

BAB II

LANDASAN TEORI

2.1 Definisi Sistem

Terdapat dua kelompok pendekatan didalam mendefinisikan sistem, yaitu menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut :

Jogiyanto HM (2005,1) menjelaskan “Suatu sistem adalah suatu jaringan kerja dari prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu”.

Pendekatan sistem yang merupakan kumpulan dari elemen-elemen atau komponen-komponen merupakan definisi yang lebih luas sedangkan pendekatan sistem yang menekankan pada prosedur merupakan definisi yang lebih sempit. Komponen-komponen atau subsistem-subsistem di dalam suatu sistem tidak dapat berdiri sendiri, karena saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat dicapai

2.1.1 Karakteristik Sistem

Menurut Jogiyanto HM (2005:4) suatu sistem mempunyai karakteristik atau sifat- sifat yang tertentu, sebagai berikut :

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari jumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-

komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Suatu sistem dapat mempunyai sistem yang lebih besar, yang disebut dengan *supra system*.

2. Batas Sistem (*Boundary*)

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

3. Lingkungan Luar Sistem (*Environment*)

Adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar ini dapat bersifat menguntungkan atau merugikan sistem tersebut.

4. Penghubungan Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukkan Sistem (*Input*)

Adalah energi yang dimasukkan ke dalam sistem. Masukkan dapat berupa masukkan perawatan (*maintenance input*) dan masukkan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem (*Output*)

Adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Dapat berupa masukan untuk subsistem lainnya.

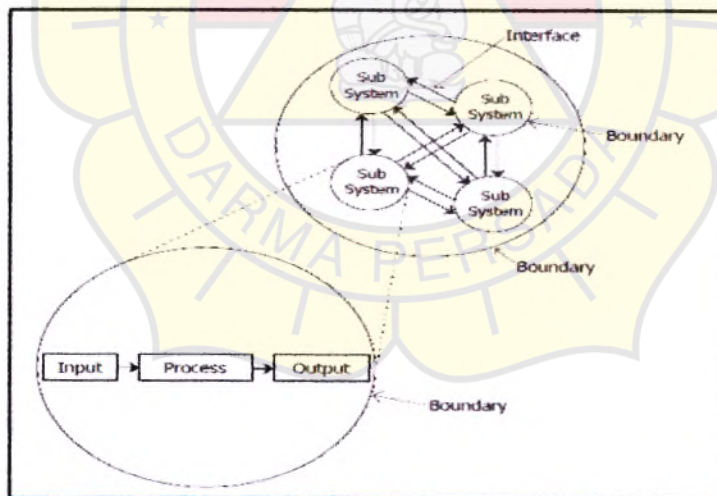
7. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran yang berguna.

8. Sasaran Sistem (*Objective*)

Suatu sistem pasti memiliki tujuan (*goal*) dan sasaran (*objective*). Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dapat dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

Karakteristik sistem juga dapat digambarkan sebagai berikut :



Gambar 2.1 Karakteristik Sistem

2.1.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

1. Sistem diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*). Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sedangkan sistem fisik merupakan sistem yang ada secara fisik.
2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Sedangkan sistem buatan manusia adalah sistem yang dirancang oleh manusia.
3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
4. Sistem dikalsifikasikan menjadi dua yaitu sistem tertutup (*closed system*) dan terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

2.2 Definisi Informasi

Menurut Yakub (2012,8) Informasi adalah data yang telah diolah menjadi bentuk yang lebih berguna dan berarti bagi yang menerimanya. Di dalam dunia

usaha saat ini yang penuh persaingan, informasi merupakan suatu yang berharga karena dengan penguasaan yang baik atas informasi tersebut perusahaan akan dapat mengambil atau memperoleh kesempatan yang lebih luas untuk berkembang. Dengan adanya informasi yang memadai suatu perusahaan akan dapat mengambil keputusan yang akan mendukung kemajuan perusahaan itu sendiri.

Kualitas dari suatu informasi (*quality of information*) tergantung dari tiga hal, yaitu:

1. Akurat (*accurate*), berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.
2. Tepat pada waktunya (*timeliness*), berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan.
3. Relevan (*relevance*), berarti informasi tersebut mempunyai manfaat untuk pemakaiannya.

2.3 Definisi Sistem Informasi

Jogiyanto HM menjelaskan (2005,11) “Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan”. Sistem informasi merupakan sistem pembangkit informasi, kemudian

dengan integrasi yang dimiliki antarsubsystem, maka sistem informasi akan mampu menyediakan informasi yang berkualitas, tepat, cepat dan akurat sesuai dengan manajemen yang membutuhkannya.

Dapat disimpulkan bahwa dengan adanya suatu sistem informasi maka suatu organisasi dapat menyediakan kepentingan aktifitas-aktifitasnya. Sistem informasi itu sendiri menurut John Burch dan Gary Grudnitski mengemukakan bahwa sistem informasi terdiri dari komponen-komponen, yaitu:

1. Blok masukan, *input* mewakili data yang masuk ke dalam sistem informasi. *Input* disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.
2. Blok Model, blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.
3. Blok keluaran, produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Blok teknologi, teknologi merupakan "kotak alat" (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

5. Blok basis data (*database*), kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.
6. Blok kendali, beberapa pengendalian perlu dirancang dan diterapkan untuk menyelesaikan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun dapat diatasi dengan cepat.

2.4 Definisi Analisa Sistem Informasi

Menurut Jogiyanto HM (2005,129), analisa sistem informasi dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

Untuk mendapatkan hasil yang baik dalam merancang suatu sistem, maka diperlukan terlebih dahulu adanya sistem informasi tentang sistem yang sedang berjalan, meliputi deskripsi prosedur dan analisa dokumen yang digunakan.

Beberapa alasan perlunya menganalisa sistem yang sedang berjalan adalah sebagai berikut :

1. Sistem yang sedang berjalan akan dijadikan dasar untuk perancangan sistem baru atau yang akan dikembangkan.
2. Untuk bahan pertimbangan dalam perancangan sistem yang baru.
3. Sistem yang sedang berjalan bisa dijadikan bahan acuan untuk menentukan karakteristik sistem yang baru.

Dalam analisa sistem informasi terdapat pula langkah-langkah dasar yang harus dilakukan oleh analisa sistem sebagai berikut :

1. *Identify*, yaitu mengidentifikasi masalah
2. *Understand*, yaitu memahami kerja dari sistem yang ada
3. *Analyze*, yaitu menganalisa sistem
4. *Report*, yaitu membuat laporan hasil analisis.

2.5 UML

2.5.1 Definisi UML

Menurut Munawar (2005:17) UML (*Unified Modelling Language*) adalah salah satu alat bantu yang saat ini sering digunakan di dunia pengembangan sistem yang berorientasi obyek. Pada saat ini dalam pengembangan suatu *web* banyak menggunakan alat bantu berupa UML (*Unified Modelling Language*) dalam merancang suatu sistem yang akan dibuat, dikarena dapat mempermudah bagi perancangan dalam merancanganya.

UML (*Unified Modelling Language*) merupakan suatu alat yang telah menjadi standar dalam merancang suatu sistem visualisasi dan mendokumentasikan sistem piranti lunak. Serta menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakannya kita dapat membuat model untuk semua jenis aplikasi lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax/semantik*. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya : Grady Booch OOD (*Object Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object Oriented Software Engineering*). Sebagai bahasa pemrograman, UML dapat menterjemahkan diagram yang ada menjadi code program yang siap untuk dijalankan. UML mempunyai sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. Diagram menggambarkan atau mendokumentasikan beberapa aspek dari sebuah sistem.

2.5.2 Model – Model Diagram UML

Beberapa literature menyebutkan bahwa *UML* menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram perwaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

1. Diagram Kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, Antarmuka-Antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan system berorientasi

objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.

2. Diagram paket (*Package Diagram*). Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.
3. Diagram *Use Case*. Bersifat statis. Diagram ini memperlihatkan himpunan *Use Case* dan Aktor-Aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.
4. Diagram interaksi dan *Sequence* (urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
5. Diagram Komunikasi (*Communication Diagram*). Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML 1.4 yang menekankan organisasi structural dari objek-objek yang menerima serta mengirim pesan.
6. Diagram *Statechart* (*Statechart Diagram*). Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), transisi, kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.

7. Diagram aktivitas (*Activity Diagram*). Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas yang lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali antar objek.
8. Diagram komponen (*Component Diagram*). Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka, serta kolaborasi-kolaborasi.
9. Diagram *Deployment* (*Deployment Diagram*). Bersifat statis. Diagram ini memperlihatkan konfigurasi saat konfigurasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen yang ada didalamnya. Diagram *Deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*Distributed Computing*).

Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semua dibuat sesuai kebutuhan. Pada *UML* dimungkinkan kita menggunakan diagram-diagram yang lainnya (misalnya *Data Flow Diagram*, *Entity Relationship Diagram* dan sebagainya).

2.5.3 Function Modelling

2.5.3.1 Activity Diagram

Activity diagram adalah teknik untuk mendiskripsikan logika *procedural*, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa (Munawar, 2005:109).

2.5.3.2 Use Case Diagram

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna (Munawar, 2005:63). *Use Case Diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem Yang ditekankan adalah “apa” yang diperbuat sistem bukan “Bagaimana”.

Sebuah *Use Case* mempresentasikan sebuah interaksi antar aktor dengan sistem. Seorang aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem agar dapat melakukan sesuatu hal yang telah ditentukan. *Use Case* dapat membantu memprestasikan sebuah rancangan kepada *klien*.

2.5.3.3 Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) diantaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya

komponen terbentuk dari beberapa *class* dan / atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

2.5.3.4 Deployment Diagram

Deployment / physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur system, dimana komponen akan terletak (pada mesin, server atau piranti keras), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya.

2.6 Bahasa Pemrograman PHP

PHP singkatan dari PHP Hypertext Preprocessor yaitu bahasa pemrograman web server-side yang bersifat open source. PHP merupakan script yang menyatu dengan HTML dan berada pada server (server side HTML embedded scripting). PHP adalah script yang digunakan untuk membuat halaman web yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru/up to date. Semua script PHP dieksekusi pada server dimana script tersebut dijalankan.

2.7 Database MYSQL

Database menurut Uus Rusmawan (2011), adalah kumpulan data yang terintegrasi satu sama lain. Setiap user akan diberi wewenang untuk dapat mengakses data didalam database. Database biasanya terorganisasi dalam beberapakomponen yang terdiri dari satu atau lebih table. Table digunakan untuk menyimpan data yang terdiri dari baris dan kolom. Akses terhadap data dapat berupa menampilkan, memodifikasi, dan menambah atau menghapus data yang telah tersimpan. Akses tersebut di atas dapat dilakukan oleh MySQL yaitu salah satu database yang memiliki koneksitas yang baik.

MySQL adalah sebuah sistem manajemen database relasi (*relational database management system*) yang bersifat “terbuka” (*open source*). Terbuka maksudnya adalah MySQL boleh didownload oleh siapa saja. Baik versi kode program aslinya (*source code program*) maupun versi biner nya (*executable program*) dan bisa digunakan secara (relatif) gratis baik untuk dimodifikasi sesuai dengan kebutuhan seseorang maupun sebagai program aplikasi komputer. MySQL menggunakan bahasa standar SQL (*structure query language*) sebagai bahasa intraktif dalam mengolah data. Perintah SQL juga sering disebut query, karena menggunakan bahasa standar yang sama maka tidak akan menjadi kendala besar bila suatu saat nanti berhubungan dengan database selain MySQL. Tentu saja tetap ada perbedaan sintaks bahasa paket program SQL tersebut. Adapun beberapa kelebihan MySQL adalah sebagai berikut:

1. Kecepatan. Dengan adanya fitur tambahan *Query Chacing* pada MySQL rilis 4.0, kinerja query secara umum akan naik rata-rata 200% dari kinerja biasanya.

2. Open Source. Dengan konsep ini siapapun dapat berpartisipasi mengembangkan MySQL dan hasil pengembangan itu diserahkan kepada umum atau kepada komunitas Open Source.
3. Kapabilitas. MySQL telah digunakan untuk mengelola database dengan jumlah 50 juta record. Bahkan sanggup untuk mengelola 60.000 tabel dengan jumlah baris 5.000.000.000. mendukung penggunaan index hingga 32 index per tablenya, sedangkan setiap indexnya terdiri dari 1 hingga 16 kolom kriteria. Informasi ini dapat diperoleh dari manual MySQL yang dapat di download dari situs MySQL.
4. Konektifitas dan keamanan. MySQL mendukung dan menerapkan sistem keamanan dan ijin akses tingkat lanjut, termasuk dukungan keamanan dengan pengacakan lapisan data (*SSL transport layer encryption*).
5. Fleksibilitas dan portabilitas. MySQL mendukung perintah-perintah ANSI SQL 99 dan beberapa perintah database alternatif lainnya sehingga mempermudah untuk beralih dari dan ke MySQL.
6. Lintasan platform sistem operasi. MySQL dapat dijalankan pada beberapa sistem operasi yang berbeda seperti linux, microsoft windows, freeBSD, sun solaris, mac OS, HP-UX, AIX, QNX, novell netware, SCO openunix, SGI iric, dan dec OSF.

2.8 XAMPP

Xampp menurut Uus Rusmawan (2011), merupakan sebuah tool yang menyediakan paket perangkat lunak kedalam satu buah paket. Dengan menginstal XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server

Apache, dan MySQL secara manual. XAMPP akan menginstalasi dan mengkonfigurasikannya secara otomatis untuk anda atau auto konfigurasi.

Software XAMPP terdiri atas:

1. APACHE. Tugas utama apache adalah menghasilkan halaman web yang benar kepada peminta, maka dapat saja suatu database diakses terlebih dahulu (misalnya dalam MySQL) untuk mendukung halaman yang dihasilkan.
2. MySQL. MySQL dapat digunakan untuk membuat dan mengelola database beserta isinya. Dengan MySQL dapat digunakan untuk menambahkan, mengubah dan menghapus data yang berada didalam database. Artinya data-data yang dikelola dalam database akan diletakkan dalam beberapa tabel yang terpisah sehingga manipulasi data akan menjadi jauh lebih cepat.
3. PHPMyAdmin. Pengelolaan database dengan MySQL harus dilakukan dengan mengetikkan baris-baris perintah yang sesuai (*command line*) untuk setiap maksud tertentu. Jika ingin membuat database, ketikkan baris perintah sesuai untuk membuat database. Jika ingin menghapus tabel ketikkan baris perintah yang sesuai untuk menghapus tabel. Hal tersebut tentu menyulitkan karna harus hafal dan mengetikkan perintahnya satu persatu. Banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola database dalam MySQL, salah satunya adalah phpMyAdmin. Dengan phpMyAdmin kita dapat membuat tabel, mengisi data dan lain-lain dengan mudah tanpa harus menghafal perintahnya.