukan penilaian secara lebih tepat, karena didasarkan pada nilai kriteria dan bobot preferensi yang sudah ditentukan.

2. Total perubahan nilai yang dihasilkan lebih banyak, sehingga sangat relevan untuk menyelesaikan masalah pengambilan keputusan.

3. Mampu menyeleksi alternatif terbaik dari sejumlah alternatif yang ada.

4.4.2 Kekurangan

Bukan hanya memiliki kelebihan metode SAW juga memiliki kelemahan, antara lain :

1. Harus menentukan bobot pada setiap atribut.

2. Data yang dimasukkan harus benar dan tepat agar tidak menimbulkan kesalahan pada saat pembobotan dan perankingan kriteria.

3. Harus membuat matriks keputusan.

4. Keakuratan hasil kurang, hal ini dikarenakan kriteria yang ditentukan harus dinamis dan memiliki cakupan yang luas.

4.5 Studi Kasus dan Teknik Penyelesaian SAW

Metode *Simple Additice Weighting* (SAW) dapat memudahkan menyelesaian masalah dalam pemilihan pegawai teladan dengam menggunakan metode *Multiple Atribut Decision Making* (MADM) menggunakan model *Simple Assitive Weighting*. Guna mendapatkan pegawai teladan terbaik, diperlukan syarat kriteria dari setiap pegawai. Adapun sysrat yang nasabah yang harus dipenuhi berdasarkan kriteria, subkriteria dan bobot yang telah ditentukan oleh perusahaan. Untuk menyelesaikan masalah, algoritma yang digunakan dapat dilihat pada gambar 4.1.

Kriteria pemilhan pegawai teladan terbagi dalam tiga kondisi dimana masing-masing kondisi terbagi atas beberapa dimensi dengan indikator-indikator tertentu. Dalam menentuka kriteria pegawai teladan dengan menggunakan *Multiple Attribute Decision Making* (MADM) menggunakan model *Simple Additive Weighting* diperlukan syarat kriteria. Adapun syarat pegawai harus memenuhi kriteria, subkriteria, bobot, dan sifat indikatornya adalah sebagai berikut :

1. Disiplin

Indikator yang digunakan dalam penentuan pegawai teladan berdasarkan kategori disiplin dimana perusahaan melalui *Supervisor* menganalisis kandidat atau alternatif pegawai nya untuk mengetahui kedisiplinan pagawai dalam lingkungan kerja.

Kode	Kriteria	Sub Kriteria	Nilai	Sifat	Bobot
C1	Kehadiran	Sangat Baik	4	*Benefit*	0,1
		Baik	3	*Benefit*	
		Cukup	2	*Benefit*	
		Kurang	1	*Benefit*	
C2	Tidak meninggalkan tempat tugas	Sangat Baik	4	*Benefit*	0,05
		Baik	3	*Benefit*	
		Cukup	2	*Benefit*	
		Kurang	1	*Benefit*	
C3	Ketaatan pada peraturan	Sangat Baik	4	*Benefit*	0,1
		Baik	3	*Benefit*	
		Cukup	2	*Benefit*	
		Kurang	1	*Benefit*	
C4	Ketepatan waktu kerja	Sangat Baik	4	*Benefit*	0,05
		Baik	3	*Benefit*	
		Cukup	2	*Benefit*	
		Kurang	1	*Benefit*	
C5	Kualitas kerja	Sangat Baik	4	*Benefit*	0,1
		Baik	3	*Benefit*	
		Cukup	2	*Benefit*	
		Kurang	1	*Benefit*	

Tabel 4.1 Kategori Disiplin

2. Penampilan
   Indikator yang digunakan dalam penentuan pegawai teladan berdasarkan kategori penampilan yang berhubungan dengan kerapihan dan kebersihan di lingkungan kerja.

Kode	Kriteria	Sub Kriteria	Nilai	Sifat	Bobot
C6	Kerapihan lingkun-gan kerja	Sangat Baik Baik Cukup Kurang	4 3 2 1	*Benefit* *Benefit* *Benefit* *Benefit*	0,05
C7	Kebersihan	Sangat Baik Baik Cukup Kurang	4 3 2 1	*Benefit* *Benefit* *Benefit* *Benefit*	0,05
C8	Kerapihan berpakaian	Sangat Baik Baik Cukup Kurang	4 3 2 1	*Benefit* *Benefit* *Benefit* *Benefit*	0,05

Tabel 4.2 Kategori Penampilan

3. Sikap dan Komunikasi
   Indikator yang digunakan dalam penentuan pegawai teladan berdasarkan kategori penampilan yang berhubungan dengan kerapihan dan kebersihan di lingkungan kerja.

Kode	Kriteria	Sub Kriteria	Nilai	Sifat	Bobot
C9	Kerjasama dalam tim	Sangat Baik	4	*Benefit*	0,05
		Baik	3	*Benefit*	
		Cukup	2	*Benefit*	
		Kurang	1	*Benefit*	
C10	Tanggap terhadap situasi	Sangat Baik	4	*Benefit*	0,05
		Baik	3	*Benefit*	
		Cukup	2	*Benefit*	
		Kurang	1	*Benefit*	
C11	Kemauan untuk mengem-bangkan diri	Sangat Baik	4	*Benefit*	0,05
		Baik	3	*Benefit*	
		Cukup	2	*Benefit*	
		Kurang	1	*Benefit*	
C12	Pencapaian target kerja	Sangat Baik	4	*Benefit*	0,1
		Baik	3	*Benefit*	
		Cukup	2	*Benefit*	
		Kurang	1	*Benefit*	
C13	Pelaksanaan tugas pokok	Sangat Baik	4	*Benefit*	0,05
		Baik	3	*Benefit*	
		Cukup	2	*Benefit*	
		Kurang	1	*Benefit*	
C14	Inisiatif/kreatifitas	Sangat Baik	4	*Benefit*	0,05
		Baik	3	*Benefit*	
		Cukup	2	*Benefit*	
		Kurang	1	*Benefit*	
C15	Tanggung jawab	Sangat Baik	4	*Benefit*	0,1
		Baik	3	*Benefit*	
		Cukup	2	*Benefit*	
		Kurang	1	*Benefit*	

Tabel 4.3 Kategori Penampilan

Dari banyaknya pegawai pada perusahaan, diambil 10 pegawai sebagai pegawai kandidat dengan masing-masing 1 divisi adalah 1 kandidat. Data nilai kandidat pegawai dimasukkan kedalam tabel di bawah ini.

Kode Kriteria	Nama	Sub Kriteria	Nilai
C1	Agus	Baik Sekali	4
	Nina	Baik Sekali	4
	Dody	Baik Sekali	4
	Aditya	Baik Sekali	4
	Yando	Baik Sekali	4
	Jali	Baik Sekali	4
	Dedi	Baik Sekali	4
	Hanna	Baik Sekali	4
	Bambang	Baik Sekali	4
	Anwar	Baik Sekali	4

Tabel 4.4 Nilai Kehadiran

Kode Kriteria	Nama	Sub Kriteria	Nilai
C2	Agus	Baik	3
	Nina	Baik	3
	Dody	Cukup	2
	Aditya	Baik Sekali	4
	Yando	Cukup	2
	Jali	Cukup	2
	Dedi	Baik	3
	Hanna	Cukup	2
	Bambang	Baik Sekali	4
	Anwar	Baik	3

Tabel 4.5 Nilai Tidak Meninggalkan Tempat Tugas

Kode Kriteria	Nama	Sub Kriteria	Nilai
C3	Agus	Baik Sekali	4
	Nina	Baik Sekali	4
	Dody	Baik	3
	Aditya	Baik Sekali	4
	Yando	Baik	3
	Jali	Baik Sekali	4
	Dedi	Baik Sekali	4
	Hanna	Baik	3
	Bambang	Baik Sekali	4
	Anwar	Baik	3

Tabel 4.6 Nilai Ketaatan pada Peraturan

Kode Kriteria	Nama	Sub Kriteria	Nilai
C4	Agus	Baik Sekali	4
	Nina	Baik Sekali	4
	Dody	Baik	3
	Aditya	Baik Sekali	4
	Yando	Baik	3
	Jali	Baik Sekali	4
	Dedi	Cukup	2
	Hanna	Baik	3
	Bambang	Cukup	2
	Anwar	Baik	3

Tabel 4.7 Nilai Ketepatan Waktu Kerja

Kode Kriteria	Nama	Sub Kriteria	Nilai
C5	Agus	Baik	3
	Nina	Baik	3
	Dody	Baik Sekali	4
	Aditya	Baik	3
	Yando	Baik Sekali	4
	Jali	Baik	3
	Dedi	Baik Sekali	4
	Hanna	Baik Sekali	4
	Bambang	Baik	3
	Anwar	Baik Sekali	4

Tabel 4.8 Nilai Kualitas Kerja

Kode Kriteria	Nama	Sub Kriteria	Nilai
C6	Agus	Cukup	2
	Nina	Cukup	2
	Dody	Baik Sekali	4
	Aditya	Cukup	2
	Yando	Baik Sekali	4
	Jali	Cukup	2
	Dedi	Cukup	2
	Hanna	Baik Sekali	4
	Bambang	Cukup	2
	Anwar	Baik	3

Tabel 4.9 Nilai Kerapihan Lingkungan Kerja

Kode Kriteria	Nama	Sub Kriteria	Nilai
C7	Agus	Cukup	2
	Nina	Cukup	2
	Dody	Cukup	2
	Aditya	Cukup	2
	Yando	Baik	3
	Jali	Cukup	2
	Dedi	Cukup	2
	Hanna	Cukup	2
	Bambang	Cukup	2
	Anwar	Baik	3

Tabel 4.10 Nilai Kebersihan

Kode Kriteria	Nama	Sub Kriteria	Nilai
C8	Agus	Baik	3
	Nina	Baik Sekali	4
	Dody	Baik	3
	Aditya	Baik Sekali	4
	Yando	Baik	3
	Jali	Baik	3
	Dedi	Baik Sekali	4
	Hanna	Baik	3
	Bambang	Baik Sekali	4
	Anwar	Baik	3

Tabel 4.11 Nilai Kerapihan Berpakaian

Kode Kriteria	Nama	Sub Kriteria	Nilai
C9	Agus	Baik Sekali	4
	Nina	Baik	3
	Dody	Baik	3
	Aditya	Baik Sekali	4
	Yando	Baik	3
	Jali	Baik Sekali	4
	Dedi	Baik	3
	Hanna	Baik	3
	Bambang	Baik Sekali	4
	Anwar	Baik	3

Tabel 4.12 Nilai Kerjasama Dalam Tim

Kode Kriteria	Nama	Sub Kriteria	Nilai
C10	Agus	Baik	3
	Nina	Baik	3
	Dody	Baik Sekali	4
	Aditya	Baik Sekali	4
	Yando	Baik Sekali	4
	Jali	Baik	3
	Dedi	Baik	3
	Hanna	Baik Sekali	4
	Bambang	Baik	3
	Anwar	Baik Sekali	4

Tabel 4.13 Nilai Tanggap Terhadap Situasai

Kode Kriteria	Nama	Sub Kriteria	Nilai
C11	Agus	Baik	3
	Nina	Baik	3
	Dody	Baik Sekali	4
	Aditya	Baik Sekali	4
	Yando	Baik Sekali	4
	Jali	Baik	3
	Dedi	Baik	3
	Hanna	Baik Sekali	4
	Bambang	Baik	3
	Anwar	Baik Sekali	4

Tabel 4.14 Nilai Tanggap Terhadap Situasai

Kode Kriteria	Nama	Sub Kriteria	Nilai
C12	Agus	Baik	3
	Nina	Baik	3
	Dody	Baik Sekali	4
	Aditya	Baik Sekali	4
	Yando	Baik Sekali	4
	Jali	Baik	3
	Dedi	Baik	3
	Hanna	Baik Sekali	4
	Bambang	Baik	3
	Anwar	Baik Sekali	4

Tabel 4.15 Nilai Tanggap Terhadap Situasai

Kode Kriteria	Nama	Sub Kriteria	Nilai
C13	Agus	Baik	3
	Nina	Baik	3
	Dody	Baik Sekali	4
	Aditya	Baik Sekali	4
	Yando	Baik Sekali	4
	Jali	Baik	3
	Dedi	Baik	3
	Hanna	Baik Sekali	4
	Bambang	Baik	3
	Anwar	Baik Sekali	4

Tabel 4.16 Nilai Tanggap Terhadap Situasai

Kode Kriteria	Nama	Sub Kriteria	Nilai
C14	Agus	Baik	3
	Nina	Baik	3
	Dody	Baik Sekali	4
	Aditya	Baik Sekali	4
	Yando	Baik Sekali	4
	Jali	Baik	3
	Dedi	Baik	3
	Hanna	Baik Sekali	4
	Bambang	Baik	3
	Anwar	Baik Sekali	4

Tabel 4.17 Nilai Tanggap Terhadap Situasai

Kode Kriteria	Nama	Sub Kriteria	Nilai
C15	Agus	Baik	3
	Nina	Baik	3
	Dody	Baik Sekali	4
	Aditya	Baik Sekali	4
	Yando	Baik Sekali	4
	Jali	Baik	3
	Dedi	Baik	3
	Hanna	Baik Sekali	4
	Bambang	Baik	3
	Anwar	Baik Sekali	4

Tabel 4.18 Nilai Tanggap Terhadap Situasai

Dari data nilai kandidat pegawai yang telah dipaparkan, selanjutnya dilakukan normalisasi berdasarkan sifatnya. Untuk kriteria dengan sifat *benefit* menggunakan rumus 4.1 dan untuk *cost* menggunakan rumus normalisasi 4.2.

Kode	Nama	Nilai	Normalisasi	
			Perhitungan	Hasil
C1	Agus	4	$\frac{4}{4}$	1
	Nina	4	$\frac{4}{4}$	1
	Dody	4	$\frac{4}{4}$	1
	Aditya	4	$\frac{4}{4}$	1
	Yando	4	$\frac{4}{4}$	1
	Jali	4	$\frac{4}{4}$	1
	Dedi	4	$\frac{4}{4}$	1
	Hanna	4	$\frac{4}{4}$	1
	Bambang	4	$\frac{4}{4}$	1
	Anwar	4	$\frac{4}{4}$	1

Tabel 4.19 Normalisasi Kehadiran

Kode	Nama	Nilai	Normalisasi	
			Perhitungan	Hasil
C2	Agus	3	$\frac{3}{4}$	0,75
	Nina	3	$\frac{3}{4}$	0,75
	Dody	2	$\frac{2}{4}$	0,5
	Aditya	4	$\frac{4}{4}$	1
	Yando	2	$\frac{2}{4}$	0,5
	Jali	2	$\frac{2}{4}$	0,5
	Dedi	3	$\frac{3}{4}$	0,75
	Hanna	2	$\frac{2}{4}$	0,5
	Bambang	4	$\frac{4}{4}$	1
	Anwar	3	$\frac{3}{4}$	0,75

Tabel 4.20 Normalisasi Tidak Meninggalkan Tempat Tugas

Kode	Nama	Nilai	Normalisasi	
			Perhitungan	Hasil
C3	Agus	4	$\frac{4}{4}$	1
	Nina	4	$\frac{4}{4}$	1
	Dody	3	$\frac{3}{4}$	0,75
	Aditya	4	$\frac{4}{4}$	1
	Yando	3	$\frac{3}{4}$	0,75
	Jali	3	$\frac{3}{4}$	0,75
	Dedi	3	$\frac{3}{4}$	0,75
	Hanna	3	$\frac{3}{4}$	0,75
	Bambang	4	$\frac{4}{4}$	1
	Anwar	3	$\frac{3}{4}$	0,75

Tabel 4.21 Normalisasi Ketaatan pada Peraturan

Kode	Nama	Nilai	Normalisasi	
			Perhitungan	Hasil
C4	Agus	4	$\frac{4}{4}$	1
	Nina	4	$\frac{4}{4}$	1
	Dody	3	$\frac{3}{4}$	0,75
	Aditya	4	$\frac{4}{4}$	1
	Yando	3	$\frac{3}{4}$	0,75
	Jali	4	$\frac{4}{4}$	1
	Dedi	2	$\frac{2}{4}$	0,5
	Hanna	3	$\frac{3}{4}$	0,75
	Bambang	2	$\frac{2}{4}$	0,5
	Anwar	3	$\frac{3}{4}$	0,75

Tabel 4.22 Normalisasi Ketepatan Waktu Kerja

Kode	Nama	Nilai	Normalisasi	
			Perhitungan	Hasil
C5	Agus	3	$\frac{4}{4}$	1
	Nina	3	$\frac{3}{4}$	0,75
	Dody	4	$\frac{4}{4}$	1
	Aditya	3	$\frac{3}{4}$	0,75
	Yando	4	$\frac{4}{4}$	1
	Jali	3	$\frac{3}{4}$	0,75
	Dedi	4	$\frac{4}{4}$	1
	Hanna	4	$\frac{4}{4}$	1
	Bambang	3	$\frac{3}{4}$	0,75
	Anwar	4	$\frac{4}{4}$	1

Tabel 4.23 Normalisasi Kualitas Kerja

Kode	Nama	Nilai	Normalisasi	
			Perhitungan	Hasil
C6	Agus	2	$\frac{2}{4}$	0,5
	Nina	2	$\frac{2}{4}$	0,5
	Dody	4	$\frac{4}{4}$	1
	Aditya	2	$\frac{2}{4}$	0,5
	Yando	4	$\frac{4}{4}$	1
	Jali	2	$\frac{2}{4}$	0,5
	Dedi	2	$\frac{2}{4}$	0,5
	Hanna	4	$\frac{4}{4}$	1
	Bambang	2	$\frac{2}{4}$	0,5
	Anwar	3	$\frac{2}{4}$	0,75

Tabel 4.24 Normalisasi Kerapihan Lingkungan Kerja

Kode	Nama	Nilai	Normalisasi	
			Perhitungan	Hasil
C7	Agus	2	$\frac{2}{4}$	0,5
	Nina	2	$\frac{2}{4}$	0,5
	Dody	2	$\frac{2}{4}$	0,5
	Aditya	2	$\frac{2}{4}$	0,5
	Yando	3	$\frac{3}{4}$	0,75
	Jali	2	$\frac{2}{4}$	0,5
	Dedi	2	$\frac{2}{4}$	0,5
	Hanna	2	$\frac{2}{4}$	0,5
	Bambang	2	$\frac{2}{4}$	0,5
	Anwar	3	$\frac{3}{4}$	0,75

Tabel 4.25 Normalisasi Kebersihan

Kode	Nama	Nilai	Normalisasi	
			Perhitungan	Hasil
C8	Agus	3	$\frac{3}{4}$	0,75
	Nina	4	$\frac{4}{4}$	1
	Dody	3	$\frac{3}{4}$	0,75
	Aditya	4	$\frac{4}{4}$	1
	Yando	3	$\frac{3}{4}$	0,75
	Jali	3	$\frac{3}{4}$	0,75
	Dedi	4	$\frac{4}{4}$	1
	Hanna	3	$\frac{3}{4}$	0,75
	Bambang	4	$\frac{4}{4}$	1
	Anwar	3	$\frac{4}{4}$	1

Tabel 4.26 Normalisasi Kerapihan Berpakaian

Kode	Nama	Nilai	Normalisasi	
			Perhitungan	Hasil
C9	Agus	4	$\frac{4}{4}$	1
	Nina	3	$\frac{3}{4}$	0.75
	Dody	3	$\frac{3}{4}$	0,75
	Aditya	4	$\frac{4}{4}$	1
	Yando	3	$\frac{3}{4}$	0,75
	Jali	4	$\frac{4}{4}$	1
	Dedi	3	$\frac{3}{4}$	0,75
	Hanna	3	$\frac{3}{4}$	0,75
	Bambang	4	$\frac{4}{4}$	1
	Anwar	3	$\frac{3}{4}$	0,75

Tabel 4.27 Normalisasi Kerjasama Dalam Tim

Kode	Nama	Nilai	Normalisasi	
			Perhitungan	Hasil
C10	Agus	3	$\frac{3}{4}$	0,75
	Nina	3	$\frac{3}{4}$	0,75
	Dody	4	$\frac{4}{4}$	1
	Aditya	4	$\frac{4}{4}$	1
	Yando	4	$\frac{4}{4}$	1
	Jali	3	$\frac{3}{4}$	0,75
	Dedi	3	$\frac{3}{4}$	0,75
	Hanna	4	$\frac{4}{4}$	1
	Bambang	3	$\frac{3}{4}$	0,75
	Anwar	4	$\frac{4}{4}$	1

Tabel 4.28 Normalisasi Tanggap Terhadap Situasai

Kode	Nama	Nilai	Normalisasi	
			Perhitungan	Hasil
C11	Agus	3	$\frac{3}{4}$	0,75
	Nina	3	$\frac{3}{4}$	0,75
	Dody	4	$\frac{4}{4}$	1
	Aditya	4	$\frac{4}{4}$	1
	Yando	4	$\frac{4}{4}$	1
	Jali	3	$\frac{3}{4}$	0,75
	Dedi	3	$\frac{3}{4}$	0,75
	Hanna	4	$\frac{4}{4}$	1
	Bambang	3	$\frac{3}{4}$	0,75
	Anwar	4	$\frac{4}{4}$	1

Tabel 4.29 Normalisasi Tanggap Terhadap Situasai

Kode	Nama	Nilai	Normalisasi	
			Perhitungan	Hasil
C12	Agus	3	$\frac{3}{4}$	0,75
	Nina	3	$\frac{3}{4}$	0,75
	Dody	4	$\frac{4}{4}$	1
	Aditya	4	$\frac{4}{4}$	1
	Yando	4	$\frac{4}{4}$	1
	Jali	3	$\frac{3}{4}$	0,75
	Dedi	3	$\frac{3}{4}$	0,75
	Hanna	4	$\frac{4}{4}$	1
	Bambang	3	$\frac{3}{4}$	0,75
	Anwar	4	$\frac{4}{4}$	1

Tabel 4.30 Normalisasi Tanggap Terhadap Situasai

Kode	Nama	Nilai	Normalisasi	
			Perhitungan	Hasil
C13	Agus	3	$\frac{3}{4}$	0,75
	Nina	3	$\frac{3}{4}$	0,75
	Dody	4	$\frac{4}{4}$	1
	Aditya	4	$\frac{4}{4}$	1
	Yando	4	$\frac{4}{4}$	1
	Jali	3	$\frac{3}{4}$	0,75
	Dedi	3	$\frac{3}{4}$	0,75
	Hanna	4	$\frac{4}{4}$	1
	Bambang	3	$\frac{3}{4}$	0,75
	Anwar	4	$\frac{4}{4}$	1

Tabel 4.31 Normalisasi Tanggap Terhadap Situasai

Kode	Nama	Nilai	Normalisasi	
			Perhitungan	Hasil
C14	Agus	3	$\frac{3}{4}$	0,75
	Nina	3	$\frac{3}{4}$	0,75
	Dody	4	$\frac{4}{4}$	1
	Aditya	4	$\frac{4}{4}$	1
	Yando	4	$\frac{4}{4}$	1
	Jali	3	$\frac{3}{4}$	0,75
	Dedi	3	$\frac{3}{4}$	0,75
	Hanna	4	$\frac{4}{4}$	1
	Bambang	3	$\frac{3}{4}$	0,75
	Anwar	4	$\frac{4}{4}$	1

Tabel 4.32 Normalisasi Tanggap Terhadap Situasai

Kode	Nama	Nilai	Normalisasi	
			Perhitungan	Hasil
C15	Agus	3	$\frac{3}{4}$	0,75
	Nina	3	$\frac{3}{4}$	0,75
	Dody	4	$\frac{4}{4}$	1
	Aditya	4	$\frac{4}{4}$	1
	Yando	4	$\frac{4}{4}$	1
	Jali	3	$\frac{3}{4}$	0,75
	Dedi	3	$\frac{3}{4}$	0,75
	Hanna	4	$\frac{4}{4}$	1
	Bambang	3	$\frac{3}{4}$	0,75
	Anwar	4	$\frac{4}{4}$	1

Tabel 4.33 Normalisasi Tanggap Terhadap Situasai

Perankingan		
Nama	**Perhitungan**	**Hasil**
Agus	$(1 \times 0,1) + (0,75 \times 0,05) + (1 \times 0,1) + (1 \times 0,05) + (0,75 \times 0,1) + (0,5 \times 0,05) + (0,67 \times 0,05) + (0,75 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,05) + (1 \times 0,05) + (1 \times 0,1) + (1 \times 0,05) + (0,67 \times 0,05) + (0,75 \times 0,1)$	0,85
Nina	$(1 \times 0,1) + (0,75 \times 0,05) + (1 \times 0,1) + (1 \times 0,05) + (0,75 \times 0,1) + (0,5 \times 0,05) + (0,67 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,05) + (0,75 \times 0,05) + (0,75 \times 0,05) + (1 \times 0,1) + (1 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,1)$	0,86
Dody	$(1 \times 0,1) + (0,5 \times 0,05) + (0,75 \times 0,1) + (0,75 \times 0,05) + (1 \times 0,1) + (1 \times 0,05) + (0,67 \times 0,05) + (0,75 \times 0,05) + (0,75 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,05) + (0,75 \times 0,1) + (0,75 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,1)$	0,82
Aditya	$(1 \times 0,1) + (1 \times 0,05) + (1 \times 0,1) + (1 \times 0,05) + (0,75 \times 0,1) + (0,5 \times 0,05) + (0,67 \times 0,05) + (1 \times 0,05) + (1 \times 0,05) + (1 \times 0,05) + (1 \times 0,05) + (0,75 * 0,1) + (0,67 \times 0,05) + (1 \times 0,05) + (0,67 \times 0,1)$	0,89
Yando	$(1 \times 0,1) + (0,5 \times 0,05) + (0,75 \times 0,1) + (0,75 \times 0,05) + (1 \times 0,1) + (1 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,05) + (0,75 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,05) + (0,75 \times 0,1) + (1 \times 0,05) + (0,67 \times 0,05) + (1 \times 0,1)$	0,86
Jali	$(1 \times 0,1) + (0,5 \times 0,05) + (1 \times 0,1) + (1 \times 0,05) + (0,75 \times 0,1) + (0,5 \times 0,05) + (0,67 \times 0,05) + (0,75 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,05) + (1 \times 0,05) + (1 \times 0,1) + (1 \times 0,05) + (0,67 \times 0,05) + (0,75 \times 0,1)$	0,84
Dedi	$(1 \times 0,1) + (0,75 \times 0,05) + (1 \times 0,1) + (0,5 \times 0,05) + (1 \times 0,1) + (0,5 \times 0,05) + (0,67 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,05) + (0,75 \times 0,05) + (0,75 \times 0,05) + (1 \times 0,1) + (1 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,1)$	0,86
Hanna	$(1 \times 0,1) + (0,5 \times 0,05) + (0,75 \times 0,1) + (0,75 \times 0,05) + (1 \times 0,1) + (1 \times 0,05) + (0,67 \times 0,05) + (0,75 \times 0,05) + (0,75 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,05) + (0,75 \times 0,1) + (0,75 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,1)$	0,82
Hanna	$(1 \times 0,1) + (0,5 \times 0,05) + (0,75 \times 0,1) + (0,75 \times 0,05) + (1 \times 0,1) + (1 \times 0,05) + (0,67 \times 0,05) + (0,75 \times 0,05) + (0,75 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,05) + (0,75 \times 0,1) + (0,75 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,1)$	0,82
Anwar	$(1 \times 0,1) + (0,75 \times 0,05) + (0,75 \times 0,1) + (0,75 \times 0,05) + (1 \times 0,1) + (0,75 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,05) + (0,75 \times 0,05) + (1 \times 0,05) + (0.75 \times 0,05) + (0,75 \times 0,1) + (0,75 \times 0,05) + (0,67 \times 0,05) + (1 \times 0,1)$	0,85

Tabel 4.34 Perankingan

Berdasarkan tabel 4.34 maka dapat disimpulkan bahwa pegawai teladan yang terpilih adalah Aditya dengan nilai perolehan tertinggi yaitu 0,89.

BAB 5

SISTEM

Setelah memahami cara penyelesaian studi kasus menggunakan *Simple Additive Weighting* (SAW) sebagai mana dijelaskan pada Bab IV. Selanjutnya pada bab ini akan menjelaskan tentang apa itu sistem, konsep dasar sistem, bentuk umum, elemen, karakteristik, klasifikasi, anaisis, komponen, dan cohtoh dari sistem.

5.1 Sistem

Sistem adalah sekelompok komponen dan elemen yang digabungkan menjadi satu untuk mencapai tujuan tertentu. Sistem berasal dari bahasa Latin (*systma*) dan bahasa Yunani (*sustma*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat. Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya seperti negara. Negara merupakan suatu kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu

By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

negara dimana yang berperan sebagai penggeraknya yaitu rakyat yang berada dinegara tersebut.

Ada banyak pendapat tentang pengertian dan definisi sistem yang dijelaskan oleh beberapa ahli. Berikut pengertian dan definisi sistem menurut beberapa ahli:

1. Jogianto (2005:2), Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata, seperti tempat, benda dan orang-orang yang betul-betul ada dan terjadi.

2. Indrajit (2001:2), Sistem adalah kumpulan-kumpulan dari komponen-komponen yang memiliki unsur keterkaitan antara satu dengan lainnya.

3. Lani Sidharta (1995:9), Sistem adalah himpunan dari bagian-bagian yang saling berhubungan, yang secara bersama mencapai tujuan-tujuan yang sama.

4. Murdick, R. G (1991:27), Sistem adalah seperangkat elemen yang membentuk kumpulan atau prosedur-prosedur atau bagan-bagan pengolahan yang mencari suatu tujuan bagian atau tujuan bersama dengan mengoperasikan data dan/atau barang pada waktu rujukan tertentu untuk menghasilkan informasi dan/atau energi dan/atau barang.

5. Davis, G. B (1991:45), Sistem adalah kumpulan dari elemen-elemen yang beroperai bersama-sama untuk menyelesaikan suatu sasaran.

5.2 Kendala Sistem

Suatu sistem harus memiliki tiga kendala dasar:

1. Suatu sistem harus memiliki beberapa struktur dan perilaku yang dirancang untuk mencapai tujuan yang telah ditentukan.

2. Interkonektivitas dan saling ketergantungan harus ada di antara komponen sistem.

3. Tujuan organisasi memiliki prioritas lebih tinggi daripada tujuan subsistemnya.

Misalnya, sistem manajemen lalu lintas, sistem penggajian, sistem perpustakaan otomatis, sistem informasi sumber daya manusia.

5.3 Konsep Dasar Sistem

Suatu sistem terdiri dari sistem-sistem bagian (subsystems). Masing- masing subsistem terdiri dari subsistem-subsistem yang lebih kecil lagi atau terdiri dari komponen-komponen.interaksi dari subsistem-subsistem sedemikian rupa, sehingga

dicapai suatu kesatuan yang terpadu atau terintegrasi (integrated). Keterpaduan sistem ini memungkinkan terciptanya kerjasama untuk menghasilkan informasi yang cepat, tepat dan akurat.

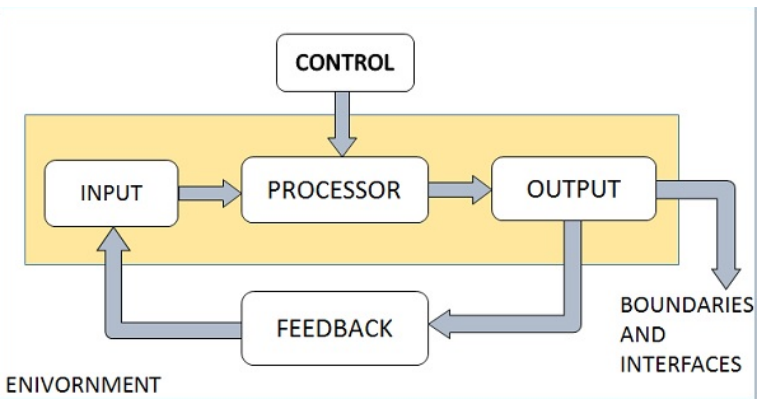
Menurut Jugiyono, sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

Pendekatan sistem yang merupakan jarngan kerja dari prosedur lebih menekankan urut-urutan operasi di dalam sistem. Prosedur *(procedure)* didefinisikan oleh Richard F. Neuschel yang disadur oleh Jogiyanto mendefinisikan Prosedur adalah suatu urut-urutan operasi klerikal (tulis-menuls) biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis terjadi.

Suatu prosedur adalah urutan-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakannya, kapan (*when*) dikerjakan dan bagaimana (*how*) mengerjakannya. Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

5.4 Bentuk Umum Sistem

Bentuk umum sistem dari suatu sistem terdiri atas masukan *(Input)*, proses dan keluaran *(Output)*, dalam bentuk umum sistem ini terdapat satu atau lebih masukan yang akan diproses dan akan menghasilkan suatu keluaran.



Gambar 5.1 Bentuk Umun

1. *Outputs and Inputs*
   - Tujuan utama suatu sistem adalah untuk menghasilkan keluaran yang bermanfaat bagi penggunanya.
   - *Input* adalah informasi yang masuk ke dalam sistem untuk diproses.

- *Output* adalah hasil dari pemrosesan.

2. *Processor(s)*

   - Prosesor adalah elemen dari suatu sistem yang melibatkan transformasi input menjadi output yang sebenarnya.
   - Ini adalah komponen operasional dari suatu sistem. Prosesor dapat memodifikasi input baik seluruhnya atau sebagian, tergantung pada spesifikasi output.
   - Ketika spesifikasi output berubah, demikian juga pemrosesan. Dalam beberapa kasus, input juga dimodifikasi untuk memungkinkan prosesor menangani transformasi.

3. *Control*

   - Elemen kontrol memandu sistem.
   - Subsistem pengambilan keputusan yang mengontrol pola kegiatan yang mengatur *input*, pemrosesan, dan *output*.
   - Perilaku Sistem komputer dikendalikan oleh Sistem Operasi dan perangkat lunak. Untuk menjaga keseimbangan sistem, apa dan berapa banyak input yang dibutuhkan ditentukan oleh Spesifikasi Output.

4. *Feedback*

   - *Feedback* menyediakan kontrol dalam sistem yang dinamis.
   - *Feedback* positif bersifat rutin yang mendorong kinerja sistem.
   - *Feedback* negatif bersifat informasi yang memberikan pengontrol informasi untuk bertindak.

5. *Environment*

   - *Environment* adalah ”supersistem” di mana suatu organisasi beroperasi.
   - Ini adalah sumber elemen eksternal yang menyerang sistem.
   - Ini menentukan bagaimana suatu sistem harus berfungsi. Misalnya, vendor dan pesaing dari lingkungan organisasi, dapat memberikan kendala yang memengaruhi kinerja bisnis aktual.

6. *Boundaries and Interface*

   - Suatu sistem harus didefinisikan oleh batasan-batasannya. Batas adalah batas yang mengidentifikasi komponen, proses, dan keterkaitannya ketika berinteraksi dengan sistem lain.
   - Setiap sistem memiliki batas yang menentukan lingkup pengaruh dan kontrolnya.
   - Pengetahuan tentang batas-batas sistem yang diberikan sangat penting dalam menentukan sifat antarmuka dengan sistem lain untuk desain yang sukses.

5.5 Elemen Sistem

Semua sistem meliputi tiga elemen utama yaitu *input*, transformasi dan *output*. Sebagian sistem dapat mengendalikan operasi mereka sendiri yang disebut sebagai sistem lingkaran tertutup *(closed-loop system)*. Sistem lingkaran tertutup mencakup suatu mekanisme kontrol, tujuan dan lingkaran umpan balik *(feedback loop)* disamping tiga elemen utama. Sistem yang tidak memiliki kemampuan pengendalian disebut sistem lingkaran terbuka *(open-loop system)*, dalam arti mereka berhubungan dengan lingkungan mereka. Perusahaan adalah suatu contoh sistem terbuka dan sistem lingkaran tertutup.

5.6 Karakteristik Sistem

Menurut Jogiyanto sistem memiliki karakteristik atau sifat-sifat yang tertentu yaitu :

1. komponen *(Components)*
2. batas sistem *(Boundary)*
3. lingkungan *(Environments)*
4. penghubung *(Interface)*
5. masukan *(Input)*
6. keluaran *(Output)*
7. pengolah *(Process)*
8. sasaran *(Objectives)* atau tujuan *(Goals)*

Menurut Jogiyanto sistem mempunyai karasteristik atau sifat- sifat tertentu, yaitu:

1. Komponen Sistem
   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling kerjasama membentuk satu kesatuan.
2. Sifat-sifat
   Sistem itu terdiri untuk menjalankan fungsi tertentu dan mempunyai sistem yang lain secara keseluruhan.
3. Batasan *(Boundary)* Sistem
   Merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya.
4. Lingkungan Luar Sistem *(Environment)* Adalah apapun diluar batasan sistem yang mempengaruhi operasi sistem, lingkungan luar sistem dapat bersifat menguntungkan dan juga dapat bersifat merugikan sistem tersebut.

5. Penghubung *(Interface)* Sistem
   Merupakan media penghubung antara suatu subsistem dengan subsistem lainya.

6. Masukkan Sistem *(Input)*
   Adalah energi yang dimasukkan kedalam sistem, masukan dapat berupa masukkan perawatan *(Maintenace Input)*, dan masukkan sinyal *(Signal Input)*, maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Sedangkan signal input adalah energi yang di proses untuk mendapatkan keluaran.

7. Keluaran Sistem *(Output)*
   Adalah hasil energi yang diolah dan di klasifikasikan menjadi keluaran yang berguna dan sisa pembuangan keluaran dapat merupakan masukan untuk subsistem yang lain kepada supra sistem.

8. Sasaran Sistem Suatu sistem pasti memiliki sasaran atau tujuan *(Goal)*. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan di hasilkan sistem.

5.7 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya :

1. Sistem diklasifikasikan sebagai sistem abstrak dan sistem fisik. Sistem abstrak adalah sistem yang berupa pemikiran atau ide - ide yang tidak tampak secara fisik. Sistem fisik adalah sistem yang ada secara fisik.

2. Sistem diklasifikasikan sebagai sistem alamiah dan sistem buatan manusia. Sistem alamiah adalah sistem yang terjadi karena proses alam tidak dibuat oleh manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia.

3. Sistem diklasifikasikan sebagai sistem tertentu dan sistem tak tentu. Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung probabilitas.

4. Sistem diklasifikasikan sebagai sistem tertutup dan sistem terbuka. Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem terbuka adalah sistem yang hubungan dan terpengaruh dengan lingkungan luarnya.

5.8 Analisis Sistem

Tahapan analisis sistem di mulai karena adanya permintaan terhadap sistem baru. Permintaan dapat dating dari seorang manajer dan dari luar departemen sistem informasi atau dari pihak eksekutif yang melihat adanya masalah atau menemukan

adanya peluang baru. Namun, adakalanya inisiatif pengembangan sistem baru berasal dari bagian yang bertanggung jawab terhadap pengembangan sistem informasi, yang bermaksud mengembangkan sistem yang sudah ada atau mengatasi masalah-masalah yang belum tertangani.

Analisis sistem adalah untuk menentukan hal-hal detail tentang yang akan dikerjakan oleh sistem yang diusulkan (dan bukan bagaimana caranya). analisis sistem mencakup studi kelayakan dan analisis kebutuhan.

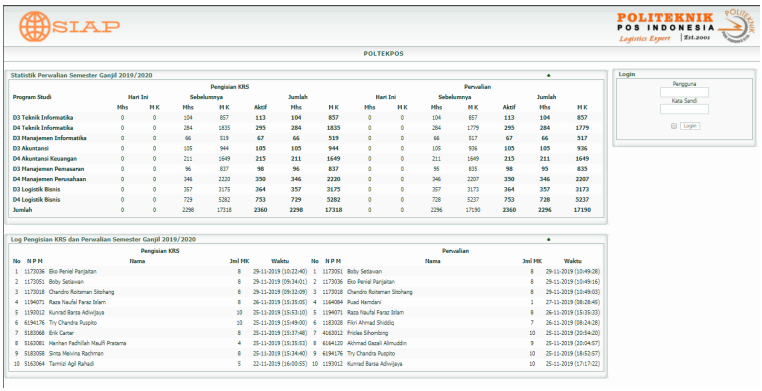
5.9 Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya saling bekerja sama dalam membentuk satu kesatuan komponen-komponen sistem atau elemen-elemen sistem dapat berupa subsistem atau bagian- bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

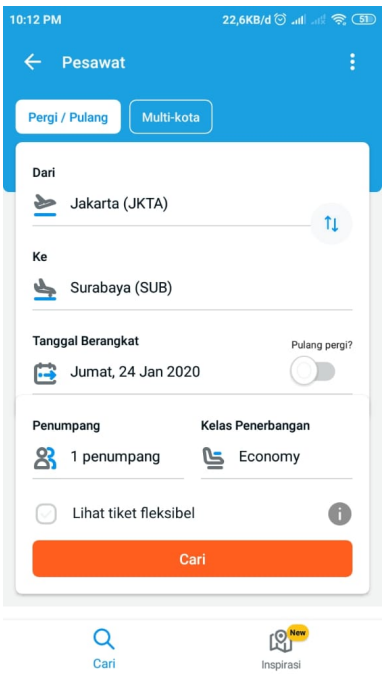
5.10 Contoh Sistem

1. *Website*
   Contoh penerepan sistem pada aplikasi *website* adalah sistem layanan/informasi akademis yang memungkinkan mahasiswa memperoleh data akademis dan mendaftar mata kuliah yang diambil pada tiap semester. Dengan adanya layanan sistem informasi ini, sangat memudahkan mahasiswa dalam setiap kepentingannya, ketika mengisi KRS kita tidak harus berangkat ke kampus hanya untuk mengisi KRS saja. Kita sudah bisa mengakses lewat internet sehingga bisa di akses dimana saja sekalipun kita sedang berada di luar kota. Semua informasi bisa kita dapatkan disini, mulai dari rekap nilai kita dalam setiap semester.

2. *Mobile*
   Contoh dari penerapan sistem pada aplikasi *mobile* adalah sistem pemesanan tiket secara *online*, misalnya pemesanan tiket kereta atau pesawat. Melalui sistem informasi ini kita tidak harus lagi cape antri di loket untuk membeli tiket, kita cukup buka internet kemudian melakukan transaksi untuk pembelian atau pemesanan tiket yang kita perlukan, sehingga menghemat waktu juga hemat tenaga dan meminimalisir kemungkinan kehabisan tiket.



Gambar 5.2 Penerapan Sistem pada Website



Gambar 5.3 Penerapan Sistem pada Mobile

BAB 6

ANALISIS

Setelah memahami apa itu sistem sebagai mana dijelaskan pada Bab IV. Selanjutnya pada bab ini akan memperkenalkan siklus hidup pengembangan sistem (SDLC), model empat fase dasar (perencanaan, analisis, desain, dan implementasi) yang umum untuk semua proyek pengembangan sistem informasi. Ini menjelaskan evolusi metodologi pengembangan sistem dan membahas peran dan keterampilan yang dibutuhkan seorang analis sistem. Bab ini kemudian meninjau karakteristik dasar sistem berorientasi objek dan dasar-dasar analisis dan desain sistem berorientasi objek dan ditutup dengan deskripsi Proses Terpadu dan serta ekstensi *Unified Modeling Language*.

6.1 *Sistem Analisis*

Sistem analisis adalah proses mengumpulkan dan menafsirkan fakta, mengidentifikasi masalah, dan penguraian suatu sistem menjadi komponen-komponennya. Analisis sistem dilakukan untuk tujuan mempelajari sistem atau bagian-bagiannya untuk mengidentifikasi tujuannya. Ini adalah teknik pemecahan masalah yang meningkatkan sistem dan memastikan bahwa semua komponen sistem bekerja secara efisien untuk mencapai tujuannya. Analisis menentukan apa yang harus dilakukan sistem.

By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

Sistem analisis merupakan profesi yg menantang karena menggabungkan banyak keahlian seperti keahlian analisis,teknis,interpersonal,dan manajerial.Hal ini bisa di lihat dari tanggung jawab seorang analis berdasarkan pendekatan Analisis Sistem yg meliputi:

1. Bagaimana membangun sistem informasi
2. Bagaimana menganalisis kebutuhan diri sistem informasi
3. Bagaimana membuat sistem informasi berbasis komputer
4. Bagaimana memecahkan masalah dalam organisasi melalui sistem informasi

Beberapa ahli membagi proses-proses pengembangan sistem ke dalam sejumlah urutan yg berbeda-beda. Tetapi semuanya akan mengacu pada proses-proses standar berikut:

1. Analisis
2. Desain
3. Implementasi
4. Pemeliharaan

Pada perkembangannya,proses-proses standar tadi di tuangkan dalam satu metode yg di kenal dengan nama *Systems Development Lifle Cycle* (SDLC) yg merupakan metodologi umum dalam pengembangan sistem yg menandai kemajuan usaha analisis dan desain.

6.2 *Systems Development Life Cycle (SDLC)*

Daur hidup pengembangan sistem (SDLC) adalah model konseptual yang digunakan dalam manajemen proyek yang menggambarkan tahapan yang terlibat dalam proyek pengembangan sistem informasi, dari studi kelayakan awal hingga pemeliharaan aplikasi yang lengkap. SDLC dapat diterapkan untuk sistem teknis dan nonteknis. Dalam kebanyakan kasus penggunaan, sistem adalah teknologi IT seperti perangkat keras dan perangkat lunak. Manajer proyek dan program biasanya mengambil bagian dalam SDLC, bersama dengan insinyur sistem dan perangkat lunak, tim pengembangan dan pengguna akhir.

Setiap perangkat keras atau sistem perangkat lunak akan melalui proses pengembangan yang dapat dianggap sebagai proses berulang dengan beberapa langkah. SDLC digunakan untuk memberikan struktur dan kerangka kerja yang kaku untuk menentukan fase dan langkah-langkah yang terlibat dalam pengembangan sistem.

SDLC juga merupakan kependekan dari *Synchronous Data Link Control* dan siklus hidup pengembangan perangkat lunak. Siklus hidup pengembangan perangkat lunak adalah proses yang sangat mirip dengan siklus hidup pengembangan sistem, tetapi berfokus secara eksklusif pada siklus hidup pengembangan perangkat lunak.

6.2.1 SDLC *Models*

Berbagai metodologi SDLC telah dikembangkan untuk memandu proses yang terlibat, termasuk metode SDLC asli, model air terjun. Model SDLC lainnya termasuk pengembangan aplikasi cepat (RAD), pengembangan aplikasi bersama (JAD), model air mancur, model spiral, bangun dan pasang, dan sinkronisasi-dan-stabilkan. Model umum lainnya saat ini disebut pengembangan perangkat lunak Agile.

Seringkali, beberapa model digabungkan ke dalam metodologi hybrid. Banyak dari model ini dibagi dengan pengembangan perangkat lunak, seperti air terjun atau gesit. Berbagai kerangka kerja model dapat disesuaikan agar sesuai dengan pengembangan perangkat lunak.

Dalam SDLC, dokumentasi sangat penting, terlepas dari jenis model yang dipilih untuk aplikasi apa pun, dan biasanya dilakukan bersamaan dengan proses pengembangan. Beberapa metode bekerja lebih baik untuk jenis proyek tertentu, tetapi dalam analisis akhir, faktor yang paling penting untuk keberhasilan suatu proyek mungkin adalah seberapa dekat rencana tertentu itu diikuti.

6.2.2 Langkah-langkah di SDLC

SDLC dapat terdiri dari beberapa langkah. Tidak ada sejumlah langkah konkret yang terlibat. Sekitar tujuh atau delapan langkah tampak umum; Namun, bisa ada di mana saja dari lima ke atas hingga 12. Biasanya, semakin banyak langkah didefinisikan dalam model SDLC, semakin granular tahapannya.

Secara umum, metodologi SDLC mengikuti langkah-langkah berikut:

1. Analisis: Sistem yang ada dievaluasi. Kekurangan diidentifikasi. Ini dapat dilakukan dengan mewawancarai pengguna sistem dan berkonsultasi dengan personel pendukung.

2. Rencana dan persyaratan: Persyaratan sistem baru didefinisikan. Secara khusus, kekurangan dalam sistem yang ada harus ditangani dengan proposal spesifik untuk perbaikan. Faktor-faktor lain yang didefinisikan termasuk fitur yang dibutuhkan, fungsi dan kemampuan.

3. Desain: Sistem yang diusulkan dirancang. Rencana disusun mengenai konstruksi fisik, perangkat keras, sistem operasi, pemrograman, komunikasi dan masalah keamanan.

4. Pengembangan: Sistem baru dikembangkan. Komponen dan program baru harus diperoleh dan diinstal. Pengguna sistem harus dilatih dalam penggunaannya.

5. Pengujian: Semua aspek kinerja harus diuji. Jika perlu, penyesuaian harus dilakukan pada tahap ini. Pengujian yang dilakukan oleh tim jaminan kualitas (QA) dapat mencakup integrasi sistem dan pengujian sistem.

6. Penempatan: Sistem ini tergabung dalam lingkungan produksi. Ini bisa dilakukan dengan berbagai cara. Sistem baru dapat secara bertahap masuk, sesuai

dengan aplikasi atau lokasi, dan sistem lama secara bertahap diganti. Dalam beberapa kasus, mungkin lebih hemat biaya untuk mematikan sistem lama dan mengimplementasikan sistem baru sekaligus.

7. Pemeliharaan dan pemeliharaan: Langkah ini melibatkan mengubah dan memperbarui sistem setelah itu ada. Perangkat keras atau perangkat lunak mungkin perlu ditingkatkan, diganti, atau diubah sedemikian rupa agar lebih sesuai dengan kebutuhan pengguna akhir secara berkelanjutan. Pengguna sistem harus selalu terbarui tentang modifikasi dan prosedur terbaru.

Langkah-langkah lain yang mungkin muncul termasuk inisiasi proyek, spesifikasi fungsional, spesifikasi terperinci, evaluasi, akhir masa pakainya dan langkah-langkah lain yang dapat dibuat dengan memisahkan langkah sebelumnya lebih jauh.

6.2.3 Keuntungan dan Kerugian SDLC

Manfaat mematuhi model SDLC yang jelas termasuk:

- Memiliki pandangan yang jelas tentang keseluruhan proyek, pekerja yang terlibat, perkiraan biaya dan jadwal.
- Memberikan manajer proyek proyeksi biaya pokok proyek.
- Tujuan dan standar didefinisikan dengan jelas.
- Pengembang dapat mundur selangkah jika sesuatu tidak berjalan seperti yang diharapkan.

Selain memiliki keuntungan, SDLC juga memiliki Kerugian yang mencakup:

- Karena asumsi yang dibuat pada awal proyek, jika keadaan tak terduga mempersulit pengembangan sistem, maka mungkin menimbun lebih banyak komplikasi di jalan. Sebagai contoh, jika perangkat keras yang baru diinstal tidak berfungsi dengan benar, maka itu dapat meningkatkan waktu pengembangan sistem, sehingga meningkatkan biaya.
- Beberapa metode tidak fleksibel.
- Mungkin rumit untuk memperkirakan biaya keseluruhan pada awal proyek.
- Pengujian di akhir pengembangan dapat memperlambat beberapa tim pengembangan.

6.3 *Systems Development Methodologies (Processes)*

”Sistem informasi dikembangkan untuk tujuan yang berbeda tergantung pada kebutuhan bisnis” (Kendall dan Kendall, 2002). Lingkungan bisnis, persyaratan teknis dan keadaan sosial dan politik seringkali sangat berbeda dan apa yang berfungsi

dalam satu organisasi mungkin tidak cocok untuk yang lain. Akibatnya, tidak ada pendekatan tunggal terbaik untuk pengembangan sistem. Masalah kecil bahkan mungkin tidak memerlukan proses formal yang mahal, namun, masalah yang lebih besar dan lebih kompleks akan membutuhkan proses yang terdefinisi dengan baik.

6.3.1 SSADM

Structured Systems Analysis and Design Method atau Metode Analisis dan Desain Sistem Terstruktur (SSADM) adalah metode pengembangan terstruktur yang dikembangkan pada awal 1980-an sebagai metode pengembangan standar domain publik (Beynon-Davies, 2009). SSADM berfokus pada aspek kelayakan, analisis dan desain siklus hidup pengembangan sistem. Ini memberikan pedoman lebih sedikit pada aspek pergantian dan pemeliharaan proyek IS. Menjelaskan SSADM dalam beberapa detail menyoroti pendekatan metodis yang diperlukan untuk proyek-proyek skala besar yang beberapa orang sebut sebagai birokrasi. Ini juga menggambarkan kontras dengan teknik alternatif seperti RAD.

6.3.1.1 Langkah-langkah SSADM

SSADM memiliki kerangka kerja lima modul di mana ada tujuh tahap, yaitu:

1. Studi kelayakan
   Proyek sudah akan melalui tahap perencanaan atau inisiasi, sehingga perlu pada titik ini untuk menentukan apakah layak secara teknis dan ekonomis. Studi kelayakan dipecah menjadi empat langkah:

   (a) Mempersiapkan studi kelayakan dengan menilai ruang lingkup proyek;

   (b) Mendefinisikan masalah (apa yang seharusnya dilakukan oleh sistem baru yang tidak dilakukan oleh sistem sekarang);

   (c) Pilih opsi kelayakan terbaik dari yang tersedia (biasanya hingga lima opsi bisnis dan sejenisnya) sejumlah opsi teknis);

   (d) Menyusun laporan kelayakan, termasuk alasan untuk opsi yang dipilih.

   Output dari tahap ini, laporan kelayakan, sekarang menyediakan input untuk modul selanjutnya; Analisa Kebutuhan.

2. Analisa Kebutuhan
   Tahap ini sangat penting karena digunakan untuk memperoleh pemahaman penuh tentang apa yang dibutuhkan sistem baru. Setiap kesalahan atau kelalaian yang dilakukan pada tahap ini akan tercermin dalam sisa proses pengembangan sistem. Berikut langkah-langkah yang diambil:

   (a) *Establish analysis framework*. Ruang lingkup proyek dinilai kembali dan kemudian direncanakan sesuai.

   (b) *Investigate and define requirements.*. Persyaratan luas akan ditentukan pada tahap kelayakan.

(c) *Investigate current processing*. Studi kelayakan akan membuat diagram alir data awal yang sekarang diperluas untuk mencakup semua proses yang ada.

(d) *Investigate current data*. Model data logis dikembangkan sehingga organisasi dapat memperoleh gambaran yang jelas tentang atribut mana yang dimiliki entitas data dan bagaimana mereka saling berhubungan.

(e) *Derive logical view of current services*. Ini melibatkan revisi model data logis sehingga mencerminkan logika bisnis dari sistem yang dipertimbangkan daripada implementasi fisik saat ini.

(f) *Assemble investigation results*. Ini adalah langkah terakhir dalam analisis lingkungan sistem saat ini. Analis akan memeriksa konsistensi dan kelengkapan sebelum melanjutkan ke tahap berikutnya.

Sejumlah solusi sistem yang mungkin untuk persyaratan bisnis yang dirasa dirumuskan dan dampak serta manfaat masing-masing akan dievaluasi. Solusi yang dipilih akan menjadi yang paling sesuai dengan persyaratan bisnis. Dua langkah itu adalah:

(a) Menentukan opsi sistem bisnis. Kegiatan di sini akan mencakup penetapan persyaratan sistem minimum, pengembangan dalam bentuk kerangka alternatif, pembuatan daftar pendek opsi, dan akhirnya evaluasi penuh dari setiap alternatif pilihan-pilihan, termasuk analisis biaya-manfaat, analisis dampak dan pengembangan sistem dan rencana integrasi untuk masing-masing.

(b) Pilih opsi sistem bisnis. Cara yang tepat dalam melakukan ini akan bervariasi di antara berbagai organisasi. Tujuannya sama, namun: untuk manajer pengguna yang tepat untuk memilih opsi sistem bisnis dari bukti yang disajikan oleh tim analisis.

3. Spesifikasi kebutuhan
Modul ini memiliki satu tahap yang pada gilirannya dibagi menjadi delapan langkah terpisah.

(a) Menentukan pemrosesan sistem yang diperlukan. Di sini, fitur-fitur sistem yang ada yang akan tetap menjadi bagian dari sistem baru ditambahkan ke rincian yang terkandung dalam katalog persyaratan.

(b) Mengembangkan model data yang diperlukan. Elemen redundan dari model data sistem yang ada dihapus (jika ada) dan elemen tambahan yang diperlukan ditambahkan. Selain itu, hubungan antara entitas lama dan baru ditinjau.

(c) Turunkan fungsi sistem. Di sini, proses yang akan diidentifikasi dan dimasukkan dalam diagram aliran data diidentifikasi lebih tepat dan didokumentasikan dengan baik.

(d) Meningkatkan model data yang diperlukan. Model data yang diperlukan yang dikembangkan sebelumnya sekarang ditingkatkan dengan melakukan analisis dan normalisasi data relasional; hasilnya harus berupa seperangkat

tabel yang dapat diimplementasikan menggunakan sistem manajemen basis data relasional.

(e) Mengembangkan prototipe spesifikasi. Ini melibatkan pembuatan prototipe untuk bagian tertentu dari spesifikasi sehingga persyaratan yang tepat dapat divalidasi dengan pengguna akhir yang dituju; elemen seperti menu, contoh layar entri data dan laporan dapat dibangun.

(f) Mengembangkan spesifikasi pemrosesan. Analis pada tahap ini prihatin dengan menggambarkan pengaruh waktu pada data yang dikenakan berbagai tindakan (mis. Penciptaan, membaca, memperbarui, dan menghapus); dua alat yang digunakan di sini adalah analisis riwayat hidup entitas dan diagram korespondensi efek. Ini adalah alat yang digunakan oleh profesional analis sistem dan berada di luar cakupan buku ini untuk membahasnya secara terperinci.

(g) Konfirmasikan tujuan sistem. Tugas kedua dari belakang adalah untuk melakukan tinjauan formal dari persyaratan sistem untuk memastikan bahwa spesifikasi persyaratan akhir yang berikut lengkap dan sepenuhnya dipahami oleh pengguna dan pengembang.

(h) Merakit spesifikasi persyaratan. Akhirnya, berbagai komponen (termasuk model data logis sistem yang diperlukan, definisi fungsi, katalog persyaratan, dan item lainnya) dirakit ke dalam dokumen spesifikasi persyaratan akhir, yang kemudian memberikan input ke modul dan tahap berikutnya.

4. Spesifikasi Logika Sistem
   Di sini, segala kendala pada pilihan lingkungan teknis ditetapkan (mis. Keamanan, kinerja, kemudahan upgrade). Opsi teknis yang sesuai dipilih; harus sesuai dengan kriteria strategis dan operasional yang diperlukan sudah ditetapkan. Proses pengembangan spesifikasi sistem dilanjutkan, dengan hasilnya satu set komponen yang dapat diterapkan. Langkah-langkah individual adalah sebagai berikut:

   (a) *Define user dialogues*. Ini berkaitan dengan mendefinisikan cara-cara di mana pengguna akan berinteraksi dengan sistem (mis. Menu dan navigasi sistem).

   (b) *Define update processes*. Di sini, definisi transaksi yang akan mengubah data ditetapkan (riwayat kehidupan entitas digunakan untuk mendukung langkah ini).

   (c) *Define enquiry processes*. Selain navigasi dan pembaruan, pengguna akan ingin melakukan pertanyaan tentang data yang disimpan dalam sistem.

   (d) *Assemble logical design*. Ini pada dasarnya adalah pemeriksaan konsistensi dan kelengkapan. Setelah desain logis selesai dan telah 'ditandatangani', tahap akhir dapat diatasi.

5. Desain fisik
   Tahap ini berkaitan dengan pengiriman cetak biru akhir dari mana sistem dapat

dikembangkan dan diimplementasikan. Ada tujuh langkah yang harus diselesaikan:

(a) *Prepare for physical design*. Lingkungan implementasi dipelajari, standar pengembangan aplikasi disusun dan strategi desain fisik disepakati.

(b) *Create physical data design*. *Logical Data Model* (LDM) yang diperlukan digunakan sebagai dasar untuk ini dan desain data khusus bisnis dihasilkan.

(c) *Create function component implementation map*. Komponen dari masing-masing fungsi sistem disusun. Ini termasuk hubungan mereka dengan komponen fungsi fisik (kegiatan bisnis aktual) yang mereka dukung.

(d) *Optimise physical data design*. Desain data fisik diuji terhadap sasaran kinerja yang diperlukan dan dioptimalkan jika perlu.

(e) *Complete function specification design*. Ini akan untuk setiap komponen fungsi yang membutuhkan pemrograman.

(f) *Consolidate process data interface*. Antarmuka data proses terletak antara desain database fisik dan desain proses. Ini membantu pemetaan database dengan persyaratan pemrosesan (terutama penting ketika database telah diubah atau persyaratan pemrosesan telah dimodifikasi).

(g) *Assemble physical design*. Tahap ini dan seluruh siklus hidup SSADM selesai dengan langkah ini. Sejumlah produk dikirimkan, termasuk definisi fungsi, desain data fisik yang dioptimalkan, katalog persyaratan, serta perkiraan ruang dan waktu.

6.3.1.2 Keuntungan dan Kekurangan SSADM

Keuntungan dari SSADM adalah sebagai berikut:

- Garis Waktu: Secara teoritis, SSADM memungkinkan seseorang untuk merencanakan, mengelola, dan mengendalikan proyek dengan baik. Poin-poin ini penting untuk mengantarkan produk tepat waktu.

- Kegunaan: Dalam SSADM penekanan khusus diberikan pada analisis kebutuhan pengguna. Secara bersamaan, model sistem dikembangkan dan analisis permintaan yang komprehensif dilakukan. Keduanya dicoba untuk melihat apakah mereka cocok satu sama lain.

- Menanggapi perubahan dalam lingkungan bisnis: Seperti dalam dokumentasi SSADM tentang kemajuan proyek ditanggapi dengan sangat serius, masalah-masalah seperti tujuan bisnis dan kebutuhan bisnis dipertimbangkan ketika proyek sedang dikembangkan. Ini menawarkan kemungkinan untuk menyesuaikan perencanaan proyek dengan persyaratan aktual bisnis.

- Penggunaan keterampilan secara efektif: SSADM tidak memerlukan keterampilan yang sangat khusus dan dapat dengan mudah diajarkan kepada staf. Biasanya, alat pemodelan dan diagram yang umum digunakan. Alat CASE komersial juga ditawarkan untuk dapat mengatur SSADM dengan mudah.

- Kualitas yang lebih baik: SSADM mengurangi tingkat kesalahan IS dengan mendefinisikan tingkat kualitas tertentu di awal dan secara konstan memeriksa sistem.

- Peningkatan produktivitas: Dengan mendorong pengiriman yang tepat waktu, memenuhi persyaratan bisnis, memastikan kualitas yang lebih baik, menggunakan sumber daya manusia secara efektif serta berusaha menghindari birokrasi, SSADM meningkatkan produktivitas keseluruhan proyek spesifik dan perusahaan.

- Biaya pemotongan: SSADM memisahkan desain sistem logis dan fisik. Jadi sistem tidak harus diimplementasikan lagi dengan perangkat keras baru atau perangkat lunak.

SSADM memberikan penekanan khusus pada analisis sistem dan dokumentasinya. Ini menyebabkan bahaya analisis yang berlebihan, yang bisa sangat memakan waktu dan biaya. Karena berbagai jenis metode deskripsi, pemeriksaan konsistensi tidak dapat dilakukan. Terutama dengan sistem besar, diagram garis besar dapat menjadi sangat tidak jelas, karena semua aliran data yang relevan harus dimasukkan.

Namun, perusahaan besar yang melakukan berbagai proyek dapat mengambil untung dari fakta bahwa SSADM memberikan kemungkinan untuk menggunakan kembali teknik dan alat tertentu untuk proyek lain. Ini mengurangi biaya dan waktu yang dihabiskan sangat besar dalam jangka panjang. Jadi, bahaya menghabiskan terlalu banyak uang untuk analisis dapat dikompensasi dengan menggunakan kembali sistem yang dikembangkan dan pengalaman yang diperoleh.

6.3.2 *Rapid Application Design* (RAD)

Rapid Application Development adalah model pengembangan yang memprioritaskan pembuatan prototipe cepat dan umpan balik cepat selama siklus pengembangan dan pengujian yang lama berlarut-larut. Dengan pengembangan aplikasi yang cepat, pengembang dapat membuat beberapa iterasi dan pembaruan ke perangkat lunak dengan cepat tanpa harus memulai jadwal pengembangan dari awal setiap kali.

RAD adalah model pengembangan yang muncul begitu pengembang menyadari model pengembangan *waterfall* tradisional tidak terlalu efektif.

Kelemahan utama dalam model *waterfall* adalah bahwa begitu program berada dalam fase pengujian, menjadi sulit untuk mengubah fungsi inti dan fitur perangkat lunak. Pada dasarnya meninggalkan Anda dengan perangkat lunak yang mungkin cocok atau tidak sesuai dengan kebutuhan Anda yang berkembang.

Rapid Application Development (RAD) disusun pada tahun 1980-an, jadi ini jelas bukan sesuatu yang baru. Tetapi tidak seperti model *waterfall*, yang tidak tunggal. Ini merupakan sebuah evolusi berkelanjutan dari filosofi pembangunan sesuai dengan kebutuhan pada waktu tertentu.

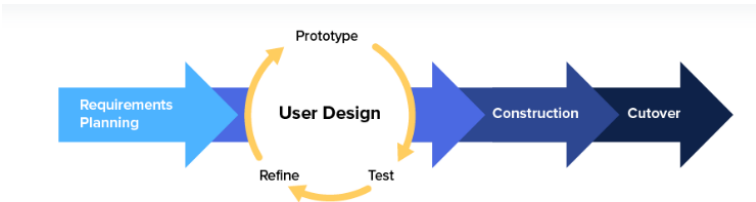
Awalnya, Barry Boehm, James Martin, dan sejumlah lainnya melihat bahwa perangkat lunak tidak terbatas pada metode teknik tradisional. Itu bukan sumber daya tunggal

yang membutuhkan struktur tetap. Itu mudah dibentuk sesuai dengan kebutuhan pengguna.

Awalnya, pengembangan *Rapid Application Development* mengambil bentuk model Spiral, di mana satu atau lebih model pengembangan digunakan untuk bekerja pada proyek tertentu.

Seiring berjalannya waktu, pengembangan aplikasi yang cepat berubah. Itu dibentuk sendiri agar sesuai dengan persyaratan waktu sambil mempertahankan beberapa pedoman pengembangan inti.

6.3.2.1 5 langkah atau fase dalam RAD



Gambar 6.1 Rapid Application Development

Meskipun RAD telah berubah selama bertahun-tahun, keempat langkah dasar ini memberikan kesinambungan selama bertahun-tahun.

1. *Define the Requirements*
   Pada awalnya, pengembangan aplikasi yang cepat membedakan dirinya dari model pengembangan perangkat lunak tradisional. Anda tidak perlu duduk dengan pengguna akhir dan mendapatkan daftar spesifikasi yang terperinci; sebaliknya, ia meminta persyaratan yang luas. Sifat luas dari persyaratan membantu Anda memberikan persyaratan khusus di berbagai titik siklus pengembangan.

2. *Prototype*
   Di sinilah perkembangan aktual terjadi. Alih-alih mengikuti serangkaian persyaratan yang ketat, pengembang membuat prototipe dengan berbagai fitur dan fungsi secepat mungkin. Prototipe ini kemudian ditampilkan kepada klien yang memutuskan apa yang mereka sukai dan apa yang tidak.
   Lebih sering daripada tidak, prototipe ini dengan cepat dibuat untuk bekerja, hanya untuk memamerkan fitur-fitur tertentu, tanpa semir yang tepat. Ini normal, dan produk akhir hanya dibuat selama tahap finalisasi di mana klien dan pengembang dapat menyetujui produk akhir.

3. *Receive Feedback*
   Pada tahap ini, *feedback* atau umpan balik tentang apa yang baik, apa yang tidak, apa yang berhasil, dan apa yang tidak dibagikan. Umpan balik tidak terbatas hanya pada fungsionalitas murni, tetapi juga visual dan antarmuka. Den-

gan umpan balik ini dalam pikiran, prototyping berlanjut. Kedua langkah ini diulangi hingga produk akhir dapat direalisasikan yang sesuai dengan persyaratan pengembang dan klien.

4. *Finalize Software*
   Di sini, fitur, fungsi, estetika, dan antarmuka perangkat lunak diselesaikan dengan klien. Stabilitas, kegunaan, dan pemeliharaan adalah yang paling penting sebelum dikirim ke klien.

6.3.2.2 Keuntungan dan Kerugian RAD

Dengan langkah-langkah ini, sepertinya pengembangan aplikasi merupakan ide bagus untuk semua proyek, tetapi itu sangat sulit. Perangkat lunak RAD sangat bagus untuk tim kecil dan proyek cepat. Tapi itu bukan solusi untuk semuanya. Berikut adalah beberapa keuntungan dan kerugian menggunakan pengembangan aplikasi yang cepat.

Berikut beberapa keuntungan manggunakan RAD:

1. Persyaratan dapat diubah kapan saja.
2. Mendorong dan memprioritaskan umpan balik pelanggan.
3. Ulasan cepat
4. Waktu pengembangan berkurang drastis.
5. Lebih banyak produktivitas dengan lebih sedikit orang.
6. Waktu antara prototipe dan iterasi adalah singkat.
7. Integrasi bukan masalah, karena terintegrasi dari awal proyek

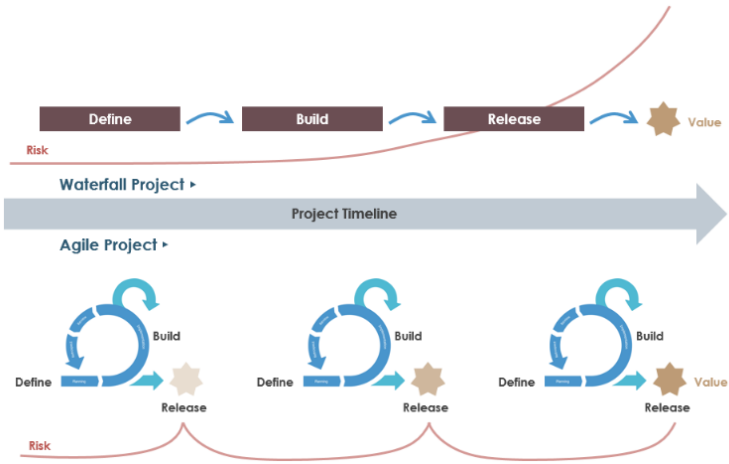
Selain memiliki kelebihan, RAD juga memiliki kekurangan diantaranya sebagai berikut:

1. Membutuhkan kolaborasi tim yang kuat.
2. Tidak dapat bekerja dengan tim besar.
3. Membutuhkan pengembang yang sangat terampil.
4. Membutuhkan kebutuhan pengguna di seluruh siklus hidup produk.
5. Hanya cocok untuk proyek yang memiliki waktu pengembangan kecil.
6. Lebih rumit untuk dikelola jika dibandingkan dengan model lain.
7. Hanya sistem yang dapat dimodulasi yang dapat dikembangkan menggunakan pengembangan aplikasi yang cepat.

6.3.3 *Agile Development*

Agile adalah istilah yang digunakan untuk menggambarkan pendekatan pengembangan perangkat lunak yang menekankan pengiriman bertahap, kolaborasi tim, perencanaan berkelanjutan, dan pembelajaran berkelanjutan, alih-alih mencoba menyampaikannya sekaligus menjelang akhir. Agile berfokus pada menjaga proses ramping dan menciptakan produk-produk minimum yang layak (MVP) yang melewati sejumlah iterasi sebelum segala sesuatu bersifat final. Umpan balik dikumpulkan dan diimplementasikan secara terus-menerus dan secara keseluruhan, ini adalah proses yang jauh lebih dinamis di mana setiap orang bekerja bersama menuju satu tujuan.

Metodologi Agile adalah praktik yang mempromosikan iterasi pengembangan dan pengujian yang berkelanjutan di seluruh siklus pengembangan perangkat lunak proyek. Aktivitas pengembangan dan pengujian bersamaan tidak seperti model Waterfall.



Gambar 6.2 Agile Software Development

Pengembangan perangkat lunak tangkas menekankan pada empat nilai inti.

1. Interaksi individu dan tim atas proses dan alat
2. Bekerja perangkat lunak melalui dokumentasi yang komprehensif
3. Kolaborasi pelanggan atas negosiasi kontrak
4. Menanggapi perubahan setelah mengikuti rencana

Ada berbagai metode yang ada dalam pengujian agile, dan yang tercantum di bawah ini:

1. Scrum
   Scrum adalah kerangka kerja di mana orang dapat mengatasi masalah adap-

tif yang kompleks, sementara secara produktif dan kreatif memberikan produk dengan nilai setinggi mungkin. Ini digunakan untuk mengelola proyek perangkat lunak dan pengembangan produk atau aplikasi. Fokusnya adalah pada strategi pengembangan produk adaptif di mana tim lintas fungsi bekerja sebagai unit untuk mencapai tujuan bersama dalam 2-4 minggu (Sprint). Ini terdiri dari kumpulan nilai, artefak, peran, upacara, aturan, dan praktik terbaik.

2. Lean
   Lean berasal dari Toyota Production System, atau TPS, yang merevolusi pembuatan barang fisik pada 1950-an, 60-an, dan seterusnya. Lean mempertahankan cengkeramannya di bidang manufaktur tetapi juga telah menemukan aplikasi baru dalam pekerjaan pengetahuan, membantu bisnis di semua industri menghilangkan limbah, meningkatkan proses, dan mendorong inovasi . Pengembangan perangkat lunak adalah aplikasi alami dari metodologi Lean karena, seperti halnya manufaktur, umumnya mengikuti proses yang ditentukan, memiliki beberapa kondisi penerimaan yang ditentukan, dan menghasilkan pengiriman nilai nyata. Konsep kunci yang memandu semua praktik metodologi Lean, yang kita sebut *Pillars of Lean*, yaitu:

   - *Continuous improvement*
   - *Respect for people*
   - *Lightweight Leadership*

3. Kanban
   Kanban adalah metode manajemen alur kerja yang sangat visual yang populer di antara tim Lean. Faktanya, 83% tim yang berlatih Lean menggunakan Kanban untuk memvisualisasikan dan secara aktif mengelola penciptaan produk dengan penekanan pada pengiriman yang berkelanjutan, sementara tidak membebani tim pengembang. Seperti Scrum, Kanban adalah proses yang dirancang untuk membantu tim bekerja sama secara lebih efektif.

4. Dynamic Systems Development Method (DSDM)
   DSDM adalah kerangka kerja yang terdiri dari delapan prinsip, siklus hidup dan produk, peran dan tanggung jawab dan beberapa teknik praktik terbaik. Ini menopang dan mendukung filosofi memberikan manfaat bisnis yang selaras secara strategis sedini mungkin untuk memberi organisasi pengembalian laba investasi (ROI) terbaik.

5. Extreme Programming (XP)
   Extreme Programming (XP), awalnya digambarkan oleh Kent Beck, telah muncul sebagai salah satu metodologi Agile paling populer dan kontroversial. XP adalah pendekatan disiplin untuk memberikan perangkat lunak berkualitas tinggi dengan cepat dan terus menerus. Hal ini dimaksudkan untuk meningkatkan kualitas perangkat lunak dan responsif dalam menghadapi perubahan kebutuhan pelanggan. Ini mempromosikan keterlibatan pelanggan yang tinggi, loop umpan balik yang cepat, pengujian berkelanjutan, perencanaan berkelanjutan, dan kerja tim

yang erat untuk memberikan perangkat lunak yang bekerja pada interval yang sangat sering, biasanya setiap 1-3 minggu.
Metodologi ini mengambil namanya dari gagasan bahwa elemen menguntungkan dari praktik rekayasa perangkat lunak tradisional dibawa ke tingkat ”ekstrem”. Sebagai contoh, tinjauan kode dianggap sebagai praktik yang bermanfaat. Dibawa ke ekstrim, kode dapat ditinjau terus menerus melalui praktik pemrograman pasangan. Metode XP asli didasarkan pada empat nilai sederhana - kesederhanaan, komunikasi, umpan balik, dan keberanian.

6. Feature Driven Development (FDD)
Feature-Driven Development (FDD) diperkenalkan pada tahun 1997 oleh Jeff De Luca ketika ia bekerja dalam proyek pengembangan perangkat lunak untuk bank besar Singapura. Ini adalah proses pengembangan perangkat lunak iteratif dan tambahan dan merupakan metode tangkas untuk mengembangkan perangkat lunak. FDD memadukan sejumlah praktik terbaik yang diakui industri menjadi satu kesatuan yang utuh. Praktik-praktik ini didorong dari perspektif fungsionalitas (fitur) yang dihargai klien. Tujuan utamanya adalah untuk memberikan perangkat lunak yang nyata dan berfungsi berulang kali secara tepat waktu. Keuntungan menggunakan FDD adalah scalable bahkan untuk tim besar karena konsep ’awalnya cukup desain’ (JEDI). Ini adalah solusi hebat untuk mempertahankan kontrol atas proyek yang gesit, bertahap, dan inheren karena prosesnya yang mengutamakan fitur. Ini terdiri dari lima kegiatan dasar:

   - Pengembangan model keseluruhan
   - Membangun daftar fitur
   - Perencanaan berdasarkan fitur
   - Merancang berdasarkan fitur
   - Bangunan berdasarkan fitur.

7. Crystal
Metode kristal adalah keluarga metodologi (keluarga Crystal) yang dikembangkan oleh Alistair Cockburn pada pertengahan 1990-an. Metode berasal dari studi bertahun-tahun dan wawancara tim oleh Cockburn. Penelitian Cockburn menunjukkan bahwa tim yang dia wawancarai tidak mengikuti metodologi formal namun mereka masih berhasil menyelesaikan proyek. Keluarga Crystal adalah cara Cockburn mengkatalogkan apa yang mereka lakukan yang membuat proyek berhasil. Metode kristal difokuskan pada:

   - Orang-orang
   - Interaksi
   - Masyarakat
   - Keterampilan
   - Bakat
   - Komunikasi

6.3.4 Memilih Metodologi Pengembangan yang Tepat

Karena ada banyak metodologi, tantangan pertama yang dihadapi oleh analis adalah memilih metodologi yang akan digunakan. Memilih metodologi tidak mudah, karena tidak ada satu metodologi yang selalu terbaik. Banyak organisasi memiliki standar dan kebijakan untuk memandu pilihan metodologi. Anda akan menemukan bahwa organisasi berkisar dari memiliki satu metodologi ”disetujui”, hingga memiliki beberapa opsi metodologi, hingga tidak memiliki kebijakan formal sama sekali.

Kemampuan Mengembangkan Sistem	Metodologi Terstruktur	Metodologi RAD	Metodologi Agile
	Waterfall	Prototyping	XP
Persyaratan Pengguna Tidak Jelas	Buruk	Sangat Baik	Sangat Baik
Teknologi tidak dikenal	Buruk	Buruk	Buruk
Kompleksitas	Baik	Buruk	Buruk
Keandalan	Baik	Buruk	Baik
Waktu yang singkat	Buruk	Sangat Baik	Sangat Baik
Visibilitas	Buruk	Sangat Baik	Baik

Tabel 6.1 Perbandingan Metodologi

- *Clarity of User Requirements*
  Ketika persyaratan pengguna untuk apa yang harus dilakukan sistem tidak jelas, sulit untuk memahaminya dengan membicarakannya dan menjelaskannya dengan laporan tertulis. Pengguna biasanya perlu berinteraksi dengan teknologi untuk benar-benar memahami apa yang dapat dilakukan sistem baru dan cara terbaik menerapkannya untuk kebutuhan mereka. Metodologi RAD berbasis prototipe dan sekali pakai biasanya lebih sesuai ketika persyaratan pengguna tidak jelas karena mereka menyediakan prototipe bagi pengguna untuk berinteraksi dengan awal di SDLC.

- *Familiarity with Technology*
  Saat sistem akan menggunakan teknologi baru yang tidak dikenal oleh para analis dan programmer (mis., Proyek pengembangan Web pertama dengan Java), aplikasi awal teknologi baru dalam metodologi akan meningkatkan peluang keberhasilan. Jika sistem dirancang tanpa mengenal teknologi dasar, risiko meningkat karena alat mungkin tidak mampu melakukan apa yang diperlukan. Metodologi berbasis prototipe Throwaway sangat sesuai untuk kurangnya pengetahuan tentang teknologi karena mereka secara eksplisit mendorong pengembang untuk mengembangkan prototipe desain untuk area dengan risiko tinggi.

Metodologi berbasis pengembangan bertahap juga bagus, karena mereka menciptakan peluang untuk menyelidiki teknologi secara mendalam sebelum desain selesai. Sementara orang mungkin berpikir metodologi berbasis prototyping juga akan sesuai, mereka jauh lebih sedikit, karena prototipe awal yang dibangun biasanya hanya menggores permukaan teknologi baru. Biasanya, hanya setelah beberapa prototipe dan beberapa bulan pengembang menemukan kelemahan atau masalah dalam teknologi baru.

- *System Complexity*
  Sistem yang kompleks membutuhkan analisis dan desain yang cermat dan terperinci. Metodologi berbasis prototyping Throwaway sangat cocok untuk analisis dan desain rinci, yang bertentangan dengan metodologi berbasis prototyping, yang tidak. Metodologi berbasis desain terstruktur tradisional dapat menangani sistem yang kompleks, tetapi tanpa kemampuan untuk mendapatkan sistem atau prototipe ke tangan pengguna sejak dini, beberapa masalah utama mungkin diabaikan. Meskipun metodologi berbasis pengembangan bertahap memungkinkan pengguna untuk berinteraksi dengan sistem di awal proses, kami telah mengamati bahwa tim proyek yang mengikuti ini cenderung kurang mencurahkan perhatian pada analisis domain masalah lengkap daripada mereka mungkin menggunakan yang lain.

- *System Reliability*
  Keandalan sistem biasanya merupakan faktor penting dalam pengembangan sistem lagipula, siapa yang menginginkan sistem yang tidak dapat diandalkan? Namun, keandalan hanyalah salah satu dari beberapa faktor. Untuk beberapa aplikasi, keandalan sangat penting (mis., Peralatan medis, sistem kendali rudal), sedangkan untuk aplikasi lain itu hanya penting (mis., Game, video Internet). Metodologi prototyping Throwaway adalah yang paling tepat ketika keandalan sistem adalah prioritas tinggi, karena menggabungkan analisis rinci dan fase desain dengan kemampuan tim proyek untuk menguji berbagai pendekatan melalui prototipe desain sebelum menyelesaikan desain. Metodologi prototipe umumnya bukan pilihan yang baik ketika keandalan sangat penting karena tidak memiliki analisis yang cermat dan fase desain yang penting untuk sistem yang dapat diandalkan.

- *Short Time Schedules*
  Proyek yang memiliki jadwal waktu singkat sangat cocok untuk metodologi berbasis RAD. Ini karena mereka dirancang untuk meningkatkan kecepatan pengembangan. Prototyping dan metodologi berbasis pengembangan adalah pilihan yang sangat baik ketika jadwal pendek karena mereka memungkinkan tim proyek untuk menyesuaikan fungsi dalam sistem berdasarkan pada tanggal pengiriman tertentu, dan jika jadwal proyek mulai tergelincir, dapat disesuaikan kembali dengan menghapus fungsionalitas dari versi atau prototipe dalam pengembangan. Metodologi berbasis air terjun adalah pilihan terburuk ketika waktu sangat mahal karena tidak memungkinkan untuk perubahan jadwal yang mudah.

- *Schedule Visibility*
  Salah satu tantangan terbesar dalam pengembangan sistem adalah menentukan apakah suatu proyek sesuai jadwal. Ini terutama berlaku untuk metodologi desain terstruktur karena desain dan implementasi terjadi pada akhir proyek. Metodologi berbasis RAD memindahkan banyak keputusan desain kritis sebelumnya dalam proyek untuk membantu manajer proyek mengenali dan mengatasi faktor risiko dan menjaga ekspektasi tetap terjaga.

6.4 *Business Analysis*

Analis bisnis membantu mengarahkan bisnis dalam meningkatkan proses, produk, layanan, dan perangkat lunak melalui analisis data. Para pekerja yang gesit mengangkangi garis antara IT dan bisnis untuk membantu menjembatani kesenjangan dan meningkatkan efisiensi.

Analis Bisnis (BA) bertanggung jawab untuk menjembatani kesenjangan antara TI dan bisnis menggunakan analitik data untuk menilai proses, menentukan persyaratan, dan memberikan rekomendasi dan laporan yang didorong data ke eksekutif dan pemangku kepentingan. BA terlibat dengan para pemimpin bisnis dan pengguna untuk memahami bagaimana perubahan yang didorong data untuk proses, produk, layanan, perangkat lunak, dan perangkat keras dapat meningkatkan efisiensi dan menambah nilai. Mereka harus mengartikulasikan ide-ide itu tetapi juga menyeimbangkannya dengan apa yang secara teknologi layak dan masuk akal secara finansial dan fungsional. Bergantung pada perannya yang memungkinkan bekerja dengan kumpulan data untuk meningkatkan produk, perangkat keras, alat, perangkat lunak, layanan atau proses.

Untuk membuat sebuah analisis bisnis, perlu adanya laporan analisis bisnis berupa alur kerja atau *workflow*. *A workflow diagram (or workflow)* adalah cara visual untuk analisis bisnis Anda untuk menunjukkan bagaimana pekerjaan dicapai. Alur kerja terdiri dari serangkaian simbol yang menunjukkan bagaimana berbagai pekerja menyelesaikan tugas dan berinteraksi satu sama lain, serta bagaimana informasi (data) mengalir melalui area bisnis. Diagram ini sudah ada sejak lama dan memiliki banyak variasi:

1. **Swimlane:** Diagram alur kerja ini berfokus pada interaksi antara unit-unit organisasi dan memperlihatkan kemacetan dan inefisiensi proses

2. **ANSI flowchart:** Gaya ini tumbuh dari flowcharting pada tahun 1970-an dan menjadi standar pertama untuk alur kerja. Simbol keluar dari American National Standards Institute (karenanya, ANSI). Banyak orang masih menggunakan simbol hari ini, sering di dalam swimlanes untuk menunjukkan pekerjaan sedang dilakukan. Misalnya, bagan alur ANSI dapat menunjukkan bagaimana permintaan liburan karyawan disetujui.

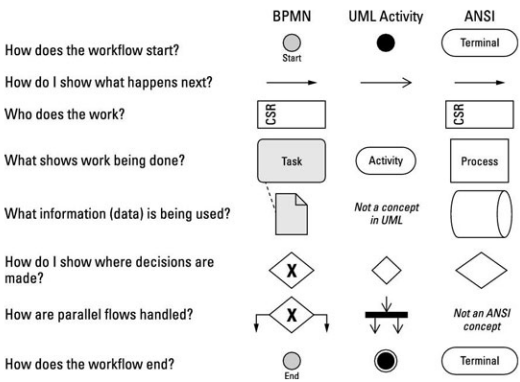
3. **UML Activity:** Diagram ini muncul dari UML *(Unified Modeling Language)*. Ini beroperasi seperti bagan alur dan menunjukkan langkah yang berbeda dan tugas yang dipesan serta aliran kontrol.

4. **BPMN:** *Business Process Modeling Notation* (BPMN) adalah standar yang dibuat untuk bisnis daripada untuk pengembangan aplikasi. Dengan kata lain, ini berfokus pada menyajikan proses bisnis dan informasi daripada solusi. Misalnya, diagram BPMN dapat menunjukkan proses persetujuan liburan yang dihamparkan pada semua agen eksternal dalam proses: kalender karyawan, manajer, dan liburan perusahaan.

5. **Geographic diagrams:** Diagram ini biasanya digunakan dalam pergudangan dan logistik untuk menunjukkan penempatan objek dalam kaitannya dengan pekerja. Misalnya, diagram geografis akan mengamanatkan bahwa produk yang paling jarang dipesan ditempatkan paling jauh dari dermaga pengiriman dan paling tinggi di rak dan bahwa produk yang paling umum dipesan ditempatkan paling dekat dengan stasiun pengiriman.

6. **SIPOC:** SIPOC adalah singkatan dari "*SupplierInputProcessOutputCustomer*." Digunakan terutama dalam Six Sigma, diagram ini menunjukkan siapa yang menciptakan data, merinci proses tingkat tinggi, dan menggambarkan siapa yang menerima data. Misalnya, perusahaan pengiriman (pemasok) memberikan nomor pelacakan (*input*) yang dilampirkan pada pesanan Anda (proses), dan *email* konfirmasi keluar (*output*) kepada Anda (pelanggan, konsumen).

Semua diagram ini menggunakan simbol untuk menunjukkan aliran proses saat bergerak melalui area bisnis dalam ruang lingkup proyek Anda. Setiap diagram menggunakan simbol yang berbeda, tetapi semuanya mendokumentasikan berbagai aspek tentang proses, seperti siapa yang melakukan pekerjaan, di mana hal itu dilakukan, data apa yang digunakan proses, di mana keputusan dibuat, siapa yang membuat keputusan, ketika proses dimulai, dan apa yang menyebabkan proses berakhir.



Gambar 6.3 Simbol Workflow

Berikut adalah hal-hal penting tentang cara membuat *workflow* diagram:

1. Tentukan titik pandang dan titik waktu Anda membuat diagram.
   Apakah Anda membuat diagram proses dari sudut pandang bisnis atau dari sudut pandang pelanggan? Apakah Anda menunjukkan cara Anda melakukan proses hari ini (disebut proses apa adanya) atau bagaimana Anda ingin melakukannya dengan implementasi baru (disebut proses menjadi). Jika Anda tidak tahu, tanyakan pemangku kepentingan Anda. Beri judul alur kerja dengan tepat.

2. Cari tahu di mana proses dimulai dan di mana itu berakhir.
   Langkah ini sangat penting karena merupakan titik awal untuk aktivitas proses Anda dan juga mengidentifikasi kapan itu berhasil (atau tidak berhasil) berakhir. Jika Anda tidak tahu - Anda dapat menebaknya - tanyakan kepada pemangku kepentingan Anda.

3. Tanyakan ”Apa yang terjadi selanjutnya?” Atau ”Lalu apa yang terjadi?” Pertanyaan dari pemangku kepentingan Anda sampai Anda memahami proses dari ujung ke ujung.
   Dokumentasikan setiap tugas dan hubungkan untuk membuat aliran (atau jalur) dari satu tugas ke tugas lainnya. Ketika bekerja dengan para pemangku kepentingan, teruslah menyusuri jalan yang paling mungkin (yang paling sering terjadi) terlebih dahulu. Kemudian kembali ke jalur alternatif (jalur yang tidak mengikuti jalur normal). Strategi ini membantu Anda menghindari terjebak dalam semua jalur ”bagaimana jika”.
   Saat Anda menemukan prosesnya, dengarkan kata-kata operatif seperti validasi atau setujui; itulah poin keputusan Anda. Ketika Anda menemukan kata-kata ini, Anda akan memiliki dua jalur keluar dari keputusan: satu untuk persetujuan dan satu untuk penolakan. Lanjutkan ke jalur yang paling umum pada titik ini, tetapi pastikan untuk kembali ke jalur pengecualian.

4. Pada poin keputusan, cari tahu informasi atau data apa yang dibutuhkan pekerja untuk membuat keputusan.
   Untuk setiap tugas, tanyakan tentang data yang mereka gunakan (atau baca, dalam bahasa data) atau hasilkan (buat). Informasi ini mengarahkan Anda ke persyaratan.

5. Pilih jenis *workflow*.
   Tidak ada solusi satu ukuran untuk semua yang berfungsi di sini. Pilihan Anda tergantung pada apa budaya perusahaan itu, apakah perusahaan telah bekerja dengan diagram alur kerja apa pun di masa lalu, dan opsi mana yang paling baik berkomunikasi dengan pemangku kepentingan mereka, serta pada pengalaman dan tingkat kenyamanan Anda sendiri dengan berbagai alur kerja.

6.5 BPMN

Business Process Modeling Notation (BPMN) adalah metode diagram alir yang memodelkan langkah-langkah proses bisnis yang direncanakan dari ujung ke ujung.

Kunci Manajemen Proses Bisnis, secara visual menggambarkan urutan rinci kegiatan bisnis dan arus informasi yang diperlukan untuk menyelesaikan suatu proses.

Tujuannya adalah untuk memodelkan cara-cara untuk meningkatkan efisiensi, memperhitungkan keadaan baru atau mendapatkan keunggulan kompetitif. Metode ini telah mengalami dorongan standardisasi dalam beberapa tahun terakhir dan sekarang sering disebut dengan nama yang sedikit berbeda: Model Proses Bisnis dan Notasi, masih menggunakan akronim BPMN. Ini berbeda dari *Unified Modeling Language* (UML) yang digunakan dalam desain perangkat lunak.

6.5.1 Tujuan dan Manfaat

Pada tingkat tinggi, BPMN ditargetkan untuk para peserta dan pemangku kepentingan lainnya dalam proses bisnis untuk mendapatkan pemahaman melalui representasi visual yang mudah dipahami dari langkah-langkah tersebut. Pada tingkat yang lebih terlibat, ditargetkan pada orang-orang yang akan mengimplementasikan proses, memberikan detail yang cukup untuk memungkinkan implementasi yang tepat. Ini memberikan standar, bahasa umum untuk semua pemangku kepentingan, baik teknis atau non-teknis: analis bisnis, peserta proses, manajer dan pengembang teknis, serta tim dan konsultan eksternal. Idealnya, ini menjembatani kesenjangan antara niat proses dan implementasi dengan memberikan detail dan kejelasan yang cukup ke dalam urutan kegiatan bisnis.

Pembuatan diagram dapat jauh lebih mudah dipahami daripada teks naratif. Ini memungkinkan komunikasi dan kolaborasi yang lebih mudah untuk mencapai tujuan proses yang efisien yang menghasilkan hasil berkualitas tinggi. Ini juga membantu komunikasi yang mengarah ke dokumen XML (Extensible Markup Language) yang diperlukan untuk menjalankan berbagai proses. Satu standar XML utama disebut BPEL atau BEPEL4WS, singkatan dari Bahasa Eksekusi Proses Bisnis untuk Layanan Web.

6.5.2 Elemen dan Simbol BPMN

BPMN menggambarkan empat jenis elemen ini untuk diagram proses bisnis:

1. ***Flow objects***
   Ini menentukan perilaku proses bisnis.

   - *Events* Model BPMN dimulai dan diakhiri dengan *event*, dan akan ada beberapa *event* antara dalam diagram juga. Suatu *event* adalah sesuatu yang terjadi. Itu bukan tugas atau kegiatan. *Event* diwakili oleh lingkaran. Lingkaran biasa adalah titik awal. Lingkaran dengan batas ganda menunjukkan acara menengah, dan lingkaran tebal mewakili titik akhir.

Gambar 6.4 Events

- *Activity*
  Kegiatan atau tugas tertentu yang dilakukan oleh seseorang atau sistem. Itu ditunjukkan oleh sebuah persegi panjang dengan sudut bulat. Mereka dapat menjadi lebih terperinci dengan sub-proses, pengulangan, kompensasi dan *multiple instances*.

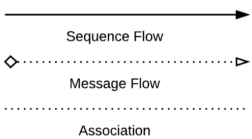
Gambar 6.5 Caption

- *Gateway*
  Gateway adalah elemen kondisional dan ditandai dengan berlian. Ada beberapa jenis *event* yang berbeda:

Gambar 6.6 Gateway

  - *Exclusive gateways* yang memindahkan alur kerja dari satu rute ke rute lain. Mereka mengindikasikan bahwa hanya ada satu opsi.
  - *Event-based gateways* yang bergantung pada sesuatu (selain tugas) yang terjadi.
  - *Parallel* di mana dua jalur berdampingan dan tidak bergantung pada suatu peristiwa.
  - *Inclusive gateways, complex gateways,* dan *and parallel event based gateways*. Masing-masing juga memiliki sendiri.

2. ***Connecting objects***
   Ini menghubungkan objek aliran satu sama lain atau ke informasi lain.

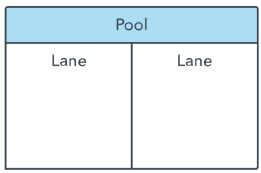


Gambar 6.7 Connecting objects

- *Sequence flows* :
  Elemen ini menunjukkan urutan kegiatan dilakukan.
- *Message flows* :
  Menampilkan pesan dan urutan aliran antara peserta.

- *Associations* :
  Elemen ini digunakan untuk menghubungkan informasi dan artefak (lihat di bawah).
- *Data associations* :
  Memiliki panah untuk menunjukkan arah aliran dalam suatu asosiasi

3. ***Swimlanes***
   Terkadang, departemen yang berbeda dan bahkan organisasi yang berbeda perlu bekerja bersama untuk mencapai sesuatu. Pool terdiri dari fungsi atau organisasi yang berpartisipasi, dan masing-masing memiliki swim lane sendiri. Di dalam setiap lane, terdapat *flows, objects,* dan *artifacts*.



Gambar 6.8 Swimlanes

4. ***Artifacts***
   Artefak menunjukkan bahwa ada informasi yang harus dirujuk di luar diagram BPMN. Setiap jenis informasi tambahan memiliki ikonnya sendiri. Informasi tambahan yang ditambahkan pengembang untuk membawa tingkat detail yang diperlukan ke diagram. Ada tiga jenis artefak: objek data, grup, atau anotasi. Objek data menunjukkan data apa yang diperlukan untuk suatu kegiatan. Grup menunjukkan pengelompokan kegiatan yang logis tetapi tidak mengubah aliran diagram. Anotasi memberikan penjelasan lebih lanjut ke bagian diagram.

Gambar 6.9 Artifacts

6.6 UML

UML atau *Unified Modeling Language*, adalah bahasa pemodelan standar yang terdiri dari serangkaian diagram terintegrasi, yang dikembangkan untuk membantu pengembang sistem dan perangkat lunak untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak sistem perangkat lunak, serta untuk pemodelan bisnis dan sistem non-perangkat lunak lainnya. UML mewakili sekumpulan praktik teknik terbaik yang telah terbukti berhasil dalam pemodelan sistem besar dan kompleks. UML adalah bagian yang sangat penting dari pengembangan

perangkat lunak berorientasi objek dan proses pengembangan perangkat lunak. UML sebagian besar menggunakan notasi grafis untuk mengekspresikan desain proyek perangkat lunak. Menggunakan UML membantu tim proyek berkomunikasi, mengeksplorasi desain potensial, dan memvalidasi desain arsitektur perangkat lunak. Pada artikel ini, kami akan memberikan Anda ide-ide terperinci tentang apa itu UML, sejarah UML dan deskripsi dari setiap jenis diagram UML, bersama dengan contoh-contoh UML.

6.6.1 Sejarah UML

Tujuan dari UML adalah untuk memberikan notasi standar yang dapat digunakan oleh semua metode berorientasi objek dan untuk memilih dan mengintegrasikan elemen terbaik dari notasi prekursor. UML telah dirancang untuk berbagai aplikasi. Karenanya, ia menyediakan konstruksi untuk berbagai sistem dan aktivitas (mis., Sistem terdistribusi, analisis, desain dan penyebaran sistem).

UML adalah notasi yang dihasilkan dari penyatuan OMT dari:

1. Teknik Pemodelan Objek OMT [James Rumbaugh 1991] - adalah yang terbaik untuk analisis dan sistem informasi intensif data.

2. Booch [Grady Booch 1994] - sangat baik untuk desain dan implementasi. Grady Booch telah bekerja secara luas dengan bahasa Ada, dan telah menjadi pemain utama dalam pengembangan teknik Berorientasi Objek untuk bahasa tersebut. Meskipun metode Booch kuat, notasi kurang diterima dengan baik (banyak bentuk awan mendominasi modelnya - tidak terlalu rapi)

3. OOSE (*Object-Oriented Software Engineering* [Ivar Jacobson 1992]) - menampilkan model yang dikenal sebagai *Use Cases*. *Use Cases* adalah teknik yang ampuh untuk memahami perilaku seluruh sistem (area di mana OO secara tradisional lemah).

Pada tahun 1994, Jim Rumbaugh, pencipta OMT, mengejutkan dunia perangkat lunak ketika ia meninggalkan *General Electric* dan bergabung dengan Grady Booch di Rational Corp. Tujuan kemitraan ini adalah untuk menggabungkan ide-ide mereka menjadi satu, metode terpadu (dengan judul kerja metode "*Unified Method*"). Pada 1995, pencipta OOSE, Ivar Jacobson, juga bergabung dengan Rational, dan gagasannya (khususnya konsep "*Use Cases*") dimasukkan ke dalam Metode *Unified* baru - sekarang disebut *Unified Modelling Language1*. Tim Rumbaugh, Booch dan Jacobson dikenal sebagai "*Three Amigos*"

UML juga telah dipengaruhi oleh notasi berorientasi objek lainnya:

- Mellor and Shlaer [1998]
- Coad and Yourdon [1995]
- Wirfs-Brock [1990]
- Martin and Odell [1992]

UML juga mencakup konsep-konsep baru yang tidak ada dalam metode utama lainnya pada saat itu, seperti mekanisme ekstensi dan bahasa kendala.

Selama tahun 1996, Permintaan Proposal (RFP) pertama yang dikeluarkan oleh Object Management Group (OMG) memberikan katalis bagi organisasi-organisasi ini untuk bergabung dalam upaya menghasilkan tanggapan RFP bersama. Rasional membentuk konsorsium Mitra UML dengan beberapa organisasi yang bersedia mendedikasikan sumber daya untuk bekerja menuju definisi UML 1.0 yang kuat. Yang paling berkontribusi terhadap definisi UML 1.0 termasuk:

- Digital Equipment Corp
- HP
- i-Logix
- IntelliCorp
- IBM
- ICON Computing
- MCI Systemhouse
- Microsoft
- Oracle
- Rational Software
- TI
- Unisys

Kolaborasi ini menghasilkan UML 1.0, bahasa pemodelan yang terdefinisi dengan baik, ekspresif, kuat, dan berlaku umum. Ini diajukan ke OMG pada Januari 1997 sebagai tanggapan RFP awal. Pada Januari 1997 IBM, ObjecTime, Platinum Technology, Ptech, Taskon, Reich Technologies dan Softeam juga mengirimkan tanggapan RFP terpisah ke OMG.

Perusahaan-perusahaan tersebut bergabung dengan mitra UML untuk menyumbangkan ide-ide mereka, dan bersama-sama para mitra menghasilkan respons UML 1.1 yang direvisi. Fokus dari rilis UML 1.1 adalah untuk meningkatkan kejelasan semantik UML 1.0 dan untuk memasukkan kontribusi dari mitra baru. Itu diserahkan kepada OMG untuk pertimbangan mereka dan diadopsi pada musim gugur 1997.1 dan ditingkatkan 1,1 menjadi 1,5, dan kemudian ke UML 2.1 dari 01 hingga 06 (sekarang versi UML saat ini adalah 2.5).

6.6.2 Mengapa UML

Seiring dengan meningkatnya nilai strategis perangkat lunak bagi banyak perusahaan, industri mencari teknik untuk mengotomatisasi produksi perangkat lunak dan meningkatkan kualitas serta mengurangi biaya dan waktu ke pasar. Teknik-teknik ini termasuk teknologi komponen, pemrograman visual, pola dan kerangka kerja. Bisnis juga mencari teknik untuk mengelola kompleksitas sistem karena mereka meningkatkan ruang lingkup dan skala. Secara khusus, mereka mengenali kebutuhan untuk memecahkan masalah arsitektur berulang, seperti distribusi fisik, konkurensi, replikasi, keamanan, keseimbangan beban, dan toleransi kesalahan. Selain itu, pengembangan untuk *World Wide Web*, sementara membuat beberapa hal lebih sederhana, telah memperburuk masalah arsitektur ini. *Unified Modelling Language* (UML) dirancang untuk menanggapi kebutuhan ini. Tujuan utama dalam desain UML diringkas oleh Page-Jones dalam Desain Berorientasi Objek Dasar di UML sebagai berikut:

1. Memberi pengguna bahasa pemodelan visual ekspresif yang siap pakai sehingga mereka dapat mengembangkan dan bertukar model yang bermakna.

2. Menyediakan mekanisme ekstensibilitas dan spesialisasi untuk memperluas konsep inti.

3. Tidak bergantung pada bahasa pemrograman tertentu dan proses pengembangan.

4. Berikan dasar formal untuk memahami bahasa pemodelan.

5. Dorong pertumbuhan pasar alat OO.

6. Mendukung konsep pengembangan tingkat tinggi seperti kolaborasi, kerangka kerja, pola dan komponen.

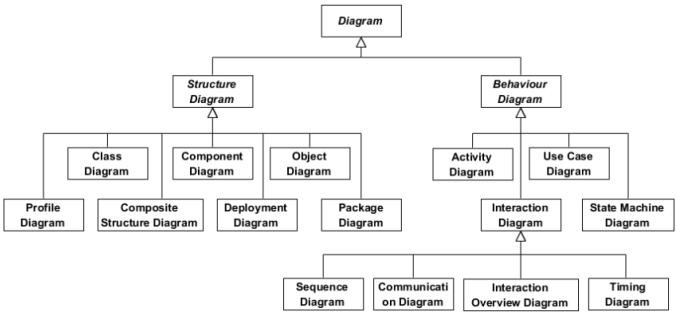
7. Integrasikan praktik terbaik.

6.6.3 Tinjauan Umum UML

Sebelum kita mulai melihat teori UML, kita akan mengambil langkah singkat melalui beberapa konsep utama UML. Hal pertama yang perlu diperhatikan tentang UML adalah bahwa ada banyak diagram (model) yang berbeda untuk digunakan. Alasannya adalah karena dimungkinkan untuk melihat suatu sistem dari berbagai sudut pandang.

Banyak orang tertarik pada berbagai aspek sistem, dan masing-masing dari mereka memerlukan tingkat detail yang berbeda. Sebagai contoh, seorang programmer perlu memahami desain sistem dan dapat mengkonversi desain ke kode tingkat rendah. Sebaliknya, seorang penulis teknis tertarik pada perilaku sistem secara keseluruhan, dan perlu memahami bagaimana fungsi produk. UML berupaya menyediakan bahasa yang sangat ekspresif sehingga semua pemangku kepentingan dapat memperoleh manfaat dari setidaknya satu diagram UML.

Berikut ini sekilas pada masing-masing dari 13 diagram ini seperti yang ditunjukkan dalam dua Struktur Diagram UML di bawah ini:



Gambar 6.10 Struktur Diagram UML

1. Diagram struktur menunjukkan struktur statis sistem dan bagian-bagiannya pada abstraksi yang berbeda dan tingkat implementasi dan bagaimana mereka saling terkait. Elemen-elemen dalam diagram struktur mewakili konsep yang berarti dari suatu sistem, dan dapat mencakup konsep abstrak, dunia nyata dan implementasi, ada tujuh jenis diagram struktur sebagai berikut:

   - Class Diagram
   - Component Diagram
   - Deployment Diagram
   - Object Diagram
   - Package Diagram
   - Composite Structure Diagram
   - Profile Diagram

2. Behavior diagram menunjukkan perilaku dinamis objek dalam suatu sistem, yang dapat digambarkan sebagai serangkaian perubahan pada sistem dari waktu ke waktu, ada tujuh jenis diagram perilaku sebagai berikut:

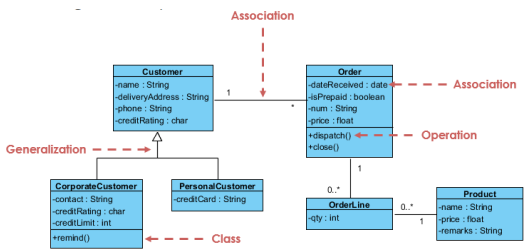
   - Use Case Diagram
   - Activity Diagram
   - State Machine Diagram
   - Sequence Diagram
   - Communication Diagram
   - Interaction Overview Diagram
   - Timing Diagram

6.6.3.1 Class Diagram

Class Diagram dalam Unified Modeling Language (UML) adalah jenis diagram struktur statis yang menggambarkan struktur suatu sistem dengan menunjukkan kelas sistem, atributnya, operasi (atau metode), dan hubungan antar objek. *Class Diagram* merupakan teknik pemodelan sentral yang berjalan melalui hampir semua metode berorientasi objek. Diagram ini menjelaskan jenis objek dalam sistem dan berbagai macam hubungan statis yang ada di antara objek.

Tujuan dari *Class Diagram* adalah:

1. Menunjukkan struktur statis pengklasifikasi dalam suatu sistem
2. Diagram menyediakan notasi dasar untuk diagram struktur lain yang ditentukan oleh UML
3. Bermanfaat untuk pengembang dan anggota tim lainnya juga
4. Analis Bisnis dapat menggunakan diagram kelas untuk memodelkan sistem dari perspektif bisnis



Gambar 6.11 Class Diagram

Notasi *class* ada 3, yaitu :

1. *Class Name*
   - Nama kelas muncul di partisi pertama.
2. *Class Attributes*
   - Atribut ditampilkan di partisi kedua.
   - Jenis atribut ditampilkan setelah titik dua.
   - Mengaitkan peta ke variabel anggota (anggota data) dalam kode.
3. *Class Operations (Methods)*
   - Operasi ditampilkan di partisi ketiga. Mereka adalah layanan yang disediakan oleh kelas.

- Jenis pengembalian metode ditampilkan setelah titik dua di akhir tanda tangan metode.
- Jenis pengembalian parameter metode ditampilkan setelah titik dua mengikuti nama parameter.
- Operasi memetakan ke metode kelas dalam kode

sebuah *class* mungkin terlibat dalam satu atau lebih relasi dengan *class* lain. Suatu relasi dapat berupa salah satu dari jenis berikut:

1. *Inheritance* (atau Generalisasi)
   Mengacu pada jenis hubungan di mana satu kelas terkait adalah anak dari kelas lain berdasarkan asumsi fungsi yang sama dari kelas induk. Dengan kata lain, kelas anak adalah tipe spesifik dari kelas induk. Untuk menunjukkan warisan dalam diagram UML, garis solid dari kelas anak ke kelas induk digambar menggunakan panah yang tidak terisi.

2. *Association*
   Association adalah istilah luas yang mencakup hampir semua koneksi logis atau hubungan antar kelas.

3. *Aggregation*
   Mengacu pada pembentukan kelas tertentu sebagai hasil dari satu kelas yang dikumpulkan atau dibangun sebagai koleksi. Misalnya, kelas ”perpustakaan” terdiri dari satu atau lebih buku, di antara bahan-bahan lainnya. Secara agregasi, kelas yang terkandung tidak sangat tergantung pada siklus hidup wadah. Dalam contoh yang sama, buku akan tetap ada meskipun perpustakaan dibubarkan. Untuk memperlihatkan agregasi dalam diagram, gambarlah garis dari kelas induk ke kelas anak dengan bentuk berlian di dekat kelas induk. Untuk memperlihatkan agregasi dalam diagram, gambarlah garis dari kelas induk ke kelas anak dengan bentuk berlian di dekat kelas induk.

6.6.3.2 Component Diagram

Dalam *Unified Modeling Language*, diagram komponen menggambarkan bagaimana komponen dihubungkan bersama untuk membentuk komponen yang lebih besar atau sistem perangkat lunak. Ini menggambarkan arsitektur komponen perangkat lunak dan ketergantungan di antara mereka. Komponen perangkat lunak tersebut termasuk komponen *run-time*, komponen yang dapat dieksekusi juga komponen kode sumber.

Komponen mewakili bagian modular dari sistem yang merangkum isi dan yang manifestasinya dapat diganti dalam lingkungannya. Dalam UML, komponen digambarkan sebagai persegi panjang dengan kompartemen opsional ditumpuk secara vertikal. Tampilan komponen pada UML dapat dimodelkan sebagai sebagai berikut:

Gambar 6.12 Komponen

1. *A rectangle with the component's name*
2. *A rectangle with the component icon A rectangle with the stereotype text and/or icon*

Secara grafis, diagram komponen adalah kumpulan simpul dan busur dan biasanya berisi komponen, antarmuka dan relasi seperti, *dependency, aggregation, constraint, generalization, association,* dan *realization*.

- Association: Menetapkan hubungan semantik yang dapat terjadi di antara instance yang diketik. Ini memiliki setidaknya dua ujung yang diwakili oleh properti, yang masing-masing terhubung ke jenis ujung. Lebih dari satu ujung asosiasi dapat memiliki tipe yang sama.

Gambar 6.13 Association

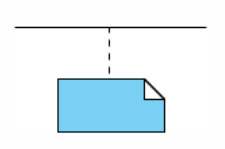
- **Composition:** Sebuah bentuk agregasi yang kuat yang membutuhkan *instance part* dimasukkan dalam paling banyak satu komposit sekaligus. Jika komposit dihapus, semua bagiannya biasanya dihapus bersamanya.

Gambar 6.14 Composition

- **Aggregation:** Suatu jenis asosiasi yang salah satu ujungnya ditandai dibagikan sebagai jenis agregasi, yang berarti bahwa ia memiliki agregasi bersama.

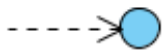
Gambar 6.15 Aggregation

- **Constraint:** Suatu kondisi atau batasan yang diekspresikan dalam teks bahasa alami atau dalam bahasa yang dapat dibaca mesin untuk tujuan mendeklarasikan beberapa semantik suatu elemen.



Gambar 6.16 Constraint

- **Dependency:** Hubungan yang menandakan bahwa satu atau serangkaian elemen model memerlukan elemen model lain untuk spesifikasi atau implementasinya. Ini berarti bahwa semantik lengkap dari elemen-elemen dependen bergantung secara semantik atau struktural pada definisi elemen-elemen pemasok.



Gambar 6.17 Dependency

- **Links:** Hubungan taksonomi antara classifier yang lebih umum dan classifier yang lebih spesifik. Setiap instance dari classifier spesifik juga merupakan instance tidak langsung dari classifier umum. Dengan demikian, classifier spesifik mewarisi fitur dari classifier yang lebih umum.

Gambar 6.18 Link

6.6.3.3 Deployment Diagram

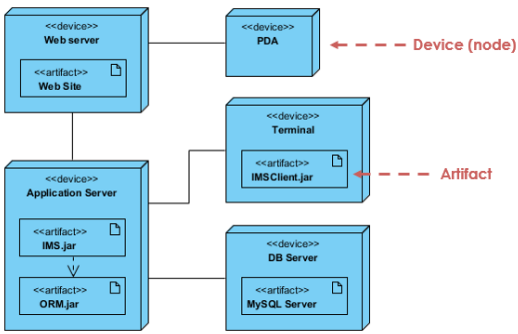
Deployment Diagram membantu memodelkan aspek fisik dari sistem perangkat lunak Berorientasi Objek. Ini adalah diagram struktur yang menunjukkan arsitektur sistem sebagai penyebaran (distribusi) artefak perangkat lunak ke target penyebaran. Artefak mewakili elemen konkret di dunia fisik yang merupakan hasil dari proses pengembangan. Ini memodelkan konfigurasi *run-time* dalam tampilan statis dan memvisualisasikan distribusi artefak dalam suatu aplikasi. Dalam kebanyakan kasus, ini melibatkan pemodelan konfigurasi perangkat keras bersama dengan komponen perangkat lunak yang hidup.

Deployment Diagram pada UML adalah diagram yang menunjukkan konfigurasi node pemrosesan waktu berjalan dan komponen yang ada di dalamnya. Deployment

diagram adalah sejenis diagram struktur yang digunakan dalam pemodelan aspek fisik dari sistem berorientasi objek. Mereka sering digunakan untuk memodelkan tampilan penyebaran statis suatu sistem (topologi perangkat keras).

Deployment Diagram memiliki tujuan sebagai berikut:

- Menunjukkan struktur sistem *run-time*.
- Menangkap perangkat keras yang akan digunakan untuk mengimplementasikan sistem dan tautan antara berbagai item perangkat keras.
- Memodelkan elemen perangkat keras fisik dan jalur komunikasi.
- Dapat digunakan untuk merencanakan arsitektur suatu sistem.
- Berguna untuk mendokumentasikan penyebaran komponen atau node perangkat lunak



Gambar 6.19 Deployment Diagram

6.6.3.4 Use Case Diagram

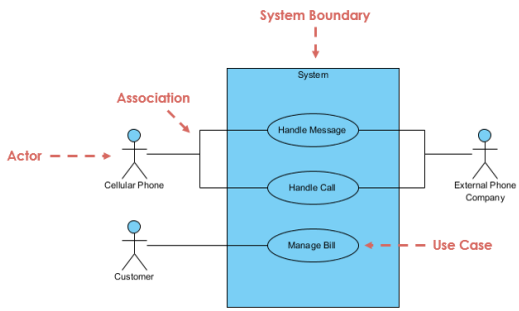
Use case Diagram pada UML adalah bentuk utama dari persyaratan sistem / perangkat lunak untuk program perangkat lunak baru yang kurang berkembang. Use case menentukan perilaku yang diharapkan (apa), dan bukan metode yang tepat untuk mewujudkannya (bagaimana). Use case yang telah ditentukan dapat dilambangkan dengan representasi tekstual dan visual. Konsep kunci pemodelan use case adalah membantu kami merancang sistem dari perspektif pengguna akhir. Ini adalah teknik yang efektif untuk mengkomunikasikan perilaku sistem dalam istilah pengguna dengan menentukan semua perilaku sistem yang terlihat secara eksternal.

Model use-case menggambarkan persyaratan fungsional sistem dalam hal use case. Ini adalah model fungsionalitas sistem yang dimaksudkan (*use cases*) dan lingkungannya (*actors*). *Use case* memungkinkan Anda untuk menghubungkan apa yang Anda butuhkan dari suatu sistem dengan bagaimana sistem memenuhi kebutuhan tersebut.

Use Case Diagram biasanya dikembangkan pada tahap awal pengembangan dan orang sering menerapkan pemodelan use case untuk tujuan berikut:

- Tentukan konteks suatu sistem
- Menangkap persyaratan sistem
- Validasi arsitektur sistem
- Mendorong implementasi dan hasilkan kasus uji

Bentuk standar diagram use case didefinisikan dalam Unified Modeling Language seperti yang ditunjukkan pada contoh Use Case Diagram di bawah ini:



Gambar 6.20 Use Case

1. ***Actor*** **:** Seseorang yang berinteraksi dengan use case (fungsi sistem).

Gambar 6.21 Actor

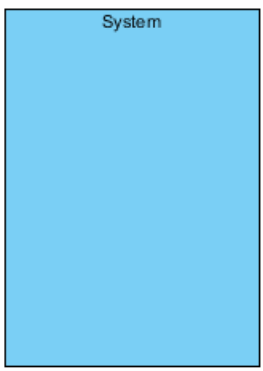
2. ***Use Case*** **:** Fungsi sistem (proses - otomatis atau manual)

Gambar 6.22 Use Case

3. ***Communication Link* :** Partisipasi aktor dalam *use case* ditunjukkan dengan menghubungkan aktor ke use case dengan tautan yang solid. Aktor dapat dihubungkan untuk menggunakan kasing oleh asosiasi, menunjukkan bahwa aktor dan kasing berkomunikasi satu sama lain menggunakan pesan.

Gambar 6.23 Communication Link

4. ***Boundary of system* :** Batas sistem berpotensi seluruh sistem seperti yang didefinisikan dalam dokumen persyaratan. Untuk sistem yang besar dan kompleks, setiap modul dapat menjadi batas sistem.

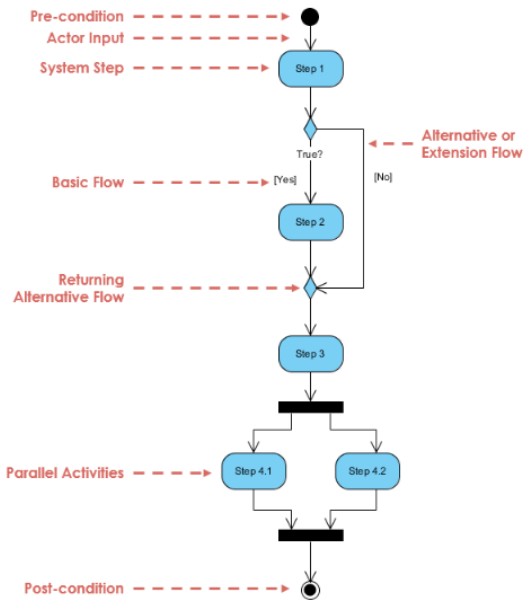


Gambar 6.24 Boundary of system

6.6.3.5 Activity Diagram

Activity diagram adalah representasi grafis dari alur kerja kegiatan dan tindakan bertahap dengan dukungan untuk pilihan, iterasi, dan konkurensi. Ini menggambarkan aliran kontrol dari sistem target, seperti mengeksplorasi aturan bisnis yang

kompleks dan operasi, menggambarkan use case juga proses bisnis. Dalam Bahasa Pemodelan Bersatu, diagram aktivitas dimaksudkan untuk memodelkan proses komputasi dan organisasi (misalnya alur kerja).

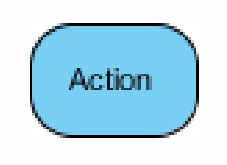


Gambar 6.25 Activity Diagram

- ***Activity* :** Digunakan untuk mewakili serangkaian tindaka.

Gambar 6.26 Activity

- ***Action* :** Tugas yang harus dilakukan.



Gambar 6.27 Action

- ***Control Flow*** **:** Memperlihatkan urutan eksekusi.

Gambar 6.28 Control Flow

- ***Object Flow*** **:** Tampilkan aliran suatu objek dari satu aktivitas (atau aksi) ke aktivitas lain (atau aksi).

Gambar 6.29 Object Flow

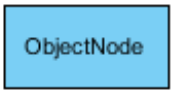
- ***Initial Node*** **:** Menggambarkan awal dari serangkaian tindakan atau kegiatan.

Gambar 6.30 Initial Node

- ***Activity Final Node*** **:** Hentikan semua aliran kontrol dan objek mengalir dalam suatu aktivitas (atau tindakan).

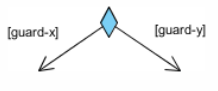
Gambar 6.31 Activity Final Node

- ***Object Node*** **:** Mewakili objek yang terhubung ke satu set Object Flows.



Gambar 6.32 Object Node

- ***Decision Node* :** Mewakili kondisi pengujian untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalur.

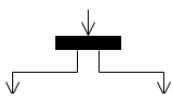


Gambar 6.33 Decision Node

- ***Merge Node* :** Menyatukan kembali jalur keputusan berbeda yang dibuat menggunakan simpul keputusan.

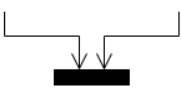
Gambar 6.34 Merge Node

- *Fork Node :* Membagi perilaku menjadi serangkaian kegiatan paralel atau bersamaan (atau tindakan)



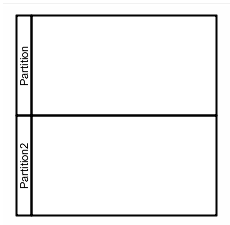
Gambar 6.35 Fork Node

- ***Join Node* :** Menyatukan kembali serangkaian kegiatan (atau tindakan) yang paralel atau bersamaan.



Gambar 6.36 Join Node

- ***Swimlane and Partition*** **:** Cara mengelompokkan aktivitas yang dilakukan oleh aktor yang sama pada diagram aktivitas atau mengelompokkan aktivitas dalam satu utas.



Gambar 6.37 Swimlane and Partition

6.6.3.6 State Machine Diagram

State Machine Diagram adalah jenis diagram yang digunakan dalam UML untuk menggambarkan perilaku sistem yang didasarkan pada konsep diagram keadaan oleh David Harel. *State machine diagram* juga dapat menunjukkan bagaimana suatu entitas merespons berbagai peristiwa dengan mengubah dari satu keadaan ke keadaan lain. State machine diagram adalah diagram UML yang digunakan untuk memodelkan sifat dinamis suatu sistem. Ini membantu untuk memvisualisasikan seluruh siklus objek dan dengan demikian membantu untuk memberikan pemahaman yang lebih baik tentang *state-based systems*.

State machine diagram biasanya digunakan untuk menggambarkan perilaku yang bergantung pada keadaan untuk suatu objek. Objek merespons secara berbeda terhadap peristiwa yang sama tergantung pada keadaannya. Diagram mesin negara biasanya diterapkan pada objek tetapi dapat diterapkan ke elemen apa pun yang memiliki perilaku terhadap entitas lain seperti: aktor, kasus penggunaan, metode, sistem subsistem, dan dll. dan biasanya digunakan bersama dengan diagram interaksi (biasanya diagram urutan).

6.6.3.7 Sequence Diagram

Sequence Diagram memodelkan kolaborasi objek berdasarkan urutan waktu. Ini menunjukkan bagaimana objek berinteraksi dengan orang lain dalam skenario tertentu dari use case. Dengan kemampuan pemodelan visual canggih, Anda dapat membuat diagram urutan yang kompleks dengan beberapa klik. Selain itu, beberapa alat pemodelan seperti Visual Paradigm dapat menghasilkan diagram urutan dari aliran peristiwa yang telah Anda tetapkan dalam deskripsi use case.

textitSequence Diagram memiliki tujuan, antara lain:

- Model interaksi tingkat tinggi antara objek aktif dalam suatu sistem.
- Model interaksi antara instance objek dalam kolaborasi yang mewujudkan use case.

- Model interaksi antara objek dalam kolaborasi yang mewujudkan operasi.

- Baik model interaksi generik (menunjukkan semua jalur yang mungkin melalui interaksi) atau contoh spesifik dari interaksi (hanya menunjukkan satu jalur melalui interaksi)

Sequence Diagram menunjukkan elemen ketika mereka berinteraksi dari waktu ke waktu dan mereka diatur sesuai dengan objek (horizontal) dan waktu (vertikal):

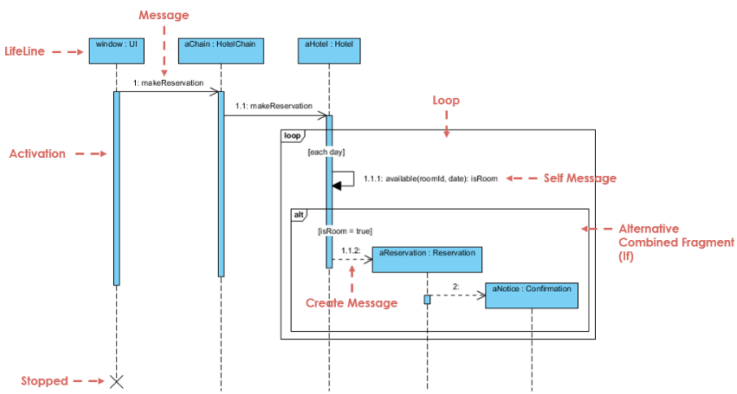
1. Dimensi Objek

   - Sumbu horizontal menunjukkan elemen-elemen yang terlibat dalam interaksi.

   - Secara konvensional, objek yang terlibat dalam operasi terdaftar dari kiri ke kanan sesuai dengan ketika mereka mengambil bagian dalam urutan pesan. Namun, elemen pada sumbu horizontal dapat muncul dalam urutan apa pun.

2. Dimensi Waktu

   - Sumbu vertikal menunjukkan proses waktu (atau progres) ke bawah halaman.



Gambar 6.38 Sequence Diagram

1. *Actor*

Gambar 6.39 Actor

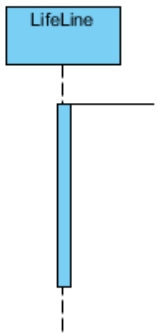
- Seorang aktor tidak selalu mewakili entitas fisik tertentu tetapi hanya peran tertentu dari entitas tertentu
- Seseorang dapat memainkan peran beberapa aktor yang berbeda dan, sebaliknya, aktor yang diberikan dapat dimainkan oleh beberapa orang yang berbeda.
- Mewakili peran yang dimainkan oleh pengguna manusia, perangkat keras eksternal, atau subjek lainnya.

2. *Lifeline*

Gambar 6.40 Lifeline

- *Lifeline* mewakili seorang individu peserta dalam Interaksi.

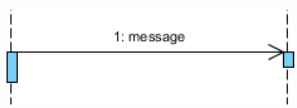
3. *Activations*



Gambar 6.41 Activations

- Sebuah persegi panjang tipis pada garis hidup) mewakili periode di mana suatu elemen melakukan operasi.
- Bagian atas dan bawah dari persegi panjang selaras dengan inisiasi dan waktu penyelesaian masing-masing.

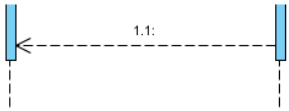
4. *Call Message*



Gambar 6.42 Call Message

- Sebuah pesan mendefinisikan komunikasi tertentu antara Lifelines dengan Interaksi.
- Pesan panggilan adalah jenis pesan yang mewakili permintaan operasi jalur kehidupan target.

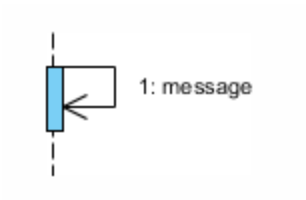
5. *Return Message*



Gambar 6.43 Return Message

- Sebuah pesan mendefinisikan komunikasi tertentu antara Lifelines dengan Interaksi.
- Pesan balik adalah jenis pesan yang mewakili informasi lewat balik ke penelepon dari pesan yang sesuai sebelumnya.

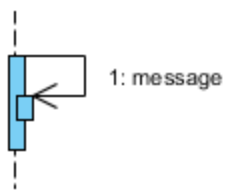
6. *Self Message*



Gambar 6.44 Self Message

- Sebuah pesan mendefinisikan komunikasi tertentu antara Lifelines dengan Interaksi.
- Pesan diri adalah jenis pesan yang mewakili doa pesan dari garis hidup yang sama.

7. *Recursive Message*



Gambar 6.45 Recursive Message

- Sebuah pesan mendefinisikan komunikasi tertentu antara Lifelines dengan Interaksi.

- Pesan rekursif adalah jenis pesan yang mewakili doa pesan dari garis hidup yang sama. Itu target menunjuk ke aktivasi di atas aktivasi tempat pesan itu berasal.

8. *Note*

Gambar 6.46 Note

Catatan (komentar) memberikan kemampuan untuk melampirkan berbagai komentar ke elemen. Sebuah komentar tidak memiliki kekuatan semantik, tetapi dapat berisi informasi yang berguna bagi pemodel.

BAB 7

WEBSITE

7.1 Sejarah dan Perkembangan

website pertama kali ditemukan oleh Sir Timothy John, Tim Berners-Lee. Pada tahun 1991 *website* terhubung dengan jaringan.Ttujuan dari dibuatnya *website* pada saat itu yakni untuk mempermudah tukar menukar dan memperbaharui informasi kepada sesama peneliti di tempat mereka bekerja. Dengan demikian pengertian *website* saat itu masih sebatas tukar menukar informasi, bukan pengertian *website* secara terminologi. *website* dipublikasikan ke publik setelah adanya pengumuman dari CERN pada tanggal 30 april 1993. CERN menyatakan bahwa *website* dapat digunakan secara gratis oleh semua orang. Pada saat ini pengertian *website* sudah masuk ke dalam ranah publik karena sudah bisa digunakan oleh semua orang dimanapun dan kapanpun.

Sejak web ditemukan tahun 1990, Web semakin populer dan menjadi layanan internet dengan pengguna terbanyak. Sebagian pengguna internet bahkan mengira kalau Web adalah satu satunya layanan di internet. Teknologi Web pun terus bergulir dari era web 1.0, web 2.0 dan trend menuju web 3.0.

By Rolly M. Awangga Copyright © 2020 Kreatif Industri Nusantara.

7.2 Pengertian

World Wide Web, yang lebih populer disingkat dengan WWW merupakan satu buah ruang informasi yang digunakan oleh pengenal global yang dinamakan dengan URL *(Uniform Resource Locator)* untuk mengenal pasti sumber daya berguna. WWW tidak jarang dianggap sama dengan Internet dengan cara total, meski sebenarnya WWW sendiri hanyalah bagian daripada Internet.

WWW merupakan kumpulan situs *server* dari seluruh dunia yang memiliki kegunaan untuk menyediakan data info untuk bisa digunakan bersama. WWW ialah sektor yang paling menarik dari Internet. Lewat web, para *customer* bisa terhubung informasi-informasi yang tidak hanya berupa teks namun mampu pula berupa gambar, nada, video, dan animasi.

Web merupakan salah satu sumber daya internet yang berkembang pesat. Informasi Web didistribusikan melalui pendekatan *hypertext*, yang memungkinkan suatu teks pendek menjadi acuan untuk membuka dokumen yang lain.

Secara garis besar, website bisa digolongkan menjadi 3 bagian yaitu:

1. *Website* Statis
   Website statis adalah web yang mempunyai halaman tidak berubah. Artinya adalah untuk melakukan perubahan pada suatu halaman dilakukan secara manual dengan mengubah *code* yang menjadi struktur dari *website* tersebut.

2. Website Dinamis
   Website dinamis merupakan *website* yang secara struktur diperuntukan untuk *update* sesering mungkin. Biasanya selain utama yang bisa diakses oleh user pada umumnya, juga disediakan halaman *backend* untuk mengubah konten dari *website*. Contoh umum mengenai *website* dinamis adalah web berita atau web portal yang didalamnya terdapat fasilitas berita, polling dan sebagainya.

3. *Website* Interaktif
   Website Interaktif adalah web yang saat ini memang sedang booming. Salah satu contoh *website* interaktif adalah blog dan forum. Di *website* ini user bisa berinteraksi dan beradu argument mengenai apa yang menjadi pemikiran mereka. Biasanya *website* seperti memiliki moderator untuk mengatur supaya topik yang diperbincangkan tidak keluar jalur.

7.3 Kelebihan dan Kekurangan

7.3.1 Kelebihan

Aplikasi ini tidak akan membebani penyimpanan *terminal* atau CPU, karena tidak perlu dilakukan penginstalan di *smartphone* atau *tablet*. Hanya dengan bermodalkan jaringan *internet* dan sebuah *device* sudah dapat dengan mudah menggunakannya. Selain itu, sesuai dengan sifat aplikasi web yang seba *online* termasuk pembuatan akunnya, kita dapat menggunakan layanan dan pengaturan yang sama pada *device* yang berbeda-beda.

Dengan kata lain, layanan aplikasi serupa dapat digunakan di tablet melalui smartphone saat di luar rumah. Selain itu, dalam kancah bisnis *file* perusahaan dapat diubah sekaligus jika menggunakan aplikasi jenis ini, sehingga akan meningkatkan kenyamanan.

7.3.2 Kekurangan

Pengoperasiannya yang hanya memerlukan internet dalam hal ini bisa menjadi kelebihan sekaligus sebagai kekurangan untuk aplikasi web. Aplikasi ini tidak akan bisa digunakan di tempat yang tidak memiliki koneksi internet. Selain itu, kinerja aplikasi Web tidak ditentukan oleh spesifikasi *devise* tetapi ditentukan di lingkungan di mana aplikasi Web ini beroperasi. Oleh karena itu, bahkan jika kita menggunakan *device* dengan spesifikasi tinggi, aplikasi ini mungkin lambat beroperasi karena keadaan lingkungan operasi aplikasi web yang tidak kondusif.

Dalam hal keamanan, karena aplikasi web selalu digunakan dengan adanya koneksi internet, menjadikannya rentan terkena virus. Selain itu kita juga harus berhati-hati karena untuk aplikasi Web yang dapat menyimpan file seperti gambar dan dokumen secara *online* memungkinkan terjadinya kebocoran data.

7.4 *Tools* yang Digunakan

Berikut beberapa beberapa *tools* yang digunakan untuk membangun sebuah *website* :

1. Bahasa Pemrograman
   Menggunakan bahasa pemrograman PHP sejauh ini bisa anda jadikan pilihan utama. Walaupun ada beberapa bahasa pemrograman lain seperti ASP (Active Server Page), JSP (Java Server Pages), Perl dan Phyton, tetapi sejauh yang saya ketahui tutorial/ebook dan penggunaan PHP lebih banyak dibandingkan pemrograman lainnya.

2. *Database*
   MySQL bisa dijadikan pilihan untuk menyimpan data-data yang telah anda buat. MySQL merupakan software database open source yang cukup populer. Banyak pengembang software dan aplikasi populer yang menggunakannya seperti google, yahoo, nokia, youtube, wordpress dan facebook. Banyak tools yang bisa anda gunakan untuk mempermudah administrasi MySQL, diantaranya: MySQL command line client, MySQL-Front, SQL Yog, MySQL Administrator dan yang paling sering digunakan hampir semua penyedia hosting adalah PHPMyAdmin.

3. *Web Server*
   Web server merupakan sebuah perangkat lunak dalam server yang berfungsi menerima request (permintaan) berupa halaman web web melalui HTTP atau

HTTPS dari klien yang dikenal dengan web browser dan mengirimkan kembali hasilnya (response) dalam bentuk dokumen html. Beberapa web server yang bisa anda gunakan diantaranya IIS (Internet infromation service) keluaran microsoft, Xitami Web server, Sun Java System Web Server dan Apache (www.apache.org). Anda bisa menggunakan Apache sebagai web server di komputer/laptop sendiri (localhost). Proses instalasi Apache, PHP dan MySQL terkadang menjadi kendala dalam pembuatan pemrograman web dengan menggunakan bahasa pemrograman PHP. Untuk itu gunakanlah aplikasi paket gratis yang menyediakan ketiga software tersebut. Untuk hal ini, saya lebih menyarankan XAMPP (versi windows) dan LAMPP (linux) untuk dijadikan pilihan.

4. *Browser*
   Berfungsi untuk menjalankan program berbasis web dan menampilkan aplikasi yang telah dibuat.

7.5 PHP

PHP atau singkatan dari *Hypertext Preprocessor* adalah bahasa *scripting Web HTML-embedded*. PHP adalah bahasa *scripting* dengan tujuan umum *open source* yang banyak digunakan dan sangat cocok untuk pengembangan web dan dapat disematkan ke dalam HTML. Ini berarti kode PHP dapat dimasukkan ke HTML halaman Web. Saat halaman PHP diakses, kode PHP dibaca atau ”diurai” oleh server tempat halaman itu berada. *Output* dari fungsi PHP pada halaman biasanya dikembalikan sebagai kode HTML, yang dapat dibaca oleh *browser*. Karena kode PHP diubah menjadi HTML sebelum halaman dimuat, pengguna tidak dapat melihat kode PHP pada halaman. Ini membuat halaman PHP cukup aman untuk mengakses *database* dan informasi aman lainnya. Banyak sintaks PHP yang dipinjam dari bahasa lain seperti C, Java dan Perl. Namun, PHP juga memiliki sejumlah fitur unik dan fungsi khusus. Tujuan dari bahasa ini adalah untuk memungkinkan pengembang Web untuk menulis halaman yang dihasilkan secara dinamis dengan cepat dan mudah. PHP juga bagus untuk membuat Website berbasis *database*. Jika Anda ingin mempelajari lebih lanjut tentang PHP, situs resminya adalah PHP.net.

Contoh penggunaan PHP :

```
<!DOCTYPE html>
<html>
    <head>
        <title>Example</title>
    </head>
    <body>

        <?php
            echo "Hi, I'm a PHP script!";
        ?>

    </body>
</html>
```

7.6 *Text Editor*

Text Editor adalah suatu software aplikasi atau suatu program komputer yang memungkinkan Anda sebagai penggunanya untuk membuat, mengubah atau mengedit file teks yang ada berupa plain text. Text editor ini sebenarnya bisa digunakan untuk membuat program-program komputer dan mengedit *source code* dari bahasa pemograman. Untuk menuliskan bahasa pemrograman PHP dapat menggunakan beberapa *tools* seperti:

- Notepad
- notepad++
- Dreamweaver
- Sublime
- Visual Code
- Dan lain-lain.

7.6.1 Visual Studio Code

Visual Studio Code merupakan salah satu *text editor* yang paling populer dikalangan Web *text editor* diseluruh dunia. Para web *developer* itu yang mengembangkan aplikasi web menggunakan ASP.NET, Node.js, HTML, CSS, Less, Sass, dan JSON. Seperti editor pada umumnya VSCode memiliki fitur *syntax coloring* dan *bracket matching*. Bahasa pemrograman yang mendukung fitur tadi adalah Batch, C++, Closure, Coffee Script, DockerFile, F, Go, Jade, Java, HandleBars, Ini, Lua, Makefile, Markdown, Objective-C, Perl, PHP, PowerShell, Python, R, Razor, Ruby, SQL, Visual Basic, dan XML.

Berikut spesifikasi atau *requirements* yang dibutuhkan untuk menginstal VSCode :

1. ***Hardware***, Perangkat keras rekomendasi yang dibutuhkan yaitu memiliki *processor* 1.6 GHz or faster *processor*, dan minimal memiliki 1 GB RAM.
2. **Sistem operasi**, Sistem operasi minimal yang digunakan yaitu :
   (a) Windows minimal windows 7 wajib Menginstal .NET Framework 4.5.2
   (b) OS X Yosemite
   (c) Windows 7, 8.0, 8.1 and 10 (32-bit and 64-bit)
   (d) Linux (Debian): Ubuntu Desktop 14.04, Debian 7
   (e) Linux (Red Hat): Red Hat Enterprise Linux 7, CentOS 7, Fedora 23

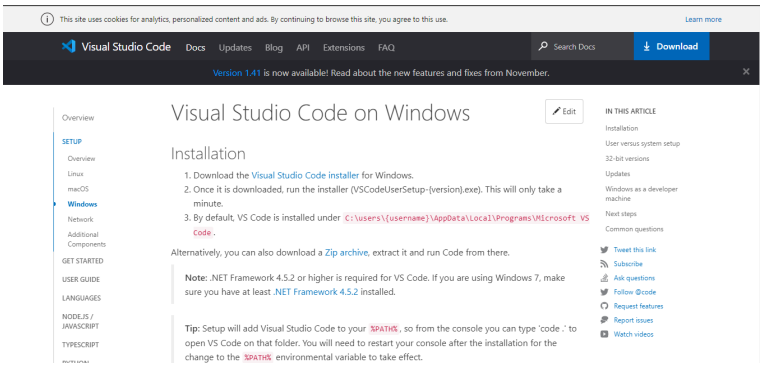
7.6.1.1 Tutorial Instalasi Visual Code

Berikut tutorial instalasi visual code pada sistem operasi widows :

1. *Download* Visual Studio Code untuk windows pada

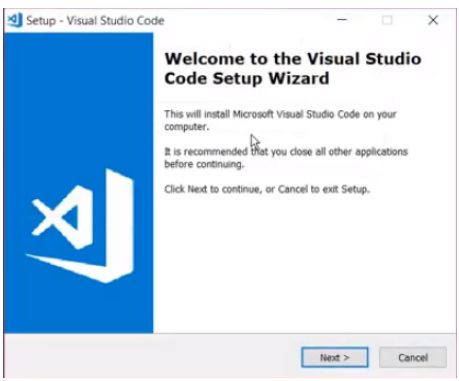
$$https://code.visualstudio.com/docs/setup/windows$$

kemudian klik *download* di pojok kanan atas.



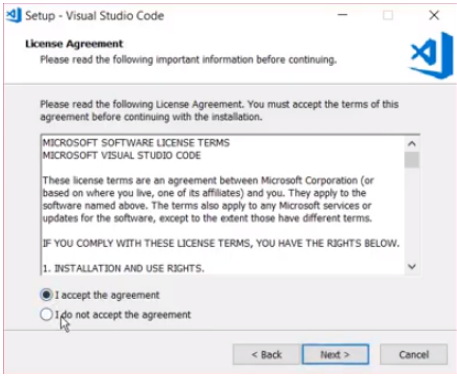
Gambar 7.1 Tutorial instalasi Visual Studio Code

2. Setelah diunduh, jalankan penginstal $VSCodeUserSetup - version.exe$ sehingga akan muncul seperti gambar 7.2. Selanjutnya klik *next*.



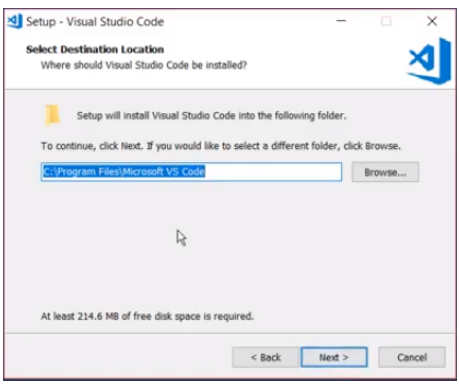
Gambar 7.2 Tutorial instalasi Visual Studio Code

3. Kemudian pilih *I accept the agreement* pada halaman *Lisence Agreement* untuk menyetujui kebijakan dari VSCode lalu klik *Next*.



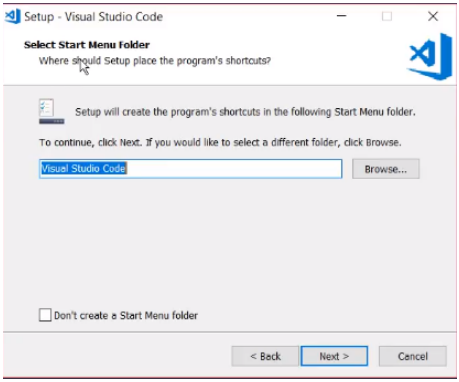
Gambar 7.3 Tutorial instalasi Visual Studio Code

4. Selanjutnya akan menampilkan halaman *Select Destination Location* dimana pada bagian ini dapat memilih lokasi hasil instalasi visual studio code. Disarankan lokasi dibiarkan *default*, namun jika ingin mengubah dengan mengklik *browse* lalu pilih lokasi hasil instalasi. Setelah itu, pilih *next*.



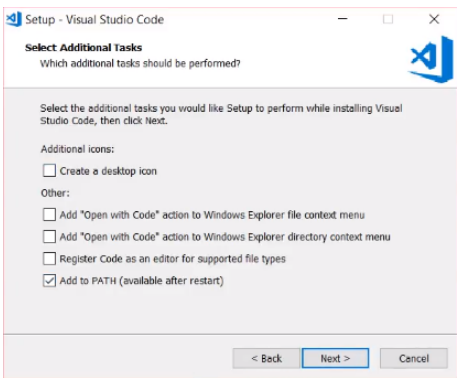
Gambar 7.4 Tutorial instalasi Visual Studio Code

5. Kemudian akan menampilkan halaman *Select Start Menu Folder* lalu *Next* untuk membuat foldernya.



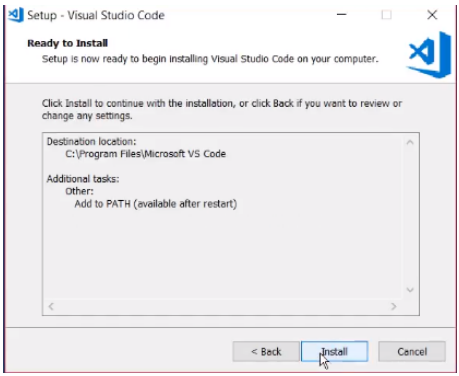
Gambar 7.5 Tutorial instalasi Visual Studio Code

6. Kemudian Ceklis *Create Desktop Icon* jika ingin membuat *shortcut* VSCode nya kemudian ceklis "*Add to PATH (available after restart)*" jika sudah klik *Next*.



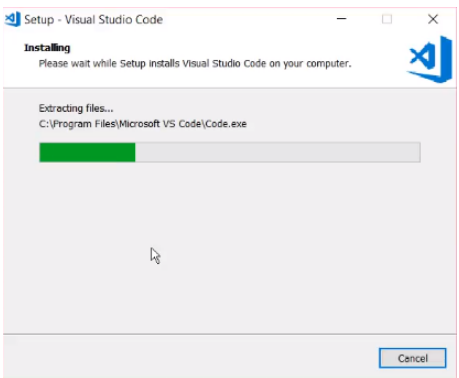
Gambar 7.6 Tutorial instalasi Visual Studio Code

7. Kemudian Klik Install



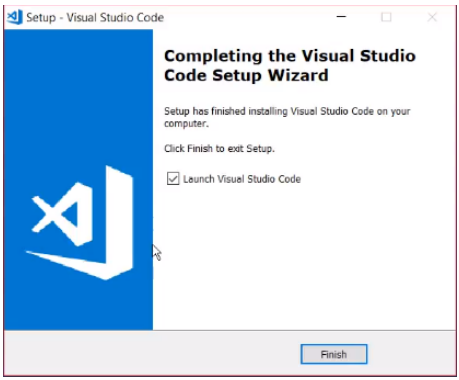
Gambar 7.7 Tutorial instalasi Visual Studio Code

8. tunggu sampai Setup has finished installing Visual Studio Code on your komputer .



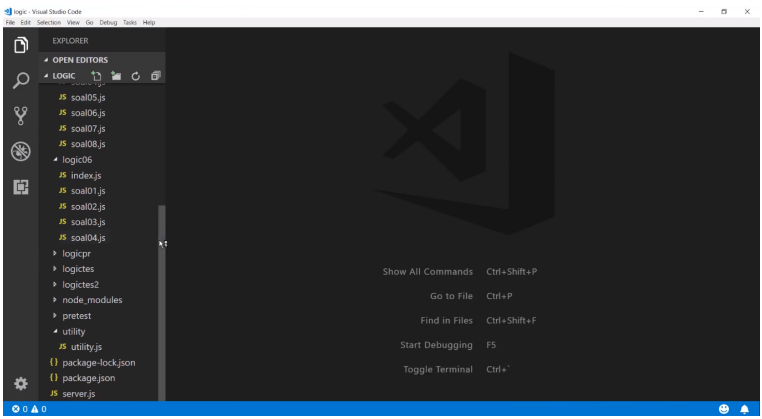
Gambar 7.8 Tutorial instalasi Visual Studio Code

9. Kemudian klik finish



Gambar 7.9 Tutorial instalasi Visual Studio Code

10. Terakhir akan menampilkan halaman Visual Studio Code seperti pada gambar 7.9

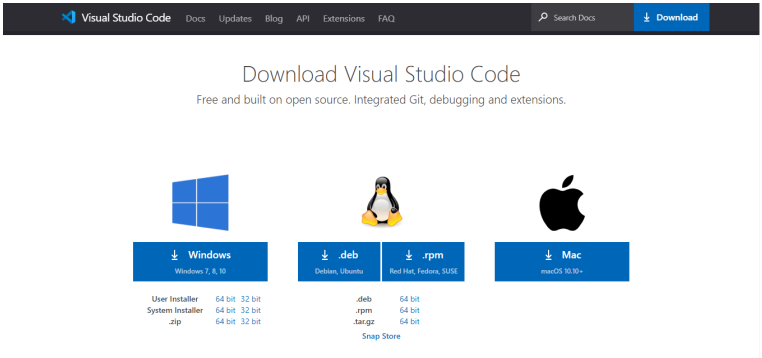


Gambar 7.10 Tutorial instalasi Visual Studio Code

Terdapat 2 Cara untuk memasang VSCODE dpada sistem perasi Linux, pertama dengan menggunakan metode GUI (grafis) dan yang kedua adalah menggunakan metode *command line* melalui terminal. Menginstal dengan metode grafis sangatlah mudah, Anda hanya perlu pergi ke halaman *download* VSCODE

$$https://code.visualstudio.com$$

kemudian mengunduhnya seperti biasa. Pastikan untuk memilih opsi .deb jika Anda menggunakan Ubuntu/Debian.



Gambar 7.11 Tutorial instalasi Visual Studio Code

Instalasi dengan metode grafis hampir sama dengan instalasi pada wodows, ketika aplikasi telah selesai didownload, maka akan dihadapkan dengan dua opsi, *Open with (Software install)* dan *Save file*. Untuk lebih mudahnya, Anda dapat memilih opsi pertama agar secara otomatis membuka dan menginstalnya dengan mudah.

Sedangkan untuk menginstal file .deb melalui terminal, berikut tutorial instalasi visual code yang harus dilakukan :

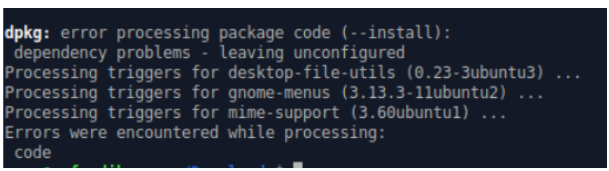
1. Masuk ke folder Downloads

```
cd /home/[user]/Downloads

```

Atau buka folder Downloads melalui File Manager, kemudian klik kanan pilih Open Terminal Here.

```
sudo dpkg -i nama-file-vscode.deb

```

Jika muncul pesan error seperti berikut:



Gambar 7.12 Tutorial instalasi Visual Studio Code

2. Ketik perintah berikut untuk menyelesaikan:

```
sudo apt install libgconf-2-4

```

```
sudo apt --fix-broken install

```

3. Kemudian ulang langkah dpkg :

```
sudo dpkg -i nama-file-vscode.deb
```

7.6.2 Sublime Text

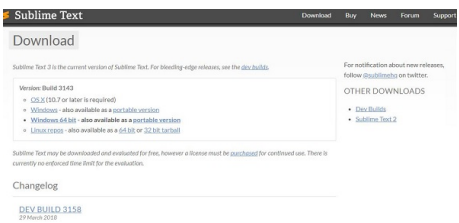
Sublime dapat diunduh dari Website resmi. Versi 3 saat ini tersedia dalam versi beta untuk OS X, Windows dan Ubuntu. Ada juga versi portabel yang dapat dengan mudah dijalankan dari stik USB. Anda dapat mengunduh Teks Luhur secara gratis, namun secara resmi tidak. Setelah periode evaluasi Anda seharusnya mendapatkan lisensi untuk penggunaan berkelanjutan. Namun, halaman unduhan itu sendiri menyatakan bahwa tidak ada batasan waktu untuk evaluasi yang membuatnya lebih bersifat sukarela. Jika Anda memutuskan untuk mendapatkan lisensi itu akan dikenakan biaya 70 US dolar. Setelah memilih versi yang Anda inginkan, unduhan dilakukan dengan cepat. Sublime cukup kecil, arsip instalasi adalah 8MB dan setelah instalasi program hanya berjumlah 22MB dan berubah.

7.6.2.1 Instalasi

Berikut tata cara instalasi sublime text pada sistem operasi windows :

1. Unduh paket .exe dari Website resmi seperti yang ditunjukkan di bawah ini

$$https://www.sublimetext.com/3$$

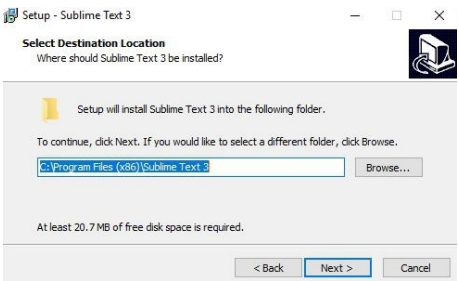


Gambar 7.13 Tutorial instalasi Sublime

2. Lalu, jalankan file yang dapat dieksekusi. Ini mendefinisikan variabel lingkungan. Ketika Anda menjalankan file yang dapat dieksekusi, Anda dapat mengamati jendela berikut di layar Anda. Klik Selanjutnya.

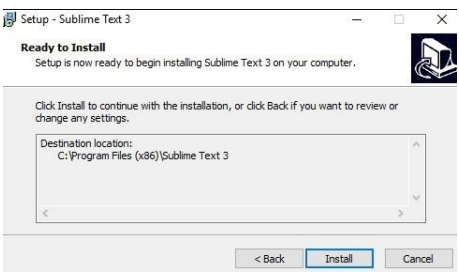
Gambar 7.14 Tutorial instalasi Sublime

3. Kemudian, pilih lokasi tujuan untuk menginstal Sublime Text3 dan klik Next.



Gambar 7.15 Tutorial instalasi Sublime

4. Verifikasi folder tujuan dan klik Instal.

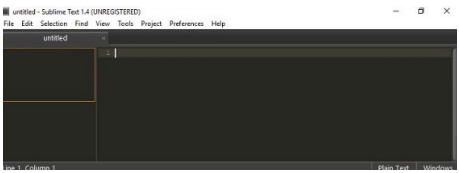


Gambar 7.16 Tutorial instalasi Sublime

5. Selanjutnya, klik Selesai untuk menyelesaikan instalasi.

Gambar 7.17 Tutorial instalasi Sublime

6. Setelah instalasi selesai, saatnya untuk startup pertama. Ketika Anda melakukannya, Anda melihat layar berikut:



Gambar 7.18 Tutorial instalasi Sublime

Anda harus mengikuti langkah-langkah yang ditunjukkan di bawah ini untuk menginstal Sublime Text pada distribusi Linux

1. Buka terminal melalui Ctrl + Alt + T atau dengan mencari ”Terminal” dari peluncur aplikasi desktop. Kemudian, instal paket untuk editor Teks Sublime, menggunakan perintah yang diberikan di bawah ini.

```
sudo add-apt-repository ppa:webupd8team/Sublime-Text-3

```

2. Perbarui paket menggunakan perintah berikut.

```
sudo apt-get update

```

3. Instal repositori Teks Sublim dengan menggunakan perintah berikut.

```
sudo apt-get install Sublime-Text

```

7.6.2.2 Fitur dan Keuntungan

Berikut beberapa fitur serta keuntungan menggunakan sublime teks :

1. *Autocompletion, Syntax Highlight, Code Folding*
   Dari desain umum, Atom dan Sublime Text sangat mirip. Ini tidak mengherankan karena banyak orang berpikir bahwa Atom sebenarnya berdasarkan pada Sublime. Karena alasan itu, juga tidak mengejutkan bahwa keduanya disertai dengan pelengkapan otomatis kode dan sorotan sintaksis. Meskipun sebenarnya, mungkin tidak ada editor kode modern yang layak melakukannya. Sublime memiliki dukungan asli untuk beberapa lusin bahasa pemrograman dan akan mewarnai elemen kode dengan tepat (bahkan memiliki beberapa tema warna yang berbeda - lebih dari itu di bawah). Editor kode bahkan akan memperingatkan Anda tentang kesalahan parsing secara real time! Selain itu, Sublime dilengkapi dengan pelengkapan otomatis kode - termasuk untuk variabel yang dibuat pengguna - dan pelipatan kode. Yang terakhir membuat potongan kode besar lebih kompak untuk dilihat dengan menyembunyikan bagian-bagiannya.

2. *Customizability*
   Seperti Atom, Sublime Text dapat sepenuhnya dikustomisasi melalui plugin. Package Manager yang terintegrasi dengan cepat memungkinkan pengguna untuk menemukan, menginstal, memutakhirkan, dan menghapus plugin langsung dari dalam editor - biasanya tanpa perlu restart. Pengguna dapat mengunduh paket dari Github dan BitBucket serta PackageControl.io. Yang terakhir adalah repositori milik Sublime sendiri di mana komunitas di belakang editor kode menerbitkan banyak plugin untuk meningkatkan kemampuannya. Semua dalam semua, ada lebih dari 3.800 paket yang tersedia dipesan oleh label baru, tren, baru-baru ini diperbarui, populer dan lainnya. Dengan bantuan mereka, Anda dapat mengubah Sublime dari editor kode sederhana menjadi IDE lengkap dan membangun persis jenis lingkungan pengembangan yang Anda butuhkan. Misalnya, Anda dapat menambahkan kemampuan untuk menyinkronkan file pada server FTP jarak jauh, kode pratinjau yang ditulis dalam Markdown dan kontrol bilah sisi lanjutan. Seperti direktori plugin WordPress, paket menyertakan informasi luas tentang fungsionalitasnya, instalasi dan penggunaannya termasuk tangkapan layar. Satu-satunya downside: Informasi itu tidak tersedia di dalam Sublime Text itu sendiri, hanya di Website. Di sisi lain, instalasi melalui Kontrol Paket sangat cepat dan Anda dapat mengontrol pengaturan untuk semua paket di bawah Pengaturan¿ Pengaturan Paket. Tema juga tersedia sebagai paket tambahan. Itu merupakan tambahan dari 22 tema pra-instal yang ditawarkan Sublime di luar kotak. Masing-masing memungkinkan pengguna untuk mengubah skema warna kode mereka dengan satu sentuhan tombol. Namun, berbeda dengan Atom, seluruh UI tetap seperti apa adanya.

3. *Lightweight, Fast and Stable*
   Sesuatu yang dengan cepat menjadi jelas ketika menggunakan Sublime adalah

seberapa cepat programnya. Dibangun dengan C / C ++ dan Python, editor kode sangat ringan. Bahkan, ketika memilihnya dari menu mulai Windows, itu memuat hampir secara instan. Berbeda dengan itu, Atom membutuhkan waktu beberapa detik untuk muncul di layar. Perbedaan ini juga terlihat saat menangani file besar. Sublime Text dapat melakukan lebih banyak tugas berat tanpa mengalami masalah. Dalam pengujian saya, saya tidak punya masalah sama sekali dan juga tidak dapat menemukan keluhan tentang masalah kinerja online. Bahkan, kestabilannya adalah salah satu alasan yang paling banyak dikutip orang untuk lebih memilih Sublime Text daripada editor lain. Membeku dan macet tampaknya sedikit dan jarang.

4. *Powerful Search*
   Fitur lain yang menonjol dari Sublime adalah fungsi pencarian yang kuat. Fungsi pencarian dan penggantian regulernya dapat mencari ekspresi reguler, melakukan pencarian case sensitif atau mencari seluruh kata. Plus, itu memungkinkan Anda menambahkan hasil pencarian ke buffer untuk digunakan kembali nanti. Selain itu, ada fungsi Goto. Setelah memilihnya, Sublime membuka bilah pencarian. Saat Anda mulai mengetikkan nama file atau direktori di proyek atau folder aktif Anda, editor akan menampilkan opsi yang cocok secepat kilat. Bagian terbaik: ketika Anda memindahkan daftar, Sublime akan secara otomatis membuka file yang disorot di latar belakang sehingga Anda dapat melihat apakah itu yang benar. Ketika Anda menemukannya, klik atau menekan enter akan membuka file di tab baru sehingga Anda dapat segera mulai mengerjakannya. Akhirnya, ada palet perintah. Setiap orang yang tidak ingat cara pintas kata kunci dapat membuka koleksi perintah yang tersedia ini, gunakan pencarian fuzzy untuk menemukan apa yang mereka cari dan jalankan tanpa pernah mengangkat tangan mereka dari keyboard. Peasy mudah.

7.7 *Web Server*

7.7.1 XAMPP

XAMPP adalah perangkat lunak sumber terbuka yang dikembangkan oleh teman-teman Apache. Paket perangkat lunak XAMPP berisi distribusi Apache untuk server Apache, MariaDB, PHP, dan Perl. Dan itu pada dasarnya adalah tuan rumah lokal atau server lokal. Server lokal ini berfungsi di komputer desktop atau laptop Anda sendiri. Penggunaan XAMPP adalah untuk menguji klien atau Website Anda sebelum mengunggahnya ke server web jarak jauh. Perangkat lunak server XAMPP ini memberi Anda lingkungan yang cocok untuk menguji proyek MYSQL, PHP, Apache dan Perl di komputer lokal. Bentuk lengkap XAMPP adalah X singkatan dari *Cross-platform*, (A) server Apache, (M) MariaDB, (P) PHP dan (P) Perl. *Cross-platform* biasanya berarti dapat berjalan di komputer mana saja dengan sistem operasi apa pun.

Proses instalasi XAMPP sangat sederhana dan cepat. Setelah XAMPP diinstal pada komputer lokal, ia bertindak sebagai server lokal atau localhost. Anda da-

pat menguji Website sebelum mengunggahnya ke server web jarak jauh. Perangkat lunak server XAMPP ini memberi Anda lingkungan yang cocok untuk menguji aplikasi MYSQL, PHP, Apache dan Perl di komputer lokal.

Setelah instalasi XAMPP selesai, Anda dapat memulai dan menghentikan setiap modul dengan menggunakan Panel Kontrol XAMPP. Misalnya, menguji aplikasi PHP di komputer Anda, Anda dapat memulai dua modul Apache dan MySQL. Ini akan memungkinkan program PHP dijalankan di komputer Anda. Perangkat lunak XAMPP ini mengemulasi server jauh seperti lingkungan di komputer lokal Anda. Sebagai pengembang aplikasi, Anda perlu menguji aplikasi sebanyak mungkin untuk menemukan dan memperbaiki bug. Jika Anda menguji di lingkungan lokal seperti XAMPP, itu akan mempercepat proses pengembangan Anda. Sebelum XAMPP setiap kali Anda perlu mengunggah file ke server jauh untuk tujuan pengujian. Akan sangat sulit untuk menguji pada server langsung dan dapat dilihat oleh pengunjung Anda. Tetapi di XAMPP Anda dapat dengan mudah menguji dan membuat pembaruan apa pun di localhost Anda. Kapan saja, Anda dapat memperbarui dan menguji di XAMPP. Setelah selesai, Anda dapat mengunggah file baru yang diperbarui ini ke server jarak jauh.

7.7.1.1 Main Tools pada XAMPP

XAMPP berisi alat-alat seperti Apache, MYSQL, PHP, dan Perl.

1. Apache
   Server Apache adalah perangkat lunak bebas sumber terbuka yang pada awalnya dikembangkan oleh sekelompok pengembang perangkat lunak dan sekarang dikelola oleh yayasan perangkat lunak Apache. Apache HTTP adalah server jarak jauh (komputer) jika seseorang meminta file, gambar atau dokumen menggunakan browser mereka, mereka akan melayani file-file itu kepada klien menggunakan server HTTP.

2. MYSQL
   MYSQL adalah perangkat lunak sumber terbuka. Ini sebenarnya adalah sistem manajemen basis data relasional (RDBMS). SQL ini adalah singkatan dari Structured Query Language. Ini adalah RDBMS paling populer dan terbaik yang digunakan untuk mengembangkan berbagai aplikasi perangkat lunak berbasis web. Dengan bantuan MYSQL, dimungkinkan untuk mengatur informasi, mengelola, mengambil, dan memperbarui data kapan pun Anda mau.

3. PHP
   Sebagaimana telah dijelaskan pada bab sebelumnya, bentuk lengkap PHP adalah Hypertext Preprocessor. Ini adalah bahasa skrip sisi server yang membantu Anda membuat Website dinamis. Bahasa ini terutama digunakan untuk membangun aplikasi perangkat lunak berbasis web. Ini adalah perangkat lunak sumber terbuka dan berfungsi baik dengan MYSQL. Apa yang sebenarnya terjadi adalah, kode PHP akan dieksekusi di server dan di sisi browser kode HTML-nya akan ditampilkan.

4. Perl
   Perl biasanya dikatakan sebagai bahasa pemrograman tujuan umum. Bahasa Perl ini ditafsirkan dan sangat dinamis. Sebenarnya, bahasa ini digunakan untuk pengembangan web, pengembangan GUI, administrasi sistem, dll. Perl mampu bekerja dengan HTML, XML, dan bahasa markup lainnya. Di versi terbaru XAMPP, ada alat tambahan seperti Server surat Merkurius, OpenSSL, phpMyAdmin, dll. Dengan alat di atas, Anda dapat membuat server desktop lengkap.

7.7.1.2 Proses Instalasi untuk XAMPP pada Windows

XAMPP telah dirancang untuk menjadi cara termudah untuk menginstal dan menjalankan server pengembangan di komputer lokal Anda. Ada banyak XAMPP lain seperti perangkat lunak yang tersedia, tetapi XAMPP adalah salah satu perangkat lunak paling terkenal. Selain Apache, MySQL, PHP, dan Perl, XAMPP termasuk alat yang sangat berguna lainnya seperti alat administrasi basis data phpMyAdmin, server FileZilla FTP, server surat Merkurius, dan server JSP Tomcat. XAMPP juga dapat menginstal situs administrasi sebagai halaman utama server. Dari mana Anda dapat melakukan segala macam tugas administratif, seperti memeriksa status dan keamanan server, meluncurkan alat seperti analisis phpMyAdmin dan Webalizer.

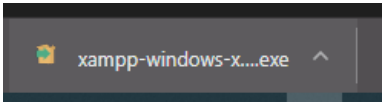
1. Di browser web, Anda dapat dengan mudah mengunduh XAMPP dari

$$http://www.apachefriends.org/$$

   Karena ini merupakan sumber terbuka, Anda dapat mengunduhnya secara gratis.

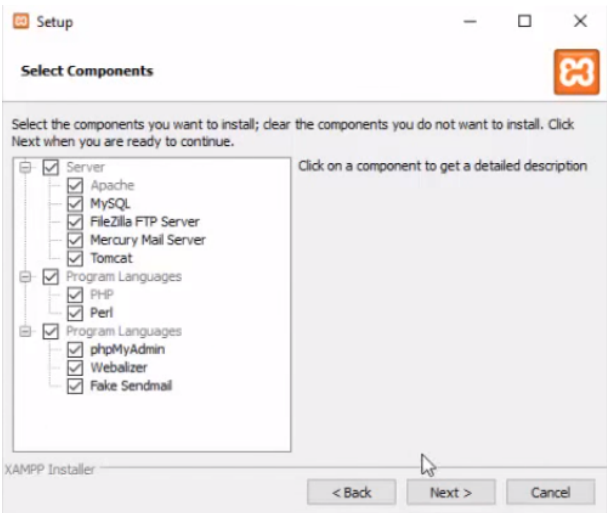
Gambar 7.19 Tutorial instalasi XAMPP

2. Setelah perangkat lunak diunduh, Anda harus menginstal dengan mengklik dua kali file .exe



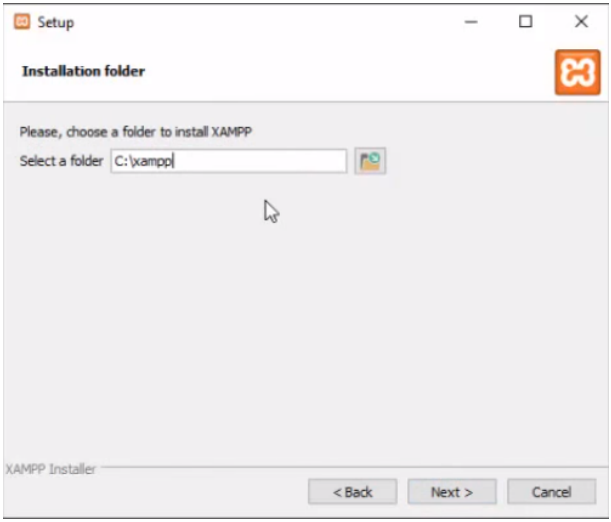
Gambar 7.20 Tutorial instalasi XAMPP

3. Setelah file dieksekusi, jendela pengaturan muncul. Di file pengaturan, pilih komponen yang diperlukan. Sebagai contoh (jika Anda ingin menginstal WordPress di XAMPP, komponen yang diperlukan adalah MySQL, Apache, PHP-MyAdmin).



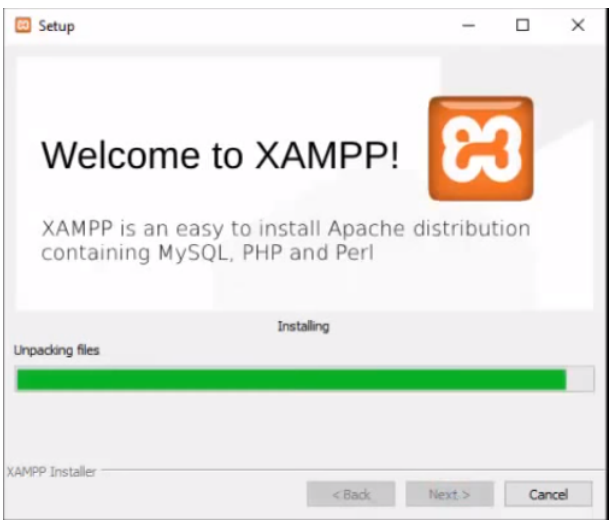
Gambar 7.21 Tutorial instalasi XAMPP

4. Langkah selanjutnya adalah memilih folder tempat file itu berada. Disarankan untuk memilih ”drive C” default dan kemudian klik tombol ”Next”.



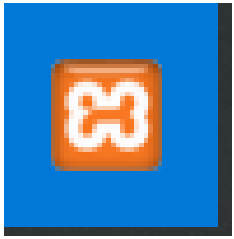
Gambar 7.22 Tutorial instalasi XAMPP

5. Mengklik pada selanjutnya proses instalasi Anda akan dimulai. Wizard pengaturan akan membongkar dan menginstal semua komponen yang dipilih dan akan menyimpannya ke direktori yang ditentukan. Proses instalasi membutuhkan beberapa menit untuk menyelesaikannya.



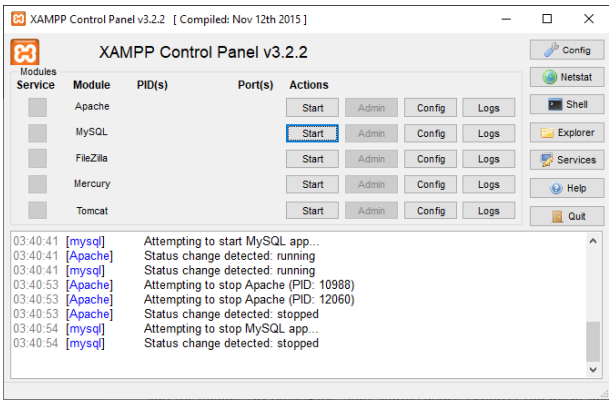
Gambar 7.23 Tutorial instalasi XAMPP

6. Setelah proses instalasi selesai, klik tombol ”Selesai”.



Gambar 7.24 Tutorial instalasi XAMPP

7. Setelah selesai, ikon XAMPP akan muncul di desktop Anda atau menu mulai. Dengan mengklik dua kali ikon XAMPP, jendela panel kontrol XAMPP muncul.



Gambar 7.25 Tutorial instalasi XAMPP

Pada panel kontrol, Anda akan dapat mengontrol masing-masing komponen server teks Anda. Di sini Anda dapat memulai atau menghentikan modul individual dengan mengeklik tombol di bawah Tindakan. Panel kontrol berisi berbagai tombol seperti config, stat net, shell, explorer, layanan, bantuan, keluar.

(a) *Config*, memungkinkan Anda untuk mengonfigurasi XAMPP serta komponen individual

(b) *Netstat*, Membantu untuk menunjukkan semua proses yang sedang berjalan pada sistem lokal

(c) *Shell*, memungkinkan membuka shell UNIX

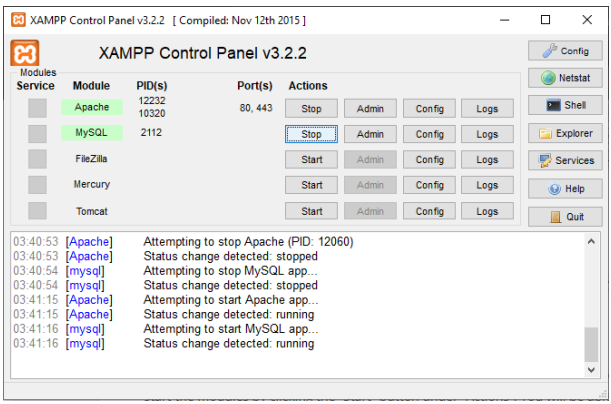
(d) *Explorer*, membuka folder XAMPP di windows explorer

(e) *Services*, Bantuan untuk menunjukkan semua layanan yang saat ini berjalan di back-end

(f) *Help*, Sesuai namanya, memberi Anda tautan ke forum pengguna

(g) *Exit*, Dengan mengklik tombol panel kontrol XAMPP akan keluar

8. Di panel kontrol XAMPP, klik modul yang diperlukan yang diperlukan agar Anda dapat bekerja. Anda dapat memulai modul dengan mengklik tombol "*Start*" di bawah "Actions". Anda akan dapat melihat modul yang diinisiasi disorot dengan warna hijau. Misalnya, jika Anda menggunakan platform WordPress maka komponen yang diperlukan adalah Apache dan MYSQL. Jika Anda tidak dapat memulai modul karena kesalahan, maka akan ditandai dengan font merah. Mereka akan memberi Anda laporan terperinci untuk mengidentifikasi penyebab kesalahan.



Gambar 7.26 Tutorial instalasi XAMPP

selain dapat menginstall secara keseluruhan, kita juga dapat menginstal Apache, PHP dan MariaDB di Windows secara manual. Berikut adalah langkah yang perlu diambil untuk menginstal pengaturan pengembangan web (lokal) di Windows. Kadang langkah ini dilakukan untuk untuk mengubah versi perangkat lunak dari masing-masing komponen dengan lebih mudah dan mendaftarkannya sebagai layanan Windows.

1. Unduh versi perangkat lunak terbaru: Apache, PHP, dan MariaDB. Pilihlah versi PHP yang tepat, pilih *Thread-Safe*, dan arsitektur yang sama dengan Apache.

2. Dipasang atau diekstraksi ke lokasi khusus $C: WebDevTools$.

3. SRVROOTVariabel yang dimodifikasi dalam $conf/httpd.conffile$ apache untuk mencerminkan jalur pemasangan berbeda - $C: WebDevTools\ Apache24$.

4. Juga $conf/httpd.conf$ setelah semua arahan *Load Module* kemudian tambahkan baris berikut untuk menggunakan php versi 7.3 dan mengekstraknya $C: WebDevTools\ php73$.

```
LoadModule php7_module "C:\WebDevTools\php73\
php7apache2_4.dll"
```

```
<IfModule php7_module>
    Aplikasi AddHandler / x-httpd-php .php
    Aplikasi AddType / x-httpd-php .php .html
    PHPIniDir "C: \ WebDevTools \ php73"
</IfModule>
```

5. Dalam folder instalasi php salin $php.ini - development$ ke $php.ini$ dan dalam uncommented $directivedirectory_ext = "ext"$ dan atur nilainya $C : WebDevTools\ php73\ ext$. Dan karena akan mengakses *database* juga mengaktifkan ekstensi yang sesuai $mysqli, pdo_mysql$ dengan membatalkan komentar mereka.

6. Pergi ke direktori trash Apache24, kamudian buka PowerShell sebagai Administrator dan menjalankan perintah $..exe - kinstall$ guna memasang Apache sebagai layanan.

7. Menginstal MariaDB dengan opsi default (tanpa mengubah *password root*).

7.7.2 LAMP

LAMP adalah kombinasi server web dari Apache, MySQL, dan PHP di server Linux. Ini banyak digunakan untuk hosting Website berbasis PHP dan MySQL di *web world wide* (WWW).

7.7.2.1 Fitur LAMP

Berikut beberapa layanan yang terdapat di LAMP :

1. Apache WebServer
2. DBMS mySQL
3. FTP proFTPD

7.7.2.2 Cara Instalasi

Instalasi server LAMP di Linux mudah. Cukup ikuti instruksi ini:

1. Buka terminal dan ketik:

```
    sudo apt-get install apache2 php5-mysql libapache2-mod-
php5 mysql-server phpmyadmin php5-curl
```

2. Katakan ya [Y] ketika manajer paket meminta Anda mengunduh dan menginstal paket. Langkah ini akan memakan waktu tergantung pada kecepatan koneksi Anda.

3. Pada titik tertentu, penginstal akan menanyakan kata sandi root MySQL Anda. Gunakan kata sandi yang Anda suka. Untuk contoh ini kita akan menggunakan ”myadmin”.

4. Pemasang akan meminta ”server Web yang harus dikonfigurasi secara otomatis untuk menjalankan phpmyadmin”. Tekan [bilah spasi] untuk memilih ”apache2” dan tekan [enter]. **CATATAN**: Pastikan pemilihan ditandai dengan tanda bintang [*].

5. Pemasang akan meminta ”Konfigurasikan basis data untuk PHPMyAdmin dengan dbconfig-common”. Pilih ”¡Ya¿” dan tekan [enter].

6. Pemasang akan meminta ”kata sandi pengguna administratif basis data”. Gunakan kata sandi apa pun yang Anda suka, tetapi untuk contoh ini kita akan menggunakan ”myadmin”.

7. Pemasang akan meminta ”kata sandi aplikasi mysql untuk PHPMyAdmin”. Gunakan kata sandi apa pun yang Anda suka, tetapi untuk contoh ini kita akan menggunakan ”myadmin”.

8. Jika tidak ada kesalahan yang ditampilkan, instalasi selesai.

selain cara instalasi tersebut, terdapat juga cara instalasi lain dengan menginstal masing-masing komponen secara terpisah seperti berikut :

1. Menginstal Apache dan Memperbarui *Firewall*
   Server web Apache adalah salah satu server web paling populer di dunia. Ini didokumentasikan dengan baik dan telah digunakan secara luas untuk sebagian besar sejarah web, yang menjadikannya pilihan default yang bagus untuk hosting Website.
   Install Apache menggunakan manajer paket Debian, apt:

```
    $ sudo apt update
    $ sudo apt install apache2

```

   Karena ini adalah sudoperintah, operasi ini dijalankan dengan hak akses root. Ini akan menanyakan kata sandi pengguna reguler Anda untuk memverifikasi niat Anda.
   Setelah memasukkan kata sandi, Anda aptakan diberi tahu paket mana yang akan dipasang dan berapa banyak ruang disk tambahan yang akan digunakan. Tekan Ydan tekan ENTERuntuk melanjutkan, dan instalasi akan dilanjutkan.
   Selanjutnya, dengan asumsi bahwa Anda telah mengikuti instruksi pengaturan server awal dengan menginstal dan mengaktifkan firewall UFW, pastikan firewall Anda memungkinkan lalu lintas HTTP dan HTTPS.
   Ketika diinstal pada Debian 9, UFW hadir dengan profil aplikasi yang dapat Anda gunakan untuk mengubah pengaturan firewall Anda. Lihat daftar lengkap profil aplikasi dengan menjalankan:

```
$ sudo ufw app list

```

The WWW profil yang digunakan untuk mengelola pelabuhan yang digunakan oleh server web:

```
Output
Available applications:
. . .
WWW
WWW Cache
WWW Full
WWW Secure
. . .

```

Jika Anda memeriksa WWW Fullprofil, itu menunjukkan bahwa itu memungkinkan lalu lintas ke port 80dan 443:

```
$ sudo ufw app info "WWW Full"

```

```
Output
Profile: WWW Full
Title: Web Server (HTTP,HTTPS)
Description: Web Server (HTTP,HTTPS)

Ports:
80,443/tcp

```

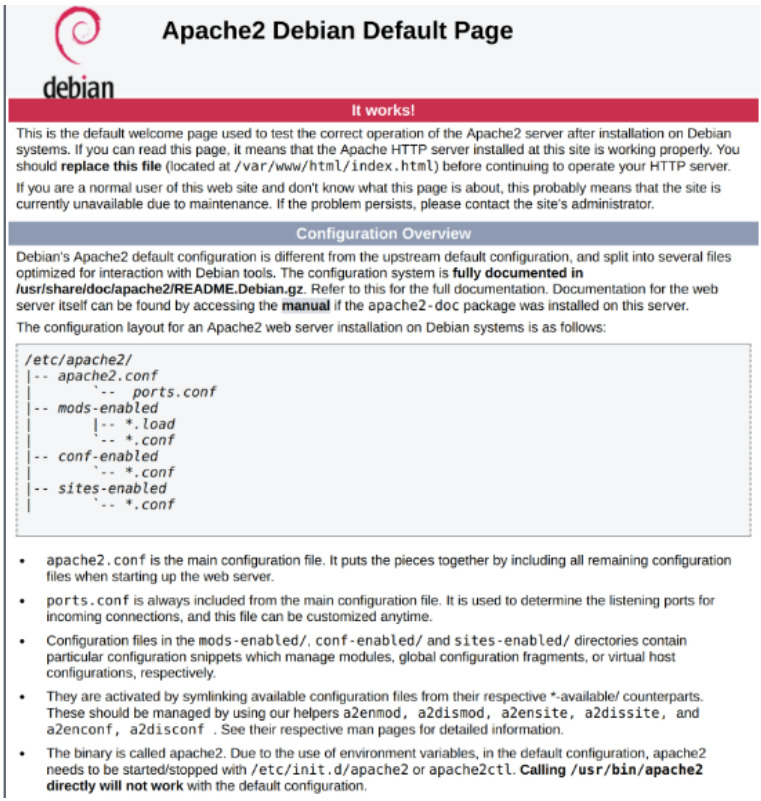
Izinkan lalu lintas HTTP dan HTTPS yang masuk untuk profil ini:

```
$ sudo ufw allow in "WWW Full"

```

Anda dapat melakukan pemeriksaan langsung untuk memverifikasi bahwa semuanya berjalan sesuai rencana dengan mengunjungi alamat IP publik server Anda di browser web Anda:

```
http://your_server_ip

```

Anda akan melihat halaman web Debian 9 Apache default, yang ada untuk tujuan informasi dan pengujian. Seharusnya terlihat seperti ini:



Apache2 Debian Default Page

debian

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Gambar 7.27 Pengujian Apche2

Jika Anda melihat halaman ini, maka server web Anda sekarang terpasang dengan benar dan dapat diakses melalui firewall Anda.

Jika Anda tidak tahu apa alamat IP publik server Anda, ada sejumlah cara untuk menemukannya. Biasanya, ini adalah alamat yang Anda gunakan untuk terhubung ke server Anda melalui SSH.

Ada beberapa cara berbeda untuk melakukan ini dari baris perintah. Pertama, Anda bisa menggunakan iproute2alat untuk mendapatkan alamat IP Anda dengan mengetik ini:

```
$ ip addr show eth0 | grep inet | awk '{ print $2; }' | sed 's/\/.*$//'
```

Ini akan memberi Anda dua atau tiga baris kembali. Semua itu adalah alamat yang benar, tetapi komputer Anda mungkin hanya dapat menggunakan salah satunya, jadi silakan mencoba masing-masing.

Metode alternatif adalah menggunakan curlutilitas untuk menghubungi pihak

luar untuk memberi tahu Anda bagaimana ia melihat server Anda. Ini dilakukan dengan menanyakan pada server tertentu apa alamat IP Anda:

```
$ sudo apt install curl
$ curl http://icanhazip.com

```

pa pun metode yang Anda gunakan untuk mendapatkan alamat IP, ketikkan itu di bilah alamat browser web Anda untuk melihat halaman Apache default.

2. Menginstal MariaDB
Sekarang setelah server web Anda aktif dan berjalan, sekarang saatnya untuk menginstal MariaDB. MariaDB adalah sistem manajemen basis data. Pada dasarnya, ini akan mengatur dan menyediakan akses ke database tempat situs Anda dapat menyimpan informasi.
MariaDB adalah fork yang dibangun komunitas dari MySQL. Dalam Debian 9, server MySQL default adalah MariaDB 10.1, dan mysql-serverpaket, yang biasanya digunakan untuk menginstal MySQL, adalah paket transisi yang benar-benar akan menginstal MariaDB. Namun, Anda disarankan untuk menginstal MariaDB menggunakan paket aktual program mariadb-server,.
Sekali lagi, gunakan aptuntuk mendapatkan dan menginstal perangkat lunak ini:

```
$ sudo apt install mariadb-server

```

> **Catatan** : Dalam hal ini, Anda tidak harus menjalankan sudo apt update sebelum perintah. Ini karena Anda baru-baru ini menjalankannya di perintah di atas untuk menginstal Apache, dan indeks paket di komputer Anda harus sudah terbaru.

Perintah ini juga akan menampilkan daftar paket yang akan diinstal, bersama dengan jumlah ruang disk yang akan digunakan. Masuk Yuntuk melanjutkan.
Ketika instalasi selesai, jalankan skrip keamanan sederhana yang datang pra-instal dengan MariaDB yang akan menghapus beberapa pengaturan default tidak aman dan mengunci akses ke sistem database Anda. Mulai skrip interaktif dengan menjalankan:

```
$ sudo mysql_secure_installation

```

Ini akan membawa Anda melalui serangkaian konfirmasi di mana Anda dapat membuat beberapa perubahan pada opsi keamanan instalasi MariaDB Anda. Prompt pertama akan meminta Anda untuk memasukkan kata sandi root database saat ini . Ini adalah akun administratif di MariaDB yang telah meningkatkan hak istimewa. Anggap saja mirip dengan akun root untuk server itu sendiri (walaupun yang Anda konfigurasikan sekarang adalah akun khusus MariaDB). Karena Anda baru saja menginstal MariaDB dan belum melakukan perubahan

konfigurasi, kata sandi ini akan kosong, jadi cukup tekan ENTER saat diminta. Prompt berikutnya menanyakan apakah Anda ingin mengatur kata sandi root basis data . Ketik Nlalu tekan ENTER. Di Debian, akun root untuk MariaDB terkait erat dengan pemeliharaan sistem otomatis, jadi kami tidak boleh mengubah metode otentikasi yang dikonfigurasi untuk akun itu. Melakukan hal itu akan memungkinkan pembaruan paket untuk memutus sistem database dengan menghapus akses ke akun administratif. Nantinya, kami akan membahas cara mengatur akun administratif tambahan untuk akses kata sandi jika otentikasi soket tidak sesuai untuk kasus penggunaan Anda.

Dari sana, Anda dapat menekan Ydan kemudian ENTERmenerima default untuk semua pertanyaan berikutnya. Ini akan menghapus beberapa pengguna anonim dan basis data pengujian, menonaktifkan login root jarak jauh , dan memuat aturan baru ini sehingga MariaDB segera menghormati perubahan yang telah Anda buat.

Dalam pemasangan baru pada sistem Debian, pengguna root MariaDB diatur untuk mengotentikasi menggunakan $unix_socket$ plugin secara default daripada dengan kata sandi. Ini memungkinkan beberapa keamanan dan kegunaan yang lebih besar dalam banyak kasus, tetapi juga dapat mempersulit hal-hal ketika Anda perlu mengizinkan program eksternal (misalnya, phpMyAdmin) hak administratif.

Karena server menggunakan akun root untuk tugas-tugas seperti rotasi log dan memulai dan menghentikan server, yang terbaik adalah tidak mengubah rincian otentikasi akun root . Mengubah kredensial akun pada $/etc/mysql/debian.cnf$ awalnya mungkin berhasil, tetapi pembaruan paket berpotensi menimpa perubahan tersebut. Alih-alih memodifikasi akun root , pengelola paket menyarankan untuk membuat akun administratif terpisah jika Anda perlu mengatur akses berbasis kata sandi.

Untuk melakukannya, kami akan membuat akun baru yang disebut admindengan kemampuan yang sama dengan akun root , tetapi dikonfigurasi untuk otentikasi kata sandi. Untuk melakukan ini, buka prompt MariaDB dari terminal Anda:

```
$ sudo mariadb

```

Sekarang, kita dapat membuat pengguna baru dengan hak akses root dan akses berbasis kata sandi. Ubah nama pengguna dan kata sandi agar sesuai dengan preferensi Anda:

```
MariaDB [(none)]> GRANT ALL ON *.* TO 'admin'@'localhost'
IDENTIFIED BY 'password' WITH GRANT OPTION;

```

Flush hak istimewa untuk memastikan bahwa mereka disimpan dan tersedia di sesi saat ini:

```
MariaDB [(none)]> FLUSH PRIVILEGES;

```

Setelah ini, keluar dari shell MariaDB:

```
MariaDB [(none)]> exit

```

Sekarang, setiap kali Anda ingin mengakses database Anda sebagai pengguna administratif baru, Anda harus mengautentikasi sebagai pengguna dengan kata sandi yang baru saja Anda atur menggunakan perintah berikut:

```
$ mariadb -u admin -p

```

Pada titik ini, sistem basis data Anda sudah diatur dan Anda dapat melanjutkan untuk menginstal PHP, komponen terakhir dari tumpukan LAMP.

3. Menginstal PHP
PHP adalah komponen pengaturan Anda yang akan memproses kode untuk menampilkan konten dinamis. Itu dapat menjalankan skrip, terhubung ke database MariaDB Anda untuk mendapatkan informasi, dan menyerahkan konten yang diproses ke server web Anda untuk ditampilkan.
Sekali lagi, manfaatkan aptsistem untuk menginstal PHP. Selain itu, sertakan beberapa paket pembantu saat ini sehingga kode PHP dapat berjalan di bawah server Apache dan berbicara dengan database MariaDB Anda:

```
$ sudo apt install php libapache2-mod-php php-mysql

```

Ini harus menginstal PHP tanpa masalah. Kami akan menguji ini sebentar lagi. Dalam kebanyakan kasus, Anda ingin memodifikasi cara Apache menyajikan file ketika direktori diminta. Saat ini, jika pengguna meminta direktori dari server, Apache akan mencari file bernama $index.html$. Kami ingin memberi tahu server web untuk lebih memilih file PHP daripada yang lain, jadi buat Apache mencari $index.php$ file terlebih dahulu.
Untuk melakukan ini, ketik perintah ini untuk membuka dir.conffile dalam editor teks dengan hak akses root:

```
sudo nano /etc/apache2/mods-enabled/dir.conf

```

Ini akan terlihat seperti ini:

```
<IfModule mod_dir.c>
DirectoryIndex index.html index.cgi index.pl index.php index.
xhtml index.htm
</IfModule>

```

Pindahkan file indeks PHP (disorot di atas) ke posisi pertama setelah DirectoryIndexspesifikasi, seperti ini:

```
<IfModule mod_dir.c>
DirectoryIndex index.php index.html index.cgi index.pl index.
xhtml index.htm
</IfModule>

```

Setelah selesai, simpan dan tutup file dengan menekan CTRL+X. Konfirmasikan simpan dengan mengetik Ydan kemudian tekan ENTERuntuk memverifikasi file simpan lokasi.
Setelah ini, mulai ulang server web Apache agar perubahan Anda dikenali. Lakukan ini dengan mengetik ini:

```
$ sudo systemctl restart apache2

```

Anda juga dapat memeriksa status apache2layanan menggunakan systemctl:

```
sudo systemctl status apache2

```

Kemudian akan mengeluarka *output* seperti berikut :

```
        apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled;
     vendor preset: enabled)
   Active: active (running) since Tue 2018-09-04 18:23:03 UTC; 9s
     ago
  Process: 22209 ExecStop=/usr/sbin/apachectl stop (code=exited,
     status=0/SUCCESS)
  Process: 22216 ExecStart=/usr/sbin/apachectl start (code=exited
     , status=0/SUCCESS)
 Main PID: 22221 (apache2)
    Tasks: 6 (limit: 4915)
   CGroup: /system.slice/apache2.service
               22221 /usr/sbin/apache2 -k start
               22222 /usr/sbin/apache2 -k start
               22223 /usr/sbin/apache2 -k start
               22224 /usr/sbin/apache2 -k start
               22225 /usr/sbin/apache2 -k start
               22226 /usr/sbin/apache2 -k start

```

Untuk meningkatkan fungsionalitas PHP, Anda memiliki opsi untuk menginstal beberapa modul tambahan. Untuk melihat opsi yang tersedia untuk modul dan pustaka PHP, kirimkan hasil apt search ke less, pager yang memungkinkan Anda menggulir output dari perintah lain:

```
apt search php- | less

```

Gunakan tombol panah untuk menggulir ke atas dan ke bawah, dan tekan Quntuk keluar.
Hasilnya semua komponen opsional yang dapat Anda instal. Ini akan memberi Anda deskripsi singkat untuk masing-masing:

```
Sorting...
Full Text Search...
bandwidthd-pgsql/stable 2.0.1+cvs20090917-10 amd64
Tracks usage of TCP/IP and builds html files with graphs

bluefish/stable 2.2.9-1+b1 amd64
```

```
        advanced Gtk+ text editor for web and software
    development

    cacti/stable 0.8.8h+ds1-10 all
        web interface for graphing of monitoring systems

    cakephp-scripts/stable 2.8.5-1 all
        rapid application development framework for PHP (scripts)

    ganglia-webfrontend/stable 3.6.1-3 all
        cluster monitoring toolkit - web front-end

    haserl/stable 0.9.35-2+b1 amd64
        CGI scripting program for embedded environments

    kdevelop-php-docs/stable 5.0.3-1 all
        transitional package for kdevelop-php

    kdevelop-php-docs-l10n/stable 5.0.3-1 all
        transitional package for kdevelop-php-l10n

    :

```

Untuk mempelajari lebih lanjut tentang apa yang dilakukan setiap modul, Anda dapat mencari di internet untuk informasi lebih lanjut tentang mereka. Atau, lihat deskripsi panjang paket dengan mengetik:

```
    $ apt show package_name

```

Akan ada banyak output, dengan satu bidang yang disebut Descriptionyang akan memiliki penjelasan lebih lama tentang fungsi yang disediakan modul. Misalnya, untuk mengetahui apa yang dilakukan php-climodul, Anda bisa mengetik ini:

```
    $ apt show php-cli

```

Seiring dengan sejumlah besar informasi lain, Anda akan menemukan sesuatu yang terlihat seperti ini:

```

    Description: command-line interpreter for the PHP scripting
    language (default)
    This package provides the /usr/bin/php command interpreter,
    useful for
    testing PHP scripts from a shell or performing general shell
    scripting tasks.
    .
    PHP (recursive acronym for PHP: Hypertext Preprocessor) is a
    widely-used
    open source general-purpose scripting language that is
    especially suited
    for web development and can be embedded into HTML.
```

```
.
This package is a dependency package, which depends on Debian's default
PHP version (currently 7.0).


```

Jika, setelah meneliti, Anda memutuskan untuk menginstal paket, Anda dapat melakukannya dengan menggunakan apt installperintah seperti yang telah Anda lakukan untuk perangkat lunak lain.
Jika Anda memutuskan itu php-cliadalah sesuatu yang Anda butuhkan, Anda dapat mengetik:

```
$ sudo apt install php-cli

```

Jika Anda ingin menginstal lebih dari satu modul, Anda dapat melakukannya dengan mendaftar masing-masing, dipisahkan oleh spasi, mengikuti apt installperintah, seperti ini:

```
$ sudo apt install package1 package2 ...

```

Pada titik ini, tumpukan LAMP Anda diinstal dan dikonfigurasi. Namun, sebelum membuat perubahan lagi atau menggunakan aplikasi, akan sangat membantu untuk secara proaktif menguji konfigurasi PHP Anda jika ada masalah yang harus diatasi.

4. Menguji Pemrosesan PHP di Server Web Anda Untuk menguji apakah sistem Anda dikonfigurasikan dengan benar untuk PHP, buat skrip PHP yang sangat mendasar yang disebut info.php. Agar Apache menemukan file ini dan menyajikannya dengan benar, itu harus disimpan ke direktori yang sangat spesifik yang disebut web root .
Di Debian 9, direktori ini terletak di $/var/www/html/$. Buat file di lokasi itu dengan menjalankan:

```
$ sudo nano /var/www/html/info.php

```

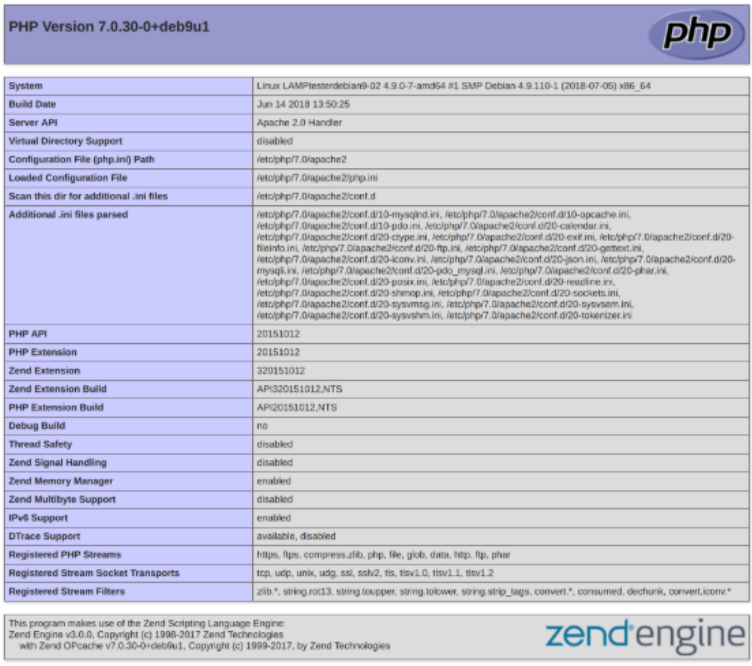
Ini akan membuka file kosong. Tambahkan teks berikut, yang merupakan kode PHP yang valid, di dalam file:

```
<?php
phpinfo();
?>

```

Setelah selesai, simpan dan tutup file.
Sekarang Anda dapat menguji apakah server web Anda dapat menampilkan konten yang dihasilkan oleh skrip PHP ini dengan benar. Untuk mencoba ini, kunjungi halaman ini di browser web Anda. Anda akan memerlukan alamat IP publik server Anda lagi.
Alamat yang ingin Anda kunjungi adalah:

```
http://your_server_ip/info.php
```

Halaman yang Anda datangi akan terlihat seperti ini:



PHP Version 7.0.30-0+deb9u1

System	Linux LAMPtesterdebian9-02 4.9.0-7-amd64 #1 SMP Debian 4.9.110-1 (2018-07-05) x86_64
Build Date	Jun 14 2018 13:50:25
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-mysqli.ini, /etc/php/7.0/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS
PHP Extension Build	API20151012,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv2, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.0.0, Copyright (c) 1998-2017 Zend Technologies
with Zend OPcache v7.0.30-0+deb9u1, Copyright (c) 1999-2017, by Zend Technologies

Gambar 7.28 Informasi PHP

Halaman ini menyediakan beberapa informasi dasar tentang server Anda dari perspektif PHP. Ini berguna untuk debugging dan untuk memastikan bahwa pengaturan Anda diterapkan dengan benar.

Jika Anda dapat melihat halaman ini di browser Anda, maka PHP Anda berfungsi seperti yang diharapkan.

Anda mungkin ingin menghapus file ini setelah tes ini karena sebenarnya bisa memberikan informasi tentang server Anda kepada pengguna yang tidak sah. Untuk melakukan ini, jalankan perintah berikut:

```
$ sudo rm /var/www/html/info.php
```

Anda selalu dapat membuat kembali halaman ini jika Anda perlu mengakses informasinya lagi nanti.

Sekarang setelah Anda memiliki tumpukan LAMP yang terinstal, Anda memiliki banyak pilihan untuk apa yang harus dilakukan selanjutnya. Pada dasarnya, Anda telah menginstal platform yang memungkinkan Anda menginstal sebagian besar Website dan perangkat lunak web di server Anda.

7.7.3 Perbedaan *Auto Installer* dengan *Manually Installing*

Auto Installer memiliki fitur hebat untuk pemula tanpa pengetahuan teknis yang diperlukan, sayangnya ia memiliki beberapa kelemahan karena pengaturan instalasi *default*. Nama basis data dapat dengan mudah ditebak serta pengguna basis data dan awalan basis data, karena penginstal otomatis menggunakan awalan basis data yang sama untuk semua instalasi dan polanya mudah ditebak.

Ada beberapa perbedaan dalam hal fitur jika Anda membandingkan xampp dengan instalasi manual. Katakanlah Anda sedang mengembangkan di lokal Anda dan daripada menggunakan situs Anda menggunakan ftp. Juga selama pengembangan Anda harus memeriksa email. Jadi jika Anda menggunakan xampp.

- Terlepas dari tumpukan LAMP, Anda memiliki server filezilla sehingga Anda tidak perlu menginstal secara terpisah.

- Anda dapat menguji email Anda di lingkungan lokal dengan mengkonfigurasi server surat merkuri yang menjadi default dengan xampp.

- Anda telah menginstal phpMyAdmin sehingga Anda tidak harus menginstalnya secara terpisah.

Jadi, jika Anda seorang pemula, daripada menggunakan xampp akan menjadi ide yang baik tetapi ketika Anda maju dalam karir Anda, Anda harus tahu cara mengkonfigurasi mereka secara manual karena itu akan membantu Anda mempelajari konfigurasi secara mendalam. Setelah Anda mulai meletakkan situs Anda di server langsung akan ada konfigurasi manual sehingga Anda harus tahu cara mengkonfigurasinya secara manual.

Untuk semua pengembang PHP, pengaturan lingkungan lokal adalah praktik umum. XAMPP pada Windows membatasi pengembang atau *developer* untuk meningkatkan keterampilannya.

- Menggunakan XAMPP berarti Anda berbagi lingkungan di antara semua proyek Anda yang meninggalkan banyak bug potensial.

- Menggunakan XAMPP berarti Anda menghemat waktu pengaturan lingkungan tetapi membuat dinding bata ke dunia Linux.

- Menggunakan XAMPP berarti Anda tidak sering berlatih Perintah Linux, bahkan Anda tidak akan pernah melakukannya ketika Anda menggunakan Website Anda melalui FTP pada host bersama dengan panel kontrol dukungan.

- Menggunakan XAMPP berarti Anda tidak bekerja pada sistem yang sama dengan server produksi.

- Menggunakan XAMPP berarti Anda akan cukup sering menggunakan PHP-MyAdmin untuk praktik administrasi basis data. Untuk pengalaman saya, ini bukan praktik terbaik. itu menghabiskan banyak waktu saya mengimpor database

besar tidak berhasil untuk mengaktifkan proyek baru yang saya bisa menyelesaikannya dengan cepat tanpa gangguan dengan menggunakan perintah impor mysql.

- Menggunakan XAMPP berarti Anda akan bergantung padanya untuk memperbarui perpustakaan baru, pembaruan dari komunitas LAMP yang dinamis. Seperti halnya menguji kinerja WordPress atau Magento di PHP7.

Ketika Anda menginstal Apache melalui Xampp, kita dapat mengaktifkan dan menonaktifkannya dengan satu klik, selain itu membantu memuat komputer lebih cepat, menggunakan lebih sedikit sumber daya, karena ketika Anda tidak menggunakan maka Anda dapat mematikan server Apache atau layanan mysql. Ketika menginstal melalui Xampp Anda menginstal LAMP Stack dalam 1 klik, di sini LAMP adalah singkatan dari Apache Mysql dan PHP plus PHPMyAdmin, Anda tidak perlu menginstal Mysql atau PHP atau PHPMyAdmin sendiri, mereka dapat diakses melalui panel kontrol XAMPP. Untuk melihat file dalam 127.0.0.1 atau localhost, yang perlu Anda lakukan adalah menambahkan file html atau php ke folder htdocs yang secara harfiah dalam direktori instalasi folder Xampp.

Dalam kasus instalasi Apache manual, Anda pada dasarnya memiliki Apache yang diinstal pada komputer dan ketika Anda menyalakan komputer dan OS boot, server apache dimulai secara otomatis, maka Anda hanya perlu mengetikkan localhost di browser web. Setelah menginstal Apache kita harus menginstal PHP, MySql dan PHPMyAdmin sendiri. Untuk melihat file dalam 127.0.0.1 atau localhost, yang perlu Anda lakukan adalah menambahkan file html atau php ke folder / var / www / html /.

Auto installer direkomendasikan untuk instalasi pada laptop atau PC, sedangkan untuk *Manually Installing* disarankan untuk menginstal server Apache di VPS *(Virtual Private Server)* untuk meng-*host* Website melalui internet

7.8 Basis Data (*Database*)

Basis data merupakan sekumpulan informasi yang saling berkaitan pada suatu subjek tertentu pada tujuan tertentu pula. Basis data adalah susunan *record* data operasional lengkap dari suatu organisasi atau perusahaan, yang diorganisir dan disimpan secara terintegrasi dengan menggunakan metode tertentu dalam komputer sehingga mampu memenuhi informasi yang optimal yang dibutuhkan oleh para pengguna banyak sekali kegiatan manusia yang menggunakan komputer sebagai sarana pengolahan data, sehingga diperlukan suatu perangkat lunak *database*. Jika dikaji lebih mendasar tentang batasan suatu basis data, maka dapat disebutkan bahwa segala bentuk koleksi data adalah suatu basis data. Mulai dari kelompok data pegawai, sampai dengan kelompok *file*, merupakan basis data.

Basis data terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah resepresentasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa dan sebagainya, yang direkam dalam bentuk angka, huruf,

symbol, teks, gambar, bunyi, atau kombinasi. Basis Data sendiri dapat didefinisikan dalam jumlah sudut pandang seperti:

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

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. prinsip utamanya adalah pengaturan data/arsip dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Perbedaanya hanya terletak pada media penyimpanan yang digunakan. Jika lemari arsip menggunakan lemari dari besi dan kayu sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektonik seperti *disk* (disket atau *harddisk*). Hal merupakan konsekuensi yang logis, karena lemari arsip langsung dikelola/ditangani langsung oleh manusia, sementara basis data dikelola/ditangai memalui perantara alat/mesin pintar elektronika (yang kita kenal sebagai komputer).

Perbedaan media ini yang selanjutnya melahirkan perbedaan perbedaan yang lainya menyangkut jumlah dan jenis metode/cara yang dapat digunakan dalam upaya penyimpanan. Satu hal yang juga diperhatikan, bahwa basis data bukanya hanya sekedar penyimpanan data secara elektronis (dengan menggunakan komputer), artinya tidak semua bentuk penyimpanan data secara elektoronis bisa disebut basis data. Kita dapat menyimpan dokumen berisi data dalam *file* teks (dengan program pengolah kata), *file spread sheet* dan lain-lain tetapi tidak bisa disebut sebagai basis data, karena didalamnya tidak ada pemilahan dan pengelompokan data sesuai jenis/fungsi data. Sehingga akan menyulitkan pencarian data kelak. Yang di tonjolkan dalam basis data adalah pengaturan, pemilahan, pengelompokan dan pengorganisasian data yang akan kita simpan sesuai dengan fungsi/jenisnya. Pengaturan, pemilahan, pengelompokan dan pengorganisasian ini dapat berbentuk sejumlah *file/table* terpisah atau dalam bentuk definisian kolom-kolom *field*, *field* data dalam setiap *table*.

Tipe-Tipe database pada mysql diantaranya :

1. DDL (*Data Definition Language*)
   DDL merupakan perintah SQL yang berhubungan dengan pendefinisian suatu struktur database yang digunakan untuk membuat, mengubah dan menghapus struktur dan definisi metadata dari objek-objek database. Beberapa perintah dasar yang termasuk DDL ini antara lain:

   (a) *Create*
   Perintah ini digunakan untuk membuat, termasuk diantaranya membuat database baru, tabel baru, view baru, dan kolom.

   (b) *Alter*
   Perintah ini digunakan untuk mengubah struktur tabel yang telah dibuat.

Pekerjaannya mencakup mengganti nama tabel, menambah kolom, mengubah kolom, menghapus kolom, maupun memberikan atribut pada kolom.

(c) *Rename*
Perintah yang digunakan untuk merubah nama Objek

(d) *Drop*
Perintah ini digunakan untuk menghapus database dan tabel.

2. DML (*Data Manipulation Language*)
DML merupakan perintah SQL yang digunakan untuk proses pengolahan isi data di dalam table seperti memasukkan, merubah dan menghapus isi data dan tidak terkait dengan perubahan struktur dan definisi tipe data dari objek database. Perintah SQL yang termasuk dalam DML antara lain:

(a) *Update*
Perintah ini digunakan untuk memperbaharui data lama menjadi data terkini. Jika anda memiliki data yang salah atau kurang up to date dengan kondisi sekarang, maka dapat diubah isi datanya dengan menggunakan perintah update.

(b) *Insert*
Perintah ini digunakan untuk menyisipkan atau memasukkan data baru ke dalam tabel. Penggunaannya setelah database dan tabel selesai dibuat.

(c) *Select*
Perintah ini digunakan untuk mengambil data atau menampilkan data dari satu tabel atau beberapa tabel dalam relasi. Data yang diambil dapat kita tampilkan dalam layar prompt MySQL secara langsung maupun ditampilkan pada tampilan aplikasi.

(d) *Delete*
Perintah ini digunakan untuk menghapus data dari tabel. Biasanya data yang dihapus adalah data yang tidak diperlukan lagi. Pada saat menghapus data, perintah yang telah dijalankan tidak dapat digagalkan, sehingga data yang telah hilang tidak dapat dikembalikan lagi.

3. DCL (*Data Control Language*)
DCL merupakan perintah SQL yang berhubungan dengan manipulasi user dan hak akses (*priviledges*). Perintah SQL yang termasuk dalam DCL antara lain:

(a) *Grant*
Perintah ini digunakan untuk memberikan hak/izin akses oleh administrator (pemilik utama) server kepada *user* (pengguna biasa). Hak akses tersebut berupa hak membuat (*create*), mengambil (*select*), menghapus (*delete*), mengubah (*update*) dan hak khusus berkenaan dengan sistem databasenya.

(b) *Revoke*
Perintah ini memiliki kegunaan terbalik dengan *grand*, yaitu untuk menghilangkan atau mencabut hak akses yang telah diberikan kepada *user* oleh administrator.

7.8.1 SQL

SQL (diucapkan ”ess-que-el”) adalah singkatan dari *Structured Query Language*. SQL digunakan untuk berkomunikasi dengan database. Menurut ANSI *(American National Standards Institute)*, yang merupakan bahasa standar untuk sistem manajemen basis data relasional. Pernyataan SQL digunakan untuk melakukan tugas-tugas seperti memperbarui data pada *database*, atau mengambil data dari *database*. Beberapa sistem manajemen basis data relasional umum yang menggunakan SQL adalah: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, dll. Meskipun sebagian besar sistem database menggunakan SQL, kebanyakan dari mereka juga memiliki ekstensi kepemilikan tambahan sendiri yang biasanya hanya digunakan pada sistem mereka. . Namun, perintah SQL standar seperti ”Select”, ”Insert”, ”Update”, ”Delete”, ”Create”, dan ”Drop” dapat digunakan untuk mencapai hampir semua yang harus dilakukan dengan basis data.

7.8.1.1 Dasar-Dasar Tabel

Sistem basis data relasional berisi satu atau lebih objek yang disebut tabel. Data atau informasi untuk database disimpan dalam tabel ini. Tabel diidentifikasi secara unik dengan namanya dan terdiri dari kolom dan baris. Kolom berisi nama kolom, tipe data, dan atribut lainnya untuk kolom. Baris berisi catatan atau data untuk kolom. Berikut adalah contoh tabel yang disebut ”cuaca”.

kota, negara bagian, tinggi, dan rendah adalah kolom. Baris berisi data untuk tabel ini:

Cuaca			
Kota	Negara	Tinggi	Rendah
Phoenix	Arizona	105	90
Tucson	Arizona	101	92
Flagstaff	Arizona	88	69
San Diego	California	77	60
Albuquerque	New Mexico	80	b72

Tabel 7.1 Tabel Dasar

Select statement digunakan untuk *query database* dan mengambil data yang dipilih yang cocok dengan kriteria yang Anda tentukan. Berikut adalah format pernyataan pilih sederhana:

```
select "column1"
  [,"column2",etc]
  from "tablename"
  [where "condition"];
  [] = optional
```

Nama kolom yang mengikuti kata kunci pilih menentukan kolom mana yang akan dikembalikan dalam hasil. Anda dapat memilih sebanyak mungkin nama kolom yang Anda inginkan, atau Anda dapat menggunakan "*" untuk memilih semua kolom.

Nama tabel yang mengikuti kata kunci dari menentukan tabel yang akan diminta untuk mengambil hasil yang diinginkan.

Di mana klausa (opsional) menentukan nilai atau baris data mana yang akan dikembalikan atau ditampilkan, berdasarkan kriteria yang dijelaskan setelah kata kunci di mana.

Pilihan bersyarat yang digunakan dalam klausa mana :

- = (Sama dengan atau *Equal*)
- > Lebih besar dari
- < Kurang dari
- >= Lebih besar atau sama
- <= Kurang dari atau sama
- <> Tidak sebanding dengan
- LIKE

Operator LIKE pencocokan pola juga dapat digunakan dalam pemilihan bersyarat dari klausa mana. Like adalah operator yang sangat kuat yang memungkinkan Anda memilih hanya baris yang seperti yang Anda tentukan. Tanda persen "%" dapat digunakan sebagai kartu liar untuk mencocokkan dengan karakter apa pun yang mungkin muncul sebelum atau setelah karakter yang ditentukan. Sebagai contoh:

```
select first, last, city
   from empinfo
   where first LIKE 'Er%';
```

Pernyataan SQL ini akan cocok dengan nama depan yang dimulai dengan 'Er'. String harus dalam tanda kutip tunggal. Atau Anda dapat menentukan,

```
select first, last
   from empinfo
   where last LIKE '%s';
```

Pernyataan ini akan cocok dengan nama belakang apa pun yang berakhiran 's'.

```
select * from empinfo
   where first = 'Eric';
```

Ini hanya akan memilih baris di mana nama pertama sama dengan 'Eric'.

Empinfo					
pertama	terakhir	Id	usia	kota	negara
John	Jones	99980	45	Payson	Arizona
Mary	Jones	99982	25	Payson	Arizona
Eric	Edwards	88232	32	San Diego	California
Mary Ann	Edwards	88233	32	Phoenix	Arizona
Jahe	Howell	98002	42	Cottonwood	Arizona
Sebastian	Smith	92001	23	Gila Bend	Arizona
Gus	Gray	22322	35	Baghdad	Arizona
Mary Ann	May	32326	52	Tucson	Arizona
Erica	Williams	32327	60	Show Low	Arizona
Leroy	Brown	32380	22	Pinetop	Arizona
Elroy	Cleaver	32382	22	Globe	Arizona

Tabel 7.2 Tabel contoh

Masukkan pernyataan pilih sampel berikut dalam Formulir SQL Interpreter di bagian bawah halaman ini. Sebelum Anda menekan ”kirim”, tuliskan hasil yang Anda harapkan. Tekan ”kirim”, dan bandingkan hasilnya.

```
select first, last, city from empinfo;

select last, city, age from empinfo
       where age > 30;

select first, last, city, state from empinfo
       where first LIKE 'J%';

select * from empinfo;

select first, last, from empinfo
       where last LIKE '%s';

select first, last, age from empinfo
       where last LIKE '%illia%';

select * from empinfo where first = 'Eric';
```

7.8.1.2 Membuat Tabel

create table statement digunakan untuk membuat tabel baru. Berikut adalah format pernyataan tabel *create* sederhana:

```
create table "tablename"
("column1" "data type",
 "column2" "data type",
 "column3" "data type");
```

Format buat tabel jika Anda menggunakan batasan opsional:

```
create table "tablename"
("column1" "data type"
          [constraint],
 "column2" "data type"
          [constraint],
 "column3" "data type"
         [constraint]);
 [ ] = optional
```

Catatan: Anda mungkin memiliki kolom sebanyak yang Anda inginkan, dan batasannya adalah opsional.

Contoh :

```
create table employee
(first varchar(15),
 last varchar(20),
 age number(3),
 address varchar(30),
 city varchar(20),
 state varchar(20));
```

Untuk membuat tabel baru, masukkan kata kunci buat tabel diikuti dengan nama tabel, diikuti dengan tanda kurung terbuka, diikuti dengan nama kolom pertama, diikuti oleh tipe data untuk kolom itu, diikuti oleh kendala opsional apa pun, dan diikuti oleh penutup kurung. Penting untuk memastikan Anda menggunakan tanda kurung terbuka sebelum tabel awal, dan tanda kurung penutup setelah akhir definisi kolom terakhir. Pastikan Anda memisahkan setiap definisi kolom dengan koma. Semua pernyataan SQL harus diakhiri dengan ";".

Nama tabel dan kolom harus dimulai dengan huruf dan dapat diikuti oleh huruf, angka, atau garis bawah - tidak melebihi total 30 karakter. Jangan gunakan kata kunci yang disediakan SQL sebagai nama untuk tabel atau nama kolom (seperti "select", "create", "insert", dll).

Tipe data menentukan tipe data apa yang bisa untuk kolom tertentu. Jika kolom yang disebut "Last_Name", digunakan untuk menyimpan nama, maka kolom tersebut harus memiliki tipe data "varchar" (*variable-length character*).

Berikut adalah tipe data yang paling umum:

Tipe Data	Keterangan
char(size)	String karakter dengan panjang tetap. Ukuran ditentukan dalam tanda kurung. Maksimal 255 byte.
varchar(size)	String karakter panjang variabel. Ukuran maks ditentukan dalam kurung.
number(size)	Nilai angka dengan jumlah maksimum digit kolom yang ditentukan dalam tanda kurung.
date	Nilai tanggal
number(size,d)	Nilai angka dengan jumlah maksimum jumlah "*size*", dengan jumlah maksimum "d" digit di sebelah kanan desimal.

Tabel 7.3 Tipe Data

Ketika tabel dibuat, biasanya satu atau beberapa kolom memiliki kendala yang terkait dengannya. Batasan pada dasarnya adalah aturan yang terkait dengan kolom yang harus diikuti oleh data yang dimasukkan ke dalam kolom itu. Misalnya, batasan "unik" menetapkan bahwa tidak ada dua rekaman yang dapat memiliki nilai yang sama di kolom tertentu. Mereka semua harus unik. Dua kendala paling populer lainnya adalah "bukan nol" yang menentukan bahwa kolom tidak dapat dibiarkan kosong, dan "kunci utama". Batasan "kunci utama" mendefinisikan identifikasi unik dari setiap catatan (atau baris) dalam sebuah tabel. Semua ini dan lebih banyak akan dibahas dalam rilis Advanced Tutorial ini di masa depan. Kendala dapat dimasukkan dalam penerjemah SQL ini, namun, mereka tidak didukung dalam tutorial penerjemah Intro to SQL ini. Mereka akan dibahas dan didukung dalam rilis lanjutan tutorial SQL Lanjutan - yaitu, jika "respons" baik.

Sekarang saatnya bagi Anda untuk merancang dan membuat tabel Anda sendiri. Anda akan menggunakan tabel ini sepanjang tutorial lainnya. Jika Anda memutuskan untuk mengubah atau mendesain ulang tabel, Anda bisa melakukan *drop* dan membuatnya kembali atau Anda bisa membuat yang sama sekali berbeda. *Drop* pernyataan SQL akan dibahas nanti.

7.8.1.3 Insert Tabel

Statement insert digunakan untuk menyisipkan atau menambahkan deretan data ke dalam tabel.

Untuk menyisipkan catatan ke dalam tabel, masukkan kata kunci yang dimasukkan ke diikuti oleh nama tabel, diikuti oleh tanda kurung terbuka, diikuti oleh daftar nama kolom yang dipisahkan dengan koma, diikuti oleh tanda kurung penutup, diikuti oleh nilai kata kunci, diikuti oleh nilai kata kunci, diikuti oleh daftar nilai yang dilampirkan dalam tanda kurung. Nilai yang Anda masukkan akan disimpan di baris dan mereka akan cocok dengan nama kolom yang Anda tentukan. String harus dilampirkan dalam tanda kutip tunggal, dan angka tidak boleh.

```
insert into "tablename"
 (first_column ,... last_column)
  values (first_value ,... last_value);
```

Pada contoh di bawah ini, nama kolom pertama akan cocok dengan nilai 'Luke', dan negara nama kolom akan cocok dengan nilai 'Georgia'. Contoh :

```
    insert into employee
  (first, last, age, address, city, state)
  values ('Luke', 'Duke', 45, '2130 Boars Nest',
          'Hazard Co', 'Georgia');

```

Catatan: Semua string harus tertutup di antara tanda kutip tunggal: 'string'

7.8.1.4 Update Record

*Update statement*Pernyataan pembaruan digunakan untuk memperbarui atau mengubah catatan yang cocok dengan kriteria yang ditentukan. Ini dicapai dengan secara hati-hati membangun klausa di mana.

```
update "tablename"
set "columnname" =
    "newvalue"
 [,"nextcolumn" =
   "newvalue2"...]
where "columnname"
  OPERATOR "value"
 [and|or "column"
  OPERATOR "value"];

 [] = optional
```

Contoh :

```
update phone_book
  set area_code = 623
  where prefix = 979;

update phone_book
  set last_name = 'Smith', prefix=555, suffix=9292
  where last_name = 'Jones';

update employee
  set age = age+1
  where first_name='Mary' and last_name='Williams';
```

7.8.1.5 Delete Record

Delete statement digunakan untuk menghapus catatan atau baris dari tabel.

```
delete from "tablename"

where "columnname"
  OPERATOR "value"
[and|or "column"
  OPERATOR "value"];

[ ] = optional
```

Contoh :

```
delete from employee;
```

Catatan: jika Anda meninggalkan klausa, semua catatan akan dihapus!

```
delete from employee
  where lastname = 'May';

delete from employee
  where firstname = 'Mike' or firstname = 'Eric';
```

Untuk menghapus seluruh catatan / baris dari sebuah tabel, masukkan "hapus dari" diikuti dengan nama tabel, diikuti dengan klausa mana yang berisi ketentuan untuk dihapus. Jika Anda meninggalkan klausa di mana, semua catatan akan dihapus.

7.8.1.6 Drop Table

Perintah drop table digunakan untuk menghapus tabel dan semua baris dalam tabel. Untuk menghapus seluruh tabel termasuk semua barisnya, keluarkan perintah drop table diikuti oleh tablename. drop table berbeda dari menghapus semua record dalam tabel. Menghapus semua catatan dalam tabel meninggalkan tabel termasuk kolom dan informasi kendala. Menjatuhkan tabel menghapus definisi tabel serta semua barisnya.

```
drop table "tablename"
```

Contoh :

```
drop table myemployees_ts0211;
```

7.8.1.7 Table Join

Semua pertanyaan hingga saat ini bermanfaat dengan pengecualian satu batasan utama - yaitu, Anda hanya memilih dari satu tabel pada satu waktu dengan pernyataan SELECT Anda. Saatnya untuk memperkenalkan Anda ke salah satu fitur yang paling bermanfaat dari sistem *database SQL and relasional* - "Join". Sederhananya, "Bergabung" membuat sistem basis data relasional "relasional". Joining atau penggabungan memungkinkan Anda untuk menautkan data dari dua tabel atau

lebih bersama-sama ke dalam hasil kueri tunggal - dari satu pernyataan SELECT tunggal. ”Join” dapat dikenali dalam pernyataan SQL SELECT jika memiliki lebih dari satu tabel setelah kata kunci FROM. Sebagai contoh:

```
SELECT "list-of-columns"

FROM table1, table2

WHERE "search-condition(s)"
```

Join dapat dijelaskan dengan lebih mudah dengan menunjukkan apa yang akan terjadi jika Anda bekerja dengan satu tabel saja, dan tidak memiliki kemampuan untuk menggunakan ”*join*”. *Database* tabel tunggal ini kadang-kadang juga disebut sebagai ”*flat table*”. Katakanlah Anda memiliki *database* satu-tabel yang digunakan untuk melacak semua pelanggan Anda dan apa yang mereka beli dari toko Anda:

id	first	last	address	city	state	zip	date	item	price

Tabel 7.4 Flat Table

Setiap kali baris baru dimasukkan ke dalam tabel, semua kolom akan diperbarui, sehingga menghasilkan ”*redundant data*” yang tidak perlu. Misalnya, setiap kali Wolfgang Schultz membeli sesuatu, baris berikut akan dimasukkan ke dalam tabel:

id	first	last	address	city	state	zip	date	item	price
10982	Wolfgang	Schultz	300 N. 1st Ave	Yuma	AZ	85002	032299	snowboard	45.00
10982	Wolfgang	Schultz	300 N. 1st Ave	Yuma	AZ	85002	082899	snow shovel	35.00
10982	Wolfgang	Schultz	300 N. 1st Ave	Yuma	AZ	85002	091199	gloves	15.00
10982	Wolfgang	Schultz	300 N. 1st Ave	Yuma	AZ	85002	100999	lantern	35.00
10982	Wolfgang	Schultz	300 N. 1st Ave	Yuma	AZ	85002	022900	tent	85.00

Tabel 7.5 Redundant Data

Basis data yang ideal memiliki dua tabel, seperti berikut:

1. Tabel ”Customer_info” :

customer_number	firstname	lastname	address	city	state	zip

Tabel 7.6 Customer info

2. Tabel ”Purchases” :

customer_number	date	item	price

Tabel 7.7 Purchases

Sekarang, setiap kali pembelian dilakukan dari pelanggan berulang, tabel 7.7, ”Purchases” hanya perlu diperbarui! Kami baru saja menghilangkan data berlebihan yang tidak berguna, yaitu, kita hanya perlu menormalkan *database* ini! Perhatikan bagaimana masing-masing tabel memiliki kolom ”cusomer_number” yang umum. Kolom ini, yang berisi nomor pelanggan unik akan digunakan untuk menggabungkan ke dua tabel. Menggunakan dua tabel baru, katakanlah Anda ingin memilih nama pelanggan, dan barang yang telah mereka beli. Berikut adalah contoh pernyataan bergabung untuk mencapai ini:

```
SELECT customer_info.firstname, customer_info.lastname, purchases.
    item

FROM customer_info, purchases

WHERE customer_info.customer_number = purchases.customer_number;
```

”join” khusus ini dikenal sebagai ”Inner Join” atau ”Equijoin”. Ini adalah jenis ”join” yang paling umum digunakan. Perhatikan bahwa setiap kolom selalu diawali dengan nama tabel dan titik. Ini tidak selalu diperlukan, sehingga Anda tidak perlu bingung dengan mana kolom dijoinkan dengan tabel apa. Diperlukan jika nama kolom nama sama antara dua tabel. Saya sarankan mendahului semua kolom Anda dengan nama tabel saat menggunakan join.

Catatan: Sintaks yang diuraikan di atas akan bekerja dengan sebagian besar Sistem Database - termasuk yang dengan tutorial ini. Namun, jika ini tidak bekerja dengan Anda, silakan periksa dokumentasi database spesifik Anda.

Meskipun hal di atas mungkin akan berhasil, berikut adalah spesifikasi sintaks ANSI SQL-92 untuk Inner Join menggunakan pernyataan sebelumnya di atas yang mungkin ingin Anda coba:

```
SELECT customer_info.firstname, customer_info.lastname, purchases.
    item

FROM customer_info INNER JOIN purchases
```

```
4
5 ON customer_info.customer_number = purchases.customer_number;
```

Contoh lain :

```
1
2 SELECT employee_info.employeeid, employee_info.lastname,
      employee_sales.comission
3
4 FROM employee_info, employee_sales
5
6 WHERE employee_info.employeeid = employee_sales.employeeid;
```

Statement ini akan memilih employeeid, lastname (dari tabel employee_info), dan nilai komisi (dari tabel employee_sales) untuk semua baris di mana employeeid dalam tabel employee_info cocok dengan employeeid dalam tabel employee_sales.

7.8.2 MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basis data (DBMS) dapat diketahui dari cara kerja pengoptimasinya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basis data transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya.

Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basis data transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multi-thread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael ”Monty” Widenius.

7.9 Application Framework

Application Framework adalah pustaka perangkat lunak yang menyediakan struktur dasar untuk mendukung pengembangan aplikasi untuk lingkungan tertentu. Kerangka kerja aplikasi bertindak sebagai dukungan kerangka untuk membangun aplikasi. Tujuan dari merancang kerangka kerja aplikasi adalah untuk mengurangi masalah umum yang dihadapi selama pengembangan aplikasi. Ini dicapai melalui penggunaan kode yang dapat dibagi di berbagai modul aplikasi. Kerangka kerja aplikasi digunakan tidak hanya dalam pengembangan antarmuka pengguna grafis (GUI), tetapi juga di bidang lain seperti aplikasi berbasis web.

Mengingat fungsionalitas aplikasi web modern yang kaya dan luas, gagasan untuk membuatnya tampak luar biasa. Tidak peduli seberapa baik Anda mengetahui tentang semua langkah yang diperlukan, mereka tetap merupakan langkah yang harus Anda ikuti dengan menyelesaikan tugas-tugas tertentu. Namun Anda tidak harus melakukan itu sendiri. Untungnya, ada alat yang dapat membuat pengembangan aplikasi web lebih mudah, dan tidak hanya melalui penjelasan tetapi dengan memberikan dasar yang mendalam untuk membangun aplikasi web. *Web Application Framework* adalah salah satu alat tersebut.

Kerangka kerja web adalah alat perangkat lunak yang menyediakan cara untuk membangun dan menjalankan aplikasi web. Akibatnya, Anda tidak perlu menulis kode sendiri dan membuang waktu mencari kemungkinan kesalahan perhitungan dan bug.

Pada awal pengembangan web, semua aplikasi diberi kode tangan, dan hanya pengembang aplikasi tertentu yang dapat mengubah atau menyebarkannya. Kerangka kerja web memperkenalkan cara sederhana untuk keluar dari jebakan ini. Sejak 1995, semua kerumitan yang berhubungan dengan mengubah struktur aplikasi telah diurus karena penampilan kinerja umum. Dan saat itulah bahasa spesifik web muncul. Variasi mereka sekarang bekerja dengan baik untuk halaman web statis dan dinamis. Anda dapat memilih satu kerangka kerja yang mencakup semua kebutuhan Anda atau menggabungkan beberapa, tergantung pada tugas Anda.

7.9.1 Web application framework vs. content management system

Kerangka kerja aplikasi web dan sistem manajemen konten (CMS) dikelilingi oleh pertanyaan yang membingungkan dari calon pengembang web. Apakah mereka berbeda? Bagaimana mereka berbeda? Manakah dari keduanya yang lebih baik?

Panduan tertulis yang dimulai dengan menjelaskan prinsip kerja kerangka pengembangan web dan akhirnya memberikan daftar yang penuh dengan CMS sebagai contoh, biarkan kebingungan tetap ada.

Pada kenyataannya, baik kerangka kerja dan CMS meletakkan dasar untuk aplikasi web masa depan dan merujuk pada teknologi yang sama; misalnya, Symfony (kerangka kerja aplikasi web) dan Joomla (a CMS) didukung oleh PHP. Kemungkinannya sangat mirip, dan Anda dapat membuat aplikasi web yang sama kuatnya dengan keduanya.

Perbedaan utama ada pada pendekatan. Bayangkan menavigasi komputer Anda melalui baris perintah dan penjelajah. Yang pertama lebih dekat untuk membuat aplikasi web dengan kerangka kerja aplikasi web, yang terakhir - dengan CMS.

Karena kerangka biasanya seperangkat perpustakaan dan alat yang membantu untuk membangun aplikasi web, itu membutuhkan keterampilan pemrograman yang lebih tinggi. Menawarkan lebih banyak kebebasan untuk implementasi dari awal, kerangka kerja aplikasi web sangat cocok untuk pengembang yang berpengalaman.

Namun, Anda tidak boleh meremehkan pengembang yang lebih suka membuat aplikasi web dalam CMS. Seseorang dapat berargumen bahwa dengan CMS, seseorang tidak memerlukan bahasa pemrograman sama sekali, tetapi ini hanya berlaku jika kita berbicara tentang mengelola Website yang sudah ada. Untuk mengatur aplikasi web melalui CMS, seseorang harus tahu cara bekerja dengan server dan dapat membaca / mengedit berbagai potongan kode.

7.9.2 Jenis Web Frameworks

Ada dua fungsi utama kerangka kerja: untuk bekerja di sisi server (backend), atau di sisi klien (frontend), sesuai dengan tipenya. Kerangka kerja frontend sebagian besar berurusan dengan bagian eksternal aplikasi web. Secara singkat, itulah yang dilihat pengguna saat mereka membuka aplikasi. Hal-hal di dalam adalah karya backend.

Mari kita lihat lebih detail:

- **Server-side frameworks.** Aturan dan arsitektur kerangka kerja ini memungkinkan Anda untuk membuat halaman sederhana, pendaratan, dan bentuk dari berbagai jenis. Namun, untuk membangun aplikasi web dengan antarmuka yang dikembangkan dengan baik, Anda harus memiliki fungsionalitas yang lebih luas. Kerangka kerja ini juga dapat membentuk data keluaran dan meningkatkan keamanan jika terjadi serangan web. Semua ini pasti dapat menyederhanakan proses pengembangan. Kerangka kerja sisi server sebagian besar bekerja pada detail tertentu tetapi penting yang tanpanya aplikasi tidak dapat berfungsi dengan baik. Berikut adalah kerangka kerja backend atas dan bahasa tempat mereka bekerja:
  - Django Python
  - Zend PHP
  - Express.js Javascript

  - Ruby on Rails Ruby

- **Client-side frameworks.** Tidak seperti sisi server, kerangka kerja sisi klien tidak ada hubungannya dengan logika bisnis. Pekerjaan mereka berlangsung di dalam browser. Dengan demikian, seseorang dapat meningkatkan dan mengimplementasikan antarmuka pengguna baru. Sejumlah fitur animasi dapat dibuat dengan kerangka kerja frontend serta SPA (aplikasi satu halaman). Setiap kerangka kerja sisi klien berbeda dalam fungsi dan penggunaan. Untuk tujuan perbandingan, inilah mereka:

  - Backbone+Marionette
  - Angular
  - Ember.js
  - Vue.js
  - Bootstrap
  - React.js
  - Semantic-UI.

  Semua kerangka kerja yang disebutkan di atas menggunakan Javascript sebagai bahasa pengkodeannya. Sementara dengan kerangka kerja sisi-server pilihan Anda terutama tergantung pada bahasa yang Anda rasa nyaman untuk dikembangkan, di sini Anda harus memperhatikan kemampuan khusus dari kerangka kerja sisi-klien yang berbeda. Karena terdapat perberbedaan dalam cakupan fungsionalitas yang didukung, carilah yang sesuai dengan kebutuhan aplikasi web Anda.

- **Cross-functional frameworks.** Meteor dikenal sebagai kerangka web tumpukan penuh. Ini berarti bahwa itu mencakup hampir semua kebutuhan dari sisi klien dan server, yang membuat Meteor sangat populer dan banyak digunakan. Anda tidak perlu membuang waktu untuk menggabungkan dua kerangka kerja menggunakan REST API; Anda cukup memilih yang ini dan mempercepat proses pengembangan Anda. Tapi ini bukan fitur utama Meteor. Kedua sisi bekerja dalam satu bahasa, sehingga Anda dapat membuat dan menggunakan kode yang sama untuk keduanya. Hal berikutnya adalah "*mode real-time*" - ketika perubahan dibuat untuk satu antarmuka, itu terjadi pada semua antarmuka lainnya juga. Salah satu contohnya adalah dokumen yang dapat dibagikan atau spreadsheet. Ketika Anda menambahkan beberapa komentar ke halaman yang Anda baca atau mengubah, pengguna lain juga akan melihatnya.

7.9.3 Web Frameworks: Features and Architecture

Terlepas dari kenyataan bahwa kerangka kerja semuanya berbeda dan bisa sangat sulit untuk memilih satu, mereka semua memiliki kesamaan. Arsitektur dan fitur-fiturnya, yang sama pentingnya dengan fungsi.

7.9.3.1 Architecture

Arsitektur hampir semua kerangka kerja pengembangan web paling populer didasarkan pada dekomposisi dari beberapa lapisan terpisah (aplikasi, modul, dll), yang berarti bahwa Anda dapat memperluas fungsionalitas sesuai dengan kebutuhan Anda dan mengintegrasikan perubahan Anda dengan kode kerangka kerja, atau menggunakan ketiga aplikasi pihak yang dirancang oleh vendor eksternal. Fleksibilitas ini merupakan manfaat utama kerangka kerja lainnya. Ada banyak komunitas open-source dan organisasi komersial yang menghasilkan aplikasi atau ekstensi untuk kerangka kerja populer mis., Django REST Framework, ng-bootstrap, dll.).

Lebih dari 80% dari semua kerangka kerja aplikasi web bergantung pada arsitektur Model View Controller. Rahasia popularitas pola ini adalah dalam cara rasional memisahkan logika aplikasi dari antarmuka yang membentuk 3 komponen yang tercermin dalam nama arsitektur. MVC - yaitu, Model, View, dan Controller - adalah tiga hal yang dibuat dari kerangka kerja masing-masing web. Ini dianggap sebagai struktur dasar, tetapi bisa ada beberapa kontras di antara mereka.

- Model
  Model mengetahui semua tentang konten dan struktur aplikasi. Setelah menerima data input pengguna dari Controller, itu mengkomunikasikan cara antarmuka yang diperbarui harus melihat langsung ke View.

- View
  Ini adalah aplikasi frontend. Ia tahu tata letak dan cara pengguna dapat berinteraksi dengan bagian-bagiannya. View menerima input pengguna, mengkomunikasikannya ke Controller untuk analisis dan memperbarui atau menyusun kembali sendiri sesuai dengan instruksi Model (atau Controller, jika perubahan kecil).

- Controller
  Controller adalah perantara antara Model dan View. Controller menerima input pengguna dari View, memprosesnya dan memberi tahu Model (atau View) perubahan apa yang harus dilakukan. Beberapa orang menganjurkan bahwa Controller tidak selalu diperlukan dan yang paling penting adalah memisahkan logika dari antarmuka, yaitu Model dan View. Namun, menugaskan pemrosesan input untuk Model atau Tampilan mengganggu ideologi awal pola Presentasi Terpisah, di mana tugas didistribusikan berdasarkan jenisnya.

Ketika setiap komponen dalam arsitektur bertanggung jawab atas satu baris tugas, proyek transparan, fleksibel dan mudah dirawat. Selain itu, arsitektur MVC memungkinkan:

- Pengembangan paralel (lebih sedikit waktu untuk mengirim)
- Penggunaan kembali kode
- Memperbaiki atau memodifikasi salah satu komponen tanpa harus memperbarui yang lain

- Menetapkan URL yang SEO-friendly.

7.9.3.2 Features

Sekarang mari kita lihat beberapa fitur umum yang membantu pengembang menerapkan kerangka kerja web populer sebagai alat multifungsi dan praktis.

- Web Caching
  Caching web hanya membantu menyimpan berbagai dokumen dan menghindari fenomena server berlebih yang mengganggu. Pengguna dapat menggunakannya di berbagai sistem jika beberapa kondisi terpenuhi. Ini juga berfungsi di sisi server. Misalnya, Anda mungkin melihat tautan konten yang di-cache di SERP (Halaman Hasil Mesin Pencari) dari mesin pencari seperti Google.

- Scaffolding
  Ini adalah teknik penting lainnya untuk diketahui dan digunakan, yang didukung oleh beberapa kerangka kerja MVC. Bagian khas aplikasi atau seluruh struktur proyek (dalam hal inisialisasi) dapat dihasilkan oleh kerangka kerja secara otomatis. Pendekatan ini meningkatkan kecepatan siklus pengembangan dan membakukan basis kode.

- Web template system
  Sistem template web adalah serangkaian metodologi dan perangkat lunak yang berbeda yang diimplementasikan untuk membangun dan menggunakan halaman web. Mesin templat digunakan untuk memproses templat web. Mereka adalah alat untuk penerbitan web dalam suatu kerangka kerja.

- Security
  Keamanan online memiliki banyak kriteria untuk mengidentifikasi dan mengizinkan atau menolak akses ke berbagai fungsi dalam kerangka kerja web. Ini juga membantu mengenali profil yang menggunakan aplikasi untuk menghindari clickjacking. Akibatnya, kerangka itu sendiri otentik dan resmi.

- URL Mapping
  Jika Anda ingin menyederhanakan pengindeksan Website Anda dengan mesin pencari sambil membuat nama situs yang jelas dan menarik, fitur kerangka kerja web ini dibuat khusus untuk itu. Pemetaan URL juga dapat memfasilitasi akses ke URL situs Anda.

- Applications
  Berbagai jenis aplikasi web didukung oleh kerangka kerja web. Kerangka kerja yang paling umum dan terbaik untuk pengembangan aplikasi mendukung pembangunan blog, forum, general-purpose websites, content management systems, dll.

Semua fitur ini umum untuk semua kerangka kerja. Namun, sebuah paradoks sekarang muncul: seorang pengembang memiliki begitu banyak alat dan fungsi sehingga ia bisa hilang begitu saja. Itu sebabnya Anda harus berpikir hati-hati dan

membuat kriteria yang akan membuat pekerjaan Anda nyaman dan mudah. Misalnya, pilihan Anda mungkin bergantung pada bahasa yang Anda pilih untuk digunakan saat coding. Seperti disebutkan sebelumnya, ada kerangka kerja yang ditulis dalam semua bahasa. Yang kedua adalah ruang lingkup alat yang dimiliki masing-masing kerangka kerja. Jika itu mencakup semua proses kerja Anda dan membantu Anda mengelola semua tugas Anda, Anda berada di jalur yang benar. Berbicara tentang preferensi, poin ini mungkin bermanfaat atau merusak. Tentu saja, lebih baik menggunakan kerangka kerja yang mudah dipelajari, tetapi kadang-kadang aturan sekolah lama dan kerangka kerja yang jarang digunakan tetapi cocok dapat mengarahkan Anda menuju kesuksesan.

7.9.4 Frameworks for Web Applications

Web Application Framework atau sekadar ”kerangka kerja web” adalah kerangka kerja perangkat lunak yang dirancang untuk mendukung pengembangan aplikasi web termasuk layanan web, sumber daya web, dan API web. Kerangka kerja adalah, singkatnya, perpustakaan yang membantu Anda mengembangkan aplikasi Anda lebih cepat dan lebih cerdas! Saat ini, jumlah Kerangka Kerja Web telah meningkat pesat. Untuk membantu Anda mengambil yang paling cocok untuk Aplikasi Web Anda, kami telah menyusun daftar 10 kerangka kerja terbaik yang tersedia online, dalam bahasa pilihan Anda.

1. Ruby on Rails
   Ruby on Rails adalah kerangka kerja aplikasi web yang sangat produktif yang ditulis oleh David Heinemeier Hansson. Seseorang dapat mengembangkan aplikasi setidaknya sepuluh kali lebih cepat dengan Rails daripada kerangka kerja Java yang khas. Selain itu, Rails mencakup semua yang diperlukan untuk membuat aplikasi web berbasis database, menggunakan pola Model-View-Controller.

   - Bahasa: Ruby
   - Versi Terbaru: Rails 5.0.0.beta2
   - Kerangka Tautan: http://rubyonrails.org
   - Tautan Github: https://github.com/rails/rails
   - Website menggunakan Rails: GroupOn , UrbanDictionary , AirBnb , Shopify , Github

2. Django
   Django adalah kerangka kerja lain yang membantu dalam membangun aplikasi web yang berkualitas. Itu diciptakan untuk memenuhi tenggat waktu ruang berita yang bergerak cepat, sambil memenuhi persyaratan keras pengembang Web berpengalaman. Pengembang Django mengatakan aplikasinya sangat cepat, aman, terukur, dan serbaguna.

   - Bahasa: Python

- Versi Terbaru: Django 1.9.2
- Link Framework: https://www.djangoproject.com
- Link Github: https://github.com/django/django
- Website menggunakan Django: Disqus , Pinterest , Instagram , Quora

3. Angular(previously Angular JS)
   Angular adalah kerangka kerja oleh Google (awalnya dikembangkan oleh Misko Hevery dan Adam Abrons) yang membantu kami dalam membangun Aplikasi Web yang kuat. Ini adalah kerangka kerja untuk membangun aplikasi web berskala besar dan berkinerja tinggi sambil menjaganya agar tetap mudah dirawat.

   - Bahasa: JavaScript Versi Terbaru: Angular 7.1.5
   - Link Framework: https://angular.io/
   - Link Github: https://github.com/angular/angular
   - Website menggunakan Angular: Youtube di PS3, Weather , Netflix

4. ASP.NET
   ASP.NET adalah kerangka kerja yang dikembangkan oleh Microsoft, yang membantu kami membangun aplikasi web yang kuat untuk PC, serta perangkat seluler. Ini adalah kinerja tinggi dan kerangka kerja ringan untuk membangun Aplikasi Web menggunakan .NET. Secara keseluruhan, kerangka kerja dengan Daya, Produktivitas, dan Kecepatan.

   - Bahasa: C
   - Versi Terbaru: ASP.NET 5 (ASP.NET Core 1.0)
   - Link Framework: http://www.asp.net/
   - Website menggunakan ASP.NET: GettyImages , TacoBell , StackOverflow

5. METEOR
   Meteor atau MeteorJS adalah kerangka kerja lain yang memberikan satu cara yang lebih sederhana untuk membangun aplikasi seluler dan web secara real-time. Ini memungkinkan untuk prototyping cepat dan menghasilkan kode lintas platform (Web, Android, iOS). Platform cloud-nya, Galaxy, sangat menyederhanakan penyebaran, penskalaan, dan pemantauan.

   - Bahasa: JavaScript
   - Versi Terbaru: Meteor 1.2.1
   - Link Framework: https://www.meteor.com/
   - Link Github: https://github.com/meteor/meteor
   - Website menggunakan Meteor: HaggleMate, WishPool, Telescope

6. Laravel
   Laravel adalah kerangka kerja yang dibuat oleh Taylor Otwell pada tahun 2011

dan seperti semua kerangka kerja modern lainnya, ia juga mengikuti pola arsitektur MVC. Laravel menghargai Keanggunan, Kesederhanaan, dan Keterbacaan. Orang dapat langsung mulai belajar dan mengembangkan Laravel dengan Laracasts yang memiliki ratusan tutorial di dalamnya.

- Bahasa: PHP
- Versi Terbaru: Laravel 5.2
- Link Framework: https://laravel.com/
- Link Github: https://github.com/laravel/laravel
- Website menggunakan Laravel: Deltanet Travel , Neighborhood Lender

7. Express
   Express atau Expressjs adalah kerangka kerja minimal dan fleksibel yang menyediakan serangkaian fitur yang kuat untuk aplikasi web dan seluler. Relatif minim artinya banyak fitur yang tersedia sebagai plugin. Express memfasilitasi pengembangan aplikasi Web berbasis Node.js dengan cepat. Express juga merupakan salah satu komponen utama bundel perangkat lunak MEAN.

   - Bahasa: JavaScript
   - Link Framework: http://expressjs.com/
   - Link Github: https://github.com/strongloop/express
   - Website menggunakan Express: Storify , Myspace , LearnBoost

8. Spring
   Spring, dikembangkan oleh Pivotal Software, adalah kerangka pengembangan aplikasi yang paling populer untuk perusahaan Java. Banyak pengembang di seluruh dunia menggunakan Spring untuk membuat aplikasi Web yang berkinerja tinggi dan kuat. Spring membantu dalam menciptakan sistem dan aplikasi berbasis JVM yang sederhana, portabel, cepat dan fleksibel.

   - Language : Java
   - Versi Terbaru: Spring 4.3.0
   - Framework Link: http://projects.spring.io/spring-framework/
   - Link Github: https://github.com/spring-projects/spring-framework
   - Website menggunakan Spring: Mascus, Allocine

9. PLAY
   Play adalah salah satu kerangka kerja aplikasi web modern yang ditulis dalam Java dan Scala. Ini mengikuti arsitektur MVC dan bertujuan untuk mengoptimalkan produktivitas pengembang dengan menggunakan konvensi atas konfigurasi, pemuatan ulang kode panas, dan menampilkan kesalahan pada browser. Mainkan kutipan sendiri sebagai ”Kerangka Kerja Web Berkecepatan Tinggi”

   - Bahasa : Scala and Java

- Versi Terbaru : Play 2.4.6
- Framework Link: https://www.playframework.com/
- Github Link : https://github.com/playframework/playframework
- Websites yang menggunakan PLAY : LinkedIn, Coursera, LendUp

10. CodeIgniter
    CodeIgniter, dikembangkan oleh EllisLab, adalah kerangka kerja aplikasi web terkenal untuk membangun Website yang dinamis. Ini secara longgar didasarkan pada arsitektur MVC karena kelas Controller diperlukan tetapi model dan pandangan adalah opsional. CodeIgnitor menjanjikan dengan kinerja luar biasa, konfigurasi hampir nol dan tidak ada perpustakaan monolitik skala besar.

    - Bahasa : PHP
    - Versi Terbaru : CodeIgnitor 3.0.4
    - Framework Link: https://codeigniter.com/
    - Github Link : https://github.com/EllisLab/CodeIgniter
    - Websites yang menggunakan CodeIgnitor : Bufferapp, The Mail and Guardian