

BAB II

TINJAUAN PUSTAKA

2.1 Manajemen Proyek Konstruksi

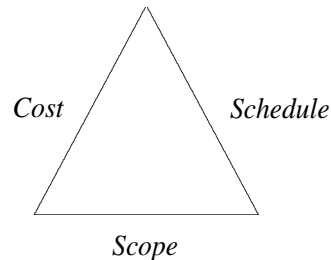
Manajemen proyek konstruksi adalah suatu metode untuk mencapai suatu hasil dalam bentuk bangunan atau infrastruktur yang dibatasi oleh waktu dengan menggunakan sumber daya yang ada secara efektif. Pada hakekatnya manajemen proyek konstruksi menurut Ervianto (2005) ada dua pemahaman yang pada pelaksanaannya menjadi satu kesatuan dalam mencapai tujuan proyek yaitu:

1. Teknologi Konstruksi (*Construction Technology*) yaitu mempelajari metode atau teknik tahapan melaksanakan pekerjaan dalam mewujudkan bangunan fisik di suatu lokasi proyek, sesuai dengan spesifikasi teknik yang disyaratkan.
2. Manajemen Konstruksi (*Construction Management*) adalah bagaimana sumber daya (*man, material, machine, money, method*) yang terlibat dalam pekerjaan dapat dikelola secara efektif dan efisien untuk mencapai tujuan proyek, sesuai dengan ketentuan/hukum yang berhubungan dengan konstruksi.

Manajemen konstruksi telah diakui sebagai suatu cabang manajemen yang khusus, yang dikembangkan dengan tujuan untuk dapat melakukan koordinasi dan pengendalian atas beberapa kegiatan pelaksanaan proyek yang sifatnya kompleks. Dengan demikian, teknik/manajemen yang dapat mengakomodasi kebutuhan sumber daya konstruksi selalu dilakukan peninjauan dan penyesuaian terus menerus, setiap saat dalam menyelesaikan pelaksanaan pekerjaan yang sedang berjalan.

Manajemen konstruksi memerlukan pengelolaan yang baik dan terarah karena suatu proyek memiliki keterbatasan hingga tujuan akhir dari suatu proyek konstruksi bisa tercapai. Pengelolaan yang diperlukan meliputi tiga hal yang dikenal dengan istilah *triple constraint* yaitu biaya (*cost*), mutu (*scope*) dan waktu (*schedule*). Ketiga batasan tersebut saling mempengaruhi dalam keberhasilan sebuah proyek.

Gambar 2.1
Project Control Triangle



Sumber : US Dept. of Transportation, *Construction Project Management Handbook*, September 2009.

Pada gambar di atas digambarkan biaya (*cost*), mutu (*scope*) dan waktu (*schedule*) sebagai sisi-sisi dari segitiga sama sisi yang saling terkait. Perubahan pada satu sisi akan berdampak pada sisi lainnya. Oleh karena itu dibutuhkan pengelolaan dari ketiga hal tersebut. Selain pengelolaan biaya, mutu dan waktu, dibutuhkan pula pengelolaan berupa manajemen sumberdaya, lingkungan, resiko dan sistem informasi. Kegiatan pengelolaan tersebut diwujudkan melalui kegiatan perencanaan (*planning*), pengorganisasian (*organizing*), pelaksanaan (*actuating*), dan pengendalian (*controlling*).

1. Perencanaan (*Planning*)

Sebuah proyek memerlukan suatu perencanaan yang matang untuk mencapai tujuan, yaitu dengan meletakkan dasar tujuan dan sasaran dari suatu proyek sekaligus menyiapkan segala program teknis dan administrasi agar dapat diimplementasikan. Hasil dari perencanaan sebagai acuan dari pelaksanaan dan pengendalian harus terus disempurnakan untuk menyesuaikan dengan perubahan dan perkembangan yang terjadi pada proses selanjutnya.

2. Pengorganisasian (*Organizing*)

Pada kegiatan ini dilakukan identifikasi dan pengelompokan jenis jenis pekerjaan, menentukan pendelegasian wewenang dan tanggung jawab perorangan serta meletakkan dasar bagi hubungan masing-masing unsur organisasi.

3. Pelaksanaan (*Actuating*)

Merupakan implementasi dari perencanaan yang telah ditetapkan. Berupa tindakan menyelaraskan seluruh anggota organisasi dalam kegiatan pelaksanaan, serta agar seluruh anggota organisasi dapat bekerja sama dalam pencapaian tujuan bersama. Proses *monitoring* dan *updating* selalu dilakukan untuk mendapatkan jadwal pelaksanaan yang realistis agar sesuai dengan tujuan proyek. Jika terjadi penyimpangan terhadap rencana semula, maka dilakukan evaluasi dan tindakan koreksi agar proyek tetap berada di jalur yang diinginkan.

4. Pengendalian (*Controlling*)

Pengendalian mempengaruhi hasil akhir suatu proyek. Tujuan utama dari kegiatan pengendalian yaitu meminimalisasi segala penyimpangan yang dapat terjadi selama berlangsungnya proyek. Kegiatan yang dilakukan dalam proses pengendalian yaitu berupa pengawasan, pemeriksaan dan koreksi yang dilakukan selama proses implementasi.

2.2 Organisasi Proyek

Kegiatan manajemen proyek tidak dapat lepas dari organisasi. Organisasi menurut Ranupandojo dan Husnan (1990) bisa dipandang sebagai suatu unit yang merubah berbagai input menjadi output yang dibutuhkan. Pengertian bentuk organisasi yang paling sederhana menurut Ervianto (2005) adalah bersatunya kegiatan-kegiatan dari dua individu atau lebih di bawah satu koordinasi, dan berfungsi mempertemukan mereka menjadi satu tujuan. Semakin banyak individu yang terlibat dengan berbagai macam kegiatan maka bentuk organisasi akan menjadi semakin kompleks.

Proses mengorganisir proyek menurut Soeharto (1995) mengikuti urutan-urutan sebagai berikut :

a. Melakukan Identifikasi dan Klasifikasi Pekerjaan

Lingkup Proyek terdiri dari sejumlah besar pekerjaan. Sebagai contoh adalah tahap implementasi fisik proyek, mulai dari menyiapkan gambar desain, pembelian material sampai dengan pelaksanaan konstruksi. Semua hal tersebut

perlu diidentifikasi dan diklasifikasi untuk mengetahui kebutuhan sumber daya dan jadwal yang dibutuhkan.

b. Mengelompokkan Pekerjaan

Setelah diidentifikasi dan diklasifikasi, selanjutnya dilakukan pengelompokan pekerjaan tersebut ke dalam unit atau paket yang masing-masing telah diidentifikasi biaya, jadwal dan mutunya. Selanjutnya dapat diserahkan kepada individu atau kelompok yang akan menangani.

c. Menyiapkan Pihak yang Akan Menangani Pekerjaan

Pada tahap ini dimulai persiapan pihak-pihak yang akan menerima tugas. Persiapan tersebut seperti memilih kelompok dengan ketrampilan dan keahlian yang sesuai dengan kebutuhan pekerjaan dan memberitahukan sasaran yang ingin dicapai terkait dengan pekerjaan yang akan diserahkan.

d. Mengetahui Wewenang dan Tanggung Jawab serta Melakukan Pekerjaan

Kelompok yang telah menerima pekerjaan harus mengetahui batas wewenang dan tanggung jawabnya agar hasil pekerjaan terlaksana sesuai harapan. Hal ini sangat penting untuk menghindari tumpang tindih wewenang.

e. Menyusun Mekanisme Koordinasi

Mekanisme koordinasi dibutuhkan mengingat banyaknya kelompok yang ikut menangani pekerjaan dan adanya keterkaitan antara pekerjaan satu dengan yang lain.

2.3 Pengendalian Proyek

Sebagai salah satu fungsi dalam kegiatan manajemen proyek, pengendalian mempunyai tujuan utama untuk meminimalisasi segala penyimpangan yang dapat terjadi selama proyek berlangsung. Menurut Mockler (dikutip oleh Husen, 2009) pengendalian dapat didefinisikan sebagai usaha yang sistematis untuk menentukan standar yang sesuai dengan sasaran dan tujuan perencanaan, merancang sistem informasi, membandingkan pelaksanaan dengan standar-standar yang telah ditetapkan, menganalisa kemungkinan terjadinya penyimpangan, kemudian melakukan tindakan koreksi yang diperlukan agar sumber daya dapat digunakan secara efektif dan efisien dalam rangka mencapai

sasaran dan tujuan. Selain agar mendapatkan produk yang memuaskan, pengendalian juga dimaksudkan untuk memastikan bahwa program dan aturan kerja yang telah ditetapkan dapat dicapai dengan penyimpangan atau kesalahan yang paling minimal.

Kegiatan pengendalian dilakukan dalam bentuk-bentuk kegiatan sebagai berikut (Husen, 2009) :

1. Supervisi, yaitu melakukan tindakan koordinasi pengawasan sesuai dengan wewenang dan tanggung jawab organisasi yang telah ditetapkan, agar dalam pelaksanaannya dapat dilakukan secara bersama-sama oleh semua personel dengan kendali pengawas.
2. Inspeksi, yaitu melakukan pemeriksaan terhadap hasil pekerjaan dengan tujuan menjamin spesifikasi mutu dan produk sesuai dengan yang direncanakan.
3. Tindakan koreksi, yaitu melakukan perubahan dan perbaikan terhadap rencana yang telah ditetapkan untuk menyesuaikan dengan kondisi pelaksanaan.

Pengelola proyek seharusnya mempunyai acuan sebagai sasaran dan tujuan pengendalian untuk memudahkan proses pengendalian. Oleh karena itu, indikator-indikator tujuan akhir pencapaian proyek harus ditampilkan dan dijadikan pegangan selama pelaksanaan proyek. Indikator-indikator yang biasanya menjadi sasaran pencapaian tujuan akhir proyek berupa *triple constrain* yaitu biaya (*cost*), mutu (*scope*) dan waktu (*schedule*). Keterlambatan dalam suatu proyek konstruksi dapat mengakibatkan peningkatan biaya dan waktu operasional pembangunan.

2.4 Analisa Jaringan Kerja (*Network Analysis*)

Pengelolaan proyek-proyek berskala besar memerlukan perencanaan, penjadwalan, dan pengordinasian yang hati-hati dari berbagai aktivitas yang saling berkaitan. Oleh karena itu kemudian dikembangkan prosedur-prosedur formal yang didasarkan atas penggunaan jaringan kerja (*network*) dan teknik-teknik *network*. Analisa jaringan kerja merupakan suatu perpaduan pemikiran yang logis, digambarkan dengan suatu jaringan yang berisi lintasan-lintasan kegiatan dan memungkinkan pengolahan secara analitis. Analisa jaringan kerja

memungkinkan suatu perencanaan yang efektif dari suatu rangkaian yang mempunyai interaktivitas. Keuntungan dari penggunaan analisa jaringan kerja adalah:

1. Dapat merencanakan suatu proyek secara keseluruhan.
2. Penjadwalan pekerjaan dalam urutan yang praktis dan efisien.
3. Pengadaan pengawasan dan pembagian kerja maupun biaya.
4. Penjadwalan ulang untuk mengatasi hambatan dan keterlambatan.
5. Menentukan kemungkinan pertukaran antara waktu dan biaya.

Salah satu prosedur yang telah dikembangkan berdasarkan jaringan kerja untuk mengatasi permasalahan pengelolaan suatu proyek adalah CPM (*Critical Path Method*). Penggambaran suatu jaringan kerja digunakan tiga buah simbol sebagai berikut :

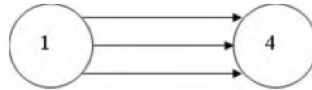
1. Anak panah (*arrow*), menyatakan sebuah kegiatan atau aktivitas. Kegiatan di sini didefinisikan sebagai hal yang memerlukan jangka waktu tertentu dalam pemakaian sejumlah sumber daya (tenaga kerja, peralatan, material, biaya)
2. Lingkaran kecil (*node*), menyatakan sebuah kejadian atau peristiwa atau *event*. Kejadian didefinisikan sebagai ujung atau pertemuan dari satu atau beberapa kegiatan.
3. Anak panah terputus-putus, menyatakan kegiatan semu atau *dummy* . *Dummy* tidak mempunyai jangka waktu tertentu, karena tidak memakai sejumlah sumber daya.

Penggunaan simbol-simbol tersebut mengikuti aturan-aturan sebagai berikut:

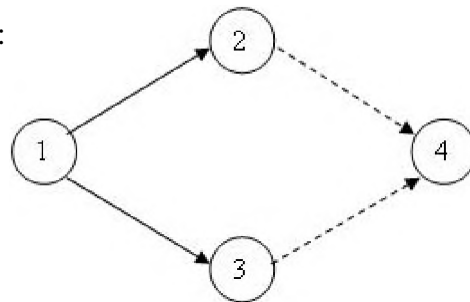
1. Di antara dua *event* yang sama hanya boleh digambarkan satu anak panah.
2. Nama suatu aktivitas dinyatakan dengan huruf atau nomor urut *event*.
3. Aktivitas harus mengalir dari *event* bernomor rendah ke *event* bernomor tinggi.
4. Diagram hanya memiliki sebuah initial *event* dan sebuah *terminal event*.

Aktivitas *dummy* adalah aktivitas yang sebenarnya tidak ada, sehingga tidak memerlukan pemakaian sumber daya. *Dummy* bisa terjadi karena:

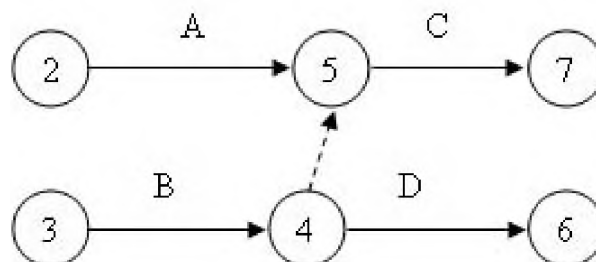
- a. Terdapat lebih dari satu kegiatan yang mulai dan selesai pada *event* yang sama, diilustrasikan pada gambar berikut.



Untuk membedakan ketiga kegiatan tersebut, maka harus digunakan *dummy* seperti berikut:



- b. Apabila suatu kegiatan, misal A dan B, harus selesai sebelum kegiatan C dapat dimulai, tetapi kegiatan D sudah dapat dimulai bila kegiatan B sudah selesai, maka digunakan *dummy* seperti berikut :



Setelah jaringan kerja digambar, kemudian dapat diestimasi waktu yang diperlukan untuk menyelesaikan masing-masing aktivitas dan menganalisis seluruh diagram *network* untuk menentukan waktu terjadinya masing-masing *event*. Dalam mengestimasi dan menganalisis waktu ini, akan terdapat satu atau beberapa lintasan tertentu dari kegiatan-kegiatan pada jaringan kerja tersebut yang menentukan jangka waktu penyelesaian seluruh proyek. Lintasan ini disebut lintasan kritis (*critical path*). Lintasan atau jalur kritis adalah jalur yang memiliki rangkaian komponen kegiatan dengan total jumlah waktu terlama dan menunjukkan kurun waktu penyelesaian yang tercepat. Pada jalur ini terletak kegiatan-kegiatan yang bila pelaksanaannya terlambat akan menyebabkan keterlambatan proyek secara keseluruhan.

Selain lintasan kritis terdapat lintasan-lintasan lain yang mempunyai jangka waktu yang lebih pendek daripada lintasan kritis. Dengan demikian, maka lintasan yang tidak kritis ini mempunyai jangka waktu untuk bisa terlambat, yang disebut *float/slack*. *Float/slack* memberikan sejumlah kelonggaran waktu dan elastisitas pada sebuah jaringan kerja, dan ini dipakai pada waktu penggunaan *network* dalam praktek, atau digunakan pada waktu penentuan jumlah material, peralatan, dan tenaga kerja. *Float* terbagi menjadi dua jenis, yaitu:

- a. *Total float/slack*, adalah selisih antara waktu yang tersedia untuk pelaksanaan kegiatan dengan waktu yang diperlukan untuk pelaksanaan kegiatan tersebut.
- b. *Free float/slack* untuk suatu kegiatan adalah waktu yang tersisa bila suatu kegiatan dilaksanakan pada waktu yang paling awal.

Untuk mempermudah perhitungan penentuan waktu digunakan notasi-notasi sebagai berikut:

TE = *earliest event occurrence time*, yaitu saat tercepat terjadinya *event*.

TL = *latest event occurrence time*, yaitu saat paling lambat terjadinya *event*.

ES = *earliest activity start time*, yaitu saat paling cepat dimulainya aktivitas.

EF = *earliest activity finish time*, yaitu saat paling cepat diselesaikannya aktivitas.

LS = *latest activity start time*, yaitu saat paling lambat dimulainya aktivitas.

LF = *latest activity finish time*, yaitu saat paling lambat diselesaikannya aktivitas.

t = *activity duration time*, yaitu waktu yang diperlukan untuk menyelesaikan suatu aktivitas.

S = *total slack/float*

SF = *free slack/float*

Asumsi yang digunakan dalam melakukan perhitungan adalah:

1. Proyek hanya memiliki satu *initial event* dan satu *terminal event*.
2. Saat tercepat terjadinya *initial event* adalah hari ke-nol
3. Saat paling lambat terjadinya *terminal event* adalah $TL = TE$ untuk *event* ini.

Adapun cara perhitungan yang harus dilakukan terdiri atas dua cara, yaitu:

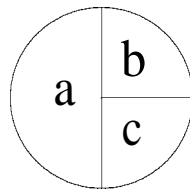
1. Perhitungan maju (*forward computation*)

Pada perhitungan ini, perhitungan bergerak dari *initial event* menuju ke *terminal event*. Tujuannya adalah untuk menghitung saat yang paling cepat terjadinya *events* dan saat paling cepat dimulainya serta diselesaikannya aktivitas-aktivitas (*earliest event time*).

2. Perhitungan mundur (*backward computation*)

Pada perhitungan ini, perhitungan bergerak dari *terminal event* menuju ke *initial event*. Tujuannya adalah untuk menghitung saat paling lambat terjadinya *events* dan saat paling lambat dimulainya dan diselesaikannya aktivitas-aktivitas (*latest event time*).

Untuk melakukan perhitungan maju dan perhitungan mundur, lingkaran *event* di bagi atas tiga bagian, yaitu:



Keterangan:

a = nomor *event*

b = saat tercepat terjadinya *event*, yang merupakan hasil perhitungan maju

c = saat paling lambat terjadinya *event*, yang merupakan hasil perhitungan mundur

Setelah kedua perhitungan di atas selesai, kemudian dilakukan perhitungan untuk mencari nilai *slack/float*. Adapun cara perhitungannya adalah sebagai berikut:

1. *Total float/slack* dihitung dengan cara mencari selisih antara saat paling lambat dimulainya aktivitas dengan saat paling cepat dimulainya aktivitas, atau dengan mencari selisih antara saat paling lambat diselesaikannya aktivitas dengan saat paling cepat diselesaikannya aktivitas.

2. *Free float/slack* aktivitas dihitung dengan cara mencari selisih antara saat tercepat terjadinya event di ujung aktivitas dengan saat tercepat diselesaikannya aktivitas tersebut.

2.5 Bagan Balok (*Bar Chart*)

Menurut Handoko (1995) Metode paling terkenal dan tertua dalam proses perencanaan dan pengawasan yang berhubungan dengan waktu adalah bagan balok atau bagan *Gantt*. Bagan balok ditemukan oleh Henry L. Gantt dalam bentuk bagan balok dengan panjang balok sebagai representasi dari durasi setiap kegiatan. Pada arah vertikal disusun daftar-daftar kegiatan, sedangkan pada kolom horisontal menunjukkan skala waktu. Saat mulai dan akhir proyek dari sebuah kegiatan dapat terlihat dengan jelas. Contoh penggunaan bagan balok ditunjukkan pada gambar berikut.

Gambar 2.2
Penggunaan Bagan Balok

Kegiatan	1	2	3	4	5	6	7	8	9	10	11	12
A												
B												
C												
D												
E												
F												
G												
H												

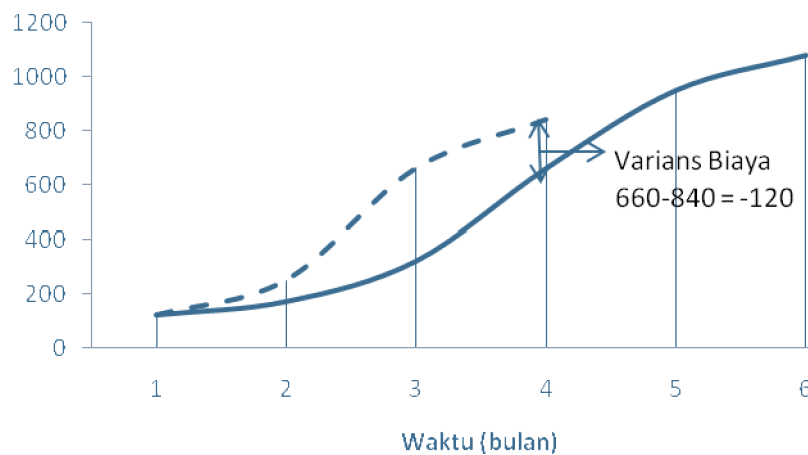
Sumber : Jay Heizer dan Barry Render, Manajemen Operasi, 2006

2.6 Kurva S

Menurut Husen (2009) kurva S atau *Hanumm curve* adalah sebuah grafik yang dikembangkan oleh Warren T. Hanumm atas dasar pengamatan terhadap sejumlah besar proyek sejak awal hingga akhir proyek. Kurva S dapat menunjukkan kemajuan proyek berdasarkan kegiatan, waktu dan bobot pekerjaan

yang direpresentasikan sebagai persentase kumulatif dari seluruh kegiatan proyek. Visualisasi dari kurva S dapat memberikan informasi mengenai kemajuan proyek dengan membandingkan antara kurva S rencana dengan realisasi. Untuk membuat kurva S, jumlah persentase kumulatif bobot masing-masing kegiatan pada suatu periode di antara durasi proyek diplotkan terhadap sumbu vertikal sehingga bila hasilnya dihubungkan dengan garis akan membentuk kurva yang berbentuk huruf S. Bentuk demikian terjadi karena volume kegiatan pada bagian awal proyek biasanya masih sedikit, kemudian pada bagian pertengahan meningkat dalam jumlah cukup besar, lalu pada akhir proyek volume kegiatan kembali mengecil. Gambar pada halaman berikut menunjukkan penggunaan kurva S.

Gambar 2.3
Penggunaan Kurva S untuk Menganalisis Varians



Sumber : Iman Soeharto, Manajemen Proyek, 1995.

2.7 Analisa Nilai Hasil (*Earned Value Analysis*)

Sumber daya proyek khususnya proyek konstruksi terdiri dari material, tenaga kerja, pendanaan, metode pelaksanaan dan peralatan. Sumber daya direncanakan untuk mencapai sasaran proyek dengan batasan waktu, biaya dan mutu. Tantangan pada pelaksanaan proyek adalah bagaimana merencanakan jadwal waktu yang efektif dan perencanaan biaya yang efisien tanpa mengurangi mutu. Waktu dan biaya merupakan dua hal penting dalam pelaksanaan pekerjaan

konstruksi selain mutu, karena biaya yang akan dikeluarkan pada saat pelaksanaan sangat erat kaitannya dengan waktu pelaksanaan pekerjaan.

Biaya proyek pada proyek konstruksi menurut Soeharto (1997) dibedakan menjadi dua jenis yaitu biaya langsung (*Direct Cost*) dan biaya tidak langsung (*Indirect Cost*). Biaya langsung adalah semua biaya yang langsung berhubungan dengan pelaksanaan pekerjaan konstruksi di lapangan. Biaya-biaya yang dikelompokkan dalam biaya langsung adalah biaya bahan/material, biaya pekerja / upah dan biaya peralatan (*equipment*). Biaya tak langsung adalah semua biaya proyek yang tidak secara langsung berhubungan dengan konstruksi di lapangan tetapi biaya ini harus ada dan tidak dapat dilepaskan dari proyek tersebut. Biaya-biaya yang termasuk dalam biaya tak langsung adalah biaya *overhead*, biaya tak terduga (*contingencies*), keuntungan/profit, pajak dan lainnya.

Hubungan biaya langsung dan biaya tak langsung terhadap waktu memiliki kecenderungan bertolak belakang. Jika waktu pelaksanaan proyek dipercepat akan mengakibatkan peningkatan biaya langsung tetapi pada biaya tidak langsung terjadi penurunan. Berdasarkan gambaran tersebut pengendalian waktu dan biaya perlu dilakukan secara terpadu atau terintegrasi. Metode pengendalian biaya dan waktu terpadu ini dikenal dengan Analisa Nilai Hasil (*Earned Value Analysis*).

Earned Value Analysis dihitung dengan cara mengkalikan biaya yang dianggarkan per pekerjaan dengan persentase penyelesaian dari pekerjaan dan menjumlahkan hasil dari semua pekerjaan dalam proyek. Proses ini lebih sulit daripada yang dibayangkan. Persentase aktual dari anggaran suatu aktivitas pada suatu saat tertentu, secara umum bukan merupakan indikator persentase penyelesaian aktivitas tersebut. Contohnya, biaya terbesar dari suatu pekerjaan adalah pengadaan alat, suatu biaya yang terjadi sebelum adanya kemajuan dalam pekerjaan tersebut. Atau mungkin biaya terbesar akan dibebankan pada saat pekerjaan selesai.

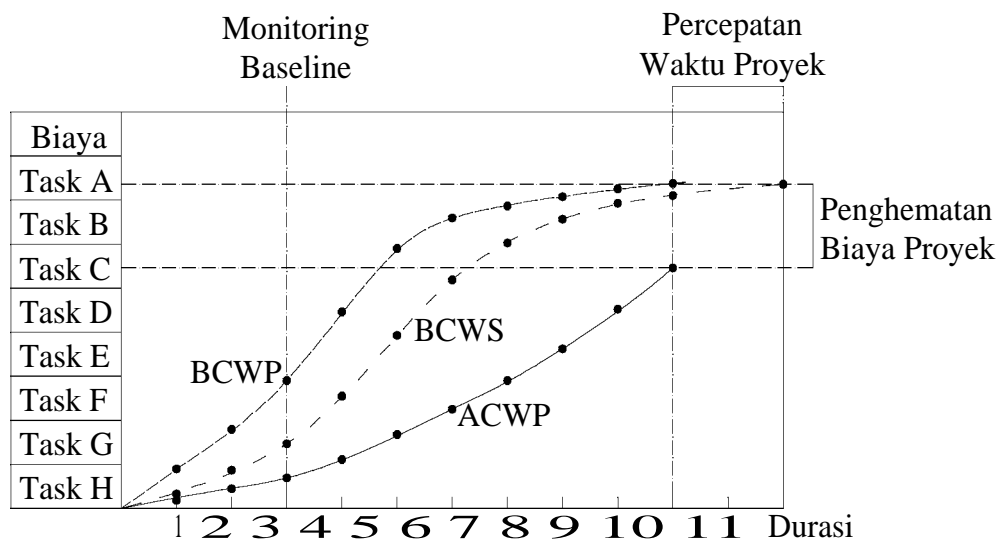
Parameter-parameter yang digunakan dalam *Earned Value Analysis* :

1. *Budgeted Cost of Work Scheduled* (BCWS). Nilai anggaran dari pekerjaan yang harus diselesaikan berdasarkan jadwal dalam suatu titik dalam proyek.

2. *Budgeted Cost of Work Performed* (BCWP). Nilai anggaran dari pekerjaan yang benar-benar telah diselesaikan pada titik tertentu pada proyek.
3. *Actual Cost of Work Performed* (ACWP). Jumlah pengeluaran yang sebenarnya dari pekerjaan yang benar-benar telah diselesaikan pada titik tertentu pada proyek.

Bentuk monitoring dan evaluasi proyek dalam pengendalian waktu dan biaya adalah dalam bentuk kurva S yang selanjutnya dimodifikasi dengan tambahan parameter-parameter di atas.

Gambar 2.4
Grafik Kinerja Biaya dan Waktu pada *Earned Value Analysis*



Sumber : Abrar Husen, Manajemen Proyek, 2009

Gambar di atas memperlihatkan grafik BCWP, BCWS dan ACWP secara bersamaan yang merupakan konsep *earned value*. Cara membaca grafik tersebut adalah dengan menentukan garis BCWP (progress proyek sebenarnya) kemudian dibandingkan dengan garis BCWS (rencana jadwal pekerjaan) dan garis ACWP (biaya yang sebenarnya dikeluarkan) dengan garis BCWS. Dapat dilihat dari grafik di atas bahwa BCWP (progress proyek sebenarnya) berada di atas BCWS (rencana pekerjaan), ini berarti progress proyek mengalami percepatan. Kemudian jika dibandingkan antara garis ACWP (biaya yang sebenarnya dikeluarkan) dengan garis BCWS terlihat bahwa garis ACWP berada di bawah garis BCWS,

ini artinya biaya yang sebenarnya dikeluarkan untuk menyelesaikan pekerjaan yang telah benar-benar dikerjakan jumlahnya lebih kecil dari biaya yang direncanakan semula. Hal tersebut menunjukkan kondisi ideal dari sebuah proyek, dimana biaya dan waktu yang digunakan untuk menyelesaikan pekerjaan ternyata lebih kecil dari yang direncanakan.

Dari ketiga parameter di atas, yaitu BCWP, BCWS dan ACWP, menurut Husen (2009) selanjutnya dilakukan analisa terhadap penyimpangan yang terjadi pada biaya dan waktu/jadwal dengan cara menghitungnya seperti diuraikan di bawah ini.

1. Penyimpangan Jadwal

a. SV (*Schedule Variance*) = $BCWP - BCWS$

$SV > 0$, *progress* aktual > rencana; terjadi percepatan proyek (*schedule overrun*)

$SV < 0$, *progress* aktual < rencana; terjadi perlambatan proyek (*schedule underrun*)

b. SPI (*Schedule Performance Index*) = $BCWP / BCWS$

$SPI > 1$, *progress* aktual > rencana; terjadi percepatan proyek (*schedule overrun*)

$SPI < 1$, *progress* aktual < rencana; terjadi perlambatan proyek (*schedule underrun*)

2. Penyimpangan Biaya

a. CV (*Cost Variance*) = $BCWP - ACWP$

$CV > 0$, pembiayaan aktual < rencana (*cost underrun*)

$CV < 0$, pembiayaan aktual > rencana (*cost overrun*)

b. CPI (*Cost Performance Index*) = $BCWP / ACWP$

$CPI > 1$, pembiayaan aktual < rencana (*cost underrun*)

$CPI < 1$, $CV < 0$, pembiayaan aktual > rencana (*cost overrun*)

Setelah mendapatkan indikasi proyek apakah akan terlambat atau lebih cepat dan apakah biaya yang harus dikeluarkan akan melebihi atau kurang dari yang dianggarkan, maka kemajuan proyek untuk waktu yang akan datang perlu diramalkan dengan cara seperti di bawah ini.

1. Perkiraan waktu penyelesaian proyek (*Estimate Completion Date* = ECD)

$ECD = (\text{sisa waktu} / SPI) + \text{waktu yang telah terpakai}$

$ECD = ((\text{jadwal rencana} - \text{waktu yang telah terpakai}) / SPI) + \text{waktu yang telah terpakai}$

2. Perkiraan biaya penyelesaian proyek (*Estimate at Completion* = EAC)

$EAC = ETC + ACWP$, dimana ETC (*Estimate to Complete*) adalah perkiraan biaya untuk menyelesaikan proyek.

$ETC = \text{sisa anggaran} / CPI$

$ETC = (\text{total anggaran biaya} - BCWP) / CPI$

2.8 Sistem Informasi Manajemen (*Management Information System*)

Menurut Handoko (1995) sistem informasi manajemen memainkan peranan penting dalam pelaksanaan fungsi-fungsi manajemen perencanaan dan pengawasan dengan efektif. Sistem informasi manajemen dapat didefinisikan sebagai suatu metode formal pengadaan dan penyediaan bagi manajemen berupa informasi yang akurat dan tepat waktu untuk membantu proses pembuatan keputusan dan memungkinkan fungsi-fungsi perencanaan, pengawasan dan operasional organisasi dilaksanakan secara efektif. Konsep sistem informasi manajemen sangat berhubungan erat dengan teknologi komputer, yang mencakup program dan bahasa pemrograman, terminal jarak jauh, *database* dan lain sebagainya. Pada dasarnya sistem informasi dapat membantu manajemen berupa informasi yang tepat pada waktu yang tepat. Selain itu, menurut Charvat (2003) sistem informasi juga sebagai sarana komunikasi yang efektif. Komunikasi adalah inti dari kesuksesan proyek. Tanpa komunikasi, proyek akan mengalami konflik, penundaan dan kegagalan.

2.9 Perencanaan Sistem Informasi

Pengertian Informasi Elektronik menurut Undang-Undang Republik Indonesia Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik pasal 1 adalah satu atau sekumpulan data elektronik, termasuk tetapi tidak terbatas pada tulisan, suara, gambar, peta, rancangan, foto, *electronic data interchange (EDI)*, surat elektronik (*electronic mail*), telegram, teleks, *telecopy* atau

sejenisnya, huruf, tanda, angka, kode akses, simbol, atau perforasi yang telah diolah yang memiliki arti atau dapat dipahami oleh orang yang mampu memahaminya. Dan pengertian Sistem Elektronik dalam Undang-Undang yang sama adalah serangkaian perangkat dan prosedur elektronik yang berfungsi mempersiapkan, mengumpulkan, mengolah, menganalisis, menyimpan, menampilkan, mengumumkan, mengirimkan, dan/atau menyebarkan Informasi Elektronik.

Sistem elektronik secara umum disebut juga sebagai sistem informasi. Suatu sistem informasi dapat dikembangkan karena adanya perencanaan terlebih dahulu. Tanpa adanya perencanaan sistem yang baik, pengembangan sistem tidak akan dapat berjalan sesuai dengan yang diharapkan. Perencanaan sistem termasuk di dalamnya adalah estimasi sumberdaya, kebutuhan-kebutuhan fisik dan biaya. Perencanaan sistem pada suatu instansi biasanya ditangani oleh staf perencanaan sistem, departemen pengembangan sistem atau departemen pengolahan data.

2.10 Program Komputer

Program komputer adalah rangkaian kata perintah yang telah dimengerti oleh komputer untuk dikerjakan. Kata-kata perintah tersebut membentuk suatu bahasa yang disebut dengan bahasa pemrograman. Sebagaimana bahasa pada manusia, bahasa pemrograman juga terdiri atas banyak macam bahasa, dan memiliki aturannya masing-masing.

Secara umum, struktur suatu program komputer terdiri dari beberapa bagian yaitu :

1. Input

Bagian ini merupakan proses untuk memasukkan data ke komputer melalui *device* yang ada misalnya *keyboard, mouse, scanner* dll. Program melakukan proses membaca data yang akan diolah dari *device* tersebut.

2. Output

Bagian ini merupakan proses untuk menampilkan data yang telah diolah, melaporkan hasil pengolahan data melalui *device* seperti *monitor, printer* dll. Program melakukan proses mencetak data ke *device* tersebut.

3. Proses Pengolahan Data

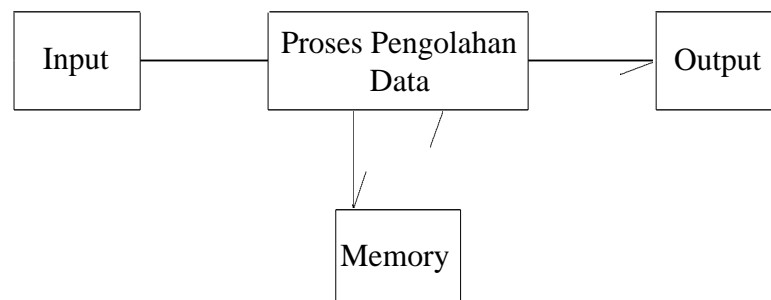
Bagian ini merupakan proses mengolah data yang di- *input* dengan menerapkan metode-metode, teknik-teknik, algoritma-algoritma yang ada. Proses ini menghasilkan data *output* yang akan ditampilkan kepada pengguna program.

4. Penyimpanan Data

Bagian ini merupakan proses menyimpan data dalam memori atau *device* penyimpanan data seperti *disket*, *harddisk*, *USB removable disk*, *Compact-Disc* dan lain-lain.

Berikut ini adalah diagram struktur dari suatu program secara umum.

Gambar 2.5
Diagram Struktur Program Komputer

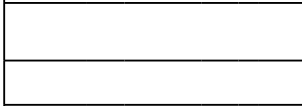
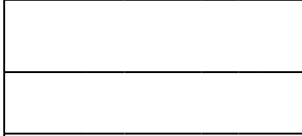
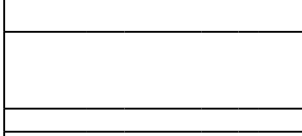
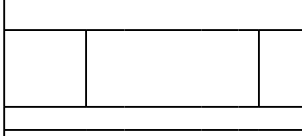
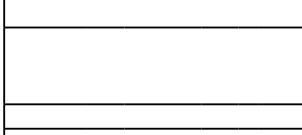
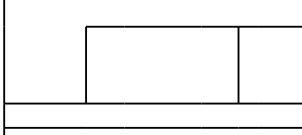


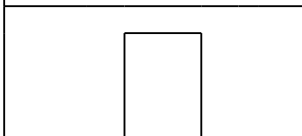


2.11 Algoritma dan Flowchart

Algoritma adalah urutan langkah-langkah logika yang menyatakan suatu tugas dalam menyelesaikan suatu masalah atau problem. Sedangkan *Flowchart* adalah gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut.

Berikut adalah beberapa simbol yang digunakan dalam menggambar suatu *flowchart* :

Tabel 2.1
Tabel simbol dan fungsi dalam *flowchart*

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inisialisasi/pemberian harga awal
	PROSES	Proses perhitungan/proses pengolahan data
	INPUT/OUTPUT DATA	Proses <i>input/output</i> data, parameter, informasi
	PREDEFINED PROCESS (SUB PROGRAM)	Permulaan sub program/proses menjalankan sub program
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	ON PAGE CONNECTOR	Penghubung bagian-bagian <i>flowchart</i> yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman berbeda

Sumber : Wiryanto Dewobroto, Aplikasi Rekayasa Konstruksi dengan Visual Basic 6.0, 2005.

2.12 Bahasa Pemrograman PHP

PHP merupakan singkatan dari *Hypertext Preprocessor*, adalah sebuah bahasa *server-side scripting* yang terpasang pada HTML (*Hypertext Markup Language*) untuk membuat sistem informasi berupa halaman *web* yang dinamis.

Sebagian besar sintaks mirip dengan bahasa C, Java dan Perl, ditambah beberapa fungsi PHP yang spesifik. Tujuan utama bahasa ini adalah untuk memungkinkan perancang *web* menulis halaman *web* yang dinamis dengan cepat.

Halaman *web* biasanya disusun dari kode-kode HTML yang disimpan dalam sebuah *file* berekstensi *.html*. *File* HTML ini dikirimkan oleh *server* ke *browser*, kemudian *browser* menerjemahkan kode-kode tersebut sehingga menghasilkan suatu tampilan yang indah. Lain halnya dengan program PHP, program ini harus diterjemahkan oleh *web-server* sehingga menghasilkan kode HTML yang dikirim ke *browser* agar dapat ditampilkan. Program ini dapat berdiri sendiri ataupun disisipkan di antara kode-kode HTML sehingga dapat langsung ditampilkan bersama dengan kode-kode HTML tersebut. Program PHP dapat ditambahkan dengan mengapit *script* program tersebut di antara tanda `<?php` dan `?>`. Tanda-tanda tersebut biasanya disebut tanda untuk *escaping* (kabur) dari kode HTML. *File* HTML yang telah dibubuhi program PHP harus diganti ekstensi-nya menjadi *.php*.

PHP merupakan bahasa pemrograman *web* yang bersifat *server-side scripting*, dimana *script*-nya menyatu dengan HTML dan berada di server. Artinya adalah sintaks dan perintah-perintah yang kita berikan akan sepenuhnya dijalankan di *server* tetapi disertakan HTML biasa. PHP dikenal sebagai bahasa *scripting* yang menyatu dengan tag HTML, dieksekusi di *server* dan digunakan untuk membuat halaman *web* yang dinamis seperti ASP (*Active Server Pages*) dan JSP (*Java Server Pages*).

PHP pertama kali dibuat oleh Rasmus Lerdorf, seorang *programmer* C. Semula PHP digunakan olehnya untuk menghitung jumlah pengunjung di dalam *web*-nya. Kemudian ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis. Versi ini pertama kali keluar pada tahun 1995. Isinya adalah sekumpulan *script* PERL yang dibuatnya untuk membuat halaman *web*-nya menjadi dinamis. Kemudian pada tahun 1996 ia mengeluarkan PHP versi 2.0 yang kemampuannya telah mampu mengakses *database* dan dapat terintegrasi dengan HTML. Pada

tahun 1998 tepatnya pada tanggal 6 Juni 1998 keluarlah PHP versi 3.0 yang dikeluarkan oleh Rasmus sendiri bersama kelompok pengembang *software*-nya.

Ketika kegiatan perdagangan melalui sistem elektronik (*electronic commerce*) semakin berkembang, situs-situs yang statis pun semakin ditinggalkan karena dianggap sudah tidak memenuhi keinginan pasar karena situs tersebut harus tetap dinamis selama setiap hari. Pada saat ini bahasa PERL dan CGI sudah jauh ketinggalan jaman sehingga sebagian besar *designer web* banyak beralih ke bahasa *server-side scripting* yang lebih dinamis seperti PHP.

Seluruh aplikasi berbasis *web* dapat dibuat dengan PHP. Namun kekuatan yang paling utama PHP adalah pada konektivitasnya dengan sistem *database* di dalam *web*. Sistem *database* yang dapat didukung oleh PHP adalah :

- a. Oracle
- b. MySQL
- c. Sybase
- d. PostgreSQL

PHP dapat berjalan di berbagai sistem operasi seperti Windows, UNIX/LINUX, Solaris maupun Macintosh. PHP merupakan *software* yang *open source* yang dapat di-*download* secara gratis dari situs resminya yaitu <http://www.php.net>. *Software* ini juga dapat berjalan pada *web server* seperti PWS (*Personal Web Server*), Apache, IIS, AOLServer, fhttpd, phttpd dan sebagainya. PHP juga merupakan bahasa pemrograman yang dapat kita kembangkan sendiri seperti menambah fungsi-fungsi baru.

Keunggulan lain dari PHP adalah PHP juga mendukung komunikasi dengan layanan seperti *protocol* IMAP, SNMP, NNTP, POP3 bahkan HTTP. PHP dapat di-*install* sebagai bagian atau uraian dari Apache *web server* atau sebagai *script* yang mandiri. Banyak keuntungan yang dapat diperoleh jika menggunakan PHP sebagai uraian dari Apache, diantaranya adalah :

- a. Tingkat keamanan yang cukup tinggi.

- b. Waktu eksekusi yang lebih cepat dibandingkan dengan bahasa pemrograman web lainnya yang berorientasi pada *server-side scripting*.
- c. Akses ke sistem *database* yang lebih fleksibel, seperti MySQL.

2.13 Web Server Apache

Salah satu *software* untuk *web server* yang biasa digunakan oleh banyak *web master* di dunia adalah Apache. *Software* tersebut dapat kita *download* secara gratis dari *web* resmi Apache yaitu <http://www.apache.org> . Apache merupakan *software open source* yang sekarang ini sudah merebut pasar dunia lebih dari 50%. *Web server* ini fleksibel terhadap berbagai sistem operasi seperti windows ataupun UNIX/LINUX. Dalam penggunaannya, Apache merupakan turunan dari *web server* yang dikeluarkan oleh NCSA yaitu NCSA HTTPd pada sekitar tahun 1995.

Kelebihan *web server* Apache :

1. *Freeware* (*software* gratisan)
2. Mudah di-*install*.
3. Mampu beroperasi pada berbagai *platform* sistem operasi.
4. Mudah mengkonfigurasinya.
5. Apache *web server* mudah dalam menambahkan periferal lainnya ke dalam *platform web server*-nya, misalnya : untuk menambahkan uraian, cukup hanya mengatur *file* konfigurasinya agar mengikutsertakan uraian itu ke dalam kumpulan uraian lain yang sudah dioperasikan.
6. Dapat dijadikan pengganti bagi NCSA *web server*.
7. Perbaikan terhadap kerusakan dan *error* pada NCSA 1.3 dan 1.4
8. Merespon *client* lebih cepat daripada *server* NCSA.
9. Mampu dikompilasi sesuai dengan spesifikasi HTTP yang sekarang.
10. Menyediakan *feature* untuk *multihomed* dan *virtual server*.
11. Kita dapat men-*setting* respon *error* yang akan dikirim *web server* dengan menggunakan *file* atau *script*.

12. Secara otomatis menjalankan *file index.html*, halaman utamanya, untuk ditampilkan secara otomatis pada *client*-nya.
13. Lebih aman karena memiliki level-level pengamanan.
14. Apache mempunyai komponen dasar terbanyak di antara *web server* lain, yang berarti bahwa *webservice* Apache termasuk salah satu dari *webservice* yang lengkap.
15. Performansi dan konsumsi sumberdaya (*resource*) dari *web server* Apache tidak terlalu banyak, standar hanya 20 MB untuk *file-file* dasarnya dan setiap *daemon*-nya hanya memerlukan seandar 950 KB memory per- *child*.
16. Mendukung transaksi yang aman (*secure transaction*) menggunakan SSL (*Secure Socket Layer*).
17. Mempunyai dukungan teknis melalui *web*.
18. Mempunyai kompatibilitas *platform* yang tinggi.
19. Mendukung *third party* berupa uraian-uraian tambahan.

2.14 PHP Pada Apache

Apache *web server* mendukung penambahan uraian-uraian. Di antara uraian yang sering dipakai adalah uraian php. PHP (*Personal Home Page Tool*) adalah salah satu jenis alat yang digunakan untuk membuat halaman *web* menjadi lebih menarik, lebih aman, dan lebih dinamis. Pada dasarnya PHP mirip dengan bahasa *script* lainnya seperti ASP, Javascript, Visual Basic atau yang lain, namun keuntungannya adalah pada PHP tidak diperlukan tambahan pada sisi *web-client* seperti halnya Javascript dan Visual Basic *script*, sehingga lebih luas penggunaannya.

Beberapa keunggulan PHP dibandingkan dengan yang lain :

1. Autentikasi http menggunakan PHP menggunakan fungsi *header()* yang ada dalam uraian apache PHP.
2. Pembuatan *file* GIF menggunakan *library* GD yang dikompilasi saat *menginstall* php.
3. PHP dapat menerima metode *upload* file.
4. Mendukung penggunaan *cookie*.

5. Mendukung integrasi dengan *database*.
6. Mendukung *ekspresi regular seperti* `ereg()`, `ereg_replace()` dan lainnya.
7. Penanganan kesalahan berdasarkan tingkat kesalahan.

2.15 PhpMyAdmin

Salah satu kata yang tidak pernah lepas dari kata PHP adalah MySQL. PHP dan MySQL kadang sebuah padanan yang tidak dapat dilepaskan satu sama lain, karena keduanya sering dipadukan menjadi satu. Perpaduan ini merupakan salah satu bagian kompatibilitas dari PHP. PHP memiliki kemampuan mengakses hampir keseluruhan basis data yang ada. Contoh umum adalah basis data MySQL. MySQL sendiri merupakan basis data yang *reliable*, yang dapat digunakan sebagai *database server*. MySQL pada dasarnya ialah sebuah aplikasi *undershell*, artinya untuk meng-konfigurasi MySQL diperlukan perintah-perintah tertentu.

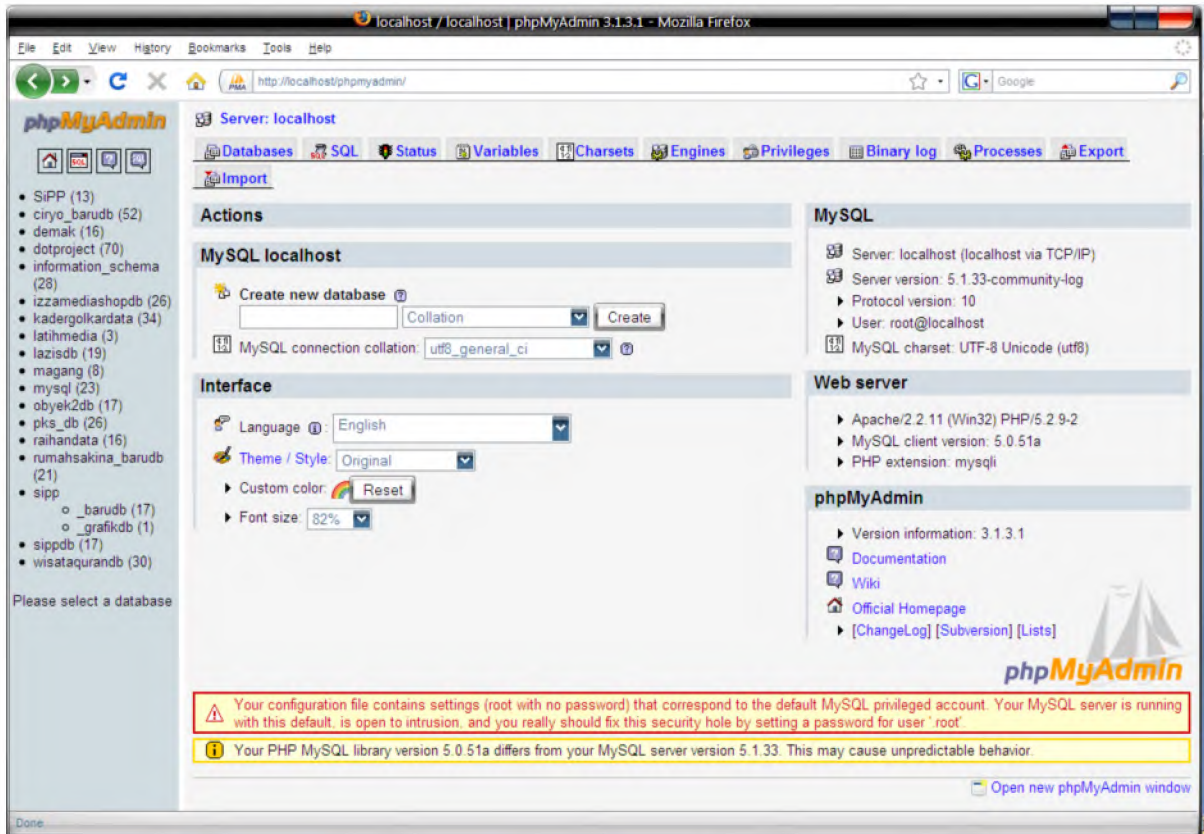
PHPTriad atau AppServ telah menyertakan phpMyAdmin di dalamnya. Ketika pertama kali halaman phpMyAdmin ditampilkan, pada *frame* sebelah kiri tampil seluruh basis data, sedangkan *frame* di sebelah kanan terdapat *textbox* untuk membuat *database*. Gambar pada halaman berikut menampilkan tampilan *interface* phpMyAdmin.

Hal pertama yang perlu dilakukan untuk membuat *database* adalah memberi nama untuk *database* tersebut. Nama tersebut digunakan sebagai syarat untuk pengaksesan ke dalam tabel-tabel yang ada dalam *database*. Nama *database* tidak boleh sama dengan *database* sebelumnya.

1. Jendela Query

Apabila sudah terdapat *query* untuk membuat tabel, maka dapat di *-paste* pada bagian ini. Selanjutnya klik *Go*. Apabila *query* tidak valid atau salah, secara otomatis akan mendapat pesan peringatan.

Gambar 2.6
Halaman Antarmuka PhpMyAdmin



2. *Browse File* yang di dalamnya terdapat sintaks MySQL

Sebuah *file* yang di dalamnya terdapat sintaks untuk membuat tabel atau jenis *query* yang lain dapat di-*browse* pada bagian ini. Setelah itu klik *Go*.

3. *Text Box*

Bagian yang terakhir ini adalah tempat membuat tabel. Setelah *database* dibuat, pastikan membuat tabel. Beberapa istilah yang harus diketahui mengenai pembuatan tabel, di antaranya :

a. *Field*

Field adalah judul yang menjadi pengelompok dalam sebuah tabel.

b. *Record*

Record dalam arti yang sederhana adalah data yang terdapat dalam sebuah tabel.

c. Tipe data

Inisialisasi jenis data. Penggunaan tipe data masing-masing *field* harus benar-benar sesuai dengan yang diinginkan.

Gambar di bawah ini merupakan atribut-atribut yang terdapat dalam suatu tabel.

Gambar 2.7
Atribut-Atribut Dalam Tabel Database

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> no	int(11)			No	None	auto_increment	[Icons]
<input type="checkbox"/> nama	varchar(100)	latin1_swedish_ci		No	None		[Icons]
<input type="checkbox"/> module	varchar(50)	latin1_swedish_ci		No	None		[Icons]
<input type="checkbox"/> urutan	int(5)			Yes	0		[Icons]
<input type="checkbox"/> admin	enum('Y','N')	latin1_swedish_ci		No	Y		[Icons]
<input type="checkbox"/> owner	enum('Y','N')	latin1_swedish_ci		No	Y		[Icons]
<input type="checkbox"/> konsultan	enum('Y','N')	latin1_swedish_ci		No	Y		[Icons]
<input type="checkbox"/> kontraktor	enum('Y','N')	latin1_swedish_ci		No	Y		[Icons]
<input type="checkbox"/> gambar	varchar(225)	latin1_swedish_ci		No	None		[Icons]

Ada beberapa hal yang perlu diketahui tentang *link-link* yang ada :

1. *Browse*

Melihat seluruh isi *record*.

2. *Change*

Mengubah *record* atau mengganti konfigurasi tabel.

3. *Drop*

Menghapus *database*, tabel, *field*, dan *record* tergantung letak *link*.

4. *Primary*

Men-Set *primary key*.

5. *Index*
Men-set *Index*.
6. *Unique*
Men-set *unique*.
7. *Empty*
Menghapus seluruh *record*.
8. *Insert*
Mengisi *record*
9. *Properties*
Melihat daftar atribut tabel.

2.16 Sistem Basis Data SQL

Dewasa ini peranan teknologi pengolahan basis data sangatlah menonjol. Basis data memiliki peranan sentral dalam sistem informasi. Dalam sistem *e-learning*, basis data juga memegang peranan yang besar, antara lain menangani informasi pengguna, informasi akademik, dan informasi lainnya yang menunjang sistem. Mengingat peranannya yang cukup besar, banyak sistem basis data ditawarkan. Salah satunya adalah MySQL. Sistem basis data MySQL didukung oleh bahasa operasional SQL singkatan dari *Structured Query Language*. SQL merupakan bahasa *query* standar yang digunakan untuk mengakses basis data relasional.

Kemampuan SQL tidak hanya untuk *query* (memperoleh data) saja, tetapi juga mencakup kemampuan lain, seperti pendefinisian struktur data, pengubahan data, pengaturan sekuritas, dan lain-lain. SQL belum dapat dikatakan sebagai bahasa tetapi sebagai subbahasa. Hal ini karena SQL tidak mendukung persyaratan bahasa yang lengkap seperti pernyataan pengujian kondisi dan pengulangan atau iterasi.

Sebagai subbahasa, SQL memiliki beberapa perintah atau pernyataan untuk melakukan fungsi-fungsinya. Tabel berikut menunjukkan pernyataan-pernyataan dasar SQL.

Tabel 2.2
Pernyataan-Pernyataan Dasar SQL

Pernyataan	Keterangan
ALTER	Mengubah struktur tabel
COMMIT	Mengakhiri sebuah eksekusi transaksi
CREATE	Menciptakan tabel, indeks, atau pandangan
DELETE	Menghapus baris pada tabel
DROP	Menghapus tabel, indeks, atau pandangan
GRANT	Menugaskan hak terhadap basis data kepada pengguna atau grup pengguna
INSERT	Menambahkan sebuah baris pada tabel
REVOKE	Membatalkan hak terhadap basis data
ROLLBACK	Mengembalikan ke keadaan semula sekiranya suatu transaksi gagal dilaksanakan
SELECT	Memilih baris dan kolom pada tabel
UPDATE	Mengubah nilai pada sebuah baris

Sumber : Allen G. Taylor, *SQL for Dummies*, 6th Edition, 2006.

1. Menciptakan Tabel

Tabel diciptakan melalui pernyataan *CREATE TABLE*. Sebagai contoh, akan dibuat tabel pekerjaan pada proyek.

```
CREATE TABLE pekerjaan (
  no INT(11),
  nama VARCHAR(200),
  tglawal DATE,
  tglakhir DATE,
  biaya INT(11))
```

Dari contoh di atas, tabel pekerjaan terdiri dari 6 kolom, yaitu no, nama, tglawal, tglakhir, volume dan biaya. Kolom no, volume dan biaya bertipe INT atau integer, yaitu tipe kolom yang berisikan angka dengan nominal 0 sampai 4.294.967.295, untuk angka negatif mempunyai nominal -2.147.483.648 sampai dengan 2.147.483.647. Pada kolom nama bertipe VARCHAR, yaitu kolom yang berisikan text dengan panjang 1 sampai 255 karakter. Kolom lainnya bertipe DATE, yaitu kolom tanggal. Angka di dalam kurung yang ada pada VARCHAR (200) menunjukkan panjang karakter yang dapat disimpan pada setiap kolom dibatasi sampai 200 karakter.

Sebagai tambahan, setiap kolom pada pendefinisian tabel dapat dilengkapi dengan UNIQUE dan NOT NULL. NULL menyatakan bahwa nilai kolom bisa tidak diisi (*default*). NOT NULL berarti bahwa nilai suatu kolom harus diisi. Sedangkan UNIQUE menyatakan bahwa nilai pada kolom tidak boleh ada yang sama (harus bersifat unik). Sebagai contoh tabel pekerjaan berikut ini.

```
CREATE TABLE pekerjaan (
no      INT(11) UNIQUE NOT NULL,,
nama   VARCHAR(200) NOT NULL,
tglawal DATE,
tglakhir DATE,
biaya  INT(11))
```

Tabel pekerjaan yang dibuat di atas mensyaratkan bahwa kolom no harus diisi dan bersifat unik atau tidak ada yang sama. Selain itu kolom nama juga harus diisi, sedangkan kolom yang lainnya bisa tidak diisi.

2. Menghapus Tabel

Tabel yang tidak terpakai dapat dihapus dengan menggunakan pernyataan DROP TABLE. Sebagai contoh pada tabel pekerjaan berikut.

```
DROP TABLE pekerjaan
```

Pernyataan di atas merupakan pernyataan untuk menghapus tabel pekerjaan .

3. Mengubah Struktur Tabel

Setelah sebuah tabel tercipta, kadangkala struktur tabel perlu diubah (misalnya penambahan kolom). Hal ini dapat dikerjakan dengan menggunakan pernyataan ALTER TABLE. Sebagai contoh pada tabel pekerjaan berikut.

```
ALTER TABLE pekerjaan ADD actualtglawal DATE
```

Pernyataan di atas merupakan perintah untuk menambahkan kolom kelamin pada tabel pekerjaan yang bertipe. Sedangkan untuk menghapus sebuah kolom tertentu dapat digunakan perintah ALTER TABLE sebagai berikut.

```
ALTER TABLE pekerjaan DROP actualtglawal
```

Perintah ALTER TABLE juga memungkinkan untuk menyisipkan sebuah kolom ke dalam sebuah tabel.

4. Menambahkan Data

Salah satu operasi yang sering digunakan dalam sistem basis data adalah penambahan data. SQL menyediakan operasi ini dengan pernyataan INSERT yang berguna untuk memasukkan data dalam tabel. Sebagai contoh pada tabel pekerjaan berikut.

```
INSERT INTO pekerjaan VALUES ( Pembersihan Lahan, 20 1-07-13, 201 07-13 250000)
```

Perintah di atas menambahkan baris pada tabel pekerjaan dengan data baru. Urutan nilai yang diletakkan dalam tanda kurung disesuaikan dengan urutan kolom dalam tabel.

5. Mengubah Data

Data yang telah tersimpan pada tabel dapat diubah dengan menggunakan pernyataan UPDATE. Sebagai contoh perubahan data pada tabel berikut.

```
UPDATE pekerjaan SET nama =Pembersihan LahanWHERE n o=1
```

Pernyataan di atas digunakan untuk mengubah isi kolom pada tabel pekerjaan dengan nama Pembersihan Lahan, khusus untuk baris dengan nilai no sama dengan 1.

6. Menghapus Data

Baris-baris yang tidak terpakai dapat dihapus dengan pernyataan DELETE. Untuk menghapus baris yang nilai no-nya 1 pada tabel pekerjaan , dapat diberikan pernyataan sebagai berikut.

```
DELETE FROM pekerjaan WHERE no= 1
```

Bila WHERE tidak disebutkan, seluruh baris pada tabel akan dihapus. Sebagai contoh pada pernyataan berikut ini.

```
DELETE FROM pekerjaan
```

Pernyataan di atas akan membuat seluruh baris pada tabel pekerjaan dihapus.

7. Menampilkan Isi Tabel

Untuk menampilkan isi tabel digunakan pernyataan SELECT. Sebagai contoh pada pernyataan berikut ini.

```
SELECT no, nama FROM pekerjaan
```

Pernyataan di atas merupakan pernyataan untuk menampilkan kolom nim dan nama yang terdapat pada tabel pekerjaan. Tabel berikut menunjukkan hasil dari pernyataan tersebut.

Tabel 2.3
Hasil Dari Pernyataan Menampilkan Kolom No Dan Nama

No	Nama
1	Pembersihan Lahan
2	Pemasangan Bowplank
3	Galian Tanah
4	Urugan Tanah
5	Urugan Pasir

Untuk menampilkan semua kolom yang terdapat pada tabel pekerjaan , dapat digunakan * sesudah pernyataan SELECT. Simbol * berarti semua kolom dipilih untuk ditampilkan. Sebagai contoh pada pernyataan berikut.

SELECT * FROM pekerjaan

Pernyataan di atas akan menghasilkan tabel seperti ditunjukkan tabel berikut .

Tabel 2.4
Tampilan Isi Tabel Pekerjaan

No	Nama	Tglawal	Tglakhir	Biaya
1	Pembersihan Lahan	2011-07-13	2011-07-13	1250000
2	Pemasangan Bowplank	2011-07-13	2011-07-13	2150000
3	Galian Tanah	2011-07-13	2011-07-13	3450000
4	Urugan Tanah	2011-07-13	2011-07-13	1520000
5	Urugan Pasir	2011-07-13	2011-07-13	2455000

Pernyataan SELECT juga memungkinkan untuk menampilkan baris-baris tertentu. Hal ini dapat dilakukan dengan menggunakan klausa WHERE. Sebagai contoh pada pernyataan berikut ini.

```
SELECT no , nama FROM pekerjaan WHERE no = 1
```

Pada contoh di atas, kolom dan nama yang ditampilkan adalah khusus untuk baris-baris yang memenuhi kondisi no = 1. Tabel berikut menunjukkan hasil dari pernyataan SELECT di atas.

Tabel 2.5
Hasil Dari Pernyataan Menampilkan Kolom No Dan Nama

No	Nama
1	Pembersihan Lahan