

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 [1].

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 mendekatkan pada tujuan tersebut.

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 alternatif-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 tersebut adalah dapat berjalan lancar dan mencapai tujuan. Namun, kerap kali terjadi hambatan dalam pelaksanaannya, misalnya hanya banyakut satu masalah saja dan jika dipecahkan tidak menimbulkan akibat lain. Namun, ada kemungkinan terjadi masalah yang pemecahannya menghendaki 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 memperhitungkan untung 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 memaksimalkan 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[1].

- 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 pemilihan, 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 subjektivitas sehingga keputusan yang diambil juga tidak objektif. Untuk mengatasi hal tersebut, maka dibangunlah sistem pendukung keputusan. Sistem ini berfungsi 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 alternatif-alternatif atau pilihan-pilihan terbaik dari berbagai alternatif yang ditawarkan, namun keputusan tetap diambil oleh seorang pengambil keputusan.

Kebutuhan akan data dan informasi untuk menghasilkan 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. *Decision 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 masalah 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 memilih alternatif terbaik, namun menentukan jumlah produksi untuk masing-masing alternatif yang memaksimalkan sejumlah tujuan yang ada.

1.8 Tahapan Pengambilan Keputusan

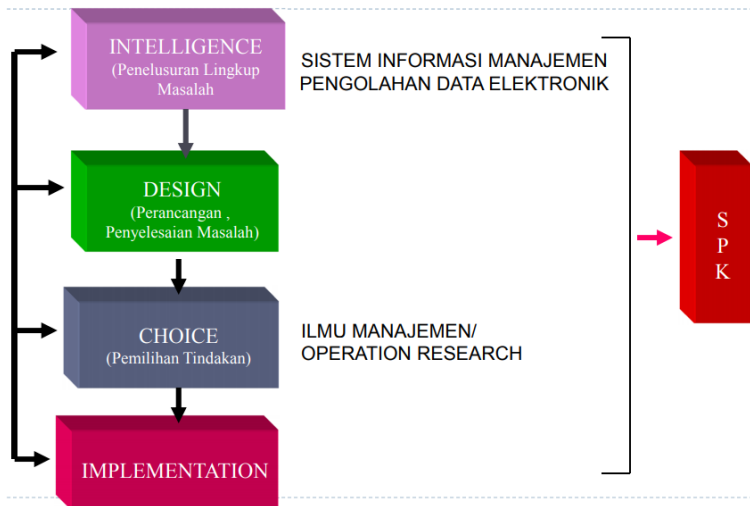
Pengambilan keputusan merupakan proses pemilihan alternatif tindakan untuk mencapai tujuan atau sasaran tertentu. Pengambilan keputusan dilakukan dengan pendekatan sistematis terhadap permasalahan melalui proses pengumpulan data menjadi informasi serta ditambah dengan faktor faktor yang perlu dipertimbangkan dalam pengambilan keputusan. Keputusan diambil berdasarkan pada keadaan lingkungan atau kondisi yang ada, seperti kondisi pasti, kondisi beresiko, kondisi tidak 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 timbulnya 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 diidentifikasi, kemudian dilakukan evaluasi terhadap masing-masing alternatif dan dipilih sebuah alternatif yang terbaik.
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 umum 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 riil 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 Perkembangan Sistem Pendukung Keputusan

Sejarah evolusi sistem pendukung keputusan dimulai 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 keuangan dan 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* [2].

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 keputusan 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 menggantikan tugas seorang pengambil keputusan. Sehingga dengan data dan informasi 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[3] :

1. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan.
2. A adanya *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.

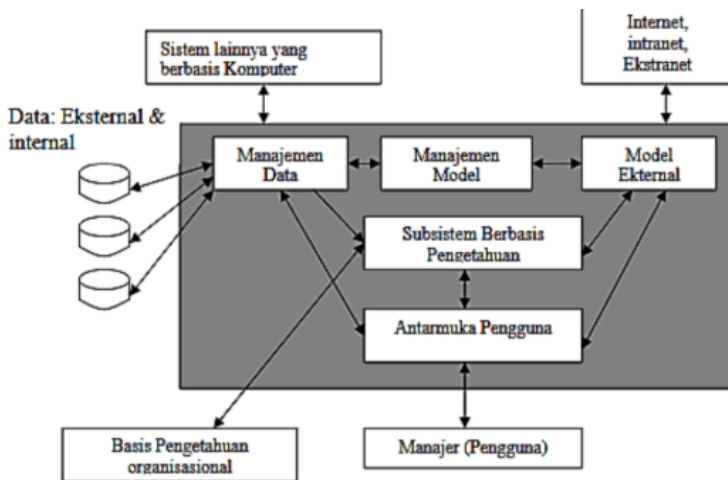
Menurut Turban karakteristik yang membedakan adalah :

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 2.1 dijelaskan sebagai berikut :

1. **Manajemen Data**
Subsistem manajemen data berisi data yang relevan untuk suatu situasi dan dikelola oleh *Database Management System (DBMS)*. Subsistem ini dapat diinterkoneksi 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 merupakan 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[4] :

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[4] :

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 suatu kumpulan perangkat keras, perangkat 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 produksi
 - (d) Menemukan cara-cara untuk meningkatkan 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 [5].). 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 [6], 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.

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 $m \times n$, berisi elemen-elemen x_{ij} , yang merepresentasikan rating dari alternatif A_i ($i = 1, 2, \dots, m$) terhadap kriteria C_j ($j = 1, 2, \dots, n$).

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 — $O_i, i=1, \dots, t$ — adalah dengan cara mendaftar konsekuensi-konsekuensi yang mungkin dari alternatif yang telah teridentifikasi — $A_i, i=1, \dots, n$ —. Selain itu juga disusun atribut-atribut yang akan digunakan — $a_k, k=1, \dots, m$ —.

3.3.2 Metode-Metode Penyelesaian Masalah MADM

Ada beberapa 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

METODE SAW (*SIMPLE ADDITIVE WEIGHTING*)

4.1 Pendahuluan Metode SAW

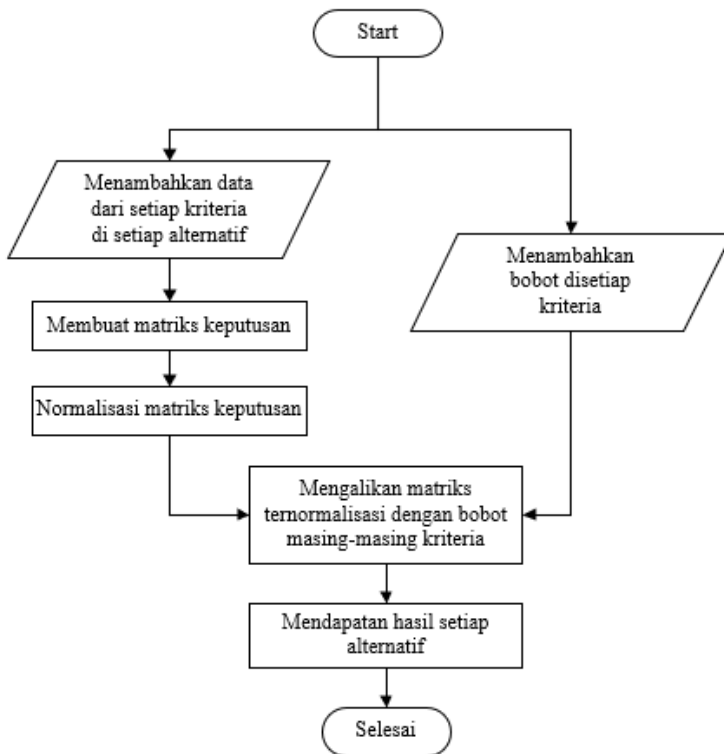
Simple additive weighting (SAW) merupakan metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua kriteria. Metode SAW membutuhkan proses normalisasi matrik keputusan, ke suatu skala yang dapat diperbandingkan dengan 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 :

- 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 alternatif 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 C_i .
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria (C_i) 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 (A_i) sebagai solusi.

Rumus untuk melakukan normalisasi tersebut ialah :

- Untuk *benefit criteria*

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

- Untuk *cost criteria*

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

Dengan :

R_{ij} = nilai rating kinerja normalisasi

x_{ij} = nilai atribut yang dimiliki dari setiap kriteria

$\max_i x_{ij}$ = nilai terbesar dari setiap kriteria i

$\min_i x_{ij}$ = nilai terkecil dari setiap kriteria i

Dimana r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j , $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$

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

$$V_i = \sum_{j=1}^n w_j r_{ij} \quad (4.3)$$

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 [7] :

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 Additive Weighting* (SAW) dapat memudahkan penyelesaian masalah dalam pemilihan pegawai teladan dengan menggunakan metode *Multiple Atribut Decision Making* (MADM) menggunakan model *Simple Assitive Weighting*. Guna mendapatkan pegawai teladan terbaik, diperlukan syarat kriteria dari setiap pegawai. Adapun syarat 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 pemilihan pegawai teladan terbagi dalam tiga kondisi dimana masing-masing kondisi terbagi atas beberapa dimensi dengan indikator-indikator tertentu. Dalam menentukan 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	<i>Benefit</i>	0,1
		Baik	3	<i>Benefit</i>	
		Cukup	2	<i>Benefit</i>	
		Kurang	1	<i>Benefit</i>	
C2	Tidak mening-galkan tempat tugas	Sangat Baik	4	<i>Benefit</i>	0,05
		Baik	3	<i>Benefit</i>	
		Cukup	2	<i>Benefit</i>	
		Kurang	1	<i>Benefit</i>	
C3	Ketaatan pada peraturan	Sangat Baik	4	<i>Benefit</i>	0,1
		Baik	3	<i>Benefit</i>	
		Cukup	2	<i>Benefit</i>	
		Kurang	1	<i>Benefit</i>	
C4	Ketepatan waktu kerja	Sangat Baik	4	<i>Benefit</i>	0,05
		Baik	3	<i>Benefit</i>	
		Cukup	2	<i>Benefit</i>	
		Kurang	1	<i>Benefit</i>	
C5	Kualitas kerja	Sangat Baik	4	<i>Benefit</i>	0,1
		Baik	3	<i>Benefit</i>	
		Cukup	2	<i>Benefit</i>	
		Kurang	1	<i>Benefit</i>	

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 lingkungan kerja	Sangat Baik	4	<i>Benefit</i>	0,05
		Baik	3	<i>Benefit</i>	
		Cukup	2	<i>Benefit</i>	
		Kurang	1	<i>Benefit</i>	
C7	Kebersihan	Sangat Baik	4	<i>Benefit</i>	0,05
		Baik	3	<i>Benefit</i>	
		Cukup	2	<i>Benefit</i>	
		Kurang	1	<i>Benefit</i>	
C8	Kerapihan berpakaian	Sangat Baik	4	<i>Benefit</i>	0,05
		Baik	3	<i>Benefit</i>	
		Cukup	2	<i>Benefit</i>	
		Kurang	1	<i>Benefit</i>	

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	<i>Benefit</i>	0,05
		Baik	3	<i>Benefit</i>	
		Cukup	2	<i>Benefit</i>	
		Kurang	1	<i>Benefit</i>	
C10	Tanggap terhadap situasi	Sangat Baik	4	<i>Benefit</i>	0,05
		Baik	3	<i>Benefit</i>	
		Cukup	2	<i>Benefit</i>	
		Kurang	1	<i>Benefit</i>	
C11	Kemauan untuk mengem-bangkan diri	Sangat Baik	4	<i>Benefit</i>	0,05
		Baik	3	<i>Benefit</i>	
		Cukup	2	<i>Benefit</i>	
		Kurang	1	<i>Benefit</i>	
C12	Pencapaian target kerja	Sangat Baik	4	<i>Benefit</i>	0,1
		Baik	3	<i>Benefit</i>	
		Cukup	2	<i>Benefit</i>	
		Kurang	1	<i>Benefit</i>	
C13	Pelaksanaan tugas pokok	Sangat Baik	4	<i>Benefit</i>	0,05
		Baik	3	<i>Benefit</i>	
		Cukup	2	<i>Benefit</i>	
		Kurang	1	<i>Benefit</i>	
C14	Inisiatif/kreatifitas	Sangat Baik	4	<i>Benefit</i>	0,05
		Baik	3	<i>Benefit</i>	
		Cukup	2	<i>Benefit</i>	
		Kurang	1	<i>Benefit</i>	
C15	Tanggung jawab	Sangat Baik	4	<i>Benefit</i>	0,1
		Baik	3	<i>Benefit</i>	
		Cukup	2	<i>Benefit</i>	
		Kurang	1	<i>Benefit</i>	

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 Situasi

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 Situasi

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 Situasi

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 Situasi

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 Situasi

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 Situasi

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 Situasi

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 Situasi

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 Situasi

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 Situasi

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 Situasi

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 Situasi

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	$(1 \times 0,1) + (1 \times 0,05) + (1 \times 0,1) + (1 \times 0,05) + (0,75 \times 0,1) + (0,5 \times 0,05) + (0,67 \times 0,05) + (1 \times 0,05) + (1 \times 0,05) + (1 \times 0,05) + (1 \times 0,05) + (0,75 \times 0,1) + (0,67 \times 0,05) + (1 \times 0,05) + (0,67 \times 0,1)$	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.

PERANCANGAN

Setelah memahami cara penyelesaian studi kasus menggunakan *Simple Additive Weighting* (SAW) sebagai mana dijelaskan pada Bab IV. Selanjutnya pada bab ini akan dijelaskan cara implementasi *Simple Additive Weighting* (SAW) ke dalam sebuah 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

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 beroperasi bersama-sama untuk menyelesaikan suatu sasaran.

5.2 Perancangan Sistem

Perancangan sistem adalah merancang atau mendesain suatu system yang baik yang isinya adalah langkah-langkah operasi dalam proses pengolahan data dan proses prosedur-prosedur untuk mendukung operasi sistem. Tujuan dari perancangan sistem adalah untuk memenuhi kebutuhan para pemakai sistem serta memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada programmer dan ahli-ahli yang terlibat didalam.

5.2.1 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.

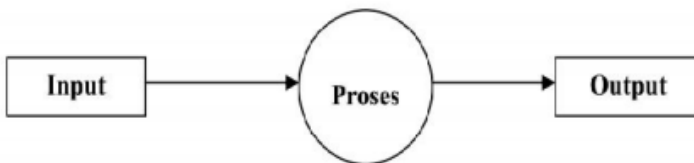
Menurut Jugiyono, sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

Pendekatan sistem yang merupakan jarngan kerja dari prosedur lebih menekankan urutan operasi di dalam sistem. Prosedur (*procedure*) didefinisikan oleh Richard F. Neuschel yang disadur oleh Jogiyanto mendefinisikan Prosedur adalah suatu urutan operasi klerikal (tuliskan-menulis) 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.

5.2.2 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 Umum

5.2.3 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.

5.2.4 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 karakteristik 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 lainnya.
6. Masukkan Sistem (*Input*)
Adalah energi yang dimasukkan kedalam sistem, masukan dapat berupa masukan 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.

5.2.5 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.

5.2.6 Analisis Sistem

Tahapan analisis sistem di mulai karena adanya permintaan terhadap sistem baru. Permintaan dapat datang 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.

5.2.7 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.

5.3 Sistem Analisis

Sistem analisis adalah profesi yg menantang karena menggabungkan banyak keahlian seperti keahlian analisis, teknis, interpersonal, dan manajerial. Hal ini bisa dilihat 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 Desain
2. Implementasi Pemeliharaan

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

1. Identifikasi dan seleksi proyek Inisiasi dan perancangan proyek
2. Analisis
3. Desain
 - (a) Desain Logical
 - (b) Desain fisik
4. Implementasi
5. Pemeliharaan

5.4 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

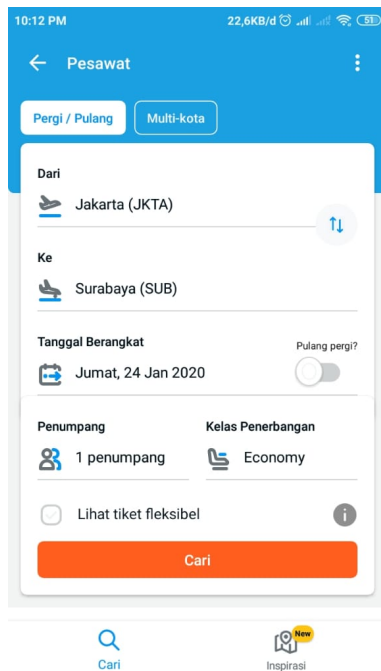
layanan sistem informasi ini, sangat memudahkan mahasiswa dalam setiap kepentingannya, ketika mengisi KRS kita tidak harus berangkat ke kampus hanya untuk mengisi KRS saja. Kita sudah bisa mengakses lewat internet sehingga bisa di akses dimana saja sekalipun kita sedang berada di luar kota. Semua informasi bisa kita dapatkan disini, mulai dari rekap nilai kita dalam setiap semester.

POLITEKNIK											
POLITEKPOS											
Statistik Penulisan Semester Ganjil 2019/2020											
Program Studi	Penerimaan KRS						Penulisan				
	Subsistem			Jumlah			Subsistem			Jumlah	
	Hbs	HK		Hbs	HK		Hbs	HK		Hbs	HK
D3 Teknik Informatika	0	0	104	857	113	104	857	0	0	104	857
D4 Teknik Informatika	0	0	284	1833	293	284	1833	0	0	284	1779
D3 Manajemen Informatika	0	0	66	119	67	66	119	0	0	66	117
D3 Akuntansi	0	0	105	944	105	944	0	0	105	936	105
D4 Akuntansi Keuangan	0	0	211	1649	212	1649	0	0	211	1649	212
D3 Manajemen Pemasaran	0	0	96	837	96	837	0	0	96	835	96
D4 Manajemen Pemasaran	0	0	344	2233	344	2233	0	0	344	2207	344
D3 Logistik Bisnis	0	0	337	3175	344	337	3175	0	0	337	3173
D4 Logistik Bisnis	0	0	729	5282	733	729	5282	0	0	728	5237
Jumlah	0	0	2296	17518	2260	2296	17518	0	0	2296	17500
Log Pengisian KRS dan Penulisan Semester Ganjil 2019/2020											
No	NPM	Penerimaan KRS			No	NPM	Penulisan			No	NPM
		Nama	Jml KRS	Waktu			Nama	Jml KRS	Waktu		
1	117036	Eko Rival Farjahan	8	28-11-2019 (18:24:01)	1	117031	Boby Setiawan	8	28-11-2019 (18:49:28)		
2	117051	Boby Setiawan	8	28-11-2019 (09:34:01)	2	117036	Eko Rival Farjahan	8	28-11-2019 (18:49:16)		
3	117038	Chandri Kusuman Sholah	8	28-11-2019 (18:52:09)	3	117038	Chandri Kusuman Sholah	8	28-11-2019 (18:49:01)		
4	119471	Raza Hafid Faza Salm	8	28-11-2019 (18:18:07)	4	119484	Razi Nuradza	1	25-11-2019 (18:24:40)		
5	119302	Kurnad Berta Adhijaya	10	25-11-2019 (18:53:10)	5	119471	Raza Hafid Faza Salm	8	26-11-2019 (18:16:33)		
6	124276	Tri Chandra Puspito	10	25-11-2019 (18:49:00)	6	119302	Piki Ahmad Sholah	7	26-11-2019 (18:24:30)		
7	118306	Erik Cumar	8	25-11-2019 (18:37:40)	7	416302	Pride Shorthing	10	25-11-2019 (18:24:00)		
8	118301	Naheem Fadhilul Haqi Prasana	4	25-11-2019 (18:35:03)	8	6164120	Almard Gazi Almuhammad	9	25-11-2019 (18:24:47)		
9	118308	Simo Nurhan Rahnema	8	25-11-2019 (18:34:40)	9	6164176	Tri Chandra Puspito	10	25-11-2019 (18:24:57)		
10	1183064	Tarmizi Agil Kurnad	5	25-11-2019 (16:40:55)	10	119302	Kurnad Berta Adhijaya	10	25-11-2019 (17:17:22)		

Gambar 5.2 Penerapan Sistem pada Website

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 5.3 Penerapan Sistem pada Mobile

BAB 6

WEBSITE

6.1 Sejarah dan Perkembangan

website pertama kali ditemukan oleh Sir Timothy John, Tim Berners-Lee. Pada tahun 1991 *website* terhubung dengan jaringan. Tujuan 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.

6.2 Pengertian

World Wide Web, yang lebih populer disingkat dengan WWW merupakan satu buah ruang informasi yang digunakan oleh pengenalan 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 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.

6.3 Kelebihan dan Kekurangan

6.3.1 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 smart-phone saat di luar rumah. Selain itu, dalam kancah bisnis *file* perusahaan dapat diubah sekaligus jika menggunakan aplikasi jenis ini, sehingga akan meningkatkan kenyamanan.

6.3.2 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 *device* 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.

6.4 Tools yang Digunakan

Berikut beberapa beberapa *tools* yang digunakan untuk membangun sebuah *web-site* :

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 Python, 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 information 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.

6.5 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 situs web berbasis *database*. Jika Anda ingin mempelajari lebih lanjut tentang PHP, situs resminya adalah PHP.net.

Contoh penggunaan PHP :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Example</title>
5   </head>
6   <body>
7
8     <?php
9       echo "Hi, I'm a PHP script!";
10    ?>
11
12   </body>
13 </html>
```


6.6 Text Editor

Text Editor adalah suatu software aplikasi atau suatu program komputer yang memungkinkan Anda sebagai pengguna 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 pemrograman. Untuk menuliskan bahasa pemrograman PHP dapat menggunakan beberapa *tools* seperti:

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

6.6.1 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/*requirements* yang dibutuhkan untuk menginstal VSCode :

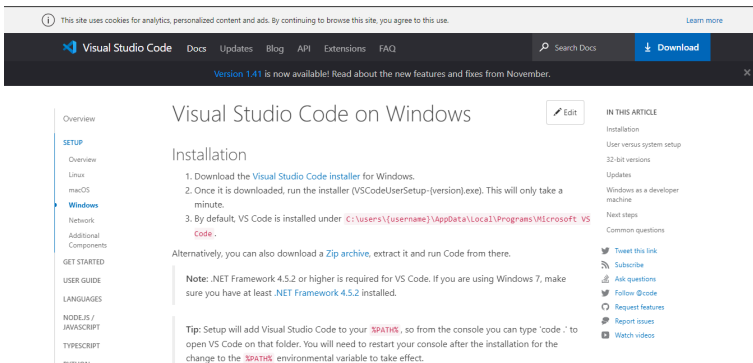
1. **Hardware**, Perangkat keras rekomendasi yang dibutuhkan yaitu memiliki *processor* 1.6 GHz or faster *processor*, dan minimal memiliki 1 GB RAM.
2. **Sistem operasi**, Sistem operasi minimal yang digunakan yaitu :
 - (a) Windows minimal windows 7 wajib Menginstal .NET Framework 4.5.2
 - (b) OS X 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

Berikut tutorial instalasi visual code pada sistem operasi widows :

1. *Download* Visual Studio Code untuk windows pada

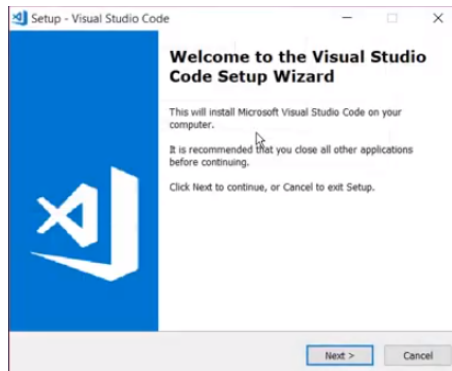
[https : //code.visualstudio.com/docs/setup/windows](https://code.visualstudio.com/docs/setup/windows)

kemudian klik *download* di pojok kanan atas.



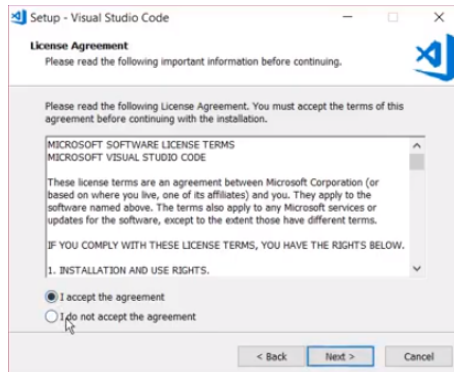
Gambar 6.1 Tutorial instalasi Visual Studio Code

2. Setelah diunduh, jalankan penginstal *VSCodeUserSetup – version.exe* sehingga akan muncul seperti gambar 6.2. Selanjutnya klik *next*.



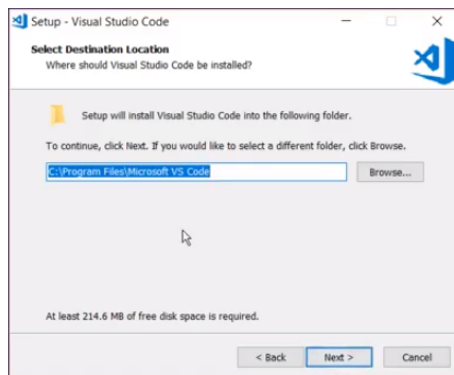
Gambar 6.2 Tutorial instalasi Visual Studio Code

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



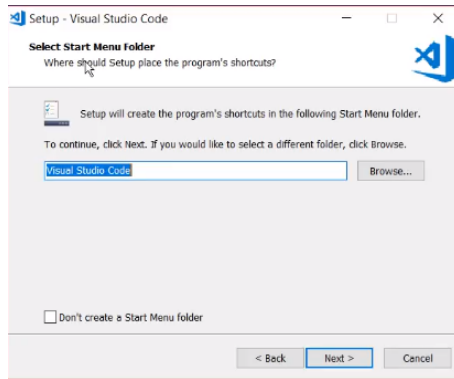
Gambar 6.3 Tutorial instalasi Visual Studio Code

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*.



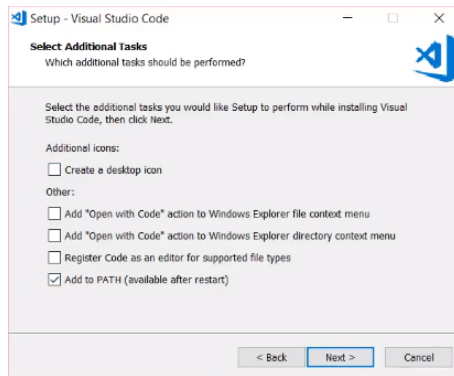
Gambar 6.4 Tutorial instalasi Visual Studio Code

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



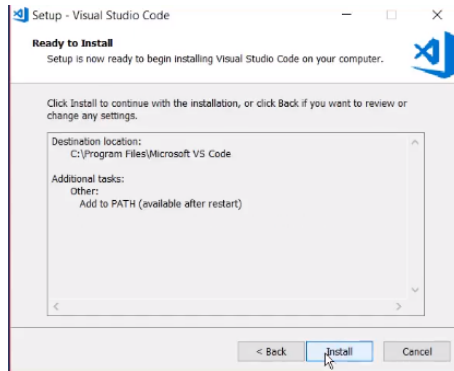
Gambar 6.5 Tutorial instalasi Visual Studio Code

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



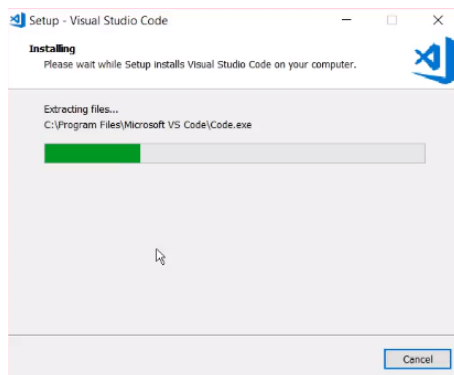
Gambar 6.6 Tutorial instalasi Visual Studio Code

7. Kemudian Klik Install



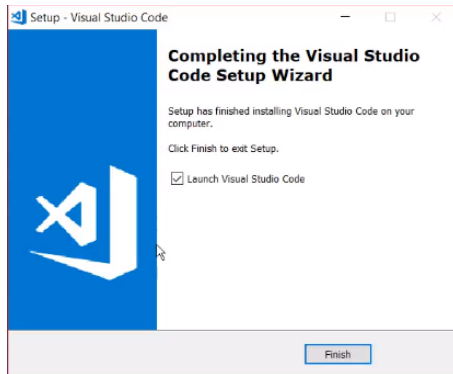
Gambar 6.7 Tutorial instalasi Visual Studio Code

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



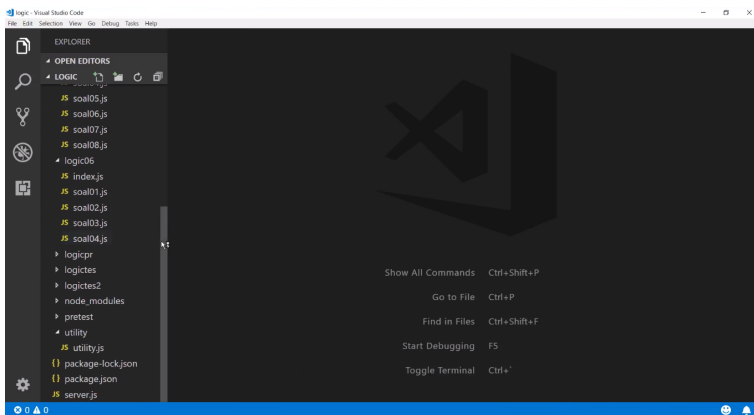
Gambar 6.8 Tutorial instalasi Visual Studio Code

9. Kemudian klik finish



Gambar 6.9 Tutorial instalasi Visual Studio Code

10. Terakhir akan menampilkan halaman Visual Studio Code seperti pada gambar 6.9



Gambar 6.10 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](https://code.visualstudio.com)

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



Gambar 6.11 Tutorial instalasi Visual Studio Code

Instalasi dengan metode grafis hampir sama dengan instalasi pada windows, 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

```
1 cd /home/[ user ]/Downloads
2
```

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

```
1 sudo dpkg --i nama-file -vscode.deb
2
```

Jika muncul pesan error seperti berikut:

```
dpkg: error processing package code (--install):
dependency problems - leaving unconfigured
Processing triggers for desktop-file-utils (0.23-3ubuntu3) ...
Processing triggers for gnome-menus (3.13.3-11ubuntu2) ...
Processing triggers for mime-support (3.60ubuntu1) ...
Errors were encountered while processing:
code
```

Gambar 6.12 Tutorial instalasi Visual Studio Code

2. Ketik perintah berikut untuk menyelesaikan:

```
1 sudo apt install libgconf-2-4
2
```

```
1 sudo apt --fix-broken install
2
```

3. Kemudian ulang langkah dpkg :

```
1 sudo dpkg --i nama-file-vscode.deb  
2
```

6.7 Web Server

6.7.1 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 situs web 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 dapat menguji situs web 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.

6.7.1.1 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 awal-

nya 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 situs web 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. 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.

6.7.1.2 *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

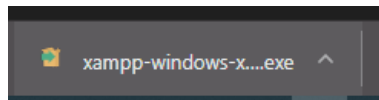
[http : //www.apachefriends.org/](http://www.apachefriends.org/)

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



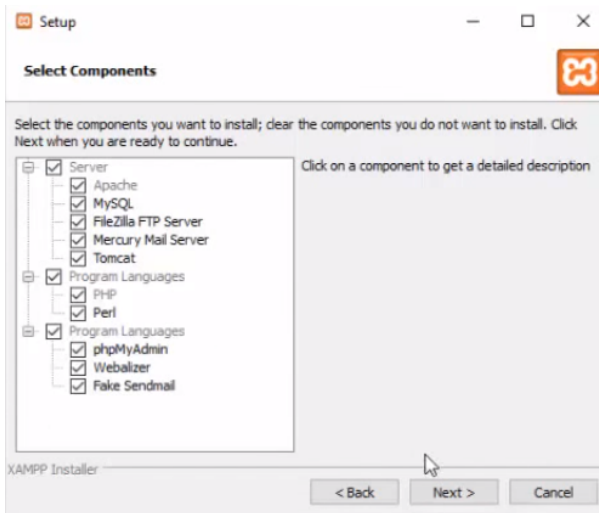
Gambar 6.13 Tutorial instalasi XAMPP

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



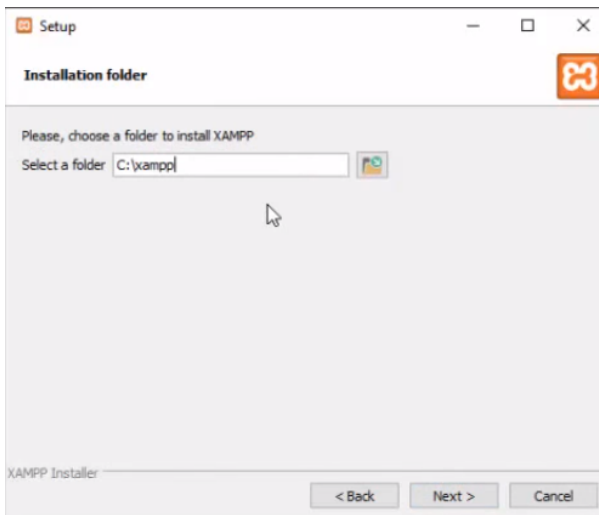
Gambar 6.14 Tutorial instalasi XAMPP

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).



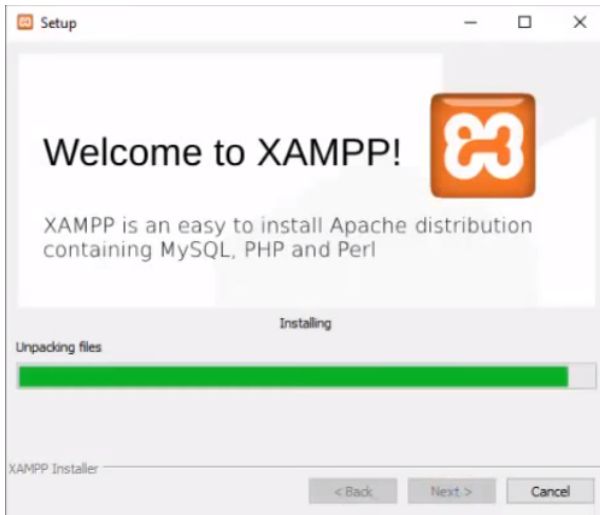
Gambar 6.15 Tutorial instalasi XAMPP

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



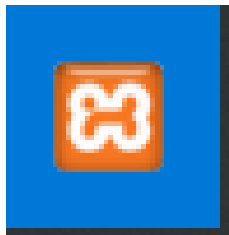
Gambar 6.16 Tutorial instalasi XAMPP

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.



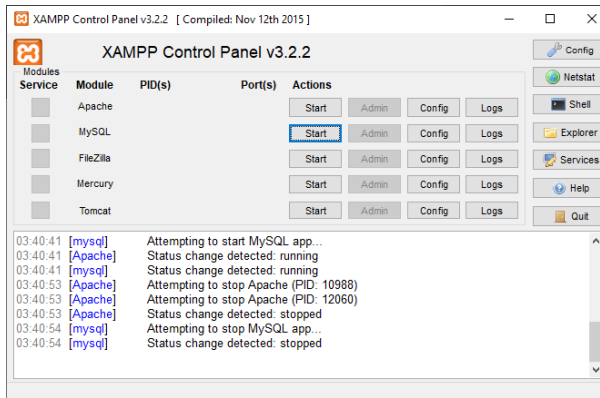
Gambar 6.17 Tutorial instalasi XAMPP

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



Gambar 6.18 Tutorial instalasi XAMPP

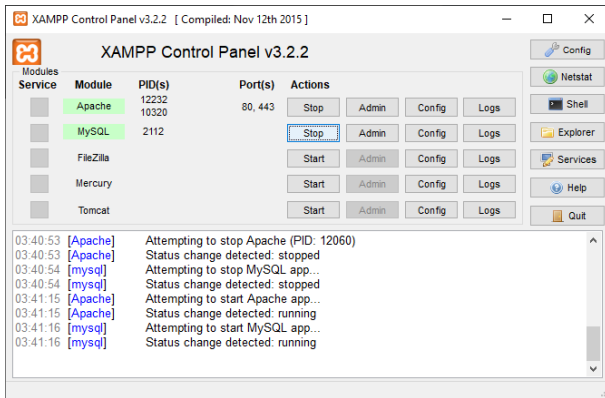
7. Setelah selesai, ikon XAMPP akan muncul di desktop Anda atau menu mulai. Dengan mengklik dua kali ikon XAMPP, jendela panel kontrol XAMPP muncul.



Gambar 6.19 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 6.20 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. *SRVROOT* Variabel yang dimodifikasi dalam *conf/httpd.conf* file 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*.

```

1 LoadModule php7_module " C: \ WebDevTools \ php73 \
  php7apache2.4.dll "
2
3 <IfModule php7_module>
4     Aplikasi AddHandler / x-httpd-php .php
5     Aplikasi AddType / x-httpd-php .php .html
6     PHPIniDir "C: \ WebDevTools \ php73"
7 </IfModule>
8

```

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, kemudian buka PowerShell sebagai Administrator dan menjalankan perintah `..exe -install` guna memasang Apache sebagai layanan.
7. Menginstal MariaDB dengan opsi default (tanpa mengubah *password root*).

6.7.2 LAMP

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

6.7.2.1 Fitur LAMP

Berikut beberapa layanan yang terdapat di LAMP :

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

6.7.2.2 Cara Instalasi

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

1. Buka terminal dan ketik:

```
1 sudo apt-get install apache2 php5-mysql libapache2-mod-
2 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 "Konfigurasi basis data untuk PHPMyAdmin dengan dbconfig-common". Pilih "iYa;" 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 situs web.

Install Apache menggunakan manajer paket Debian, apt:

```
1 $ sudo apt update
2 $ sudo apt install apache2
3
```

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 Y dan tekan ENTER untuk 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:

```
1 $ sudo ufw app list
2
```

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

```
1 Output
2 Available applications:
3 . . .
4 WWW
5 WWW Cache
6 WWW Full
7 WWW Secure
8 . . .
9
```


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

```
1 $ sudo ufw app info 'WWW Full'
2
```

```
1 Output
2 Profile : WWW Full
3 Title : Web Server (HTTP,HTTPS)
4 Description : Web Server (HTTP,HTTPS)
5
6 Ports :
7 80,443/tcp
8
```

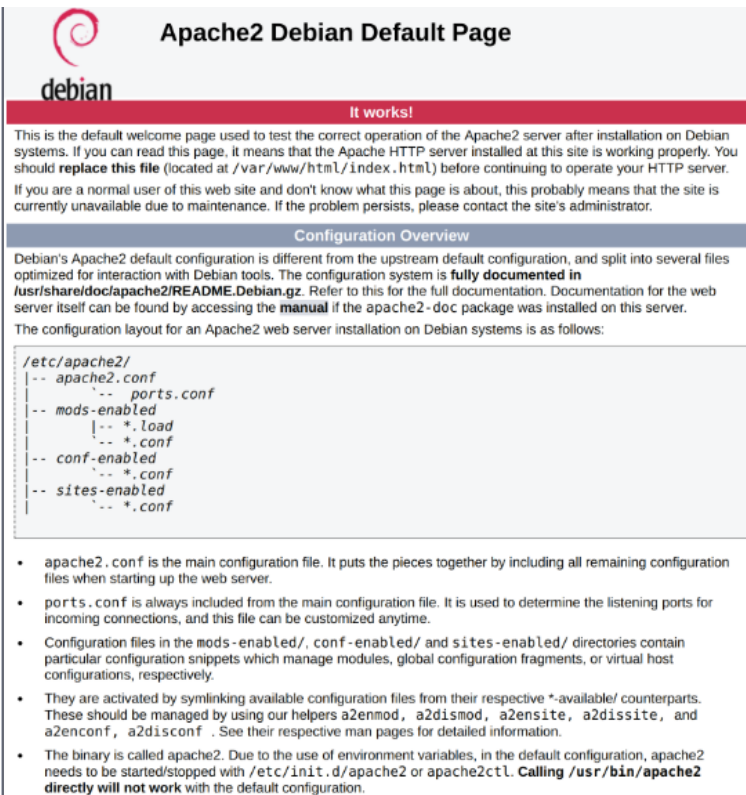
Izinkan lalu lintas HTTP dan HTTPS yang masuk untuk profil ini:

```
1 $ sudo ufw allow in 'WWW Full'
2
```

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

```
1 http://your_server_ip
2
```

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



Gambar 6.21 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 `iproute2` alat untuk mendapatkan alamat IP Anda dengan mengetik ini:

```
1 $ ip addr show eth0 | grep inet | awk '{ print $2; }' | sed '
2 s/\/.*$//'
```

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 `curl` utilitas untuk menghubungi pihak

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

```
1 $ sudo apt install curl
2 $ curl http://icanhazip.com
3
```

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-server paket, 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 apt untuk mendapatkan dan menginstal perangkat lunak ini:

```
1 $ sudo apt install mariadb-server
2
```

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:

```
1 $ sudo mysql_secure_installation
2
```

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 konfigurasi 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 socket tidak sesuai untuk kasus penggunaan Anda.

Dari sana, Anda dapat menekan Y dan kemudian ENTER menerima 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 adminden- gan kemampuan yang sama dengan akun root , tetapi dikonfigurasi untuk otentikasi kata sandi. Untuk melakukan ini, buka prompt MariaDB dari terminal Anda:

```
1 $ sudo mariadb
```

```
2
```

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:

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

```
2
```

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

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

```
2
```

Setelah ini, keluar dari shell MariaDB:

```
1 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:

```
1 $ 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:

```
1 $ 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.conf* file dalam editor teks dengan hak akses root:

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

Ini akan terlihat seperti ini:

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

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

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

Setelah selesai, simpan dan tutup file dengan menekan CTRL+X. Konfirmasikan simpan dengan mengetik Y dan kemudian tekan ENTER untuk memverifikasi file simpan lokasi.

Setelah ini, mulai ulang server web Apache agar perubahan Anda dikenali. Lakukan ini dengan mengetik ini:

```
1 $ sudo systemctl restart apache2
2
```

Anda juga dapat memeriksa status apache2 layanan menggunakan systemctl:

```
1 sudo systemctl status apache2
2
```

Kemudian akan mengeluarkan *output* seperti berikut :

```
1      apache2.service — The Apache HTTP Server
2      Loaded: loaded (/lib/systemd/system/apache2.service; enabled;
3      vendor preset: enabled)
4      Active: active (running) since Tue 2018-09-04 18:23:03 UTC; 9s
5      ago
6      Process: 22209 ExecStop=/usr/sbin/apachectl stop (code=exited ,
7      status=0/SUCCESS)
8      Process: 22216 ExecStart=/usr/sbin/apachectl start (code=exited
9      , status=0/SUCCESS)
10     Main PID: 22221 (apache2)
11     Tasks: 6 (limit: 4915)
12     CGroup: /system.slice/apache2.service
13             2 2 2 2 1 /usr/sbin/apache2 -k start
14             2 2 2 2 2 /usr/sbin/apache2 -k start
15             2 2 2 2 3 /usr/sbin/apache2 -k start
16             2 2 2 2 4 /usr/sbin/apache2 -k start
17             2 2 2 2 5 /usr/sbin/apache2 -k start
18             2 2 2 2 6 /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:

```
1 apt search php- | less
2
```

Gunakan tombol panah untuk menggulir ke atas dan ke bawah, dan tekan Q untuk keluar.

Hasilnya semua komponen opsional yang dapat Anda instal. Ini akan memberi Anda deskripsi singkat untuk masing-masing:

```
1      Sorting ...
2      Full Text Search ...
3      bandwidth-psql/stable 2.0.1+cvs20090917-10 amd64
4      Tracks usage of TCP/IP and builds html files with graphs
5
6      bluefish/stable 2.2.9-1+b1 amd64
```

```

7      advanced Gtk+ text editor for web and software
      development
8
9      cacti/stable 0.8.8h+ds1-10 all
10     web interface for graphing of monitoring systems
11
12     cakephp-scripts/stable 2.8.5-1 all
13     rapid application development framework for PHP (scripts)
14
15     ganglia-webfrontend/stable 3.6.1-3 all
16     cluster monitoring toolkit — web front-end
17
18     haser1/stable 0.9.35-2+b1 amd64
19     CGI scripting program for embedded environments
20
21     kdevelop-php-docs/stable 5.0.3-1 all
22     transitional package for kdevelop-php
23
24     kdevelop-php-docs-l10n/stable 5.0.3-1 all
25     transitional package for kdevelop-php-l10n
26
27     :
28

```

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:

```

1      $ apt show package_name
2

```

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

```

1      $ apt show php-cli
2

```

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

```

1      Description: command-line interpreter for the PHP scripting
2      language (default)
3      This package provides the /usr/bin/php command interpreter,
4      useful for
5      testing PHP scripts from a shell or performing general shell
6      scripting tasks.
7      .
8      PHP (recursive acronym for PHP: Hypertext Preprocessor) is a
9      widely-used
10     open source general-purpose scripting language that is
11     especially suited
12     for web development and can be embedded into HTML.

```

```

9      .
10     This package is a dependency package , which depends on Debian
11     's default
12     PHP version ( currently 7.0 ) .
13

```

Jika, setelah meneliti, Anda memutuskan untuk menginstal paket, Anda dapat melakukannya dengan menggunakan apt install perintah seperti yang telah Anda lakukan untuk perangkat lunak lain.

Jika Anda memutuskan itu php-cli adalah sesuatu yang Anda butuhkan, Anda dapat mengetik:

```

1      $ sudo apt install php-cli
2

```

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

```

1      $ sudo apt install package1 package2 ...
2

```

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 dikonfigurasi 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:

```

1      $ sudo nano /var/www/html/info.php
2

```

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

```

1      <?php
2      phpinfo();
3      ?>
4

```

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:

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

2

Halaman yang Anda datangi akan terlihat seperti ini:

PHP Version 7.0.30-0+deb9u1	
	
System	Linux LAMPsterdebian9-02 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-openssl.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-curl.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-dom.ini, /etc/php/7.0/apache2/conf.d/20-exception.ini, /etc/php/7.0/apache2/conf.d/20-filter.ini, /etc/php/7.0/apache2/conf.d/20-gd.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-mbstring.ini, /etc/php/7.0/apache2/conf.d/20-mysqli.ini, /etc/php/7.0/apache2/conf.d/20-mysqlnd.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-sysmsg.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	API320151012.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, compression.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, dechurk, convert.iconv.*
This program makes use of the Zend Scripting Language Engine: Zend Engine v3.6.0, Copyright (c) 1998-2017 Zend Technologies with Zend OPcache v7.0.30-0+deb9u1, Copyright (c) 1999-2017, by Zend Technologies	
	

Gambar 6.22 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:

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

2

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 situs web dan perangkat lunak web di server Anda.

6.7.3 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 situs web 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 `import 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 situs web melalui internet

6.8 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 representasi 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 agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah. Kumpulan data yang saling berhubungan disimpan bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
2. Kumpulan *file*/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik

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. Perbedaannya 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 elektronik seperti *disk* (disket atau *harddisk*). Hal merupakan konsekuensi yang logis, karena lemari arsip langsung dikelola/ditangani langsung oleh manusia, sementara basis data dikelola/ditangani melalui 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 elektronik (dengan menggunakan komputer), artinya tidak semua bentuk penyimpanan data secara elektronik 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. *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.

6.8.1 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.

6.8.1.1 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	67

Tabel 6.1 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:

```

1 select "column1"
2   [, "column2", etc ]
3   from "tablename"
4   [ where "condition" ];
5   [ ] = 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:

```
1 select first, last, city
2   from empinfo
3  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,

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

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

```
1 select * from empinfo
2  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 6.2 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.

```

1 select first , last , city from empinfo;
2
3 select last , city , age from empinfo
4     where age > 30;
5
6 select first , last , city , state from empinfo
7     where first LIKE 'J%';
8
9 select * from empinfo;
10
11 select first , last , from empinfo
12     where last LIKE '%s';
13
14 select first , last , age from empinfo
15     where last LIKE '%illia%';
16
17 select * from empinfo where first = 'Eric';

```

6.8.1.2 Membuat Tabel

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

```

1 create table "tablename"
2 ("column1" "data type",
3  "column2" "data type",
4  "column3" "data type");

```


Format buat tabel jika Anda menggunakan batasan opsional:

```
1 create table "tablename"  
2 ("column1" "data type"  
3     [constraint],  
4 "column2" "data type"  
5     [constraint],  
6 "column3" "data type"  
7     [constraint]);  
8 [ ] = optional
```

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

Contoh :

```
1 create table employee  
2 (first varchar(15),  
3 last varchar(20),  
4 age number(3),  
5 address varchar(30),  
6 city varchar(20),  
7 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
<i>char(size)</i>	String karakter dengan panjang tetap. Ukuran ditentukan dalam tanda kurung. Maksimal 255 byte.
<i>varchar(size)</i>	String karakter panjang variabel. Ukuran maks ditentukan dalam kurung.
<i>number(size)</i>	Nilai angka dengan jumlah maksimum digit kolom yang ditentukan dalam tanda kurung.
<i>date</i>	Nilai tanggal
<i>number(size,d)</i>	Nilai angka dengan jumlah maksimum jumlah "size", dengan jumlah maksimum "d" digit di sebelah kanan desimal.

Tabel 6.3 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.

6.8.1.3 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.

```

1 insert into "tablename"
2 (first_column ,... last_column)
3 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 :

```

1 insert into employee
2 (first, last, age, address, city, state)
3 values ('Luke', 'Duke', 45, '2130 Boars Nest',
4 'Hazard Co', 'Georgia');
5

```

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

6.8.1.4 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.

```

1 update "tablename"
2 set "columnname" =
3 "newvalue"
4 [, "nextcolumn" =
5 "newvalue2" ...]
6 where "columnname"
7 OPERATOR "value"
8 [and|or "column"
9 OPERATOR "value"];
10
11 [] = optional

```

Contoh :

```

1 update phone_book
2 set area_code = 623
3 where prefix = 979;
4
5 update phone_book
6 set last_name = 'Smith', prefix=555, suffix=9292
7 where last_name = 'Jones';
8
9 update employee
10 set age = age+1
11 where first_name='Mary' and last_name='Williams';

```

6.8.1.5 Delete Record

Delete statement digunakan untuk menghapus catatan atau baris dari tabel.

```

1 delete from "tablename"
2
3 where "columnname"
4     OPERATOR "value"
5 [and|or "column"
6     OPERATOR "value"];
7
8 [ ] = optional

```

Contoh :

```

1 delete from employee;

```

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

```

1 delete from employee
2   where lastname = 'May';
3
4 delete from employee
5   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.

6.8.1.6 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.

```

1 drop table "tablename"

```

Contoh :

```

1 drop table myemployees_ts0211;

```

6.8.1.7 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 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:

```
1 SELECT "list-of-columns"
2
3 FROM table1 , table2
4
5 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 6.4 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 6.5 Redundant Data

Basis data yang ideal memiliki dua tabel, seperti berikut:

1. Tabel "Customer_info" :

customer_number	firstname	lastname	address	city	state	zip
-----------------	-----------	----------	---------	------	-------	-----

Tabel 6.6 Customer info

2. Tabel "Purchases" :

customer_number	date	item	price
-----------------	------	------	-------

Tabel 6.7 Purchases

Sekarang, setiap kali pembelian dilakukan dari pelanggan berulang, tabel 6.7, "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 "customer_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:

```

1 SELECT customer_info.firstname , customer_info.lastname , purchases .
   item
2
3 FROM customer_info , purchases
4
5 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 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:

```
1 SELECT customer_info.firstname , customer_info.lastname , purchases .  
   item  
2  
3 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.

6.8.2 NoSQL

Database relasional diperkenalkan ke tahun 1970-an untuk memungkinkan aplikasi menyimpan data melalui pemodelan data standar dan bahasa *query* (*Structured Query Language*, atau SQL). Pada saat itu, penyimpanan itu mahal dan skema data cukup sederhana dan mudah. Sejak kebangkitan web, volume data yang disimpan tentang pengguna, objek, produk, dan peristiwa telah meledak. Data juga diakses lebih sering, dan diproses lebih intensif - misalnya, jejaring sosial membuat ratusan juta umpan aktivitas waktu nyata yang disesuaikan untuk pengguna berdasarkan aktivitas koneksi mereka.

Bahkan merender satu halaman web atau menjawab permintaan API tunggal dapat mengambil puluhan atau ratusan permintaan basis data saat aplikasi memproses informasi yang semakin kompleks. Interaktivitas, jaringan pengguna yang besar, dan aplikasi yang lebih kompleks semuanya mendorong tren ini.

Menanggapi permintaan ini, infrastruktur komputasi dan strategi penyebaran juga telah berubah secara dramatis. Perangkat keras *cloud* komoditas berbiaya rendah telah muncul untuk menggantikan penskalaan vertikal pada penyebaran server tunggal yang sangat kompleks dan mahal. Dan insinyur sekarang menggunakan metode pengembangan tangkas, yang bertujuan untuk penyebaran berkelanjutan dan siklus pengembangan singkat, untuk memungkinkan respon cepat terhadap permintaan pengguna untuk fitur.

NoSQL mencakup beragam teknologi basis data yang berbeda yang dikembangkan sebagai tanggapan terhadap tuntutan yang disajikan dalam membangun aplikasi modern: Pengembang bekerja dengan aplikasi yang menciptakan volume besar tipe data baru yang berubah dengan cepat - data terstruktur, semi terstruktur, tidak terstruktur, dan polimorfik. Dulu adalah siklus pengembangan *watelfall* memerlukan waktu 12-18 bulan. Sekarang tim kerja dengan skala kecil bekerja dalam sprint gesit, iterasi

dengan cepat dan mendorong kode setiap atau dua minggu, beberapa bahkan beberapa kali setiap hari. Aplikasi yang pernah melayani audiens terbatas sekarang dikirimkan sebagai layanan yang harus selalu aktif, dapat diakses dari banyak perangkat yang berbeda dan diskalakan secara global ke jutaan pengguna. Organisasi sekarang beralih ke skala arsitektur menggunakan teknologi perangkat lunak terbuka, server komoditas dan komputasi awan, bukan server monolitik besar dan infrastruktur penyimpanan. Database relasional tidak dirancang untuk mengatasi skala dan tantangan ketangkasan yang menghadapi aplikasi modern, juga tidak dibangun untuk mengambil keuntungan dari penyimpanan komoditas dan kekuatan pemrosesan yang tersedia saat ini.

NoSQL awalnya mengacu pada non SQL atau non relasional adalah database yang menyediakan mekanisme untuk penyimpanan dan pengambilan data. Data ini dimodelkan dalam cara selain hubungan tabular yang digunakan dalam database relasional. Database semacam itu muncul pada akhir 1960-an, tetapi tidak mendapatkan moniker NoSQL sampai lonjakan popularitas pada awal abad kedua puluh satu. Basis data NoSQL digunakan dalam aplikasi web waktu nyata dan data besar dan penggunaannya meningkat dari waktu ke waktu. Sistem NoSQL juga kadang-kadang disebut Tidak hanya SQL untuk menekankan fakta bahwa mereka dapat mendukung bahasa query seperti SQL.

Basis data NoSQL mencakup kesederhanaan desain, penskalaan horizontal yang lebih sederhana untuk kelompok mesin dan kontrol yang lebih baik atas ketersediaan. Struktur data yang digunakan oleh database NoSQL berbeda dari yang digunakan secara default di database relasional yang membuat beberapa operasi lebih cepat di NoSQL. Kesesuaian database NoSQL yang diberikan tergantung pada masalah yang harus dipecahkan. Struktur data yang digunakan oleh database NoSQL terkadang juga dipandang lebih fleksibel daripada tabel basis data relasional.

Banyak NoSQL *store* berkompromi meja konsistensi dalam mendukung ketersediaan, kecepatan dan toleransi partisi. Hambatan untuk adopsi yang lebih besar dari *store* NoSQL termasuk penggunaan bahasa *query* tingkat rendah, kurangnya antarmuka standar, dan investasi besar sebelumnya dalam database relasional yang ada. Sebagian besar *store* NoSQL tidak memiliki transaksi ACID (Atomicity, Konsistensi, Isolasi, Daya Tahan) yang sebenarnya tetapi beberapa basis data, seperti MarkLogic, Aerospike, c-treeACE FairCom, Google Spanner (meskipun secara teknis merupakan database NewSQL), Symas LMDB, dan OrientDB telah membuatnya pusat desain mereka.

Sebagian besar basis data NoSQL menawarkan konsep konsistensi akhirnya di mana perubahan basis data disebarkan ke semua simpul sehingga permintaan untuk data mungkin tidak segera mengembalikan data yang diperbarui atau dapat mengakibatkan pembacaan data yang tidak akurat yang merupakan masalah yang dikenal sebagai basi bacaan. Juga beberapa sistem NoSQL dapat menunjukkan penulisan yang hilang dan bentuk lain dari kehilangan data. Beberapa sistem NoSQL memberikan konsep seperti pencatatan menulis di muka untuk menghindari kehilangan data. Untuk pemrosesan transaksi terdistribusi di banyak basis data, konsistensi data adalah tantangan yang bahkan lebih besar. Ini sulit untuk NoSQL dan database relasional. Bahkan basis data relasional saat ini tidak mengizinkan batasan integritas referensial

untuk merentangkan basis data. Ada beberapa sistem yang mempertahankan standar X / Open XA dan transaksi ACID untuk pemrosesan transaksi terdistribusi.

6.8.2.1 *Kebutuhan NoSQL*

Database relasional tidak pernah dirancang untuk mengatasi skala dan tantangan ketangkasan yang menghadapi aplikasi modern - dan tidak dibangun untuk mengambil keuntungan dari penyimpanan yang murah dan kekuatan pemrosesan yang tersedia saat ini melalui *cloud*. Vendor database relasional telah mengembangkan dua pendekatan teknis utama untuk mengatasi kekurangan ini:

1. *Manual Sharding*

Tabel dipecah menjadi tabel fisik yang lebih kecil dan tersebar di beberapa server. Karena database tidak menyediakan kemampuan ini secara asli, tim pengembangan mengambil pekerjaan menyebarkan beberapa basis data relasional di sejumlah mesin. Data disimpan di setiap instance basis data secara mandiri. Kode aplikasi dikembangkan untuk mendistribusikan data, mendistribusikan pertanyaan, dan mengagregasi hasil data di semua instance database. Kode tambahan harus dikembangkan untuk menangani kegagalan sumber daya, untuk melakukan penggabungan di berbagai basis data, untuk penyeimbangan ulang data, replikasi, dan persyaratan lainnya. Selain itu, banyak manfaat dari basis data relasional, seperti integritas transaksional, dikompromikan atau dihilangkan ketika menggunakan sharding manual.

2. *Distributed Cache*

Sejumlah produk menyediakan tingkat caching untuk sistem basis data. Sistem ini dapat meningkatkan kinerja membaca secara substansial, tetapi mereka tidak meningkatkan kinerja penulisan, dan mereka menambah kerumitan untuk penyebaran sistem. Jika aplikasi Anda didominasi oleh membaca maka cache terdistribusi mungkin harus dipertimbangkan, tetapi jika aplikasi Anda didominasi oleh menulis atau jika Anda memiliki campuran membaca dan menulis yang relatif merata, maka cache terdistribusi mungkin tidak meningkatkan pengalaman anda sebagai *end user*.

Database NoSQL telah muncul sebagai tanggapan terhadap tantangan-tantangan ini dan sebagai respons terhadap peluang baru yang disediakan oleh perangkat keras komoditas berbiaya rendah dan lingkungan penyebaran berbasis cloud - dan secara alami mendukung lingkungan penerapan aplikasi modern, memenuhi kebutuhan pengembang untuk mempertahankan lapisan caching terpisah atau tulis dan pertahankan kode sharding.

6.8.2.2 *Fitur-fitur dari Database NoSQL*

NoSQL mencakup beragam teknologi basis data yang berbeda, tetapi umumnya semua basis data NoSQL memiliki beberapa fitur yang sama.

1. *Dynamic schemas*

Database relasional mengharuskan skema ditentukan sebelum Anda dapat menam-

bahkan data. Misalnya, Anda mungkin ingin menyimpan data tentang pelanggan Anda seperti nomor telepon, nama depan dan belakang, alamat, kota dan negara bagian - database SQL perlu mengetahui hal ini sebelumnya.

Ini tidak cocok dengan pendekatan pengembangan yang gesit, karena setiap kali Anda melengkapi fitur baru, skema basis data Anda sering kali perlu diubah. Jadi jika Anda memutuskan, beberapa iterasi dalam pengembangan, bahwa Anda ingin menyimpan item favorit pelanggan selain alamat dan nomor telepon mereka, Anda harus menambahkan kolom itu ke database, dan kemudian memigrasi seluruh database ke skema baru.

Jika basis datanya besar, ini adalah proses yang sangat lambat yang melibatkan downtime yang signifikan. Jika Anda sering mengubah data yang disimpan aplikasi Anda - karena Anda melakukan iterasi dengan cepat - downtime ini mungkin juga sering terjadi. Juga tidak ada cara, menggunakan database relasional, untuk secara efektif menangani data yang sepenuhnya tidak terstruktur atau tidak diketahui sebelumnya.

Database NoSQL dibangun untuk memungkinkan penyisipan data tanpa skema yang telah ditentukan. Itu membuatnya mudah untuk membuat perubahan aplikasi yang signifikan secara *real-time*, tanpa khawatir tentang gangguan layanan - yang berarti pengembangan lebih cepat, integrasi kode lebih dapat diandalkan, dan lebih sedikit waktu administrator database diperlukan.

2. *Auto-sharding, replication and integrated caching*

Karena cara mereka terstruktur, database relasional biasanya skala secara vertikal - server tunggal harus meng-host seluruh database untuk memastikan keandalan dan ketersediaan data yang berkelanjutan. Ini menjadi mahal dengan cepat, membatasi skala, dan menciptakan sejumlah kecil titik kegagalan untuk infrastruktur basis data.

Solusinya adalah skala secara horizontal, dengan menambahkan server alih-alih memusatkan lebih banyak kapasitas dalam satu server. Komputasi awan membuat ini jauh lebih mudah, dengan penyedia seperti Amazon Web Services menyediakan kapasitas yang sebenarnya tidak terbatas sesuai permintaan, dan mengurus semua tugas administrasi database yang diperlukan. Pengembang tidak perlu lagi membangun platform yang rumit dan mahal untuk mendukung aplikasi mereka, dan dapat berkonsentrasi pada penulisan kode aplikasi. Selain itu, sekelompok server komoditas dapat memberikan kemampuan pemrosesan dan penyimpanan yang sama dengan server high-end tunggal untuk sebagian kecil dari harga.

”Sharding” database di banyak contoh server dapat dicapai dengan database SQL, tetapi biasanya dilakukan melalui SAN dan pengaturan kompleks lainnya untuk membuat perangkat keras bertindak sebagai server tunggal. Database NoSQL, di sisi lain, biasanya mendukung auto-sharding, yang berarti mereka secara asli dan otomatis menyebar data ke sejumlah server yang arbitrer, tanpa memerlukan aplikasi untuk mengetahui komposisi pool server. Data dan permintaan kueri secara otomatis seimbang di seluruh server, dan ketika server turun, itu dapat dengan cepat dan transparan diganti tanpa gangguan aplikasi.

Sebagian besar basis data NoSQL juga mendukung replikasi otomatis, artinya Anda mendapatkan ketersediaan tinggi dan pemulihan bencana tanpa melibatkan aplikasi terpisah untuk mengelola tugas-tugas ini. Lingkungan penyimpanan pada dasarnya divirtualisasikan dari perspektif pengembang.

Terakhir, banyak teknologi basis data NoSQL memiliki kemampuan caching terintegrasi yang sangat baik, menyimpan data yang sering digunakan dalam memori sistem sebanyak mungkin. Ini menghilangkan kebutuhan untuk lapisan caching terpisah yang harus dipertahankan.

6.8.2.3 *Jenis Basis Data NoSQL*

Sejak "NoSQL" hanya berarti non-relasional dan bukan SQL, ada banyak cara berbeda untuk mengimplementasikan teknologi NoSQL. Secara umum, database NoSQL mencakup berikut:

1. **Key-value stores** adalah basis data NoSQL yang paling sederhana. Setiap item dalam database disimpan sebagai nama atribut, atau kunci, beserta nilainya. Contoh toko nilai kunci adalah Riak dan Voldemort. Beberapa toko nilai kunci, seperti Redis, memungkinkan setiap nilai memiliki tipe, seperti "integer", yang menambah fungsionalitas.
2. **Document databases** memasangkan setiap *key* dengan struktur data kompleks yang dikenal sebagai dokumen. Dokumen dapat berisi banyak pasangan nilai kunci, atau pasangan array kunci, atau bahkan dokumen bersarang.
3. **Wide-column stores** seperti Cassandra dan HBase dioptimalkan untuk kueri atas dataset besar, dan menyimpan kolom data secara bersamaan, bukan baris.
4. **Graph stores** digunakan untuk menyimpan informasi tentang jaringan, seperti koneksi sosial. Toko grafik termasuk Neo4J dan HyperGraphDB.

6.8.2.4 *Manfaat NoSQL*

Ketika dibandingkan dengan database relasional, database NoSQL lebih scalable dan memberikan kinerja yang unggul, dan model datanya membahas beberapa masalah yang model relasional tidak dirancang untuk mengatasi:

1. Volume besar data terstruktur, semi-terstruktur, dan tidak terstruktur dengan cepat berubah.
2. Sprint tangkas, iterasi skema cepat, dan dorongan kode yang sering.
3. Pemrograman berorientasi objek yang mudah digunakan dan fleksibel.
4. Arsitektur skala-out yang didistribusikan secara geografis alih-alih arsitektur monolitik yang mahal.

6.8.2.5 Keuntungan dari NoSQL

Ada banyak keuntungan bekerja dengan database NoSQL seperti MongoDB dan Cassandra. Keuntungan utama adalah skalabilitas tinggi dan ketersediaan tinggi.

1. Skalabilitas tinggi

Basis data NoSQL menggunakan sharding untuk penskalaan horizontal. Mem-partisi data dan menempatkannya pada banyak mesin sedemikian rupa sehingga urutan data dipertahankan. Penskalaan vertikal berarti menambahkan lebih banyak sumber daya ke mesin yang ada sedangkan penskalaan horizontal berarti menambahkan lebih banyak mesin untuk menangani data. Penskalaan vertikal tidak mudah untuk diimplementasikan tetapi penskalaan horizontal mudah untuk diterapkan. Contoh dari database penskalaan horizontal adalah MongoDB, Cassandra dll. NoSQL dapat menangani sejumlah besar data karena skalabilitas, karena data menumbuhkan skala NoSQL itu sendiri untuk menangani data tersebut secara efisien.

2. Ketersediaan tinggi

Fitur replikasi otomatis dalam basis data NoSQL membuatnya sangat tersedia karena jika ada kegagalan data mereplikasi dirinya ke keadaan konsisten sebelumnya.

6.8.2.6 Kerugian dari NoSQL

NoSQL memiliki kelemahan seperti berikut :

1. Fokus sempit

Basis data NoSQL memiliki fokus yang sangat sempit karena sebagian besar dirancang untuk penyimpanan tetapi menyediakan fungsionalitas yang sangat sedikit. Database relasional adalah pilihan yang lebih baik di bidang Manajemen Transaksi daripada NoSQL.

2. *Open-source*

NoSQL adalah basis data sumber terbuka. Belum ada standar yang dapat diandalkan untuk NoSQL. Dengan kata lain dua sistem basis data cenderung tidak sama.

3. Tantangan manajemen

Tujuan dari alat big data adalah untuk membuat pengelolaan sejumlah besar data sesederhana mungkin. Tapi itu tidak mudah. Manajemen data dalam NoSQL jauh lebih kompleks daripada database relasional. NoSQL, khususnya, memiliki reputasi sebagai tantangan untuk menginstal dan bahkan lebih sibuk untuk mengelola setiap hari.

4. GUI tidak tersedia

Alat mode GUI untuk mengakses database tidak tersedia secara fleksibel di