

DASAR PEMROGRAMAN WEB

Teori & Implementasi

(HTML, CSS, Javascript, Bootstrap, CodeIgniter)

Penulis :

Devie Rosa Anamisa

Fifin Ayu Mufarroha



DASAR PEMROGRAMAN WEB

Teori & Implementasi

(HTML, CSS, Javascript, Bootstrap, Codelgniter)

© 2020

Penulis

Devie Rosa Anamisa

Fifin Ayu Mufarroha

Desain Cover dan Penata Isi

Tim MNC Publishing

Cetakan I, Desember 2020

Diterbitkan oleh :



Media Nusa Creative

Anggota IKAPI (162/JTI/2015)

Bukit Cemara Tidar H5 No. 34, Malang

Telp. : 0812 3334 0088

E-mail : mncpublishing.layout@gmail.com

Website : www.mncpublishing.com

ISBN 978-602-462-566-5

Hak Cipta dilindungi undang-undang. Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ke dalam bentuk apapun, secara elektronik maupun mekanis, termasuk fotokopi, merekam, atau dengan teknik perekaman lainnya, tanpa izin tertulis dari Penerbit. Undang-Undang Nomor 19 Tahun 2000 tentang Hak Cipta, Bab XII Ketentuan Pidana, Pasal 72, Ayat (1), (2), dan (6)

Kata Pengantar

Alhamdulillah puji syukur kehadiran Allah SWT yang telah memberikan kekuatan kepada kami sehingga bisa menyelesaikan buku ajar berjudul “Dasar Pemrograman Web: Teori dan Implementasi (HTML, CSS, Javascript, Bootstrap, CodeIgniter)”. Kami berharap buku ini bisa menjadi panduan bagi mahasiswa, praktisi, akademisi dan pihak terkait lainnya untuk mengetahui perkembangan dari bidang ilmu Teknik Informatika. Di dalam buku ini terdapat studi kasus permasalahan dalam membangun web dasar.

Masih banyak kekurangan dan kelemahan dalam buku ini, untuk itu kami mohon maaf sebesar-besarnya.

Tim Penulis

Daftar Isi

Kata Pengantar.....	iii
Daftar Isi	iv

BAB I

Konsep Dasar Web	1
1.1 Pengertian Web.....	1
1.2 Sejarah Web.....	2
1.3 Jenis-Jenis Web	2
1.4 Manfaat Web	3
1.5 Bahasa Pemrograman.....	3
1.6 Basis Data	4
1.7 Metode Pengembangan Perangkat Lunak.....	5
1.8 Struktur Navigasi.....	12
1.9 EntityRelationshipDiagram.....	13
Rangkuman	15
Latihan Soal.....	16

BAB II

HTML	17
2.1 Pengertian HTML	17
2.2 Sejarah HTML.....	18
2.3 Struktur Dasar dan Ekstension File HTML	21
2.4 Pengertian Element, Tag, dan Atribut dan Value.....	23
2.5 Pengaturan Halaman HTML.....	25

Rangkuman	36
Latihan Soal.....	37

BAB III

CSS.....	39
3.1 Pengenalan CSS	39
3.2 Sejarah CSS.....	40
3.3 Fungsi dan Kegunaan CSS.....	42
3.4 Sintaks Dasar CSS	43
3.4 Jenis Selector	45
Rangkuman	46
Latihan Soal.....	48

BAB IV

XML	51
4.1 Pengertian XML.....	51
4.2 Sintaks XML.....	53
4.3 Element, Attribute, Entity	54
4.4 Kelebihan dan Keunggulan Menggunakan XML	55
4.5 Kekurangan dan Kerugian Menggunakan XML	58
4.6 XML Parser dan Document Type Definition (DTD).....	60
4.7 Data Elemen Objek dan Array.....	61
Rangkuman	64
Latihan Soal.....	65

BAB V

JavaScript.....	67
5.1 Pengertian Javascript	67
5.2 Penulisan Javascript	68
5.3 Variabel dan Tipe Data dalam Javascript	72
5.4 Operator pada Javascript	74
5.5 Fungsi pada Javascript	80
5.6 Pernyataan Bersyarat	82
5.7 Struktur Perulangan	88
5.8 Form	92
5.9 Event Handler	95
Rangkuman	98
Latihan Soal.....	100

BAB VI

Bootstrap	101
6.1 Pengertian Bootstrap	101
6.2 Framework CSS dengan Bootstrap	104
6.3 Pengenalan Fitur Bootstrap	106
Rangkuman	121
Latihan Soal.....	122

BAB VII

CodeIgniter.....	123
7.1 Pengertian CodeIgniter	123
7.2 MVC (Model View Controller).....	125
7.3 CodeIgniter Library dan Helper.....	129
7.4 Konfigurasi CodeIgniter	133
Rangkuman	141
Latihan Soal.....	142

BAB VIII

Implementasi Sistem.....	143
8.1 Latar Belakang.....	143
8.2 Ide Gagasan.....	144
8.3 Perancangan Sistem	144
8.4 Implementasi	149
 Daftar Pustaka	 215

BAB I

Konsep Dasar Web

1.1 Pengertian Web

Ilustrasi web sangat berkaitan erat dengan jaringan internet, dimana halaman web membutuhkan jaringan internet sebagai media penyampaian. Sedangkan jaringan internet membutuhkan halaman web sebagai isi ataupun content yang dapat dibagikan kepada para penggunanya. Menurut Wikipedia, web adalah berkas yang ditulis sebagai berkas teks biasa atau disebut plain text, yang diatur dan dikombinasikan dengan instruksi – instruksi berbasis HTML atau pun XHTML. Berkas web ini nantinya akan diterjemahkan oleh mesin browser dan ditampilkan menjadi sebuah halaman atau situs web yang biasa kita lihat. Pengertian website menurut para ahli, adalah suatu dokumen berupa sekumpulan halaman yang berisi berbagai informasi berbentuk digital. Informasi itu bisa berupa teks, gambar, animasi, video atau gabungan dari semuanya yang disediakan melalui internet dan dapat diakses oleh banyak orang di seluruh dunia selama memiliki koneksi internet. Website Menurut Gregorius adalah kumpulan halaman web yang saling terhubung dan file-filenya saling terkait. Web terdiri dari page atau halaman, dan kumpulan halaman yang dinamakan homepage. Homepage berada pada posisi teratas, dengan halaman-halaman terkait berada di bawahnya. Biasanya setiap halaman di bawah homepage disebut child page, yang berisi hyperlink ke halaman lain dalam web (Gregorius, 2000:30). Sedangkan menurut arief, Web adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, animasi, video) didalamnya yang menggunakan protokol HTTP (Hypertext Transfer Protocol) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut browser (Arief, 2011:8).

Definisi secara umum, Website adalah kumpulan dari halaman –halaman situs, yang terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam World Wide Web (WWW) di dalam Internet. Sebuah halaman web biasanya

berupa dokumen yang ditulis dalam format HTML (Hyper Text Markup Language), yang selalu bisa diakses melalui HTTP, yaitu sebuah protokol yang menyampaikan informasi dari server website untuk ditampilkan kepada para pemakai melalui web browser. Website dibuat untuk dapat diakses secara luas melalui sebuah aplikasi peramban menggunakan URL (Uniform Resource Locator). Contoh URL yang paling umum adalah <http://www.google.com>. Dalam suatu halaman web, biasanya terdapat berbagai macam jenis informasi dalam bentuk teks, video, gambar, suara, dan lain-lain. Semua data itu disimpan di dalam server hosting. Kepemilikan sebuah website bisa dalam bentuk perseorangan atau organisasi. Bentuk isi informasi yang disajikan juga beragam tergantung pada tujuan website itu dibuat.

1.2 Sejarah Web

Website dikembangkan oleh Sir Timothy John Tim Berners-Lee, dan website yang bisa terhubung dengan jaringan pertama kali dalam sejarah muncul pada tahun 1991. Pada tanggal 30 April 1993, menginformasikan bahwa World Wide Web dapat digunakan secara gratisan oleh semua orang di seluruh dunia. Kini website ditulis, secara dinamik dan dikonversikan menjadi HTML, kemudian diakses melalui program software atau disebut sebagai web browser.

1.3 Jenis-Jenis Web

Berdasarkan penampilan dan respon ketika diakses, website dapat digolongkan ke dalam tiga jenis utama yaitu static website dan dynamic website serta interaktif website.

- Web statis, adalah website yang memiliki halaman tidak berubah-ubah, karena pergantian halaman dilakukan secara manual dengan mengedit kode sebagai struktur dari web.
- Web dinamis, yaitu website yang selalu berubah sehingga harus dilakukan update sesering mungkin. Misalnya web portal berita atau situs portal polling.
- Web interaktif, merupakan website yang saat ini sedang booming di kalangan pengguna internet. Contohnya: blog dan forum.

Sedangkan untuk software editor website terdapat tiga

kategori, yaitu editor teks, WYSIWYG editor dan template editor, dan untuk contoh dari setiap editor diantaranya:

- Elemen 1 editor teks, misalnya Notepad atau TextEdit.
- Elemen 2 WYSIWYG editor. Contohnya: Macromedia Dreamweaver dan Microsoft Frontpage. Situs ini diedit menggunakan GUI (Graphical User Interface) dan format HTML.
- Elemen 3 Editor, yakni situs yang telah memiliki template. Misalnya Rapidweaver dan iWeb, yang memungkinkan editor mengupdate website langsung ke server web, tanpa harus tahu HTML tersebut.

1.4 Manfaat Web

Pada dasarnya, website memiliki berbagai manfaat yang tentunya dapat membantu berbagai masalah pekerjaan yang kita miliki. Beberapa manfaat dari website diantaranya:

1. Memperluas jangkauan promosi bisnis Anda. Dengan memiliki website maka produk atau jasa Anda lebih bisa dikenal oleh masyarakat khususnya pengguna internet.
2. Bisa menjadi media tanpa batas, sebab internet adalah media informasi yang tanpa batas. Dengan memiliki website kita berarti sama saja memiliki banyak karyawan yang mempromosikan produk kita selama 24 jam. Ini berarti website Anda dapat menghasilkan konversi dengan baik.
3. Website juga bisa digunakan sebagai tempat untuk mendapatkan komunitas yang sesuai dengan minat Anda. Melalui website, Anda bisa tergabung misalnya di dalam komunitas pecinta alam, fotografi, programmer, maupun situs bekerja secara freelance.

1.5 Bahasa Pemrograman

Website developer merupakan profesi yang bertugas untuk menciptakan sebuah program dan aplikasi berbasis web dengan menggunakan bahasa pemrograman. Seperti yang kita ketahui saat ini dunia digital sudah sangat berkembang dan peran mereka banyak dibutuhkan untuk membuat halaman web yang dinamis. Seorang website developer yang handal biasanya sudah sangat mahir dengan beragam jenis bahasa pemrograman. Bahasa

Pemrograman Website Developer, diantaranya:

- **Javascript**
Bahasa pemrograman ini sudah ada sejak tahun 1994, dan menjadi salah satu bahasa pemrograman yang cukup populer. Kelebihan Javascript ini tidak membutuhkan resource memory yang besar, serta dapat dipelajari dengan mudah untuk berbagai tujuan. Dengan Javascript ini, website developer mampu mengembangkan website yang menarik dan interaktif. Selain itu, Javascript juga digunakan untuk pengembangan aplikasi mobile, game, dan desktop.
- **PHP**
Bahasa pemrograman PHP disebut sebagai bahasa script server-side karena PHP diproses pada komputer server. Bahasa ini didesain untuk pengembangan sebuah website dan saat ini menjadi bahasa pemrograman yang banyak digunakan. PHP dapat digunakan secara gratis dan bersifat Open Source.
- **Python**
Python adalah bahasa pemrograman interpretatif multiguna dengan sintaks yang sederhana. Bahasa ini banyak digunakan untuk menulis aplikasi website, namun pada beberapa tahun terakhir juga mampu menarik perhatian dalam ruang coding IoT. Perusahaan-perusahaan besar yang menggunakan Python sebagai bahasa pemrogramannya adalah Google, Instagram, Youtube dan masih banyak lagi.
- **Ruby**
Selain Python, Ruby juga menggunakan struktur sintaks yang sederhana. Selain itu, Ruby juga mempunyai kelebihan lain yaitu memiliki exception handling yang baik, bahasa Pemrograman Berorientasi Objek, mengusung konsep single inheritance, serta bersifat Open Source.

1.6 Basis Data

Basis data (bahasa Inggris: database) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri

(query) basis data disebut sistem manajemen basis data (database management system, DBMS). Sistem basis data dipelajari dalam ilmu informasi. Jenis relational database adalah basis data yang organisasinya berdasarkan permodelan data relasional. Hampir semua relational database saat ini menggunakan SQL sebagai bahasa query dan pemeliharaan basis data. Beberapa contoh relational database yang terkenal seperti SQL, PostgreSQL, MongoDB, MariaDB, Oracle Database, SAP HANA, MemSQL, Firebird dan Interbase. Adapun berikut beberapa fungsi database, diantaranya:

- Database berfungsi untuk mengelompokkan data dan mempermudah dalam proses identifikasi data. Database akan menampilkan data sesuai dengan permintaan dari user tentang suatu informasi dengan proses yang cepat dengan bantuan software Database Management Systems (DBMS).
- Software DBMS akan menghindarkan data ganda dan inkonsistensi pada database. Database akan memberikan notif secara langsung jika terjadi duplikasi data. Duplikasi dalam database ini sangat dimungkinkan sekali karena dari banyaknya jumlah data yang diinput.
- Fungsi database yang ketiga adalah memudahkan dalam akses, edit, tambah, delete dan penyimpanan data. Admin akan lebih mudah dalam mengelola semua kegiatan tersebut diatas walaupun jumlah datanya cukup besar.
- Menjaga kualitas database yang masuk agar sesuai dengan saat proses entry data dan setelah entry. Disini peran software sangat penting terutama mengkondisikan agar data tersimpan tetap aman sampai data dibutuhkan.
- Menjadi solusi untuk mengatasi masalah penyimpanan data konvensional yang memerlukan ruang yang besar dan memakan biaya banyak.

1.7 Metode Pengembangan Perangkat Lunak

Ada beberapa model didalam mengembangkan sistem informasi berbasis web, diantaranya adalah Waterfall model, RAD (Rapid Application Development) model, Model V, Prototyping Model, Simple Interaction Design Model, Star Lifecycle model. Penjelasannya sebagai berikut:

- Model Sekuensial Linier atau Waterfall Development Model
Model Sekuensial Linier atau sering disebut Model Pengembangan Air Terjun, merupakan paradigma model pengembangan perangkat lunak paling tua, dan paling banyak dipakai. Model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem pada seluruh tahapan analisis, desain, kode, pengujian, dan pemeliharaan. Tahapan – tahapan Pengembangan Model Sekuensial Linear / Waterfall Development Model :
 1. Rekayasa dan pemodelan sistem/informasi
Langkah pertama dimulai dengan membangun keseluruhan elemen sistem dan memilah bagian-bagian mana yang akan dijadikan bahan pengembangan perangkat lunak, dengan memperhatikan hubungannya dengan Hardware, User, dan Database.
 2. Analisis kebutuhan perangkat lunak
Pada proses ini, dilakukan penganalisaan dan pengumpulan kebutuhan sistem yang meliputi Domain informasi, fungsi yang dibutuhkan untuk kerja/performansi dan antarmuka. Hasil penganalisaan dan pengumpulan tersebut didokumentasikan dan diperlihatkan kembali kepada pelanggan.
 3. Desain
Pada proses Desain, dilakukan penerjemahan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuatnya proses pengkodean (coding). Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi interface, dan detail algoritma prosedural.
 4. Pengkodean
Pengkodean merupakan proses menterjemahkan perancangan desain ke bentuk yang dapat dimengerti oleh mesin, dengan menggunakan bahasa pemrograman.
 5. Pengujian
Setelah Proses Pengkodean selesai, dilanjutkan dengan proses pengujian pada program perangkat lunak, baik Pengujian logika internal, maupun Pengujian eksternal

fungsional untuk memeriksa segala kemungkinan terjadinya kesalahan dan memeriksa apakah hasil dari pengembangan tersebut sesuai dengan hasil yang diinginkan.

6. Pemeliharaan

Proses Pemeliharaan merupakan bagian paling akhir dari siklus pengembangan dan dilakukan setelah perangkat lunak dipergunakan. Kegiatan yang dilakukan pada proses pemeliharaan antara lain :

- a. Corrective Maintenance : yaitu mengoreksi apabila terdapat kesalahan pada perangkat lunak, yang baru terdeteksi pada saat perangkat lunak dipergunakan.
- b. Adaptive Maintenance : yaitu dilakukannya penyesuaian/perubahan sesuai dengan lingkungan yang baru, misalnya hardware, periperal, sistem operasi baru, atau sebagai tuntutan atas perkembangan sistem komputer.
- c. Perfektive Maintenance : Bila perangkat lunak sukses dipergunakan oleh pemakai. Pemeliharaan ditujukan untuk menambah kemampuannya seperti memberikan fungsi-fungsi tambahan, peningkatan kinerja dan sebagainya.

- RAD (Rapid Application Development) Model

Rapid Application Development (RAD) adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek (kira-kira 60 sampai 90 hari). Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier dimana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen. Tahapan – tahapan Proses Pengembangan dalam Model Rapid Application Development (RAD), yaitu:

- a. Bussiness Modeling

Fase ini untuk mencari aliran informasi yang dapat menjawab pertanyaan tentang : Informasi apa yang mengendalikan proses bisnis, Informasi apa yang dimunculkan, Di mana informasi digunakan, Siapa yang memprosesnya.

- b. Data Modeling

Aliran informasi yang didefinisikan sebagai bagian dari fase bussiness modeling disaring ke dalam serangkaian objek data yang dibutuhkan untuk menopang bisnis tersebut. Karakteristik (atribut) masing-masing objek diidentifikasi dan hubungan antar objek-objek tersebut didefinisikan.
- c. Proses Modeling

Aliran informasi yang didefinisikan di dalam fase data modeling ditransformasikan untuk mencapai aliran informasi yang perlu bagi implementasi sebuah fungsi bisnis. Gambaran pemrosesan diciptakan untuk menambah, memodifikasi, menghapus, atau mendapatkan kembali sebuah objek data.
- d. Aplication Generation

Selain menggunakan bahasa pemrograman generasi ketiga, RAD juga memakai komponen program yang telah ada atau menciptakan komponen yang bisa dipakai lagi. Ala-alat bantu bisa dipakai untuk memfasilitasi konstruksi perangkat lunak.
- e. Testing dan Turnover

Karena proses RAD menekankan pada pemakaian kembali, banyak komponen program telah diuji. Hal ini mengurangi keseluruhan waktu pengujian. Tetapi komponen baru harus diuji dan semua interface harus dilatih secara penuh.
- Model V

Model ini merupakan perluasan dari model waterfall. Disebut sebagai perluasan karena tahap-tahapnya mirip dengan yang terdapat dalam model waterfall. Jika dalam model waterfall proses dijalankan secara linear, maka dalam model V proses dilakukan bercabang. Dalam model V ini digambarkan hubungan antara tahap pengembangan software dengan tahap pengujiannya. Tahap-tahapan model V, diantaranya:
- a. Requirement Analysis & Acceptance Testing

Tahap Requirement Analysis sama seperti yang terdapat dalam model waterfall. Keluaran dari tahap ini adalah dokumentasi kebutuhan pengguna.

Acceptance Testing merupakan tahap yang akan mengkaji apakah dokumentasi yang dihasilkan tersebut dapat diterima oleh para pengguna atau tidak.

- b. **System Design & System Testing**
Dalam tahap ini analisis sistem mulai merancang sistem dengan mengacu pada dokumentasi kebutuhan pengguna yang sudah dibuat pada tahap sebelumnya. Keluaran dari tahap ini adalah spesifikasi software yang meliputi organisasi sistem secara umum, struktur data, dan yang lain. Selain itu tahap ini juga menghasilkan contoh tampilan window dan juga dokumentasi teknik yang lain seperti Entity Diagram dan Data Dictionary.
- c. **Architecture Design & Integration Testing**
Sering juga disebut High Level Design. Dasar dari pemilihan arsitektur yang akan digunakan berdasar kepada beberapa hal seperti: pemakaian kembali tiap modul, ketergantungan tabel dalam basis data, hubungan antar interface, detail teknologi yang dipakai.
- d. **Module Design & Unit Testing**
Sering juga disebut sebagai Low Level Design. Perancangan dipecah menjadi modul-modul yang lebih kecil. Setiap modul tersebut diberi penjelasan yang cukup untuk memudahkan programmer melakukan coding. Tahap ini menghasilkan spesifikasi program seperti: fungsi dan logika tiap modul, pesan kesalahan, proses input-output untuk tiap modul, dan lain-lain.
- e. **Coding**
Dalam tahap ini dilakukan pemrograman terhadap setiap modul yang sudah dibentuk.
- **Prototyping Model**
Metode Prototype merupakan suatu paradigma baru dalam metode pengembangan perangkat lunak dimana metode ini tidak hanya sekedar evolusi dalam dunia pengembangan perangkat lunak, tetapi juga merevolusi metode pengembangan perangkat lunak yang lama yaitu sistem sekuensial yang biasa dikenal dengan nama SDLC atau waterfall development model. Dalam Model Prototype, prototype dari perangkat lunak yang dihasilkan kemudian

dipresentasikan kepada pelanggan, dan pelanggan tersebut diberikan kesempatan untuk memberikan masukan sehingga perangkat lunak yang dihasilkan nantinya betul-betul sesuai dengan keinginan dan kebutuhan pelanggan. Tahapan-tahapan Proses Pengembangan dalam Model Prototype, yaitu :

- a. Pengumpulan kebutuhan
Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.
- b. Membangun prototyping
Membangun prototyping dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat input dan format output).
- c. Evaluasi protoptyping
Evaluasi ini dilakukan oleh pelanggan, apakah prototyping yang sudah dibangun sudah sesuai dengan keinginan pelanggan atau belum. Jika sudah sesuai, maka langkah selanjutnya akan diambil. Namun jika tidak, prototyping direvisi dengan mengulang langkah-langkah sebelumnya.
- d. Mengkodekan sistem
Dalam tahap ini prototyping yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.
- e. Menguji sistem
Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, kemudian dilakukan proses Pengujian. Pengujian ini dilakukan dengan White Box, Black Box, Basis Path, pengujian arsitektur, dan lain lain.
- f. Evaluasi Sistem
Pelanggan mengevaluasi apakah perangkat lunak yang sudah jadi sudah sesuai dengan yang diharapkan. Jika ya, maka proses akan dilanjutkan ke tahap selanjutnya, namun jika perangkat lunak yang sudah jadi tidak/belum sesuai dengan apa yang diharapkan, maka tahapan sebelumnya akan diulang.

g. Menggunakan sistem

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

Model Prototyping ini sangat sesuai diterapkan untuk kondisi yang beresiko tinggi di mana masalah-masalah tidak terstruktur dengan baik, terdapat fluktuasi kebutuhan pemakai yang berubah dari waktu ke waktu atau yang tidak terduga, bila interaksi dengan pemakai menjadi syarat mutlak dan waktu yang tersedia sangat terbatas sehingga butuh penyelesaian yang segera. Model ini juga dapat berjalan dengan maksimal pada situasi di mana sistem yang diharapkan adalah yang inovatif dan mutakhir sementara tahap penggunaan sistemnya relatif singkat.

- Spiral Model

Model spiral (spiral model) yang pada awalnya diusulkan oleh Boehm adalah model proses perangkat lunak yang evolusioner yang merangkai sifat iteratif dari prototipe dengan cara kontrol dan aspek sistematis dari model sekuensial linier. Di dalam model spiral, perangkat lunak dikembangkan di dalam suatu deretan pertambahan. Selama awal iterasi, rilis inkremental bisa merupakan sebuah model atau prototipe kertas. Selama iterasi berikutnya, sedikit demi sedikit dihasilkan versi sistem rekayasa yang lebih lengkap.

- Simple Interaction Desain Model

Pada model rancangan interaksi sederhana ini input atau masukan hanya memiliki satu titik. yang mana masukan tersebut diidentifikasi apakah sesuai dengan kebutuhan, lalu didesain sesuai dengan persyaratan yang telah ditetapkan. Setelah didesain rancangan tersebut dibangun dan harus interaktif. Setelah itu barulah rancangan tersebut dievaluasi. Evaluasi dapat dilakukan dimana saja, rancangan yang telah dievaluasi dapat kembali didesain ulang atau apakah rancangan tersebut tidak sesuai dengan kebutuhan user, maka alur tersebut akan terus berputar hingga pada tahap evaluasi tidak lagi terjadi kesalahan, baik dalam penetapan kebutuhan user maupun pendesainannya, sehingga pada tahap evaluasi terciptalah sebuah hasil akhir yang valid.

- **Star Lifecycle Model**

Dalam Siklus permodelan ini pengujian dilakukan terus menerus, tidak harus dikahir. Misalnya dimulai dari menentukan kosep desain (conceptual design) dalam proses ini akan langsung terjadi evaluasi untuk langsung ternilai apakah sudah sesuai dengan kebutuhan user, bila belum maka akan terus berulang di evaluasi hingga benar-benar pas, selanjutnya apabila sudah pas, maka dari tahap evaluasi yang pertama akan lanjut ke proses yg selanjutnya yakni requirements atau specification yakni memverifikasikan persyaratan rancangan tersebut, dan pada tahap itu juga langsung terjadi pengevaluasian seperti tahap pertama, dan selanjutnya akan tetap sama terjadi pada tahapan-tahapan selanjutnya yakni task analysis/fungsion analysis, pengimplementasian, prototyping hingga pada akhirnya terciptalah sebuah aplikasi yang sesuai dengan kebutuhan user. Intinya pada rancangan model ini pengevaluasian dilakukan disetiap tahapan tidak hanya pada tahapan akhir seperti model-model rancangan yang lainnya.

1.8 Struktur Navigasi

Struktur Navigasi merupakan struktur atau alur dari suatu program yang merupakan rancangan hubungan (rantai kerja) dari beberapa area yang berbeda dan dapat membantu mengorganisasikan seluruh elemen pembuatan Website. Menentukan struktur navigasi merupakan hal yang sebaiknya dilakukan sebelum membuat suatu aplikasi. Ada 4 (empat) macam bentuk dasar dari struktur navigasi yang biasa digunakan dalam proses pembuatan aplikasi, diantaranya:

1. **Struktur Navigasi Linier**

Struktur navigasi linier hanya mempunyai satu rangkaian cerita yang berurut, yang menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada sruktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya.

2. **Struktur Navigasi Non-Linier**

Struktur navigasi non-linier atau struktur tidak berurut merupakan pengembangan dari struktur navigasi linier. Pada struktur ini diperkenankan membuat navigasi bercabang. Percabangan yang dibuat pada struktur nonlinier ini berbeda dengan percabangan pada struktur hirarki, karena pada percabangan nonlinier ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada Master Page dan Slave Page.

3. Struktur Navigasi Hirarki

Struktur navigasi hirarki biasa disebut struktur bercabang, merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama akan disebut sebagai Master Page (halaman utama pertama), halaman utama ini mempunyai halaman percabangan yang disebut Slave Page (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama Master Page (halaman utama kedua), dan seterusnya. Pada struktur navigasi ini tidak diperkenankan adanya tampilan secara linier.

4. Struktur Navigasi Campuran

Struktur navigasi campuran merupakan gabungan dari ketiga struktur sebelumnya yaitu linier, non-linier dan hirarki. Struktur navigasi ini juga biasa disebut dengan struktur navigasi bebas. Struktur navigasi ini banyak digunakan dalam pembuatan website karena struktur ini dapat digunakan dalam pembuatan website sehingga dapat memberikan ke-interaksian yang lebih tinggi.

1.9 Entity Relationship Diagram

ERD atau Entity Relationship Diagram adalah suatu bentuk diagram yang menjelaskan hubungan antar objek-objek data yang mempunyai hubungan antar relasi. ERD digunakan untuk menyusun struktur data dan hubungan antar data, dan untuk menggambarannya digunakan notasi, simbol, bagan, dan lain sebagainya. ERD atau ERM berfungsi sebagai bentuk pemodelan basis data yang kemudian dapat dikembangkan dalam berbagai project sistem informasi, contohnya rekayasa software. Dalam

membentuk suatu rekayasa software, kita memerlukan sebuah konsep. Konsep yang akan dipakai dalam project ini berbentuk ERD atau juga disebut ERM (entity relationship model). Jadi, kita dapat memanfaatkan ERD sebagai kerangka atau konsep dasar dalam suatu sistem informasi. Tools-tools yang digunakan oleh Entity Relation Diagram (ERD), diantaranya:

1. Dbdiagram
Dbdiagram adalah tools untuk mendesain database dengan cepat. Dalam pemrogramannya, dbdiagram memiliki bahasa yang mudah dipahami dan eksklusif untuk penggambaran ERD. Dbdiagram menyediakan penyimpanan online dan dapat dengan mudah dishare. Terlebihnya lagi, dbdiagram tidak berbayar untuk menikmati fitur-fiturnya.
2. Draw.io
Bagi pengguna casual, draw.io memberikan tools ERD yang cukup baik. Tidak hanya tools ERD, draw.io jug berfungsi sebagai flowchart maker atau algoritma. Draw.io sudah mendukung penyimpanan berbasis cloud seperti drive, oneplus, maupun non-cloud.
3. Lucidchart
Lucidchart dipasarkan lebih kepada designer professional. Lucidchart menawarkan tools dengan interface dan penampilan yang sangat baik. Meski berbayar, Lucidchart juga menyediakan free version yang cukup lengkap. Lucidchart mendukung penyimpanan cloud maupun storage. Tools ini sangat cocok bagi Anda yang merupakan designer ERD yang berpengalaman.
4. SQLDBM
SQLDBM merupakan tools ERD terbaik bagi anda yang menjalankan database berbasis MySQL atau MS SQL. SQLDBM dapat bekerja dimana saja pada browser apapun tanpa memerlukan database engine tambahan. Selebihnya SQLDBM bersifat gratis dan cocok bagi Anda yang menjalankan database menggunakan SQL.
5. QuickDBD
QuickDBD adalah tools ERD yang berbasis teks, dan dengan cepat menggambar diagram yang diperlukan. Meski berbayar, QuickDBD juga menyediakan free version

dimana kita dapat mencoba terlebih dahulu. QuickDBD juga menyediakan export ke berbagai format seperti SQL, PDF, maupun word. QuickDBD cocok bagi Anda yang menyukai ERD Text based dan cepat dalam mendesain.



Rangkuman

Website adalah kumpulan dari halaman – halaman situs, yang terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam World Wide Web (WWW) di dalam Internet. Sebuah halaman web biasanya berupa dokumen yang ditulis dalam format HTML (Hyper Text Markup Language), yang selalu bisa diakses melalui HTTP, yaitu sebuah protokol yang menyampaikan informasi dari server website untuk ditampilkan kepada para pemakai melalui web browser. Berdasarkan penampilan dan respon ketika diakses, website dapat digolongkan ke dalam tiga jenis utama yaitu static website dan dynamic website serta interaktif website. Sedangkan untuk software editor website terdapat tiga kategori, yaitu editor teks, WYSIWYG editor dan template editor. Pada dasarnya, website memiliki berbagai manfaat yang tentunya dapat membantu berbagai masalah pekerjaan yang kita miliki.

Seorang website developer yang handal biasanya sudah sangat mahir dengan beragam jenis bahasa pemrograman. Dan untuk menyimpan data yang ada di website dapat disimpan pada database. Basis data (bahasa Inggris: database) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (query) basis data disebut sistem manajemen basis data (database management system, DBMS). Beberapa contoh relational database yang terkenal seperti SQL, PostgreSQL, MongoDB, MariaDB, Oracle Database, SAP HANA, MemSQL, Firebird dan Interbase. ERD atau Entity Relationship Diagram adalah suatu bentuk diagram yang menjelaskan hubungan antar objek-objek data yang mempunyai hubungan antar relasi. Ada beberapa model didalam

mengembangkan sistem informasi berbasis web, diantaranya adalah Waterfall model, RAD (Rapid Application Development) model, Model V, Prototyping Model, Simple Interaction Design Model, Star Lifecycle model. Struktur Navigasi merupakan struktur atau alur dari suatu program yang merupakan rancangan hubungan (rantai kerja) dari beberapa area yang berbeda dan dapat membantu mengorganisasikan seluruh elemen pembuatan Website.



Latihan Soal

1. Jelaskan pengertian dari Website !
2. Apa sebutan halaman Web yang di tempatkan di depan situs web!
3. Software yg biasa digunakan untuk encoding adalah!
4. Apa format dokumen sebuah halaman web!
5. Apa kepanjangan dari HTML!
6. Untuk dapat diakses oleh program Web Server, ekstensi apa yang harus dipakai!
7. Jelaskan pengertian dari web designer, web Developer, web programmer dan web master!

BAB II

HTML

2.1 Pengertian HTML

HTML adalah singkatan dari Hypertext Markup Language. HTML memungkinkan seorang user untuk membuat dan menyusun bagian paragraf, heading, link atau tautan, dan blockquote untuk halaman web dan aplikasi. HTML adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web. HTML inilah yang menyusun sebuah halaman web menjadi sebagaimana yang kita lihat melalui browser (penjelajah internet). Jadi, belajar HTML dan mengerti bagaimana cara HTML bekerja akan sangat penting jika kamu ingin terjun ke dunia web development. Sebenarnya ketika bekerja dalam menampilkan dan menyusun sebuah halaman web si HTML ini nggak sendiri, dia juga dibantu oleh teman-temannya yang lain yaitu si CSS dan si JavaScript. HTML sendiri bertugas menyusun kerangka halaman web, CSS yang akan merapikan dan memperindah tampilan halaman web, sedangkan JavaScript-lah yang bertugas membuat halaman web menjadi lebih interaktif. Perlu diperhatikan, HTML bukanlah sebuah bahasa pemrograman. HTML merupakan bahasa markup yang berisi perintah-perintah dengan format tertentu yang terstruktur untuk menampilkan tampilan tertentu.

Mendesain HTML berarti melakukan suatu tindakan pemrograman. Namun HTML bukanlah sebuah bahasa pemrograman. Namun HTML hanyalah berisi perintah-perintah yang telah terstruktur berupa tag-tag penyusun. Menuliskan tag-tag HTML tidaklah sebatas hanya memasukkan perintah-perintah tertentu agar HTML kita dapat di akses oleh browser. Mendesain HTML adalah adalah sebuah seni tersendiri. Homepage yang merupakan implementasi dari HTML adalah refleksi dari orang yang membuatnya. Untuk itu kita perlu mendesainnya dengan baik agar para pengunjung homepage yang kita buat merasa senang dan bermanfaat.

Untuk dapat membuat sebuah halaman web dan dapat ditampilkan oleh browser sebagaimana yang kita inginkan tentu saja terdapat aturan dan formatnya. Untuk menulis kode HTML kita bisa menggunakan software code editor seperti Notepad, Notepad++, Sublime Text, Atom, atau yang lainnya. Dan agar kode kita mampu dibaca oleh browser dengan baik kita harus menyimpannya dalam format HTML yaitu .htm atau .html. HTML Editor merupakan sebuah program yang khusus didesain untuk membuat, melakukan editing bahkan mem-publish ke internet. Dengan kemampuannya menggabungkan kemudahan dan kecanggihan teknologi internet ke dalam dokumen HTML maka cara ini sangat disukai oleh para pemula dan desainer yang tidak ingin belajar lebih mendalam mengenai HTML. Menuliskan secara manual satu persatu tag-tag HTML, sangat disarankan sulit dikarenakan akan memakan tenaga dan waktu ekstra untuk melakukannya, ditambah lagi Anda harus melakukan cara-cara konvensional untuk melihat hasilnya pada web browser. Namun pada cara kedua adalah dasar dari segala bentuk HTML yang akan Anda pelajari, karena dengan cara itulah Anda akan lebih paham mengenai cara kerja dan berbagai perintah yang biasa dipakai pada bahasa HTML

2.2 Sejarah HTML

Sebelum suatu HTML disahkan sebagai suatu dokumen HTML standar, harus disetujui oleh W3C untuk dievaluasi secara ketat. Dengan demikian, setiap terjadi perkembangan level, HTML memiliki suatu kelebihan yang baru dalam hal penampilannya ketimbang versi-versi sebelumnya. Jika ada bug pada perintah-perintah tertentu dapat dihindari. Setiap terjadi perkembangan suatu versi HTML, maka mau tak mau browser pun harus memperbaiki diri agar bisa mendukung kode-kode HTML yang baru tersebut. HTML itu sendiri awalnya dari bahasa SGML (Standard Generalized Markup Language), yang sudah digunakan sejak dulu untuk memformat dokumen agar bersifat portabel. Tata cara penulisan HTML merupakan penyederhanaan SGML.

Sejarah HTML atau Hypertext Markup Language telah menjadi teknologi utama untuk Web sejak awal tahun 1990-an.

Tim Berners-Lee menciptakan HTML pada tahun 1989 sebagai cara sederhana namun efektif untuk mengkodekan dokumen elektronik. Bahkan, tujuan awal dari web browser adalah untuk melayani pembaca untuk membuka dokumen berformat HTML. Didalam HTML telah mengalami berbagai perkembangan dari tiap-tiap versinya, diantaranya:

1. HTML Versi 1.0

HTML Versi 1.0 merupakan pionir yang di dalamnya masih terdapat banyak sekali kelemahan hingga wajar jika tampilan yang dihasilkan sangat sederhana. Kemampuan yang dimiliki versi 1.0 ini antara lain heading, paragraf, hypertext, list, serta cetak tebal dan miring pada teks. Versi ini juga mendukung peletakan image pada dokumennya tanpa memperbolehkan teks di sekelilingnya (wrapping).

2. HTML Versi 2.0

Pada HTML Versi ini, penambahan kualitas HTML terletak pada kemampuannya untuk menampilkan suatu form pada dokumen. Dengan adanya form ini, kita dapat memasukkan nama, alamat, serta saran dan kritik. HTML versi 2.0 ini merupakan pionir dari adanya web interaktif.

3. HTML Versi 3.0

Versi HTML 3.0 menambahkan beberapa fasilitas baru seperti FIGURE yang merupakan perkembangan dari IMAGE untuk meletakkan gambar dan tabel. Selain itu, HTML ini juga mendukung adanya rumus-rumus matematika dalam dokumennya. Versi ini yang disebut HTML+- tidak bertahan lama dan segera digantikan dengan versi 3.2.

4. HTML Versi 3.2

HTML versi ini merupakan HTML yang sering digunakan. Di dalamnya terdapat suatu teknologi untuk meletakkan teks di sekeliling gambar, gambar sebagai latar belakang, tabel, frame, style sheet dan lain-lain. Selain itu pada HTML versi ini Kita bisa menggunakan script di luar HTML untuk mendukung kinerja HTML kita tersebut, seperti Javascript, VBScript dan lain-lain.

5. HTML Versi 4

HTML ini memuat banyak sekali perubahan dan revisi dari pendahulunya yaitu HTML 3.2. Perubahan ini hampir terjadi

di segala perintah HTML seperti tabel, image, link, text, meta, imagemaps, form dan lain-lain.

6. HTML Versi 4.01

HTML versi 4.01 merupakan revisi dari HTML 4.0. Versi terbaru ini memperbaiki kesalahan-kesalahan kecil (minor errors) pada versi terdahulunya. HTML 4.01 ini juga menjadi standarisasi untuk elemen dan atribut dari script XHTML 1.0.

7. HTML Versi 5.0

Teknologi ini mulai diluncurkan pada tahun 2009, tetapi pada tanggal 4 Maret 2010 terdapat sebuah informasi bahwa W3C (World Wide Web Consortium) dan IETF (Internet Engineering Task Force) yaitu sebuah organisasi yang menangani HTML sejak versi 2.0 telah mengembangkan versi HTML terbaru, yaitu versi 5.0. HTML 5 adalah sebuah prosedur pembuatan tampilan web baru yang merupakan penggabungan antara CSS, HTML itu sendiri dengan JavaScript. Beberapa kelebihan yang dimiliki oleh HTML5, diantaranya:

a. Cleaner code

Karena sebagian besar kode telah termasuk di dalam sintaks HTML5, maka kode nampak terlihat lebih sederhana daripada penggabungan antara HTML, CSS dan Java Script.

b. Greater consistency

HTML5 telah melakukan banyak sekali penambahan sintaks yang dibuat dalam struktur lebih baik dan lebih sederhana daripada sintaks-sintaks sebelumnya. Hal ini membuat developer terbantu dalam meningkatkan konsistensi dalam membangun sebuah web.

c. Improve Semantics

Berbagai elemen kode di dalam HTML5 yang telah distandarisasi, maka nilai semantik dari sebuah web dapat lebih ditingkatkan. Itu berarti bahwa bagian-bagian dari web seperti header, nav, footer dan beberapa bagian lainnya terdefinisi dengan jelas maksud serta tujuannya. Selain itu juga terbentuk dalam sebuah “machine readable format”.

d. Improved Accessibility

Teknologi HTML5 yang memudahkan struktur

pembangunan sebuah web, maka developer dapat membangun pemahaman yang lebih detil mengenai halaman web.

e. Client-side Database

HTML5 menyediakan model database SQL yang baru dengan API yang dapat dibangun dalam konsep lokal, dalam hal ini di sisi client.

f. Geolocation

HTML5 mempunyai API yang terintegrasi terhadap geolocation, fasilitas tersebut dapat diakses melalui GPS atau fasilitas lain seperti Google Latitude pada iphone.

g. Offline Application Cache

Pengguna dapat terus melakukan interaksi dengan aplikasi meskipun mereka terputus dari jaringan internet.

h. Smarter Forms

Terdapat semacam regular expression (regex) yang membuat form mampu mengenali secara lebih baik tentang input, validasi data dan interaksi dengan elemen lain (misal : format email, password dan lain lain).

i. Sharper focus on Web Application Requirments

HTML5 membuat sebuah mekanisme yang lebih mudah dalam hal pembuatan front end, aplikasi chat, tools drag and drop, video player, pengolah grafis dan masih banyak lagi. (*).

2.3 Struktur Dasar dan Ekstension File HTML

Dalam dunia website, terdapat beragam aturan penyimpanan file yang wajib ditaati. Kalau tidak, mungkin website tidak dapat diakses secara maksimal. Suatu website terdiri atas banyak file, seperti konten teks, kode, stylesheet, konten media, dan lain-lain. Aturan penulisan HTML dapat dilihat pada Gambar 1. Pada Gambar 1 terdapat beberapa tag yang harus ada pada HTML, diantaranya:

1. html

<html> : Tag untuk memulai sebuah HTML

</html> : Tag untuk menutup sebuah HTML

2. head

<head> : Tag untuk membuat bagian header

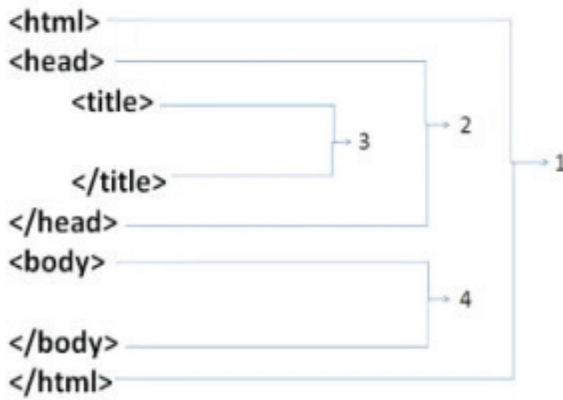
- `</head>` : Tag untuk penutup header
3. title

`<title>` : Tag untuk memberi judul, yang nantinya akan muncul di TabBrowser

`</title>` : Tag untuk menutup judul
 4. body

`<body>` : Tag pembuka bagian Body atau Isi dalam HTML

`</body>` : Tag Penutup body



Gambar 1. Aturan Dasar Penulisan HTML

Sedangkan Komentar di HTML juga sangat diperlukan. Fungsi dari komentar adalah kita dapat meninggalkan pesan pengingat atau penjelasan tag-tag untuk diri Anda sendiri dan juga penulis kode lainnya yang akan mengerjakan halaman HTML. komentar juga dapat digunakan untuk menonaktifkan satu bagian kode secara cepat ketika Anda sedang melakukan pengujian atau mengerjakan fitur baru yang belum siap. Format penulisan komentar, dapat dilihat pada Gambar 2.

`<!-- isi komentar -->`

Gambar 2. Format Penulisan Komentar

Ekstension File adalah segalanya yang mengikuti akhir nama dokumen yang menjadi indikasi dari software yang digunakan untuk membuat file. Ekstension File terdapat huruf terakhir sesudah titik, yang menunjukkan jenis file. Ekstension

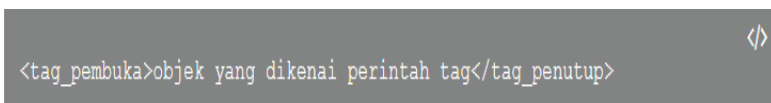
File berfungsi untuk mengetahui atau membedakan jenis file. Ekstensi pada file HTML adalah *.html atau *.htm. Selama bekerja dengan komputer, tentang ekstensi file benar-benar harus ditaati secara penuh. Di dunia website, mengenal banyak sekali ekstensi file. Beberapa ekstensi file yang paling populer, diantaranya:

1. *.htm dan *.html adalah ekstensi standar sebuah website, apabila website anda tidak mengandung teknologi berbasis server, misalnya untuk menyimpan data atau lainnya, maka pakailah format ini. Sampai saat ini, setiap browser dapat membaca file berekstensi .htm dan .html secara penuh.
2. *.shtm dan *.shtml adalah huruf 's' kependekan dari secure. Format ini hanya ideal apabila halaman web itu dipakai untuk mengirim data-data yang cukup vital sehingga harus melewati proses enkripsi (SSL). Meskipun pakai dalam kondisi normal, tapi akses atau loading menuju file .shtml ini jauh lebih lambat dibanding html itu sendiri.
3. *.css adalah format script desain web yang mampu mengatasi kelemahan-kelemahan yang ada di dunia hmtl. Apabila script CSS itu diletakkan langsung didalam script HTML maka tidak perlu menggunakan ekstensi ini, cukup menggunakan ekstensi HTML saja.
4. *.asp adalah scrpt server side yang dikembangkan oleh microsoft.
5. *.php adalah cukup populer dan banyak digunakan, PHP menuntut adanya server khusus yang mampu menangani dirinya.
6. *.cfm adalah ekstensi resmi dari script pemrograman ColdFusion. Dibandingkan script pemrograman lainnya, ColdFusion memang relatif mudah dipelajari.

2.4 Pengertian Element, Tag, dan Attribut dan Value

Sebagai sebuah bahasa markup, HTML membutuhkan cara untuk memberitahu web browser untuk apa fungsi sebuah text. Apakah text itu ditulis sebagai sebuah paragraf, list, atau sebagai link. Dalam HTML, tanda ini dikenal dengan istilah tag. Hampir semua tag di dalam HTML ditulis secara berpasangan, yakni tag pembuka dan tag penutup, dimana objek yang dikenai perintah tag berada di antara tag pembuka dan tag penutup. Objek disini

dapat berupa text, gambar, maupun video. Penulisan tag berada di antara dua kurung siku: “<” dan “>”. Format dasar penulisan tag HTML dapat dilihat pada Gambar 3. TAG biasanya dituliskan berpasangan, yang maksudnya adalah memberitahukan kepada browser dimana posisi awal dan posisi akhir sebuah format dokumen.



```
<tag_pembuka>objek yang dikenai perintah tag</tag_penutup>
```

Gambar 3. Format Dasar Tag HTML

Element adalah isi dari tag yang berada diantara tag pembuka dan tag penutup, termasuk tag itu sendiri dan atribut yang dimilikinya (jika ada). Sebagai contoh perhatikan kode HTML pada Gambar 4. Pada contoh diatas, “<p> Ini adalah sebuah paragraf </p>” merupakan element p.



```
<p> Ini adalah sebuah paragraf </p>
```

Gambar 4. Contoh Element

Atribut adalah informasi tambahan yang diber’ikan kepada tag. Informasi ini bisa berupa instruksi untuk warna dari text, besar huruf dari text, dll. Setiap atribut memiliki pasangan nama dan nilai (value) yang ditulis dengan name=”value”. Value diapit tanda kutip, boleh menggunakan tanda kutip satu (’) atau dua (”). Sebagai contoh atribut menggunakan href, dapat dilihat pada Gambar 5. href merupakan nama dari atribut, dan <https://www.trunojoyo.ac.id> adalah value atau nilai dari atribut tersebut. HTML memiliki banyak atribut yang beberapa diantaranya hanya cocok untuk tag tertentu saja. Sebagai contoh, atribut “href” diatas hanya digunakan untuk tag <a> saja (dan beberapa tag lain). Penjelasan tentang tujuan dan pengertian dari atribut seperti href ini akan kita bahas pada tutorial-tutorial selanjutnya.



```
<a href="http://trunojoyo.ac.id"> Ini adalah kampusku </a>
```

Gambar 5. Contoh Atribut

Value adalah nilai yang diberikan pada sebuah atribut. Value ada beberapa jenis, bisa value text seperti left, right, center, red, blue, bisa juga angka 1, 2, 3, bisa juga bentuk pixel: 100, 200, atau bentuk persen 100%, 30%, ada juga yang bentuk hexa seperti warna yang ditulis `color="#000"` dan lain-lain. Untuk value juga harus relevan, tidak bisa sembarangan. Contoh: align (ini untuk mengatur posisi) tidak boleh ditulis `align="1"` atau sebaliknya `size="center"`.

2.5 Pengaturan Halaman HTML

1. Mengatur Tulisan pada Halaman HTML

Agar tulisan pada halaman sebuah web terlihat rapi, maka perlu pemformatan dalam tulisan tersebut. Baik tata cara pencetakannya maupun pengaturan tulisan yang lain. Adapun pengaturan format tersebut antara lain :

a. Heading

Heading adalah hal yang penting dalam sebuah Document HTML. Heading didefinisikan dengan tag `<h1>` sampai dengan `<h6>`. Semakin besar angka dalam penulisan tag Heading semakin kecil ukuran yang tercetak dalam Browser. Contoh :

b. Paragraf

Membuat paragraf dalam HTML dapat menggunakannya dengan tag `<p>`. Pada browser, secara otomatis penulisan paragraf akan menambahkan baris kosong sebelum dan sesudah paragraf. Sebagian besar browser akan membenarkan penulisan HTML Paragraf, bahkan jika kita melupakan tag tersebut. Tetapi, tidak menuliskan tag dalam penulisan paragraf membuat hasil tulisan yang tercetak pada HTML tidak tertulis rapi. Dalam penulisan paragraf, untuk mengganti baris dalam paragraf tersebut dapat kita gunakan tag break, dapat di tulis `
` atau `</br>`. Contoh:

```

<html>
<body>
    <p>Penulisan Paragraf 1. </p>
    <p>Penulisan Paragraf 2. </p>
    <p>Penulisan <br \> Paragraf 3. </p>
</body>
</html>

```

c. Format Teks

Banyak sekali tag yang digunakan untuk menformat sebuah tulisan dalam HTML. Menformat paragraf yang di maksudkan adalah membuat text bercetak tebal, bercetak miring, subscript text, superscript text dan masih banyak format yang lainnya. Untuk lebih jelasnya lihat tabel 1, 2 & 3.

Tabel 1. HTML tags Formating Text

Tags	Description
	Mendefinisikan Teks bercetak tebal
<i>	Mendefinisikan Teks bercetak miring
<small>	Mendefinisikan Teks kecil
	Mendefinisikan Teks penting

<sub>	Mendefinisikan Teks Subscript
<sup>	Mendefinisikan Teks Superscript
<ins>	Mendefinisikan Teks yang dimasukan
	Mendefinisikan Teks yang dihapus

Tabel 2. HTML Computer Output Tags

Tags	Descriptions
<code>	Mendefinisikan Kode Komputer
<kbd>	Mendefinisikan tekx Keyboard
<samp>	Mendefinisikan Sampel kode komputer
<var>	Mendefinisikan Variabel
<pre>	Mendefinisikan teks terformat

Tabel 3. HTML Citations, Quotations, dan Definition Tags

Tags	Descriptions
<abbr>	Mendefinisikan sebuah singkatan atau akronim

<code><address></code>	Mendefinisikan informasi kontak untuk penulis / pemilik document
<code><bdo></code>	Mendefinisikan arah teks
<code><blockquote></code>	Mendefinisikan bagian yang dikutip dari sumber lain
<code><q></code>	Mendefinisikan sebuah inline (pendek) kutipan
<code><cite></code>	Mendefinisikan Judul karya
<code><dfn></code>	Mendefinisikan sebuah istilah definisi

d. Mengatur Pemerataan Paragraf

Pemerataan tulisan sangat mempengaruhi tampilan dari halaman web anda, ada kalanya tulisan tertentu perlu ditampilkan pada tengah halaman, atau pun justified. Tag-tag HTML yang mengatur pemerataan tulisan pada satu halaman:

· `<CENTER> </CENTER>`

Tag ini akan menampilkan tulisan/gambar ditengah baris.

· `<P ALIGN=LEFT|CENTER|RIGHT> </P>`

Tag paragraf digunakan untuk memulai suatu paragraf baru disertai dengan pemerataan teks ke kiri, tengah, atau kanan dari tampilan browser untuk paragraf bersangkutan. Suatu paragraf dapat terdiri dari teks, gambar, hyperlink, dan element HTML lainnya.

· `<H# ALIGN=LEFT|CENTER|RIGHT> </H#>`

Ini akan merapikan tulisan ke kiri, tengah atau kanan dari tulisan yang diberi tingkat heading.

e. Menyisipkan Gambar

```
<html>
<body>
    
</body>
</html>
```

Dalam HTML, gambar didefinisikan dengan tag ``. Tag `` kosong, yang berarti mengandung atribut saja, dan tidak memiliki tag penutup. Untuk menampilkan gambar pada halaman, Anda perlu menggunakan atribut `src`. `Src` singkatan dari “sumber”. Nilai dari atribut `src` adalah URL gambar yang ingin ditampilkan. Sintaks untuk mendefinisikan gambar:

```

```

URL menunjuk ke lokasi di mana gambar disimpan. Sebuah gambar bernama “image.jpg”, yang terletak luar direktori dari file “image.html”.

Browser menampilkan gambar di mana tag `` terjadi dalam dokumen. Jika Anda menempatkan tag gambar antara dua paragraf, browser menunjukkan paragraf pertama, maka gambar, dan kemudian paragraf kedua.

f. Link ke Dokumen atau Situs yang Lain

HTML tag `<a>` mendefinisikan hyperlink. Sebuah hyperlink (atau link) adalah kata, kelompok kata, atau gambar yang dapat Anda klik untuk melompat ke dokumen lain. Bila Anda memindahkan kursor di atas link dalam sebuah halaman Web, panah akan berubah menjadi sedikit tangan. Atribut yang paling penting dari elemen `<a>` adalah atribut `href`, yang

menunjukkan tujuan link. Secara default, link akan muncul sebagai berikut di semua browser:

- Link belum dikunjungi digarisbawahi dan biru
- Sebuah link dikunjungi digarisbawahi dan ungu
- Link aktif digarisbawahi dan merah

Contoh Penulisan tags Link :

```
<a href="http://www.trunojoyo.ac.id/">Ini nama  
Kampusku</a>
```

2. Membuat List

Agar dokumen kita mudah dibaca dan jelas, ada bagian tertentu yang perlu ditampilkan dalam list. List yang paling umum di Gunakan adalah Unordered List dan Ordered List :

a. Unordered List

Unordered list ditandai dengan tag `` dan setiap daftar item ditandai dengan ``. Hasil dari Unordered List adalah lingkaran bulat kecil.

Contoh :

```
<html>  
<body>  
    <ul>  
        <li>Unordered List 1</li>  
        <li>Unordered List 2</li>  
        <li>Unordered List 3</li>  
    </ul>  
</body>  
</html>
```

b. *Ordered List*

Ordered list dimulai dengan tag `` dan setiap item daftar dimulai dengan tag ``. Daftar item ditandai dengan angka.

Contoh :

```
<html>
<body>
    <ol>
        <li>Ordered List 1</li>
        <li>Ordered List 2</li>
        <li>Ordered List 3</li>
    </ol>
</body>
</html>
```

3. Membuat Tabel dan Pengaturannya

Tabel didefinisikan dengan tag `<table>`. Sebuah tabel dibagi menjadi baris (dengan tag `<tr>`), dan setiap baris dibagi menjadi beberapa kolom (dengan tag `<td>`). Di dalam tag `<table>` harus terdapat atribut `"border"` dan tebal dari border tersebut. Jika Anda tidak menentukan atribut border, tabel akan ditampilkan tanpa batas atau garis. Untuk tag tabel dapat dilihat pada Tabel 4.

Contoh:

```

<table border="1">
    <tr>
        <td>Baris 1, Kolom 1</td>
        <td>Baris 1, Kolom 2</td>
    </tr>
    <tr>
        <td>Baris 2, Kolom 1</td>
        <td>Baris 2, Kolom 2</td>
    </tr>
</table>

```

Tabel 4. Tag pada Tabel

Tags	Descriptions
<table>	Mendefinisikan Tabel
<th>	Mendefinisikan sel header Tabel
<tr>	Mendefinisikan Tabel baris
<td>	Mendefinisikan Tabel Kolom
<caption>	Mendefinisikan Judul Tabel
<colgroup>	Menentukan sekelompok dari satu atau lebih kolom dalam sebuah tabel untuk memformat

<col>	Menentukan sifat kolom untuk setiap kolom dalam elemen <colgroup>
<thead>	Grup isi header dalam sebuah tabel
<tbody>	isi tubuh dalam sebuah tabel
<tfoot>	isi footer dalam sebuah tabel

4. Membuat Form dan Pengaturannya
- Form digunakan untuk membuat sebuah inputan untuk pengguna yang nantinya akan dikirim ke server. Bentuk form terdiri dari beberapa elemen diantaranya teks, checkbox, radio button, tombol submit dan banyak lagi. Form juga dapat berisi daftar pilih, textarea, fieldset, legenda, dan elemen label. Untuk lebih jelasnya lihat Tabel 5 dan 6.

Tabel 5. Element pada Form

Element	Descriptions
text	Mendefinisikan satu baris field input yang pengguna dapat memasukkan teks
password	Mendefinisikan satu baris field input yang pengguna dapat memasukkan teks yang dirahasiakan
radio	Mendefinisikan satu baris field input yang pengguna dapat memilih hanya satu saja

checkbox	Mendefinisikan satu baris field input yang pengguna dapat tidak memilih atau pilih semua.
button / submit / reset	Mendefinisikan bentuk tombol untuk mengirim semua form
file	Mendefinisikan satu baris field input yang pengguna dapat upload file
hidden	mendefinisikan satu baris field input yang pengguna dapat menyembunyikan form

Tabel 6. Tags Pada Form

Tags	Descriptions
<form>	Mendefinisikan sebuah form HTML untuk input pengguna
<input>	Mendefinisikan sebuah kontrol input
<textarea>	Mendefinisikan sebuah kontrol input multiline (text area)
<label>	Mendefinisikan sebuah label untuk sebuah elemen
<fieldset>	terkait unsur-unsur dalam bentuk

<legend>	Mendefinisikan sebuah caption untuk elemen <fieldset>
<select>	Mendefinisikan sebuah daftar drop-down
<optgroup>	Mendefinisikan sebuah kelompok pilihan yang terkait dalam daftar drop-down
<option>	Mendefinisikan pilihan dalam daftar drop-down
<button>	Mendefinisikan sebuah tombol diklik
<datalist>	Baru Menentukan daftar pilihan yang telah ditetapkan untuk kontrol input
<keygen>	Baru Mendefinisikan kunci-pair bidang pembangkit (untuk formulir)
<output>	Baru Mendefinisikan hasil penghitungan

5. Menggunakan Karakter Khusus

Berikut ini tata cara pengetikkan untuk menampilkan karakter khusus pada halaman html dapat dilihat pada Tabel 7.

Tabel 7. Karakter Khusus

Simbol	Description	Character
©	Copyright symbol	©

®	registered trademark symbol	®
™	trademark symbol	™
 	nonbreaking space	
<	less-than symbol	<
>	greater-than symbol	>
&	Ampersand	&



Rangkuman

HTML adalah singkatan dari Hypertext Markup Language. HTML memungkinkan seorang user untuk membuat dan menyusun bagian paragraf, heading, link atau tautan, dan blockquote untuk halaman web dan aplikasi. HTML adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web. HTML inilah yang menyusun sebuah halaman web menjadi sebagaimana yang kita lihat melalui browser (penjelajah internet). Dalam pembuatan website, dikenal dua komponen penting yaitu www dan HTML. Dimana www merupakan suatu protokol standar dari internet, sedangkan HTML merupakan script atau program standar yang dijalankan oleh www atau internet. Proyek HTML ini dibuat dengan menggunakan media web editor notepad.

Setiap dokumen html yang ditampilkan browser pastilah memiliki struktur didalamnya termasuk html yang memiliki struktur dasar html untuk awal mula pengenalan, dimana struktur tersebut terdiri dari tag, elemen, dan atribut untuk mengatur seluruh isi informasi dan tata letak yang ditampilkan browser. Sebenarnya walaupun tanpa penulisan struktur yang benar browser dapat

saja membaca dokumen, tetapi susunan yang menjadi standar penulisan tidak akan sesuai dan mungkin akan berantakan, dari situlah akan di bahas dasar-dasar penulisan struktur dasar html yang baik agar website tampil sesuai keinginan. Struktur dasar HTML ini biasanya selalu ada dalam setiap file HTML. Penulisan struktur dasar HTML dasar juga harus berurutan atau mengikuti standar struktur HTML yang ada. Struktur dasar HTML terdiri beberapa tag-tag HTML, setiap tag HTML bisa memiliki tag pembuka dan penutup, tapi ada juga yang tidak memiliki tag penutup. Struktur dasar HTML terdiri dari:

- `<html></html>` Tag html ini menjadi awal pembuka setelah dtd, berfungsi untuk memberikan informasi kepada browser bahwa dokumen yang akan di proses adalah dokumen html, tag html ini umumnya memiliki tag pembuka dan penutup dimana seluruh elemen akan di tulis di dalamnya.
- `<head></head>` Tag head berfungsi sebagai pemberi informasi tambahan untuk diolah browser tanpa di tampilkan ke utama web browser, informasi tersebut bisa berupa judul halaman yang akan ditampilkan pada tab web browser, berupa meta informasi, dan dapat digunakan sebagai import kode pendukung seperti css dan javascript.
- `<title></title>` Tag title berada di antara tag head, dimana tag ini berfungsi untuk menampilkan judul halaman yang akan muncul pada bagian atas atau tab web browser.
- `<body></body>` Tag body berisi semua informasi yang akan di tampilkan ke layar utama web browser, sebagai contoh ketika kita membuka halaman pada web browser maka bagian dari layar yang ditampilkan secara visual tersebut diproses di dalam tag body.



Latihan Soal

1. Buat folder baru di laptop/pc anda dengan nama htmlku

2. Buka text editor yang ada di laptop / komputer anda (jika belum ada silahkan klik disini)
3. Buat file baru kemudian simpan file tersebut di dalam folder htmlku dengan nama file latihan.html
4. Jika sudah selsesai isikan kode berikut:

```
<html>
```

```
<head>
```

```
<title>Belajar HTML</title>
```

```
</head>
```

```
<body>
```

```
<!-- bagian body adalah tempat untuk menampilkan informasi -->
```

```
halo selamat belajar HTML. semangat terus ya
```

```
</body>
```

```
</html>
```

BAB III

CSS

3.1 Pengenalan CSS

CSS singkatan dari Cascading Style Sheets. Secara bahasa berarti lembaran gaya yang mengalir ke bawah. Kamu akan kebayang dengan istilah ini setelah nanti tau penggunaan dan bentuknya. Secara istilah, CSS adalah bahasa penulisan yang digunakan untuk mendeskripsikan penampilan sebuah dokumen markup. CSS selalu digunakan bersamaan dengan HTML, meskipun kita dapat juga menggunakannya untuk mengatur style dokumen markup lain seperti XML, SVG dan XUL. HTML dirancang tidak ditujukan untuk mendesain sebuah halaman web, tapi hanya untuk menampilkan konten saja. Pada saat HTML beranjak ke versi 3.2, diperkenalkan tag `` yang digunakan untuk mengatur tampilan sebuah teks, seperti jenis font yang digunakan, warna, dan ukuran. Tapi dengan adanya tag ini, malah jadi menyulitkan proses development, karena tag ini mesti disematkan di banyak halaman untuk mendapatkan hasil style yang sama. Bayangkan ketika kita sudah memiliki banyak halaman web, belasan atau puluhan, lalu kemudian hendak mengedit beberapa style saja misalkan ukuran font dan ingin semua halaman mendapatkan perubahan yang konsisten. Untuk itu, dibuatlah CSS, yang memisahkan style dari halaman web, sehingga antara konten pada HTML dan desain tampilan pada dokumen CSS dapat dikerjakan di dua tempat berbeda. Dengan kata lain, dokumen HTML cukup berisi konten saja, dan satu dokumen CSS dapat disematkan pada setiap dokumen HTML agar semuanya menampilkan style yang sama dan konsisten. Oleh karena itu bahasa CSS sangat penting untuk seorang web programmer terutama web designer. Secara singkat bahasa CSS sendiri digunakan untuk mendesign atau mengatur tampilan pada halaman web. Seiring dengan perkembangan teknologi dan pemrograman, CSS tidak hanya digunakan oleh HTML dan

XHTML saja, namun bisa juga digunakan untuk mendesign tampilan aplikasi android. CSS diakses menggunakan id atau class.

Setelah HTML selesai dituliskan, kita dapat mereferensikan CSS kepada HTML yang ada agar desain yang dispesifikasikan oleh CSS dapat diaplikasikan pada HTML. Terdapat tiga cara untuk memberikan referensi CSS, yaitu:

- **Referensi ke File Eksternal**

Kita dapat memberikan referensi ke sebuah file CSS yang berada di luar HTML. Cara referensi CSS seperti ini seringkali dianggap sebagai best practice dalam pengembangan web.

- **Penulisan CSS pada Elemen Head**

CSS yang ingin diaplikasikan pada sebuah dokumen HTML dapat juga dituliskan pada bagian head dari sebuah dokumen. Penulisan CSS seperti ini tidak disarankan, karena umumnya elemen-elemen yang ada dalam sebuah dokumen akan digunakan kembali pada dokumen lain. Penulisan CSS langsung pada bagian head akan menyebabkan elemen-elemen yang berulang harus dituliskan ulang pada dokumen lain juga.

- **CSS di dalam Atribut style pada Elemen**

Menuliskan CSS di dalam atribut style pada elemen HTML (atribut ini dimiliki oleh semua elemen) merupakan cara terakhir, yang juga tidak disarankan karena penulisan seperti ini akan “mengotori” kode HTML. HTML dibuat dengan tujuan untuk memberikan makna semantik untuk konten, bukan desain. Begitupun, metode ini biasanya digunakan untuk manipulasi gaya yang dilakukan secara dinamis, melalui Javascript (yang tidak mengotori kode HTML, karena biasanya atribut baru diisikan setelah HTML selesai dibaca oleh browser).

3.2 Sejarah CSS

Sekitar tahun 1996, Netscape dan Internet Explorer (IE) pada masa itu, saling bersaing untuk membuat dan mengembangkan tag-tag baru mengatur tampilan web. Sehingga pada tanggal 17 Desember 1996, W3C disebut juga world Wide Web Consortium menjadi CSS sebagai standars dalam pembuatan dokumen web

dengan mengurangi pembuatan tag-tag baru oleh netscape dan IE. Sejak kemunculannya tersebut, CSS mengalami perkembangan yang sangat pesat. Format dasar CSS yang banyak kita gunakan sekarang ini merupakan ide dari seorang programmer bernama Hakon Wium Lie yang tertuang dalam proposalnya mengenai Cascading HTML Style Sheet (CHSS) pada bulan Oktober 1994 (dalam konferensi W3C di Chicago, Illinois). Kemudian, beliau bersama-sama dengan seorang temannya yang bernama Bert Bos mengembangkan suatu standard CSS. Pada akhir tahun 1996, CSS telah resmi dipublikasikan (dan menyusul kemudian CSS Level 1 pada bulan Desember). Pengerjaan proyek ini juga didukung oleh seorang programmer bernama Thomas Reardon dari perusahaan software ternama, Microsoft. CSS digunakan oleh penulis maupun pembaca halaman web untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan dokumen. CSS digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan HTML atau bahasa markup lainnya) dengan presentasi dokumen (yang ditulis dengan CSS). Pemisahan ini dapat meningkatkan aksesibilitas isi, memberikan lebih banyak keleluasaan dan kontrol terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada stuktur isi. CSS memungkinkan halaman yang sama untuk ditampilkan dengan cara yang berbeda untuk metode presentasi yang berbeda, seperti melalui layar, cetak, suara (sewaktu dibacakan oleh browser basis-suara atau pembaca layar), dan juga alat pembaca braille. Halaman HTML atau XML yang sama juga dapat ditampilkan secara berbeda, baik dari segi gaya tampilan atau skema warna dengan menggunakan CSS.

1. CSS level 1

Pada tanggal 17 Agustus 1996 World Wide Web Consortium (W3C) menetapkan CSS sebagai bahasa pemrograman standard dalam pembuatan web. Tujuannya adalah untuk mengurangi pembuatan tag-tag baru oleh Netscape dan Internet Explorer, karena kedua browser tersebut sedang bersaing mengembangkan tag sendiri untuk mengatur tampilan web. CSS 1 mendukung pengaturan tampilan dalam hal: (1) Font (Jenis ketebalan), (2) Warna, teks, background dan elemen lainnya, (3) Text attributes, misalnya spasi antar

baris, kata dan huruf, (4) Posisi teks, gambar, table dan elemen lainnya, (5) Margin, border dan padding.

2. CSS level 2

W3C menyempurnakan CSS tahap awal dengan menciptakan standard CSS 2 yang menjadi standard hingga saat ini pada tahun 1998. Semua atribut dari CSS 1 dimasukkan dan diperluas dengan penekanan pada International Accessibility and Capability khususnya media-specific CSS. CSS 2 dikembangkan untuk memenuhi kebutuhan terhadap format dokumen agar bisa ditampilkan di printer.

3. CSS level 3

CSS 3 adalah versi terbaru dari CSS yang mampu melakukan banyak hal dalam mendesain website. CSS 3 dapat melakukan animasi pada halaman website, diantaranya animasi warna dan animasi 3D. Desainnya yang memudahkan dalam hal kompatibilitas websitenya pada smartphone dengan dukungan fitur baru yakni media query. Selain itu, banyak fitur baru pada CSS 3 yaitu : Multiple background, border-radius, drop-shadow, border-image, CSS-Math dan CSS Object Model. Fitur terbaru CSS 3 :

- a. Animasi, sehingga pembuatan animasi tidak memerlukan program sejenis Adobe Flash dan Microsoft Silverlight.
- b. Beberapa efek teks, seperti teks berbayang, kolom koran dan "Word-Wrap".
- c. Beberapa efek pada kotak, seperti kotak yang ukurannya dapat diubah-ubah, transformasi 2 dimensi dan 2 dimensi, sudut-sudut yang tumpul dan bayangan.

3.3 Fungsi dan Kegunaan CSS

CSS biasanya disimpan di dalam sebuah file berekstensi .css dan disematkan di dalam dokumen HTML untuk memberikan style pada halaman tersebut. Meski demikian, kita tetap dapat menambahkan sintaks CSS langsung di dalam dokumen HTML, didalam tag `<style></style>` untuk mengakomodir penerapan style yang hanya digunakan di dalam dokumen tersebut. Kita juga dapat menyematkan sintaks CSS langsung di dalam atribut style pada suatu tag HTML untuk menerapkan style yang hanya digunakan

pada tag itu saja. Sama halnya dengan objek – objek lain yang ada dimuka bumi ini. Dalam penggunaanya, CSS juga tidak luput dari kelebihan dan kekurangan. Kelebihan CSS, diantaranya:

1. CSS sangat mudah untuk dipelajari dan digunakan
2. CSS memisahkan antara Desain dan Konten Web/Blog
3. Pengaturan Desain dapat dilakukan seefisien mungkin
4. Karena satu css dapat dipakai beberapa kali ,maka dapat menghemat penulisan kode
5. Mempersingkat waktu kerja, baik saat membuat maupun saat modifikasi halaman Web/Blog
6. Ukuran file HTML jadi lebih kecil, karena biasanya CSS disimpan di file terpisah (External Stylesheet)

Kekurangannya, diantaranya:

1. CSS sangat menyita waktu karena tidak semua browser mengartikan sintaks-sintaks CSS yang sama
2. Kadang desain sudah terlihat rapih di suatu browser namun ketika dicoba di browser lain malah jadi acak-acakan
3. Kekurangan tersebut dapat diatasi dengan script-script khusus(CSS Hack)

3.4 Sintaks Dasar CSS

CSS terdiri dari aturan style yang diinterpretasikan oleh browser dan kemudian diterapkan sesuai elemen dalam dokumen Anda. Aturan style terbuat dari tiga bagian, diantaranya:

- Selector – Adalah style tag HTML yang akan diterapkan. Ini dapat dijumpain pada setiap tag seperti <h1> atau <table>dll.
- Property - Adalah atribut dari HTML tag. Sederhananya, semua atribut HTML dikonversi ke CSS property. Mereka bisa menjadi warna, border, dan lain lain.
- Value - Value yang ditugaskan ke properti. Sebagai contoh, warna properti dapat memiliki nilai berwarna merah atau #F1F1F1 dan lain sebagainya.

Untuk menentukan beberapa gaya aturan untuk satu elemen. Maka dapat menentukan aturan-aturan seperti menggabungkan beberapa properti dan nilai-nilai yang sesuai ke dalam satu blok sebagaimana didefinisikan dalam Gambar 6 sebagai contoh CSS. Di sintak CSS ini, semua pasangan properti dan nilai yang dipisahkan dengan titik koma (;). Dan dapat menyimpannya dalam satu baris

atau beberapa baris. Namun untuk dibaca lebih baik maka dapat menyimpannya ke dalam baris terpisah.

```
h1 {  
  color: #36C;  
  font-weight: normal;  
  letter-spacing: .4em;  
  margin-bottom: 1em;  
  text-transform: lowercase;  
}
```

Gambar 6. Contoh Sintaks CSS

Sebagai bahasa yang digunakan untuk memberikan gaya tampilan, CSS menggunakan metode deklaratif untuk menspesifikasikan bagian HTML yang ingin diberikan gaya tampilan. Pemilihan elemen HTML dilakukan dengan menspesifikasikan selector. Anda dapat menuliskan langsung nama tag yang ingin diberi style tanpa perlu menambahkan tanda $\langle \rangle$, dapat dilihat pada Gambar 7.



Gambar 7. Contoh Penulisan Selector

Pada contoh Gambar 7 di atas, Kita telah memberi style pada seluruh tag h1 yang terdapat dalam file HTML. Jika tag HTML yang ingin diberi style memiliki ID, Anda dapat menuliskan nama ID tersebut dengan diawali dengan tanda kress (#). Contoh : #header. Dan jika tag yang diberi style memiliki Class, maka penulisan selector bisa dilakukan dengan tanda titik (.) diikuti dengan nama class. Property adalah sifat-sifat yang ingin diterapkan pada selector, seperti warna teks, warna background, jarak antar elemen, garis pinggir dan lain sebagainya. Untuk memberikan nilai/value pada property kita gunakan tanda titik dua (:). Setiap property diakhiri dengan titik koma (;), jika Anda tidak mengakhirinya maka browser tidak akan mengetahui maksud dari property tersebut.

3.4 Jenis Selector

Selector adalah bagian CSS untuk merujuk elemen HTML yang ingin dikenakan style. Ada beberapa jenis selector yang dapat kita gunakan untuk memudahkan pekerjaan styling kita pada CSS, seperti Selektor Tag, Selektor Class, Selektor ID, Selektor Atribut, Selektor Universal.

- **Selector Tag**
Selektor Tag disebut juga Type Selector. Selektor ini akan memilih elemen berdasarkan nama tag. Misal:

```
p {  
  color: blue;  
}
```

- **Selektor Class**
Selektor class adalah selektor yang memilih elemen berdasarkan nama class yang diberikan. Selektor class dibuat dengan tanda titik di depannya. Misalnya:

```
.pink {  
  color: white;  
  background: pink;  
  padding: 5px;  
}
```

- Selektor ID

Selektor ID hampir sama dengan class. Bedanya, ID bersifat unik. Hanya boleh digunakan oleh satu elemen saja. Selektor ID ditandai dengan tanda pagar (#) di depannya. Misal :

```
#header {  
  background: teal;  
  color: white;  
  height: 100px;  
  padding: 50px;  
}
```

- Selektor Atribut

Selektor atribut adalah selektor yang memilih elemen berdasarkan atribut. Selektor ini hampir sama seperti selektor Tag. Contoh selektor Atribut:

```
p {  
  color: blue;  
}
```

- Selektor Universal

Selektor universal adalah selektor yang digunakan untuk menyeleksi semua elemen pada jangkauan (scope) tertentu. Contoh:

```
* {  
  border: 1px solid grey;  
}
```



Rangkuman

Pada dasarnya HTML dan CSS memiliki kedekatan yang sangat erat, sehingga kinerja keduanya tidak dapat dipisahkan. HTML sendiri merupakan bahasa markup dan CSS memperbaiki style, kedua bahasa pemrograman ini harus terus beriringan. Selain

pada urusan teknis, CSS berpengaruh juga pada tampilan sebuah website. Para pemilik website pun tentu tidak menginginkan tampilan artikel tanpa modifikasi apapun. Kelebihan CSS sendiri dapat mengatur semua tampilan pada aspek file yang berbeda. Kemudian Anda dapat menentukan file dan mengintegrasikannya di atas markup HTML.

Secara default CSS memakai bahasa Inggris sederhana berbasis syntax. Selanjutnya CSS didukung dengan berbagai rules yang mengaturnya. Keberadaan HTML ini pada dasarnya dirancang semata-mata untuk mendeskripsikan konten. Yang kemudian dibantu dengan keberadaan CSS yang memberikan sedikit sentuhan modifikasi. Kombinasi kinerja keduanya dapat memaksimalkan tampilan website seperti yang Anda inginkan. Struktur CSS cukup sederhana, sehingga pengaplikasiannya pun cukup mudah. Di dalam struktur CSS ini terdiri dari selector dan declaration block. Dalam hal ini Anda dapat memilih elemen yang diinginkan kemudian Anda dapat melakukan deklarasi terhadap elemen tersebut. Selector bekerja mengarahkan elemen ke HTML yang akan diubah penampilannya. Declaration block bekerja memisahkannya dengan tanda titik koma. Jenis-Jenis CSS Berdasarkan Cara Penggunaannya, diantaranya:

- **Inline Style Sheet** : Jenis CSS pertama yang kerap kali digunakan adalah Inline Style Sheet, CSS tipe ini biasa diterapkan langsung pada object yang ingin dirubah tampilannya. Misalnya, seperti teks biasa yang ingin dirubah warnanya. Penggunaan CSS jenis ini hanyamembutuhkan penambahan tag `<style>` sehingga sangat mudah diaplikasikan.
- **External Style Sheet**: Berbeda dengan Inline Style Sheet, External Style Sheet jauh lebih praktis digunakan karena peletakkannya yang berada diluar area object. CSS jenis ini juga bisa digunakan secara berulang pada beberapa object sekaligus, bahkan antar website sehingga lebih menghemat ruang.
- **Embedded Style Sheet** : CSS jenis ini memiliki ciri yang hampir sama dengan Inline Style Sheet karena diletakkan pada halaman yang sama pada suatu website. Bedanya, untuk Embedded Style Sheet posisinya berada di antara tag

<head> dan </head> serta diawali dengan tag <style>.



Latihan Soal

1. Buatlah Tabel seperti berikut dengan mengatur warna di elemen tag th!

```
<style type="text/css">
#table1 {
    background: blue;
    color: white;
    border: 3px solid red;
    padding: 3px;
}
#table2{
    background: silver;
    color: blue;
    padding: 3px;
    border-bottom: solid red;
}
</style>
```


Nama Computer	Prosesor	Ram
Clnt-1	IP 4 1,8 Ghz	10 GB
Clnt-2	IP 2 Ghz C	6GB
Clnt-3	XP 2000+	20 GB

2. Grafik yang di buat dengan SGV menggunakan format apa?
3. Bagaimana Style HTML yang benar untuk merujuk ke sheet eksternal untuk CSS?
4. Bagaimanakah sintaks CSS yang benar?
5. Sebutkan jenis selektor?

BAB IV

XML

4.1 Pengertian XML

Extensible Markup Language (XML) termasuk dalam keluarga Markup Language seperti halnya HTML atau WML. Namun dengan XML, kita bisa mendefinisikan custom tag sendiri. XML merupakan meta-language seperti tag HTML yang digunakan untuk mendeskripsikan data-data. Tag-tag pada XML dapat Anda buat sendiri sehingga tidak ada tag-tag default seperti halnya tag HTML. Kunci utama XML yaitu data yang bila diolah bisa memberikan informasi. XML memberikan cara paling dasar namun bisa diubah untuk menggambarkan isi dari dokumen. Dengan otomatis XML dapat dipakai untuk menggambarkan ngasal view database, tetapi dengan suatu cara yang standar. Beberapa hal yang perlu diingat mengenai kegunaan XML adalah sebagai berikut:

1. XML dapat memisahkan data dari tag HTML. Biasanya data yang ditampilkan dengan HTML disimpan jadi satu dengan file HTML. Kini dengan XML, data dapat dipisah dari file HTML dan disimpan pada file XML tersendiri. Dengan demikian Anda dapat menkonsentrasikan pada HTML untuk menampilkan dan memformat data, dan XML yang akan berisi data tersebut.
2. XML digunakan untuk pertukaran data meskipun berbeda platform. Pada kenyataannya dalam dunia komputer, sistem komputer dan database mengandung data yang tidak kompatibel satu sama lain. Dengan demikian tidak mungkin terjadinya pertukaran data melalui internet jika terdapat perbedaan sistem operasi dan aplikasi database yang digunakan. Dengan menggunakan XML untuk pertukaran data, masalah perbedaan platform dan aplikasi tidak perlu diresahkan lagi. karena data yang disimpan pada XML dapat dibaca oleh berbagai macam platform dan aplikasi.

3. Data pada XML berupa teks. Karena data yang disimpan pada XML berupa teks, maka data tersebut akan sangat mudah dibaca oleh berbagai aplikasi. Dan data akan tetap valid meskipun dilakukan upgrade terhadap sistem operasi, server, aplikasi baru dan browser baru
4. RSS (Really Simple Syndication): merupakan pengembangan dari XML. RSS digunakan untuk membuat standarisasi pada penggunaan tag-tag XML sehingga pembuatan tag XML tidak lagi sebebas yang Anda inginkan. Alasannya sederhana, agar pada saat mengimpor data XML dari situs lain, Anda tidak perlu mencari tahu apa nama tag XML yang digunakan. (*)

Pada XML, untuk menjelaskan apakah sebuah tag itu valid atau tidak, kita bisa menggunakan DTD (Document Type Definition). DTD juga menjelaskan struktur dari dokumen XML. Kemudian untuk menjelaskan arti-arti dari tag yang dibuat, dapat digunakan stylesheet, yaitu XSL (Extensible Stylesheet Language). XSL terdiri dari tiga bagian dan tiap bagian ada rekomendasinya sendiri-sendiri dari W3C, yaitu : Xpath (XML Path Language), bahasa untuk menunjukkan sebuah bagian dari dokumen XML. XSLT (XSL Transformation), bahasa untuk mentransformasikan sebuah dokumen XML ke dokumen lain. XSL, yaitu XSLT ditambah dengan kumpulan penjelasan mengenai formatting objects dan formatting properties. Bentuk dokumen XML terdiri dari bagian yang disebut node. Contoh program XML, dapat dilihat Gambar 8. Node itu adalah:

- Root node yaitu node yang menutup semua Pada satu dokumen XML Cuma ada satu root node. Node yang lainnya berada di dalam root node.
- Element node yaitu bagian dari dokumen XML yang ditandai dengan tag pembuka dan tag penutup, atau bisa juga sebuah tag tunggal elemen kosong seperti <anggota nama="devie"/> . Root node biasa juga disebut root element.
- Attribute note termasuk nama yang ditulis pada tag awal sebuah elemen atau pada tag tunggal.
- Text node adalah text isi dari sebuah elemen, ditulis dibagian tag pembuka dan tag penutup.
- Processing Instruction node yaitupengolahan pada dokumen

Node ini ditandai awali dengan karakter `<?` Dan diakhiri dengan `?>`. Tapi perlu diingat bahwa header standard XML `<?xml version="1.0" encoding="iso-8859-1"?>` bukanlah processing instruction node.

- Header standard bukan bagian dari hirarki pohon dokumen XML.
- Name Space Node yang mewakili deklarasi namespace.

```
<?xml version="1.0" ?>
- <product barcode="2394287410">
  <manufacturer>Verbatim</manufacturer>
  <name>DataLife MF 2HD</name>
  <quantity>10</quantity>
  <size>3.5"</size>
  <color>black</color>
  <description>floppy disks</description>
</product>
```

Gambar 8. Contoh Penggunaan XML

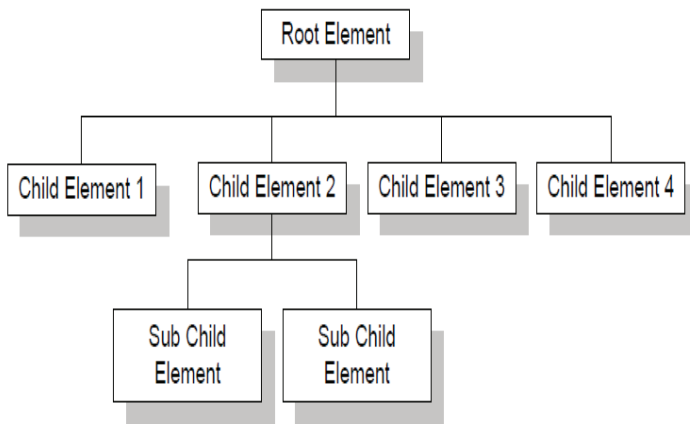
4.2 Sintaks XML

Dibandingkan dengan HTML, XML lebih cerewet. Kalau kita menulis sebuah dokumen HTML, beberapa kesalahan penulisan masih ditolerir. Misalnya kita menempatkan tag bersilangan seperti `<p>Huruf Tebal</p>` meskipun tidak dianjurkan, HTML masih bisa bekerja dan menampilkan hasil seperti yang kita inginkan. Tidak demikian dengan XML. Lebih jelasnya kita akan bahas di bawah bagaimana membuat dokumen XML yang baik. Heading standard untuk Document XML. Setiap membuat dokumen XML diawali dengan heading standard XML. Formatnya adalah sebagai berikut:

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

Dokumen XML harus memiliki Root tag. Sebuah dokumen XML yang baik harus memiliki root tag. Yaitu tag yang melingkupi keseluruhan dari dokumen. Tag-tag yang lain, disebut child tag,

berada didalam root membentuk hirarki seperti Gambar 9.



Gambar 9. Diagram Hirarki XML

Pada bahasa pemrograman atau scripting kita mengenal adanya komentar (comment). Komentar adalah kalimat/baris yang tidak dieksekusi oleh compiler, browser atau parser. Untuk menyisipkan komentar pada dokumen XML caranya adalah sebagai berikut:

```
<!-- Baris ini tidak di eksekusi oleh parser -->
```

4.3 Element, Attribute, Entity

Bila sebuah element mengandung beberapa child element, maka kita perlu mendeklarasikan child element apa saja yang dipunyai element tersebut. Pada contoh diatas kita melihat deklarasi element.

```
<!ELEMENT organisasi (anggota)>  
<!ELEMENT anggota (nama,alamat,kelamin,jabatan)>
```

Maksudnya adalah element bernama organisasi memiliki satu child element bernama anggota. Lalu element bernama anggota itu sendiri mempunyai empat child element yang bernama nama, alamat, kelamin dan jabatan. Setelah itu kita perlu medeklarasikan juga type dari element-element diatas.

```

<!ELEMENT nama (#PCDATA)>
<!ELEMENT alamat (#PCDATA)>
<!ELEMENT kelamin (#PCDATA)>
<!ELEMENT jabatan (#PCDATA)>

```

Agar dokumen XML kita valid, kita juga perlu mendefinisikan semua atribut yang akan kita gunakan dalam dokumen kita. Untuk mendefinisikannya kita akan menggunakan Attribute list declaration. Entity XML digunakan untuk menggantikan kalimat yang panjang atau satu blok elemen yang sering digunakan dengan sebuah pengenalan singkat. Misalnya ingin menggantikan kalimat “Manajemen Data dan Informasi dengan XML/XSL” dengan entity &judul. Sekali entity didefinisikan didalam DTD, maka dapat digunakan dimana saja pada seluruh dokumen XML.

```

<?xml version="1.0" encoding="iso-8859-1">
<!DOCTYPE organisasi [
<!ENTITY judul "Manajemen data dan informasi dengan XML/
XSLT"> ]>
<resensi>
  <buku judul="&judul;">
    <ulasan>Buku yang berjudul &judul; ini ditulis oleh
    Moh Junaedi..</ulasan>
  </buku>
</resensi>

```

4.4 Kelebihan dan Keunggulan Menggunakan XML

XML merupakan bahasa markup seperti halnya HTML. XML dibuat oleh World Wide Web Consortium (W3C) untuk mendefinisikan sintaks untuk menyandikan dokumen yang dapat dibaca oleh manusia dan mesin. Ini dilakukan melalui penggunaan tag yang menentukan struktur dokumen serta bagaimana dokumen harus disimpan dan dikirim. adapun kelebihan dan

keunggulan menggunakan XML, diantaranya:

1. XML adalah platform independen dan bahasa pemrograman independen sehingga dapat digunakan pada sistem apapun.
2. XML mendukung unicode. Unicode adalah standar penyandian internasional untuk digunakan dengan berbagai bahasa dan skrip.
3. Data yang disimpan dan dikirim menggunakan XML dapat diubah kapan saja tanpa mempengaruhi penyajian data. Secara umum bahasa markup lainnya seperti HTML digunakan untuk presentasi data.
4. XML memungkinkan validasi menggunakan DTD dan Skema. Validasi ini memastikan bahwa dokumen XML bebas dari kesalahan sintaks apapun.
5. XML menyederhanakan berbagai data antar sistem karena sifatnya yang independen terhadap platform. Data XML tidak memerlukan konversi apapun saat ditransfer antara sistem yang berbeda.

Selain kelebihan diatas, terdapat juga keunggulan lain dari XML, diantaranya: (1) Pintar XML dapat menangani berbagai tingkat (level) kompleksitas, (2) Dapat beradaptasi. Dapat mengadaptasi untuk membuat bahasa sendiri. Misalnya Microsoft dapat membuat bahasa MSXML dan Macromedia mengembangkan MXML, (3) Mudah pemeliharaannya, (4) XML lebih sederhana, (5) Mudah dipindah-pindahkan (Portability). XML mempunyai kemudahan perpindahan (portabilitas) yang lebih bagus. Oleh karena itu, manfaat XML, diantaranya:

1. Mempermudah pengaturan data.
2. SEO atau kepanjangan dari (Search Engine Optimization) adalah suatu cara atau teknik untuk membuat situs atau blog kita berada pada halaman/posisi satu di mesin pencarian (search engine) seperti Google, Bing, dan Yahoo. Search Engine Optimization atau sering di sebut SEO adalah suatu cara atau teknik agar weblog yang kita miliki mempunyai posisi di halaman pencarian google yang bisa meningkatkan trafik rank pada weblog kita. Lebih tinggi nilai SEO weblog berarti besar kemungkinan untuk menempati posisi pertama di pencarian google. Menurut dari beberapa sumber yang sudah di cari pengertian SEO tetap mencakup hal yang

sama yang mengoptimalkan suatu halaman weblog agar mempunyai posisi di halaman pencarian google. Istilah search engine optimization pertama kali digunakan pada 26 Juli tahun 1997 oleh sebuah pesan spam yang diposting di Usenet. Pada masa itu algoritma mesin pencari belum terlalu kompleks sehingga mudah dimanipulasi. Versi awal algoritma pencarian didasarkan sepenuhnya pada informasi yang disediakan oleh webmaster melalui meta tag pada kode html situs web mereka. Meta tag menyediakan informasi tentang konten yang terkandung pada suatu halaman web dengan serangkaian kata kunci (keyword). Sebagian webmaster melakukan manipulasi dengan cara menuliskan katakunci yang tidak sesuai dengan konten situs yang sesungguhnya, sehingga mesin pencari salah menempatkan dan memeringkat situs tersebut. Hal ini menyebabkan hasil pencarian menjadi tidak akurat dan menimbulkan kerugian baik bagi mesin pencari maupun bagi pengguna internet yang mengharapkan informasi yang relevan dan berkualitas. Keuntungan dari SEO, diantaranya: mendatangkan trafik yang banyak, meningkatkan Penjualan Online, dan meningkatkan Daya Saing.

3. Keamanan Sistem (security). Sebagai sebuah markup language, maka dokumen XML tersusun dari kumpulan teks biasa. Sedangkan maksud yang terkandung pada data dijelaskan melalui penanda yang biasa di sebut dengan tag. Dengan menggunakan tag, maka sebuah dokumen XML dapat dengan mudah di baca dan di mengerti oleh user dan bahkan oleh program komputer. Dengan semakin maraknya dan luasnya penggunaan XML pada berbagai layanan di internet, di mana penyebaran informasinya sebagian besar memakai infrastruktur jaringan umum, maka mulailah muncul hambatan dan permasalahan mengenai kebutuhan akan kewanitaan data bagi informasi yang terkandung di dalam dokumen XML. Dengan adanya hal ini mengingatkan bahwa sebuah dokumen XML tersusun dari sekumpulan teks yang sangat mudah di pahami oleh user atau bahkan program komputer.

Berdasarkan kebutuhan tersebut, maka W3C (world wide web consortium) berusaha mengembangkan beberapa spesifikasi tambahan untuk XML. Spesifikasi tersebut dikembangkan untuk memungkinkan para pengguna untuk menggunakan fasilitas pengamanan data pada sebuah dokumen XML yang akan di distribusikan. Sistem keamanan data yang terdapat pada spesifikasi XML tersebut itu di kenal dengan istilah XML security. Dengan adanya XML security ini dapat menjaga keamanan data informasi bagi pengguna. Di karenakan jika terjadi kerusakan atau data/informasi yang kita punya di ambil atau di hancurkan oleh orang-orang yang tidak bertanggung jawab, maka data akan tetap utuh. Karena pada suatu website yang di lengkapi dengan dokumen XML, biasanya tempat penyimpanan data atau informasi yang di gunakan atau di pake itu ada dua, yaitu database dan XML.

Dokumen XML dapat digunakan untuk berbagai macam tujuan, seperti: Sebagai penyimpan data (database) yang mudah dibaca oleh pengguna karena disimpan dalam bentuk teks, Standar transfer data, dapat digunakan untuk pengiriman data transaksi antar perusahaan atau mengirim data dari DBMS (Database Management System) yang berbeda (misalnya dari Oracle ke SQL Server), Sebagai acuan membuat bahasa baru, seperti WML (Wireless Markup Language) yang digunakan pada telepon seluler dengan protokol WAP (Wireless Application Protocol) atau VoiceXML yang digunakan sebagai bahasa markup untuk pengenalan suara, dialog, aplikasi interaksi respon suara maupun DTMF (Dual-Tone Multi-Frequency Signaling) seperti aplikasi pengisian pulsa atau call center dan sebagainya, Sebagai file konfigurasi, di Java dokumen-dokumen XML sering kita jumpai seperti file server.xml dan web.xml yang digunakan Tomcat atau perintah-perintah query yang disimpan dalam file XML yang dipakai pada framework iBatis atau Hibernate.

4.5 Kekurangan dan Kerugian Menggunakan XML

Saat user mengakses data website menggunakan XML, maka yang diperlukan adalah DOM XML dan DOM HTML, serta memerlukan request dari server. Sebab itu, penggunaan XML memerlukan karakter string yang lebih banyak, sehingga akan

berpengaruh pada ukuran filenya. Tidak dapat dipungkiri bahwa XML sangat fleksibel untuk pertukaran data melalui internet, namun terdapat beberapa kelemahan yang ditemukan pada XML, diantaranya:

1. XML memisahkan antara bagian data (content) dengan bagaimana data ditampilkan secara terstruktur, hal ini menyebabkan data dapat dimanipulasi oleh siapapun yang membacanya.
2. XML berbasis teks, mudah dibaca oleh manusia (human-readable), maka dokumen XML mudah untuk di-debug dan dilewatkan melalui firewall.

Secara sederhana XML adalah suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur. Secara teknis XML didefinisikan sebagai suatu bahasa meta-markup yang menyediakan format tertentu untuk dokumen-dokumen yang mempunyai data terstruktur. Bahasa markup adalah mekanisme untuk mengenal secara terstruktur di dokumen. XML adalah suatu aplikasi profil dari SGML. Seperti yang didefinisikan dari ISO 8879, SGML adalah cara standart dan vendor-independent. XML tidak mempunyai definisi secara tepat karena ada yang berpendapat bahwa XML bukanlah suatu bahasa pemrograman, melainkan XML merupakan sintaks yang digunakan untuk menjelaskan bahasa markup lain, sehingga dinamakan meta-language. Meskipun demikian pendapat yang XML bukan merupakan bahasa markup, didasarkan bahwa XML merupakan bahasa markup terpisah untuk tujuan terpisah. Kekurangan dan kerugian menggunakan XML, diantaranya:

1. Redundansi dalma sintaks XML menyebabkan biaya penyimpanan dan transportasi yang lebih tinggi ketika volume data besar
2. Dokumen XML kurang dapat dibaca dibandingkan dengan format transmisi data berbasis teks lainnya seperti JSON
3. XML tidak mendukung array
4. Ukuran file XML biasanya sangat besar kareba sifatnya yang verbase, sangat tergantung pada siapa menulisnya.

Sebagai catatan bahwa XML sebagai bahasa text base bersifat case sensitif artinya membedakan inputan antara huruf besar dan huruf kecil. Hal ini bisa menjadi kelemahan dari XML karena dokumen

XML menjadi lebih kompleks. Hal-hal lain dalam XML adalah :

- XML adalah extensible. Tidak seperti HTML yang telah memiliki tag yang pasti, XML memungkinkan kita untuk mendefinisikan beberapa tag sendiri jika diperlukan. Pada contoh diatas dokumen tersebut menggambarkan Customer namun juga dimasukan sebagai field untuk menggambarkan customer.
- XML adalah hierarchical. Elemen-elemen bisa saja memiliki subordinat dibawahnya. Sebagai contoh element Customer yang memiliki beberapa child element.
- XML adalah modular. XML dapat dikembangkan secara modular dengan memungkinkan sebuah dokumen untuk mereference dokumen lainnya.
- XML adalah bahasa pemrograman independent.
- XML menggunakan DTD (Document Type Definitions) dan XML skema. Validasi data pada XML menggunakan kedua konsep tersebut.

4.6 XML Parser dan Document Type Definition (DTD)

Untuk memproses dokumen XML dibutuhkan XML parser yakni sebuah program yang mampu melakukan decompose dokumen XML menjadi individual elemen. Ada dua kategori utama dari XML parser yakni Document Object Model (DOM) dan Simple API for XML (SAX). DOM adalah language neutral API untuk mengakses dan memodifikasi dokumen yang bercabang-cabang (tree base) seperti dokumen HTML dan XML, DOM parser bekerja pada memory. Sedangkan SAX parser adalah parser event base yang hanya digunakan untuk membaca dokumen XML. SAX bekerja dari registrasi event. SAX sebenarnya bekerja lebih sederhana dibandingkan dengan DOM hanya saja memiliki beberapa kekurangan yakni :

- Ketika dokumen XML dibaca maka tidak ada representasi dokumen di memory.
- SAX parser tidak mampu melakukan modifikasi dokumen XML.

Sesuai namanya DTD berfungsi untuk mendefinisikan tipe dokumen XML. Pada saat mempelajari salah satu bahasa pemrograman atau scripting anda diperkenalkan dengan deklarasi

variable, deklarasi fungsi dan deklarasi tipe data. Serupa dengan itu, DTD mendefinisikan struktur dokumen XML dengan daftar element yang digunakan.

DTD memungkinkan format yang unik untuk setiap file xml. DTD akan sangat berguna bila kita membuat aplikasi dalam Visual Basic, ASP atau bahasa pemrograman lain yang mendukung XML, yaitu untuk memastikan bahwa data yang diterima aplikasi itu adalah data yang valid. Atau bermanfaat juga digunakan bila satu organisasi menyepakati penggunaan satu DTD untuk tukar menukar data dan informasi. Unsur-unsur yang dideklarasikan dalam DTD adalah semua unsur yang membentuk suatu dokumen XML yaitu

- Element, satu blok data yang diawali tag pembuka dan tag penutup.
- Attribute, informasi pendukung element yang disertakan pada tag pembuka
- Entity, karakter pengganti untuk sekumpulan informasi yang didefinisikan

4.7 Data Elemen Objek dan Array

Resource string menyediakan string teks bagi aplikasi dengan pemformatan dan penataan gaya teks opsional. Ada tiga tipe resource yang bisa menyediakan string untuk aplikasi, diantaranya:

- Sumber daya XML yang menyediakan string tunggal.
- Sumber daya XML yang menyediakan larik string.
- Sumber daya XML yang membawa string berbeda untuk pluralisasi.

Semua string mampu menerapkan beberapa argumen markup penataan gaya dan pemformatan.

XML elemen adalah waktu dari (dan termasuk) tag awal sampai (dan termasuk) akhir label. Sebuah elemen dapat terdiri: unsur-unsur lain, teks, milik. elemen XML harus mengikuti aturan penamaan ini:

- Nama dapat berisi huruf, angka dan karakter lain
- Nama tidak dapat dimulai dengan angka atau tanda baca
- Nama tidak dapat dimulai dengan huruf xml (atau XML, Xml dll)
- Nama tidak dapat berisi spasi

Dokumen XML harus mengandung elemen root. Unsur adalah

induk dari semua elemen lainnya. elemen XML dalam dokumen untuk membentuk pohon dokumen. Pohon dari akar, dan meluas ke bagian bawah pohon. Contoh elemen root adalah < toko buku>. Dokumen semua <buku> elemen yang terkandung dalam < toko buku> di < Buku> elemen memiliki empat sub-elemen: <title>, <author>, <tahun>, <harga>, dapat dilihat pada Gambar 10.

```
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Gambar 10. Contoh XML membentuk struktur pohon
Elemen objek merupakan elemen dalam dokumen XML. Elemen dapat berisi atribut, unsur-unsur lain, atau teks. Jika elemen berisi teks, teks direpresentasikan dalam teks-node. Untuk elemen objek properti dapat dilihat pada Tabel 8.

Tabel 8. Elemen Object Properti

Attributes	Mengembalikan NamedNodeMap atribut untuk elemen
baseURI	Mengembalikan URI dasar mutlak dari elemen

childNodes	Mengembalikan NodeList node anak untuk elemen
firstChild	Mengembalikan anak pertama dari elemen
lastChild	Mengembalikan anak terakhir dari elemen
localName	Mengembalikan bagian lokal nama elemen
namespaceURI	Mengembalikan namespace URI dari elemen
nextSibling	Mengembalikan node segera setelah elemen
nodeName	Mengembalikan nama dari node, tergantung pada jenisnya
nodeType	Mengembalikan jenis node
ownerDocument	Mengembalikan elemen root (document object) untuk elemen
parentNode	Mengembalikan simpul induk dari elemen
prefix	Menetapkan atau mengembalikan awalan namespace elemen
previousSibling	Mengembalikan node segera sebelum elemen
schemaTypeInfo	Mengembalikan informasi jenis terkait dengan elemen
tagName	Mengembalikan nama elemen
textContent	Menetapkan atau mengembalikan isi teks dari elemen dan keturunannya



Rangkuman

XML kependekan dari eXtensible Markup Language, dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada bulan Februari 1998. Teknologi yang digunakan pada XML sebenarnya bukan teknologi baru, tapi merupakan turunan dari SGML yang telah dikembangkan pada awal 80-an dan telah banyak digunakan pada dokumentasi teknis proyek-proyek berskala besar. Ketika HTML dikembangkan pada tahun 1990, para penggagas XML mengadopsi bagian paling penting pada SGML dan dengan berpedoman pada pengembangan HTML menghasilkan markup language yang tidak kalah hebatnya dengan SGML.

Seperti halnya HTML, XML juga menggunakan elemen yang ditandai dengan tag pembuka (diawali dengan '<' dan diakhiri dengan '>'), tag penutup (diawali dengan '</' dan diakhiri '>') dan atribut elemen (parameter yang dinyatakan dalam tag pembuka misal <form name="isidata">). Hanya bedanya, HTML mendefinisikan dari awal tag dan atribut yang dipakai didalamnya, sedangkan pada XML kita bisa menggunakan tag dan atribut sesuai kehendak kita. XML untuk saat ini bukan merupakan pengganti HTML. Masing-masing dikembangkan untuk tujuan yang berbeda. Kalau HTML digunakan untuk menampilkan informasi dan berfokus pada bagaimana informasi terlihat, XML mendeskripsikan susunan informasi dan berfokus pada informasi itu sendiri. XML terutama dibutuhkan untuk menyusun dan menyajikan informasi dengan format yang tidak mengandung format standard layaknya heading, paragraph, table dan lain sebagainya. Sama dengan HTML, File XML berbentuk teks sehingga bila diperlukan kita bisa membacanya tanpa memerlukan bantuan software khusus. Hal ini memudahkan pengembang aplikasi yang menggunakan XML untuk mendebug programnya. XML lebih fleksible dibanding HTML dalam hal kemampuannya menyimpan informasi dan data. Pada XML kita bisa menyimpan data baik dalam atribut maupun sebagai isi elemen yang diletakkan diantara tag pembuka dan tag penutup. Sebuah dokumen XML terdiri dari bagian bagian yang disebut dengan node. Node-node itu adalah:

- Root node yaitu node yang melingkupi keseluruhan dokumen. Dalam satu dokumen XML hanya ada satu root node. Node-node yang lainnya berada di dalam root node.
- Element node yaitu bagian dari dokumen XML yang ditandai dengan tag pembuka dan tag penutup, atau bisa juga sebuah tag tunggal elemen kosong seperti `<anggota nama="budi"/>`. Root node biasa juga disebut root element
- Attribute note termasuk nama dan nilai atribut ditulis pada tag awal sebuah elemen atau pada tag tunggal.



Latihan Soal

1. Dari tabel data berikut, buatlah dalam bentuk XML:

nrp	nama	kota	kode pos
101	Faruq	Bogor	16132
102	Fakhri	Yogyakarta	55283
103	Nafisa	Malang	65148

2. Buat proses di php untuk membaca data XML di atas.
3. Dari file php diatas tambahkan juga bisa dimasukkan ke database dengan tabel seperti di atas (gunakan perintah insert)!
4. Untuk memahami bagaimana XML dan XSLT bekerja, perhatikan contoh berikut ini dan capture hasilnya:

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<berita>Saya sedang belajar XML</berita>
```

Simpanlah dokumen diatas dengan nama belajar.xml
XSLT yang diperlukan adalah sebagai berikut:

```

<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/TR/
WD-xsl">

    <xsl:template match ='/>

    <html>

    <head>

    <title>XML Saya yang Pertama</title>

    </head>

    <body>

    <p><xsl:value-of select="berita"/></p>

    </body>

    </html>

</xsl:template>

</xsl:stylesheet>

```

Simpanlah dokumen XSLT ini dengan nama belajar.xml, dan Bila menggunakan MSXML3 sebagai parser, maka harus ditambahkan satu baris heading standard sehingga dokumen XML akan menjadi seperti berikut:

```

<?xml version="1.0" encoding="iso-8859-1"?>
<?xml-stylesheet type="text/xsl" href="belajar.xml"
<berita>Saya sedang belajar XML</berita>

```

BAB V

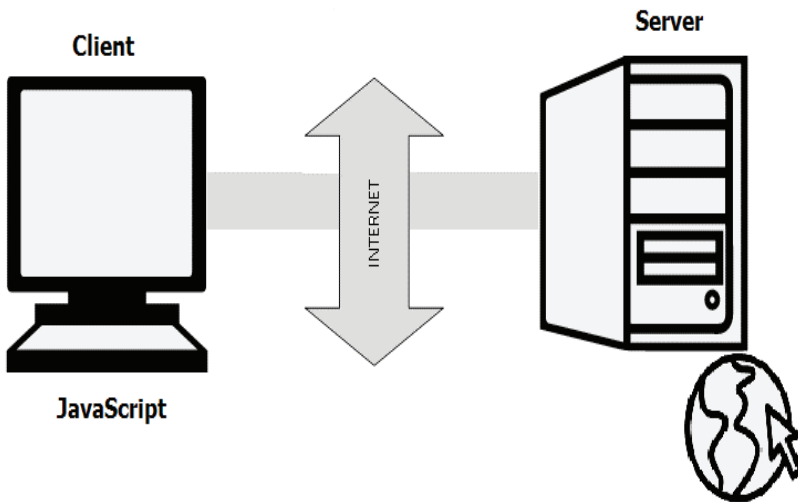
JavaScript

5.1 Pengertian Javascript

JavaScript adalah bahasa pemrograman yang populer. JavaScript saat ini sudah banyak dikembangkan menjadi beberapa pengetahuan lain, seperti AJAX. Asynchronous JavaScript dan XML (AJAX) adalah teknik yang digunakan pada halaman website dengan menggunakan bahasa pemrograman JavaScript. Metode yang diterapkan adalah menerima dan mengirimkan data dengan server tanpa harus menyegarkan (refresh) halaman itu. JavaScript sudah menjadi bahasa pemrograman yang baku. Maka dari itu, JavaScript bisa dikolaborasikan dengan banyak bahasa lain, seperti HTML, untuk membuat website yang mudah digunakan oleh pengguna. Selain itu, javascript dapat juga digunakan untuk Server, PC, Laptop, tablet dan lebih banyak lagi. Kode pemrograman javascript dapat disisipkan kedalam halaman HTML. Pada awalnya, JavaScript mulai diperkenalkan di browser Netscape Navigator 2. Namun waktu itu namanya bukan JavaScript, namun LiveScript. Mengingat pada waktu itu teknologi Java sedang populer, maka pihak Netscape memutuskan untuk mengganti namanya menjadi JavaScript, yang sepertinya nama tersebut lebih marketable dibandingkan LiveScript. Selanjutnya pihak Microsoft (rival Netscape) pun mulai ikut-ikutan memfasilitasi web browser buatannya, 'Internet Explorer', supaya bisa mendukung JavaScript. Namun mungkin karena gengsi, pihak Microsoft memberi nama bahasa yang lain, yaitu Jscript. Mulai saat itu, Netscape dan Microsoft mulai berlomba-lomba mengembangkan bahasa tersebut dalam versi yang berlainan. Oleh sebab persaingan itulah terkadang suatu JavaScript mungkin bisa bekerja dengan baik di browser Netscape, tapi tidak demikian halnya di IE, begitu pula sebaliknya.

JavaScript adalah bahasa skrip sisi klien yang sangat kuat. JavaScript digunakan terutama untuk meningkatkan interaksi pengguna dengan halaman web. Dengan kata lain, dengan

megggunakan javascript dapat membuat halaman web lebih hidup dan interaktif, dengan bantuan JavaScript. JavaScript bisa membuat berbagai fitur mulai dari yang sederhana sampai dengan yang paling kompleks, misalnya layout, galeri, tombol, causels, dan lain sebagainya. Jika dikembangkan lebih lanjut, JavaScript dapat digunakan untuk membuat baik animasi dua dimensi maupun tiga dimensi, bahkan aplikasi yang behubungan dengan database. JavaScript juga digunakan secara luas dalam pengembangan game dan pengembangan aplikasi Mobile.



Gambar 11. Klien dengan server

5.2 Penulisan Javascript

Bagi pemula yang belum memahami terkait javascript perlu diperhatikan bahwa terdapat dua jenis bagaimana javascript dibuat, pertama javascript ditulis dalam file yang terpisah dengan HTML, kedua javascript ditulis dalam HTML. Javascript yang ditulis diluar HTML disebut Eksternal Javascript dengan ekstensi file .js. Dalam HTML, penulisan script diawali dengan `<script> ... </script>`. Script yang akan dijalankan harus diletakkan diantara `<script>` dan `</script>`. Tag `<script>` memiliki beberapa atribut, namun yang terpenting adalah atribut `language` dan `type`. Karena Javascript bukan satu-satunya bahasa scripting, maka sangatlah perlu untuk memberitahukan kepada browser bahwa bahasa script

yang digunakan adalah Javascript dan selanjutnya browser akan menjalankan modul pendukung Javascript untuk memprosesnya. Sehingga untuk Javascript, pada tag <script> perlulah ditambahkan atribut berikut ini:

```
<script language="JavaScript" type="text/javascript">
```

Script dapat diletakkan di tag <body> dan atau di tag <head> pada bagian halaman HTML.

Contoh 1

```
<script language="JavaScript" type="text/javascript">  
alert("Belajar Javascript");  
</script>
```

Contoh 2

```
<!DOCTYPE html>  
<html>  
<body>  
...  
<script language="JavaScript" type="text/javascript">  
document.write("<h1> Belajar Javascript</h1>");  
</script>  
...  
</body>  
</html>
```

Sedangkan penulisan fungsi pada script dapat diletakkan didalam tag <head> dan <body>. Contoh penulisan fungsi javascript di dalam tag <head> pada halaman HTML. Fungsi akan dipanggil ketika tombol diklik.

Contoh 3

```
<!DOCTYPE html>
<html>
<head>
<script language="JavaScript" type="text/javascript">
function cobafungsi()
{
document.getElementById("coba").innerHTML="Belajar
membuat fungsi";
}
</script>
</head>
<body>
<h1>Halaman WEB</h1>
<p id="coba">A Paragraph</p>
<button type="button" onclick="cobafungsi()">Coba Fungsi</
button>
</body>
</html>
```

Contoh penulisan fungsi javascript di dalam tag <body> pada halaman HTML. Fungsi akan dipanggil ketika tombol diklik.

Contoh 4

```
<!DOCTYPE html>
<html>
<body>
<h1>Halaman Web</h1>
<p id="coba">A Paragraph</p>
<button type="button" onclick="cobafungsi()">Coba</button>
<script language="JavaScript" type="text/javascript">
Function cobafungsi()
{
document.getElementById("coba").innerHTML="Belajar Fungsi
Javascript";
}
</script>
</body>
</html>
```

Dalam javascript kita dapat pula menambahkan komentar. Komentar pada suatu script tidak akan dieksekusi oleh Javascript. Komentar ditambahkan pada script untuk memberikan penjelasan script atau membuat skrip tidak dieksekusi. Untuk memberikan komentar yang hanya satu baris gunakan `//`. Berikut contoh bagaimana memberi komentar single.

Contoh 5

```
// Write to a heading:  
document.getElementById("myH1").innerHTML="Selamat  
datang di Homepage Saya";  
// Write to a paragraph:  
document.getElementById("myP").innerHTML="Ini adalah  
paragraph saya.";
```

Contoh 6

```
var x=5; // mendeklarasikan x dan memasangkannya dengan 5  
var y=x+2; // mendeklarasikan y dan memasangkannya dengan  
x+2
```

Pada contoh diatas adalah pemberian komentar sigle atau satu baris. Untuk memberikan komentar lebih dari satu baris, gunakan `/*` dan `*/`. Jadi script yang terletak di antara `/*` dan `*/` akan dianggap sebagai komentar. Berikut contoh memberi komentar yang lebih dari satu baris.

Contoh 7

```
/*  
Kode berikut akan ditulis ke heading dan paragraph.  
Dan akan menampilkan halaman homepage  
*/  
document.getElementById("myH1").innerHTML="Selamat  
datang di Homepage Saya";  
document.getElementById("myP").innerHTML=" Ini adalah  
paragraph saya.";
```

5.3 Variabel dan Tipe Data dalam Javascript

Pada umumnya bahasa pemrograman, javascript memiliki tipe data dan variabel. Variabel pada javascript seperti kotak atau wadah yang digunakan untuk menyimpan informasi yang senantiasa dapat diload. Sedangkan tipe data terkait dengan jenis data atau nilai yang disimpan dalam variabel. Dalam javascript, setiap kali akan menggunakan variabel, langkah pertama yang harus dilakukan adalah mendeklarasikan keberadaan nama variabel. Hal ini perlu dilakukan karena dengan adanya deklarasi nama variabel, computer akan menyediakan beberapa bagian memori untuk menyimpan nilai pada nama variabel tersebut. Untuk mendeklarasikan variabel digunakan kata `var`. Dalam mendeklarasikan nama variabel ada beberapa aturan yang harus diperhatikan, yaitu :

1. Nama variabel harus dimulai dengan huruf
2. Nama variabel juga dapat dimulai dengan `$` dan `_`
3. Nama variabel adalah casesensitive (memperhatikan besar kecilnya huruf).
4. Jangan menggunakan Reserved Word atau kata tercadang sebagai nama variabel. Kata tercadang adalah kata yang sudah built in dalam javascript yang sudah bermakna khusus. Penggunaan kata tercadang ini akan mengakibatkan error.

Contoh 1

```
var namakota;  
namakota="Malang";
```

Contoh 2

```
var namakota="Malang";
```

Contoh 3

```
var namakota="malang", propinsi="Jawa Timur", kode="3",  
x=6.23;
```

Pada contoh 1 di atas, dideklarasikan variabel `namakota`. Pada awal pendeklarasian, nilai dari variabel `namakota` adalah `null` (kosong). Kemudian variabel `namakota` diberi nilai `Malang`.

Untuk memberikan nilai pada suatu variabel digunakan tanda petik dua (") apabila tipe datanya berupa string. Pada contoh 2 mendeklarasikan variabel `namakota` yang sekaligus memberi nilai pada variabel `namakota`. Contoh 3 di atas mendeklarasikan beberapa variabel sekaligus. Untuk mendeklarasikan beberapa variabel digunakan tanda koma (,) untuk memisahkan variabel satu dengan yang lainnya.

Tipe data pada javascript meliputi : String, Integer, Float, Array, Object dan Booleans. Tipe data string adalah data yang memuat karakter, misalnya "Malang". String adalah sebarang text yang ada di dalam tanda petik, baik petik ganda (") maupun petik tunggal ('). Tipe data integer dan float merupakan tipe data numerik. Dalam mendeklarasikan tipe data object digunakan tanda kurung kurawal (, ... -). Setiap property dalam tipe data object dipisahkan dengan menggunakan tanda koma (,). Tipe data Booleans terdiri dari dua nilai, yaitu `true` atau `false`. Berikut beberapa contoh penggunaan tipe data pada javascript.

Contoh 4

```
var namakota="Malang";  
var propinsi='Jawa Timur';  
var x1=34;  
var x2=3.14;  
var y=123e4;  
var x=true;  
// tipe data string  
// tipe data string  
// tipe data integer  
// tipe data float  
// tipe data integer  
// tipe data boolean
```

Contoh 5

```
//berikut beberapa cara mendeklarasikan Array  
var mobil=new Array();  
mobil[0]="Toyota";  
mobil[1]="Daihatsu";  
mobil[2]="Honda";  
var bulan=new Array("Januari","Febuari","Maret","April");  
var kampus=["UM","UNMUH","KANJURUHAN","UIN  
MAULANA MALIK","ITN"];
```

Contoh 6

```
var klien={nama:"Pamungkas", sex:"Laki-Laki", id:"5758"};
```

Pada contoh di atas dideklarasikan beberapa variabel dengan berbagai tipe data. Pada contoh 4 di atas dideklarasikan variabel dengan tipe data string, integer, float dan Boolean. Contoh 5 di atas mendeklarasikan tipe data jenis array, dan beberapa cara variasi penulisannya. Pendeklarasian variabel dengan tipe data object dicontohkan pada contoh 6.

5.4 Operator pada Javascript

Operator adalah simbol yang digunakan untuk melakukan operasi pada suatu nilai dan variabel. Operator dalam pemrograman terbagi dalam 6 jenis diantaranya aritmatika, assignment, perbandingan, logika, bitwise dan ternary.

1. Operator aritmatika

Operator aritmatika digunakan untuk melakukan operasi aritmatika antara variabel dan atau nilai. Untuk melakukan operasi perkalian, kita menggunakan simbol asterik *. Jangan gunakan x, karena simbol x bukan termasuk dalam operator di dalam pemrograman. Untuk pembagian, kita gunakan simbol garis miring /. Misal diberikan $y = 5$, tabel berikut menjelaskan tentang operator aritmatika.

Operator	Deskripsi	Contoh	Nilai X	Nilai Y
+	Penjumlahan	$x = y + 2$	7	5
-	Pengurangan	$x = y - 2$	3	5
*	Perkalian	$x = y * 2$	10	5
/	Pembagian	$x = y / 2$	2.5	5
%	Modulus (sisahasil bagi)	$x = y \% 2$	1	5
++	Increment	$x = ++y$	6	6
		$x = y++$	5	6

--	Decrement	$x = --y$	4	4
		$x = y--$	5	4

2. Operator Penugasan (Assignment)

Operator penugasan adalah operator yang digunakan untuk memberikan tugas kepada variabel. Biasanya digunakan untuk mengisi variabel atau pemberian nilai kepada variabel. Dalam contoh berikut, diberikan $x = 10$, dan $y = 5$. Tabel berikut menjelaskan operator pemberian nilai.

Operator	Deskripsi	Contoh	Sama dengan	Hasil
=	Pengisian Nilai	$x=y$		$x=5$
+=	Pengisian dan Penambahan	$x+=y$	$x = x+y$	$x=15$
-=	Pengisian dan Pengurangan	$x-=y$	$x = x-y$	$x=5$
=	Pengisian dan Perkalian	$x=y$	$x = x*y$	$x=50$
/=	Pengisian dan Pembagian	$x/=y$	$x = x/y$	$x=2$
%=	Pengisian dan Sisa bagi	$x\%=y$	$x = x\%y$	$x=0$

3. Opeartor relasi atau perbandingan

Operator pembanding digunakan dalam pernyataan logika untuk menentukan kesamaan atau perbedaan diantara nilai-nilai. Operator perbandingan akan menghasilkan sebuah nilai boolean true dan false. Diberikan nilai $X = 5$, tabel dibawah ini menjelaskan operator pembanding.

Operator	Deskripsi	Pembandingan	Balikan
<code>==</code>	Sama dengan	<code>x == 8</code> <code>x == 5</code>	False True
<code>===</code>	Sama persis dengan (nilai dan tipe data)	<code>x === "5"</code> <code>x === 5</code>	False True
<code>!=</code>	Tidak sama dengan	<code>x != 8</code>	True
<code>!==</code>	Tidak sama dengan (nilai atau tipe data)	<code>x != "5"</code> <code>x != 5</code>	True False
<code>></code>	Lebih besar dari	<code>x > 8</code>	False
<code><</code>	Lebih kecil dari	<code>x < 8</code>	True
<code>≥</code>	Lebih besar sama dengan	<code>x ≥ 8</code>	False
<code>≤</code>	Lebih kecil sama dengan	<code>(x ≤ 8)</code>	True

4. Operator Logika

Operator logika digunakan untuk melakukan operasi terhadap dua nilai boolean. Operator ini berfungsi untuk menunjukkan nilai kebenaran antara beberapa variabel atau beberapa nilai. Diberikan nilai $X = 6$ dan $Y = 3$, tabel dibawah menjelaskan operator logika.

Operator	Deskripsi	Pembandingan	Balikan
&&	Dan	$(x < 10 \ \&\& \ y > 1)$	True
	Atau	$(x == 5 \ \ y == 5)$	False
!	N e g a s i / ingkaran	$!(x == y)$	True

5. Operator Bitwise

Operator bitwise merupakan operator yang digunakan untuk operasi berdasarkan bit (biner). Operator ini berlaku untuk tipe data int, long, short, char, dan byte. Operator ini akan menghitung dari bit-ke-bit. Misalnya, kita punya variabel a = 60 dan b = 13. Bila dibuat dalam bentuk biner, akan menjadi a = 00111100 dan b = 00001101

Operator	Deskripsi	Contoh	Hasil
&	AND	$a \& b$	00001100
	OR	$a b$	00111101
^	XOR	$a \wedge b$	00110001
~	Negasi/Kebalikan	$\sim a$	11000011
<<	Left Shift		
>>	Right Shift		
<<<	Left Shift (Unsigned)		
>>>	Right Shift (Unsigned)		

6. Operator Ternary

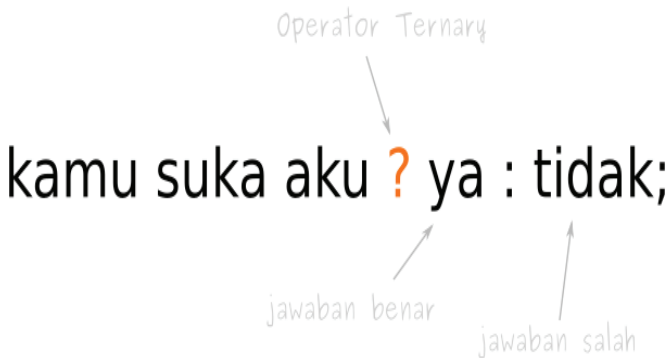
Operator ternary merupakan operator yang terdiri dari tiga bagian. Operator-operator sebelumnya hanya dua bagian saja, yaitu: bagian kiri dan kanan. Ini disebut operator binary. Sementara operator trinary ada bagian kiri, tengah, dan kanan.

bagian kiri <operator> bagian tengah <operator> bagian kanan

Operator ternary pada Javascript, biasanya digunakan untuk membuat sebuah percabangan if/else. Simbol operator ternary terdiri dari tanda tanya dan titik dua (? :). Bentuknya seperti ini:

<kondisi> ? "benar" : "salah"

Perhatikan! <kondisi> dapat kita isi dengan ekspresi yang menghasilkan nilai true dan false. Apabila kondisi bernilai true, maka “benar” yang akan dipilih dan sebaliknya—apabila false—maka “salah” yang akan dipilih.



Gambar 12. Penjelasan Operator Ternary

Pada contoh di atas, “Kamu suka aku” adalah pertanyaan atau kondisi yang akan diperiksa. Kalau jawabannya benar, maka iya. Sebaliknya akan tidak.

Selain keenam jenis operator yang telah dijabarkan diatas terdapat operator lain diantaranya:

7. Operator Bersyarat

Javascript juga memuat operator bersyarat yang memberikan suatu nilai ke suatu variabel berdasarkan kondisi yang sama.

Syntaxnya :

```
NamaVariabel=(syarat)?nilai 1: nilai 2
```

Contoh :

```
<!DOCTYPE html>
<html>
<body>
<p>Klik tombol untuk memeriksa usia</p>
Usia:<input id="usia" value="18" />
<p>Apakah Usianya mencukupi?</p>
<button onclick="myFunction()">Coba Cek Usia</button>
<p id="coba"></p>
<script>
function myFunction()
{
var usia,periksa;
usia=document.getElementById("usia").value;
periksa=(usia<18)?"Terlalu Muda":"Usia memenuhi";
document.getElementById("coba").innerHTML=periksa;
}
</script>
</body>
</html>
```

8. Operator + yang digunakan pada tipe data String

Operator + juga dapat digunakan untuk menambahkan variabel bertipe data string atau nilai text.

Contoh

```
//untuk menambahkan dua atau lebih variabel tipe String,  
gunakan operator +  
txt1="Selamat Datang";  
txt2="Di Jurusan Matematika";  
txt3=txt1+" "+txt2;  
Hasil dari script di atas adalah :  
Selamat Datang Di Jurusan Matematika
```

Contoh

```
//menambahkan String dan Bilangan  
x=5+5;  
y="5"+5;  
z="angka "+5;  
Hasil dari script di atas adalah :  
10  
55  
angka 5
```

5.5 Fungsi pada Javascript

Fungsi adalah suatu blok kode yang dijalankan ketika ada script yang memanggil fungsi tersebut. Fungsi merupakan bagian subprogram yang melaksanakan suatu tugas tertentu ketika diperlukan. Fungsi ini dapat dipanggil dengan cara menuliskan nama fungsinya saja, dan disertai dengan parameter apabila ada. Jadi parameter dalam suatu fungsi bersifat opsional. Suatu fungsi ditulis sebagai blok kode dengan syntax sebagai berikut :

```
function namafungsi(parameter)  
{  
  Beberapa code untuk dijalankan  
}
```


Contoh 1

```
<html>
<head>
</head>
<body>
  <script language="JavaScript" type="text/javascript">
    function diskon(hargaAwal)
    {
      var diskon;
      diskon = 0.5 * hargaAwal;
      hargaDiskon = hargaAwal - diskon;
      return hargaDiskon;
    }
    var harga = prompt("Masukkan harga mula-mula (Rp) : ");
    document.write("Harga stlh diskon 50% : Rp. " +
diskon(harga));
  </script>
</body>
</html>
```

Penjelasan dari script di atas ialah Bermula user diminta memasukkan harga mula-mula, harga ini diassign ke dalam variabel harga, misalnya 1000. Selanjutnya script mencetak harga setelah didiskon. Dalam hal ini, script akan memanggil function diskon(harga) atau diskon(1000). Karena function diskon memiliki parameter hargaAwal, maka nilai dari hargaAwal ini sama dengan nilai dari variabel harga, pada contoh ini adalah bernilai 1000. Selanjutnya akan dihitung nilai diskon yaitu $0.5 * 1000 = 500$, dan hargaDiskon yaitu $1000 - 500 = 500$. Perintah return hargaDiskon, artinya akan mengembalikan nilai hargaDiskon (500) ke function yang memanggilnya, yaitu diskon(harga), lihat tanda panah pada script di atas. Sehingga akan tercetak 'Harga stlh diskon 50% : Rp. 500'. Suatu function tidak harus terdapat return value. Perhatikan contoh berikut ini.

Contoh 2

```
<html>
<head>
  <script language="JavaScript" type="text/javascript">
    function cetakTeks(teks)
    {
      document.write(teks + "<br>");
    }
  </script>
</head>
<body>
  <script language="JavaScript" type="text/javascript">
    var kata = "Hallo";
    var i;
    for(i=0; i<=5; i++)
    {
      cetakTeks(kata);
    }
  </script>
</body>
</html>
```

5.6 Pernyataan Bersyarat

Pernyataan bersyarat digunakan untuk menampilkan aksi yang berbeda berdasarkan kondisi yang berbeda. Pernyataan bersyarat dalam Javascript diantaranya if, if ... else ..., if ... else if ..., dan switch.

1. Pernyataan if

Pernyataan if digunakan untuk mengeksekusi code apabila syarat yang ditentukan di penuhi (bernilai benar). Pernyataan if memiliki syntax sebagai berikut :

```
if (syarat)
{
  Kode yang dieksekusi apabila syarat dipenuhi
}
```

Contoh 1

```
<html>
<body>
<script language="JavaScript" type="text/javascript">
var bil;
var status;
bil = prompt("Masukkan sebarang bilangan : ");
if (bil > 10)
{
status = "Anda telah memasukkan bilangan lebih besar dari
10";
}
document.write(status);
</script>
</body>
</html>
```

2. Pernyataan if ... else

Pernyataan if ... else digunakan untuk mengeksekusi beberapa kode apabila syarat yang ditentukan terpenuhi dan untuk kode apabila syarat tidak dipenuhi. Syntax dari pernyataan if ... else ... sebagai berikut:

```
if (syarat)
{
Kode yang dieksekusi apabila syarat dipenuhi
}
Else
{
Kode yang dieksekusi apabila syarat tidak dipenuhi
}
```

Contoh 2

```
<html>
<body>
<script language="JavaScript" type="text/javascript">
var tanggal;
var status;
tanggal = prompt("Masukkan tanggal hari ini : ");
if (tanggal > 15)
{
status = "Tanggal " + tanggal + " adalah tanggal tua";
}
else
{
status = "Tanggal " + tanggal + " adalah tanggal muda";
}
document.write(status);
</script>
</body>
</html>
```

3. Pernyataan if ... else if ...

Pernyataan if ... else if ... else ... digunakan untuk mengeksekusi satu dari beberapa kode apabila syarat yang ditentukan terpenuhi dan untuk kode apabila syarat tidak dipenuhi. Syntax dari pernyataan if ... else ... sebagai berikut:

```
if (syarat 1)
{
Kode yang dieksekusi apabila syarat 1 dipenuhi
}
else if (syarat 2)
{
Kode yang dieksekusi apabila syarat 2 dipenuhi
}
else
{
Kode yang dieksekusi apabila syarat 1 dan syarat 2 tidak
dipenuhi
}
```

Contoh 3

```
<html>
<body>
<script language="JavaScript" type="text/javascript">
var bil;
var status;
bil = prompt("Masukkan sebarang bilangan : ");
if (bil > 10)
{
status = "Anda telah memasukkan bilangan lebih besar dari
10";
}
else if (bil < 10)
{
status = "Anda telah memasukkan bilangan lebih kecil dari 10";
}
else
{
status = "Anda telah memasukkan bilangan sama dengan 10";
}
document.write(status);
</script>
</body>
</html>
```

4. Pernyataan Switch

Pernyataan switch memiliki fungsi yang sama dengan pernyataan if ... else if ... else ... Syntax dari pernyataan ini adalah sebagai berikut :

```
Switch(variabel)
{
case option1 : statement 1
break;
case option2 : statement 2
break;
case option3 : statement 3
break;
...
...
...
default : statement x
break;
}
```

Proses dari SWITCH di atas adalah melakukan cek pada value variabel. Apabila valuenya sama dengan option1 maka statement yang dilakukan adalah statement 1. Apabila valuenya sama dengan option2 maka statement 2 yang dijalankan, begitu seterusnya. Namun apabila tidak ada value yang sama, maka statement x lah yang dijalankan. Tipe data dari variabel yang akan dicek haruslah ordinal, misalnya integer. Dalam hal ini tipe data riil tidak dapat digunakan. Untuk memastikan bahwa nilai yang akan dicek pada variabel adalah bertipe integer, sebaiknya gunakan perintah parseInt().

Contoh 4

```
<html>
<body>
<script language="JavaScript" type="text/javascript">
var nilaiAngka;
var nilaiHuruf;

nilaiAngka = parseInt(prompt("Masukkan sebuah bil
bulat 1 s/d 5 : "));
switch(nilaiAngka)

{
case 1 : document.write("Anda memasukkan bilangan
1");
        break;

case 2 : document.write("Anda memasukkan bilangan
2");

case 3 : document.write("Anda memasukkan bilangan
3");
        break;

case 4 : document.write("Anda memasukkan bilangan
4");
        break;
case 5 : document.write("Anda memasukkan bilangan
5");
        break;
default:document.write("Bilangan yang Anda
masukkan tidak 1 s/d 5");
        break;
```

```
}  
</script>  
</body>  
</html>
```

Contoh 5

```
<!DOCTYPE html>  
<html>  
<body>  
<p>Klik Tombol untuk mengetahui hari ini.</p>  
<button onclick="fungsihari()">Cek Hari</button>  
<p id="hari"></p>  
<script>  
function fungsihari()  
{  
  var x;  
  var d=new Date().getDay();  
  switch (d)  
  {  
    case 0:  
      x="Hari ini adalah hari Minggu";  
      break;  
    case 1:  
      x="Hari ini adalah hari Senin";  
      break;  
    case 2:  
  
      x="Hari ini adalah hari Selasa";  
      break;  
    case 3:  
      x="Hari ini adalah hari Rabu";  
      break;  
    case 4:  
      x="Hari ini adalah hari Kamis";  
      break;  
    case 5:  
      x="Hari ini adalah hari Jumat";  
      break;  
    case 6:  
      x="Hari ini adalah hari Sabtu";  
      break;  
  }  
}
```

```
document.getElementById("hari").innerHTML=x;  
}  
</script>  
</body>  
</html>
```

5.7 Struktur Perulangan

Perulangan adalah suatu blok perintah atau kode yang diulang selama syarat bernilai True. Dalam javascript, perulangan ini dapat dinyatakan dalam 2 bentuk, yaitu dengan For dan While. Pada bagian ini akan dijelaskan penggunaan For untuk perulangan terlebih dahulu.

1. Perulangan FOR

Perulangan For sering digunakan untuk membuat perulangan dengan jumlah perulangan telah diketahui secara pasti. Syntax dari perulangan for adalah sebagai berikut :

```
For (statement 1; statement 2; statement 3)  
{  
Kode yang dieksekusi apabila syarat dipenuhi  
}
```

Keterangan :

Statement 1 akan dieksekusi sebelum perulangan dimulai

Statement 2 mendefinisikan kondisi untuk menjalankan perulangan

Statement 3 dieksekusi setiap kali kode blok selesai dieksekusi.

Contoh 1

```
<html>
<body>
<script language="JavaScript" type="text/javascript">
var counter;
for(counter=1; counter<=5; counter++)
{
    document.write("<font size=" + counter + ">
        Ini adalah font berukuran " + counter +
    "</font><br>");
}
</script>
</body>
</html>
```

Contoh 2

```
<html>
<body>
<script language="JavaScript" type="text/javascript">
var counter;

for(counter=5; counter>=1; counter--)
{
    document.write("<font size=" + counter + ">Ini adalah font
berukuran " + counter + "</font><br>");
}
</script>
</body>
</html>
```

Dari contoh 1 dan contoh 2 di atas, dapat disimpulkan bahwa perulangan tidak selalu dalam bentuk increment, akan tetapi juga dapat dalam bentuk decrement. Selain itu kenaikan atau penurunannya tidak selalu naik 1 atau turun 1. Perhatikan contoh berikut ini :

Contoh 3

```
<html>
<body>
<script language="JavaScript" type="text/javascript">
var counter;
for(counter=1; counter<=5; counter+=2)
{
    document.write("<font size='" + counter + '">
        Ini adalah font berukuran " + counter +
"/font><br>");
}
</script>
</body>
</html>
```

2. Perulangan While

Pada prinsipnya bentuk perulangan While adalah mirip dengan perulangan For. Secara umum sintaks dari perulangan While adalah

```
For (statement 1; statement 2; statement 3)
{
    Kode yang dieksekusi apabila syarat dipenuhi
}
```

Dalam sintaks di atas, bagian statement akan terus diulang selama syarat bernilai True. Sebagai contoh, perhatikan statement dalam For dan While berikut ini. Kedua statement adalah ekuivalen atau sama (perhatikan dengan warna sama).

```
for(counter=1; counter<=5; counter++)  
{  
  document.write("Hello World <br>");  
}
```

```
-----  
For (statement 1; statement 2; statement 3)  
{  
  Kode yang dieksekusi apabila syarat dipenuhi  
}  
  
counter = 1;  
while(counter<=5)  
{  
  document.write("Hello World <br>");  
  counter++;  
}
```

Meskipun semua statement For dapat juga dinyatakan sebagai While, namun tidak berlaku sebaliknya. Tidak semua statement While dapat dinyatakan dalam For. Inilah kelebihan perulangan While. Kelebihan While dibandingkan For adalah bahwa while dapat digunakan apabila jumlah perulangan tidak diketahui. Sedangkan untuk For harus diketahui jumlah perulangannya. Sebagai contoh perumpamaan, misalkan ada statement begini: “ Saya akan terus makan selama saya masih lapar”. Statement tersebut adalah juga termasuk konsep perulangan dalam kehidupan sehari-hari. Dalam hal ini, jumlah berapa kali harus makan tidak diketahui. Melainkan hanya syarat makannya saja yang diketahui, yaitu selama masih lapar. Perhatikan contoh berikut ini. Script ini akan terus meminta user untuk mengisi suatu angka selama angka yang dimasukkan bukan angka 5.

Contoh 4

```
<html>
<body>
<script language="JavaScript" type="text/javascript">
var angka;
while(angka != 5)
{
angka = prompt("Masukkan sebuah angka : ");
}
document.write("Anda telah memasukkan angka 5");
</script>
</body>
</html>
```

Perhatikan syarat (yang dicetak biru) pada script di atas. Syarat tersebut bermakna, selama nilai angka tidak sama dengan 5, maka user akan selalu diminta memasukkan angka. Sehingga dalam kasus ini tidak diketahui berapa jumlah perulangan yang harus dilakukan, namun perulangannya tergantung dari syarat. Untuk kasus seperti ini, kita tidak bisa membuatnya dengan FOR.

5.8 Form

Pada bagian ini akan dipaparkan bagaimana memproses suatu input data yang berasal dari komponen form, seperti textbox, textarea, combo box, radio, button, dan lain lain. Pada dasarnya untuk semua tag komponen form, disisipkan dalam tag form.

```
document.namaform.namakomponen.value;
```

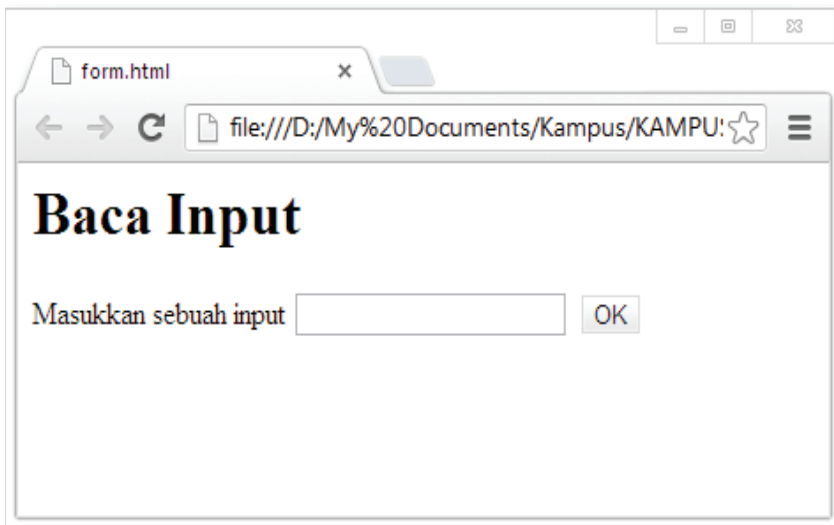
Sebagai contoh, script berikut ini adalah script untuk membaca nilai yang diinputkan user melalui komponen text lalu menampilkan sebagai alert.

Contoh 1

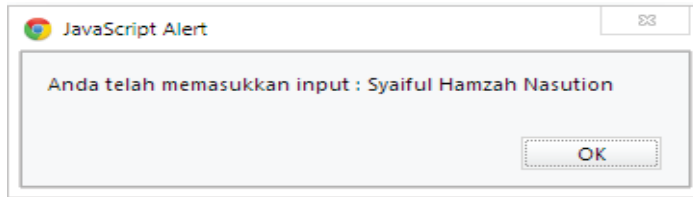
```
<html>
<head>
<script language="JavaScript" type="text/JavaScript">
function baca()
{
    var nilai = document.formku.input.value;
    alert("Anda telah memasukkan input : " + nilai);
}

</script>
</head>
<body>
    <h1>Baca Input</h1>
    <form name="formku">
        Masukkan sebuah input <input type="text" name="input">
        <input type="button" name="button" value="OK"
onclick="baca()">
    </form>
</body>
</html>
```

Apabila script di atas dijalankan, maka akan muncul tampilan seperti berikut :



Gambar 13. Tampilan form dengan masukan user



Gambar 14. Hasil tampilan setelah user melakukan inputan

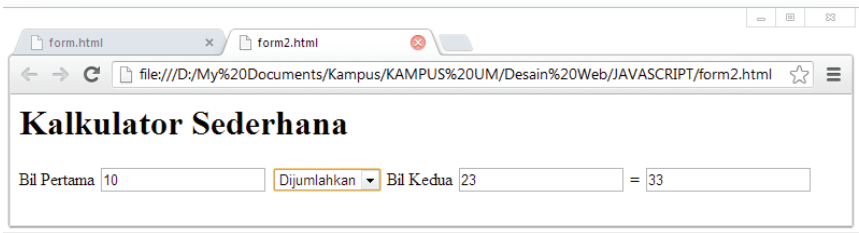
Selanjutnya, perhatikan contoh di bawah ini. Pada contoh ini, akan dibuat semacam kalkulator sederhana yang mengoperasikan penjumlahan, pengurangan, perkalian dan pembagian dari dua bilangan.

Contoh 2

```
<html>
<head>
<script language="JavaScript" type="text/JavaScript">
function hitung()
{
    var bilangan1 = parseFloat(document.formku.bil1.value);
    var bilangan2 = parseFloat(document.formku.bil2.value);

    var op = document.formku.operasi.value;
    var hasil;
    if (op == "jumlah") hasil = bilangan1 + bilangan2;
    else if (op == "kurang") hasil = bilangan1 - bilangan2;
    else if (op == "kali") hasil = bilangan1 * bilangan2;
    else if (op == "bagi") hasil = bilangan1 / bilangan2;
    document.formku.hasil.value = hasil;
}
</script>
</head>
<body>
<h1>Kalkulator Sederhana</h1>
<form name="formku">
    Bil Pertama <input type="text" name="bil1">
    <select name="operasi" onchange="hitung()">
        <option value="jumlah">Dijumlahkan</option>
        <option value="kurang">Dikurangkan</option>
        <option value="kali">Dikalikan</option>
        <option value="bagi">Dibagi</option>
    </select>
    Bil Kedua <input type="text" name="bil2"> =
    <input type="text" name="hasil">
</form>
</body>
</html>
```

Berikut tampilan dari script di atas



Gambar 15. Tampilan kalkulator sederhana

5.9 Event Handler

Pada pembahasan sebelumnya telah diberikan beberapa contoh script yang telah menggunakan even handler onclick() yaitu apabila suatu elemen diklik maka akan terjadi suatu efek/action tertentu. Apakah ada event yang selain onclick()? Jawabannya jelas ada, berikut beberapa event yang dapat kita gunakan.

Mouse Event	Description
onclick	Muncul ketika user mengklik elemen/komponen
ondblclick	Muncul ketika user mengklik ganda pada elemen/komponen
onmousedown	Muncul ketika user menekan pada mouse
onmousemove	Muncul ketika user menggerakkan pointer mouse
onmouseout	Muncul ketika user menggerakkan pointer mouse keluar dari elemen/komponen control
onmouseover	Muncul ketika user menggerakkan pointer mouse melalui/melewati elemen
onmouseup	Muncul ketika user melepas tombol mouse

keyboard Event	Description
onkeydown	Muncul ketika user menekan suatu tombol pada keyboard
onkeypress	Muncul ketika user menekan suatu tombol keyboard dan akan terus muncul sampai user melepas tombol
onkeyup	Muncul ketika user melepas tombol yang telah ditekan

HTML Control Event	Description
onblur	Muncul ketika HTML control kehilangan focus
onchange	Muncul ketika HTML control kehilangan focus dan nilainya berubah
onfocus	Muncul ketika HTML control sedang focus
onreset	Muncul ketika user menekan tombol reset dalam form
onselect	Muncul ketika user memilih teks pada HTML control
onsubmit	Muncul ketika user melakukan submit form

Window Event	Description
onload	Muncul ketika window telah selesai loading
onresize	Muncul ketika user mengubah ukuran window
onunload	Muncul ketika user keluar dari halaman/dokumen web

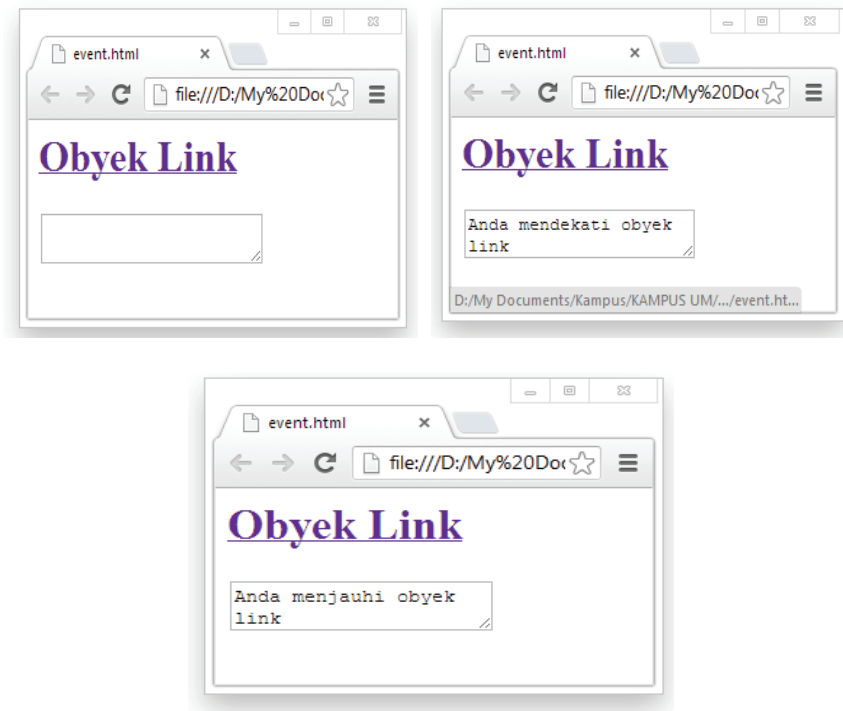
Contoh 1

```

<html>
<head>
<script language="JavaScript" type="text/JavaScript">
function dekati()
{
    document.formku.status.value = "Anda mendekati obyek
link";
}
function jauhi()
{
    document.formku.status.value = "Anda menjauhi obyek
link";
}
</script>
</head>
<body>
    <h1><a href="" onmouseover="dekati()"
onmouseout="jauhi()">
Obyek Link</a>
    </h1>
    <form name="formku">
        <textarea name="status"></textarea>
    </form>
</body>
</html>

```

Berikut tampilan dari script di atas



Gambar 16. Penerapan event handler



Rangkuman

JavaScript adalah bahasa skrip sisi klien yang sangat kuat. JavaScript digunakan terutama untuk meningkatkan interaksi pengguna dengan halaman web. Dengan kata lain, dengan menggunakan javascript dapat membuat halaman web lebih hidup dan interaktif, dengan bantuan JavaScript. JavaScript bisa membuat berbagai fitur mulai dari yang sederhana sampai dengan yang paling kompleks, misalnya layout, galeri, tombol, causels, dan lain sebagainya. Jika dikembangkan lebih lanjut, JavaScript dapat digunakan untuk membuat baik animasi dua dimensi maupun tiga dimensi, bahkan aplikasi yang berhubungan

dengan database. JavaScript juga digunakan secara luas dalam pengembangan game dan pengembangan aplikasi Mobile. Dalam javascript terdapat dua jenis penuliasannya, pertama javascript ditulis dalam file yang terpisah dengan HTML, kedua javascript ditulis dalam HTML. Javascript yang ditulis diluar HTML disebut Eksternal Javascript dengan ekstensi file .js. Dalam HTML, penulisan script diawali dengan `<script> ... </script>`. Script yang akan dijalankan harus diletakkan diantara `<script>` dan `</script>`. Dalam pendeklarasian variabel didalam javascript digunakan kata `var`. Dalam mendeklarasikan nama variabel ada beberapa aturan yang harus diperhatikan, yaitu : (1) Nama variabel harus dimulai dengan huruf, (2) Nama variabel juga dapat dimulai dengan `$` dan `_`, (3) Nama variabel adalah casesensitive (memperhatikan besar kecilnya huruf), (4) Jangan menggunakan Reserved Word atau kata tercadang sebagai nama variabel. Kata tercadang adalah kata yang sudah built in dalam javascript yang sudah bermakna khusus. Penggunaan kata tercadang ini akan mengakibatkan error. Sedangkantipe data pada javascript meliputi : String, Integer, Float, Array, Object dan Booleans.

Operator adalah simbol yang digunakan untuk melakukan operasi pada suatu nilai dan variabel. Operator dalam pemrograman terbagi dalam 6 jenis diantaranya aritmatika, assignment, perbandingan, logika, bitwise dan ternary. Selain keenam jenis operator yang telah dijabarkan diatas terdapat juga operator bersyarat dan operator `+` yang digunakan pada tipe data string. Fungsi dalam javascript dapat dipanggil dengan cara menuliskan nama fungsinya saja, dan disertai dengan parameter apabila ada. Jadi parameter dalam suatu fungsi bersifat opsional. Fungsi merupakan bagian subprogram yang melaksanakan suatu tugas tertentu ketika diperlukan. Sama halnya dengan bahasa pemrograman yang lainnya, didalam javascript terdapat pernyataan bersyarat (kondisi) dan juga struktur perulangan. Dalam melakukan interaksi dengan user javascript menyediakan form. Komponen form seperti textbox, textarea, combo box, radio, button, dan lain lain. Pada dasarnya untuk semua tag komponen form, disisipkan dalam tag form



Latihan Soal

1. Buatlah satu aplikasi web untuk mengkonversi suhu dengan menggunakan fungsi
2. Buatlah satu aplikasi web untuk menampilkan besar diskon dari pembelian barang. Inputan berupa : harga barang dan besar diskon.
3. Buatlah script Javascript untuk mengkonversi nilai angka ke nilai huruf dengan aturan sebagai berikut :

80 – 100 : A

70 – 79 : B

60 – 69 : C

50 – 59 : D

0 – 49 : E

Apabila nilai yang dimasukkan tidak dalam rentang 0 – 100, maka akan muncul keterangan nilai yang dimasukkan tidak valid.

4. Buatlah script Javascript untuk menampilkan hari dan tanggal. Contoh tampilan : Hari ini adalah Kamis, 23 Juli 2020.

BAB VI

Bootstrap

6.1 Pengertian Bootstrap

Bootstrap merupakan sebuah library framework CSS yang telah dibuat khusus untuk mengembangkan front end sebuah website. Bootstrap juga dikenal sebagai salah satu framework CSS, HTML, Javascript yang begitu populer di kalangan website developer atau pengembang website. Framework atau dalam bahasa Indonesia dapat diartikan sebagai “kerangka kerja” merupakan kumpulan dari fungsi-fungsi/prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau class dari awal. Alasan mengapa menggunakan Framework : (1) Mempercepat dan mempermudah pembangunan sebuah aplikasi web, (2) Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah framework (dengan syarat programmer mengikuti pola standar yang ada), (3) Umumnya framework menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, ORM, pagination, multiple database, scaffolding, pengaturan session, error handling, dll) (4) Lebih bebas dalam pengembangan jika dibandingkan CMS

Salah satu manfaat bootstrap adalah membuat halaman website bisa menyesuaikan dengan ukuran monitor device. Baik jika di akses lewat ponsel, tablet ataupun desktop. Awal mulanya, bootstrap sendiri bernama Twitter Blueprint, dikembangkan oleh Mark Otto dan Jacob Thornton di Twitter sebagai framework untuk mendorong konsistensi di lingkungan aplikasi internal mereka. Sebelum Bootstrap, berbagai perpustakaan digunakan untuk pengembangan antarmuka, yang menyebabkan ketidakonsistenan dan beban pemeliharaan yang tinggi. Sebagai pengguna ketika ingin menggunakan bootstrap yang dilakukan hanya perlu memanggil setiap kelas yang digunakan, contohnya

seperti tipografi, form, button, tabel, navigasi, dropdown, alert, modal, tab, accordion, carousel dan banyak ekstensi JavaScript lainnya. Keuntungan terbesar menggunakan Bootstrap adalah dilengkapi dengan perangkat gratis untuk membuat layout web yang fleksibel dan responsif serta komponen antarmuka yang umum. Selain itu, dengan menggunakan API dari Bootstrap yang kita gunakan dapat membuat komponen antarmuka tingkat lanjut seperti Scrollspy dan Typeaheads tanpa menulis satu baris JavaScript. Banyak fungsi bootstrap yang bisa dipakai untuk sebuah website. Berikut fungsinya:

1. Bisa mempercepat waktu untuk memproses pembuatan front end sebuah website.
2. Menampilkan sisi website yang lebih modern dan juga khas anak jaman sekarang.
3. Tampilan dari bootstrap sendiri sudah sangat responsive sehingga sangat mendukung untuk segala jenis resolusi, entah itu tablet, smartphone ataupun juga PC dan laptop.
4. Website yang menggunakan bootstrap umumnya lebih ringan karena lebih terstruktur.

Setelah dijabarkan tentang fungsi dari bootstrap, di bawah ini perlu diketahui juga keunggulan menggunakan bootstrap ini. Bagi pengguna yang ingin menggunakannya, berikut beberapa keunggulan yang ditawarkan:

1. Mudah digunakan
Bootstrap sangat mudah digunakan. Siapa saja yang memiliki pengetahuan dasar tentang HTML dan CSS dapat memulai pengembangan dengan Bootstrap.
2. Kompatibel pada semua browser
Bootstrap dibuat dengan browser modern dan kompatibel dengan semua browser populer seperti Mozilla Firefox, Google Chrome, Safari, Internet Explorer, dan Opera.
3. Open Source
Gratis untuk diunduh dan digunakan
4. Lebih efisien waktu
Bootstrap sendiri menyediakan banyak sekali library yang isinya berbagai macam script yang sudah siap saji untuk dipakai oleh programmer ketika mengembangkan website bagian depan. Sehingga pihak developer tidak lagi membuat

waktu untuk menulis berbagai script SCC dari awal. Tentu dengan memakai potongan script tersebut, maka akan sangat menguntungkan untuk website. Struktur website bisa lebih produktif di tangan seorang programmer handal tadi. Perlu diketahui bahwa bootstrap versi terbaru sudah dibangun dengan fitur LESS. Yang artinya, bootstrap mempunyai banyak aspek menguntungkan dalam hal website desain sehingga sangat mempermudah para programmer.

5. Lebih fleksibel

Penggunaan bootstrap website juga membuat pekerjaan developer dan programmer bisa lebih fleksibel. Dikarenakan framenya memang bisa digunakan sesuai keinginan dan bebas saja. Pengembang bisa lebih mudah untuk memodifikasikan berbagai script yang telah ada didalamnya. Tidak ada ketentuan khusus untuk memanfaatkan bootstrap twitter ini. Hal inilah yang membuatmu lantas bisa berkreasi dalam bentuk sesuai keinginan dengan memanfaatkan berbagai macam tool yang sudah ada di bootstrap twitter itu sendiri. Inilah yang membut para pengembang lebih fleksibel ingin membangun website yang ia miliki.

6. Desain yang oriented

Bootstrap untuk keperluan website umumnya mempunyai sebuah sistem baku yang sangat bagus, sistem ini disebut dengan nama grid sistem. Grid merupakan sebuah struktur dua dimensi yang menjadi perpaduan antara sumbu vertikal dan juga horizontal yang membentuk kolom serta baris. Kolom dan baris tersebut lantas merepresentasikan dimensi yang telah tersedia di sebuah browser. Untuk versi awalnya, bootstrap ini menyediakan grid yang ukuran layarnya adalah 940px dan juga margin 20px. Namun untuk era sekarang, ukuran bootstrap sendiri juga sudah mempunyai ukuran yang cukup lebar karena layarnya 768px sampai 1200px. Tentu saja ukuran tersebut bisa dimodifikasi oleh para penggunanya sendiri. Model untuk mengatur desainnya memang diatur dengan sistem yang sudah otomatis. Sehingga para developer hanya perlu melakukan pemanggilan kelas CSS dan juga meletakkan konten di script dan juga sintaks HTML yang telah dikerjakan sebelumnya.

7. Less is more

Ini merupakan hal yang begitu populer di kalangan programmer dan juga web developer karena alasan tertentu. Alasan tadi adalah kaitannya dengan fungsi responsive yang dimiliki oleh bootstrap itu sendiri. Dengan memakai bootstrap ini website yang dimiliki juga lebih responsive dan bisa ditampilkan untuk semua website dengan baik.

8. Javascript

Di dalam bootstrap sendiri sudah dilengkapi dengan yang namanya Library JavaScript yang sesuai dengan struktur dari bahasa pemrograman Javascript standar. Tentu saja dengan adanya komponen tersebut akan membuat website bisa lebih powerfull dibanding sebelumnya.

9. Dokumentasi yang lengkap

Bootstrap tidak hanya menawarkan styling untuk hampir setiap elemen yang dibutuhkan aplikasi situs web atau web biasa, namun juga menyediakan dokumentasi yang sangat lengkap dengan contoh dan demo yang membuatnya lebih mudah bagi pengguna baru sekalipun.

Selain itu, Bootstrap membutuhkan jQuery untuk berbagai komponennya. jQuery adalah perpustakaan JavaScript yang sangat populer dan banyak digunakan, yang menyederhanakan dan menambahkan kompatibilitas ke semua browser.

6.2 Framework CSS dengan Bootstrap

Membuat framework CSS yang fleksibel atau responsive, mengikuti standar terbaru dan benar-benar teruji membutuhkan waktu kerja yang panjang dan bisa menjadi proyek besar. Sebagian besar dari kita tidak punya waktu untuk melakukan hal ini. Dengan munculnya perangkat mobile, membuat situs web yang responsif dan mengikuti teknologi terbaru akan sangat menyita waktu untuk pengembangan user interfacenya. Kebutuhan untuk membuat website yang responsive adalah sebuah keharusan pada masa sekarang ini. Google pun telah mewanti-wanti pengembang web untuk selalu memperhatikan bagaimana tampilan website mereka di perangkat mobile. Tentunya tanpa mengurangi fitur dan konten yang ditampilkan pada tampilan desktop. Framework Bootstrap menjawab itu semua.

Menurut Jake Spurlock keunggulan dari penggunaan Bootstrap adalah dapat menyesuaikan dengan kebutuhan website, dimana dapat memilih fitur CSS dan JavaScript sesuai dengan kebutuhan. Contohnya CSS pada Bootstrap juga menyediakan fitur form, tombol, navigasi dan komponen-komponen lainnya dan JavaScript yang membantu pembuatan antarmuka lebih mudah dan stabil. Bootstrap juga menyediakan banyak sekali class-class CSS dan plugin yang siap dipakai untuk membantu dalam membuat tampilan sebuah website. Karena sangat membantu, maka Bootstrap menjadi salah satu front-end framework yang paling banyak digunakan. Walaupun sebagai sebuah framework, penggunaan bootstrap juga harus dikolaborasikan dengan penggunaan stylesheet. Hal tersebut dikarenakan class-class dan plugin yang tersedia didalamnya belum bisa memenuhi semua kebutuhan programmer. Jika semua programme hanya mengandalkan bootstrap dalam pembuatan tampilan sebuah website, dapat dipastikan bahwa akan terjadi kemiripan antara tampilan website satu dengan lainnya. Untuk menerapkan bootstrap, perlu adanya penambahan code berikut ini pada lembar HTML:

```
<head>
<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
</head>
```

Gambar 17. Potongan Script

Di bawah ini merupakan contoh class pada bootstrap yang dapat digunakan untuk membuat sebuah button. Dengan menerapkan potongan kode, output yang dihasilkan dapat dilihat pada gambar 16.

```
<div class="container">
<h2>Contoh penggunaan Button</h2>
<button type="button" class="btn btn-outline-primary">class Primary</button>
<button type="button" class="btn btn-outline-secondary">class Secondary</button>
<button type="button" class="btn btn-outline-success">class Success</button>
</div>
```

Gambar 18. Potongan Script

Contoh penggunaan Button



Gambar 19. Hasil tampilan dengan mengimplementasikan bootstrap

Dengan penggunaan class yang berbeda pada gambar di atas, didapatkan hasil yang berbeda pula. Class yang digunakan dapat disesuaikan dengan kebutuhan yang ada. Untuk memaksimalkan penggunaan bootstrap, maka harus diimbangi dengan penggunaan style sheet yang dapat dibuat secara manual yang disesuaikan dengan kebutuhan.

6.3 Pengenalan Fitur Bootstrap

Satu set file bootstrap dapat diperoleh di situs getbootstrap.com yang terdiri dari:

- direktori css: bootstrap.css
- direktori js: bootstrap.js
- direktori font: file-file font glyphsicons

Ditambah 1 file jquery.js yang dapat diperoleh di jquery.com. Letakkan file jquery.js ini di direktori js bersama file bootstrap.js. Letakkan semua direktori file di atas di direktori htdocs. Setelah semua file diperoleh selanjutnya menambahkan beberapa perintah di bagian head dokumen html untuk menggunakannya:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Desain Bootstrap</title>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <link rel="stylesheet" href="css/bootstrap.css">
8   <script src="js/jquery.js"></script>
9   <script src="js/bootstrap.js"></script>
10 </head>
11 <body>
12   <div class="container-fluid">
13     <h1>My First Bootstrap</h1>
14     <p>Dengan bootstrap proses layout laman web jadi lebih cepat.</p>
15   </div>
16 </body>
17 </html>

```

Gambar 20. Penggunaan Bootstrap

Penjelasan terkait Gambar 19 dapat dilihat dibawah ini.

- Baris ke-6 mengatur agar lebar laman web mengikuti lebar layar perangkat (dekstop, tablet, ponsel) dengan skala awal normal, artinya tampilan tidak diperbesar atau sebaliknya. Pengaturan ini membentuk sifat responsive Bootstrap.
- Baris ke-7 memanggil file css bootstrap.
- Baris ke-8 memanggil file jquery. File ini diperlukan bootstrap untuk menjalankan berbagai animasi, misalnya animasi silde jumbotron.
- Baris ke-9 memanggil file javascript bootstrap.
- Baris ke-12 contoh penggunaan kelas ‘container-fluid’ dalam bootstrap agar semua elemen ‘terbungkus’ mengumpul di satu wadah.

Beberapa contoh dalam menggunakan bootstrap dapat dilihat pada bagian dibawah ini.

1. Grid

Untuk memahami cara kerja bootstrap, kita perlu tahu tentang grid bootstrap. Pada dasarnya grid bootstrap adalah tentang bagaimana bootstrap membagi-bagi suatu laman web sehingga proses layout menjadi mudah dan sistematis. Bootstrap membagi laman web menjadi 12 grid (daerah vertikal laman web). Setiap grid dapat digabungkan untuk membuat grid yang lebih besar:

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

Gambar 21. Grid

Span4 merupakan gabungan dari 4 grid, span8 gabungan 8 grid, dan seterusnya. Saat menggabungkan grid perlu diingat bahwa setiap baris harus terdiri dari total 12 grid. Bootstrap memiliki 4 buah kelas grid:

- a. xs : grid untuk layout di layar ponsel
- b. sm : grid untuk layout di layar tablet
- c. md : grid untuk layout di layar dekstop
- d. lg : grid untuk layout di layar dekstop besar

Artinya ketika kita ingin mengatur grid untuk tampilan di layar ponsel maka gunakan kelas xs, untuk tablet gunakan kelas sm, dan seterusnya. Ketika menggunakan sistem grid bootstrap:

- Baris-baris harus berada didalam sebuah kelas container atau container-fluid agar obyek-obyek dalam laman teratur rapi.
- Gunakan kelas row untuk membuat kelompok kolom
- Tempatkan konten laman dalam kolom
- Setiap kolom memiliki jarak dengan kolom lain

2. Form

Ada 3 jenis tampilan form dalam bootstrap:

- Form vertical
- Form horisontal
- Form inline

Pengaturan tampilan form dalam bootstrap mengikuti aturan:

- Gunakan elemen label untuk teks setiap inputan
- Gunakan kelas form-control untuk semua elemen input, select, dan textarea
- Bungkus elemen label dan input dengan elemen div berkelas form-group

Form vertikal:

```
<div class="container">
  <h2>Vertical (basic) form</h2>
  <form action="/action_page.php">
    <div class="form-group">
      <label for="email">Email:</label>
      <input type="email" class="form-control" id="email"
        placeholder="Enter email" name="email">
    </div>
    <div class="form-group">
      <label for="pwd">Password:</label>
      <input type="password" class="form-control" id="pwd"
        placeholder="Enter password" name="pwd">
    </div>
    <div class="checkbox">
      <label><input type="checkbox" name="remember">
Remember
      me</label>
    </div>
    <button type="submit" class="btn btn
      default">Submit</button>
  </form>
</div>
```

Vertical (basic) form

Email:

Password:

☐ Remember me

Gambar 22. Form Vertical

Form inline hanya berlaku untuk layar minimal 768px. Dibawah itu akan ditampilkan seperti form vertikal. Aturan tambahan: gunakan kelas form-inline pada elemen form.

```
<div class="container">
  <h2>Inline form
    <small>Hanya berlaku untuk layar minimal 768px</small>
  </h2>
  <form class="form-inline" action="/action_page.php">
    <div class="form-group">
      <label for="email">Email:</label>
      <input type="email" class="form-control" id="email"
        placeholder="Enter email" name="email">
    </div>
    <div class="form-group">
      <label for="pwd">Password:</label>
      <input type="password" class="form-control" id="pwd"
        placeholder="Enter password" name="pwd">
    </div>
    <div class="checkbox">
      <label><input type="checkbox" name="remember">
        Remember me</label>
    </div>
    <button type="submit" class="btn btn-
      default">Submit</button>
  </form>
</div>
```

Inline form Hanya berlaku untuk layar minimal 768px

Email: Password: ☐ Remember me

Gambar 22. Form Inline

Form horisontal digunakan untuk layar kecil dibawah 768px, form horisontal akan berubah menjadi form vertikal. Aturan tambahan:

- Tambahkan kelas form-horizontal pada elemen form
- Tambahkan kelas control-label pada elemen label

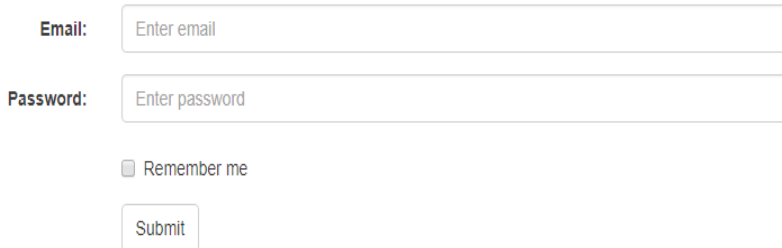
```
<div class="container">
  <form class="form-horizontal">
    <div class="form-group">
      <label class="control-label col-sm-2"

        for="email">Email:</label>
      <div class="col-sm-10">
        <input type="email" class="form-control" id="email"
          placeholder="Enter email">
      </div>
    </div>
    <div class="form-group">
      <label class="control-label col-sm-2"

        for="pwd">Password:</label>
      <div class="col-sm-10">
        <input type="password" class="form-control" id="pwd"
          placeholder="Enter password">
      </div>
    </div>
    <div class="form-group">
      <div class="col-sm-offset-2 col-sm-10">
        <div class="checkbox">
          <label><input type="checkbox"> Remember me</label>
        </div>
      </div>
    </div>
    <div class="form-group">
      <div class="col-sm-offset-2 col-sm-10">
        <button type="submit" class="btn btn-

          default">Submit</button>
      </div>
    </div>
  </form>
</div>
```

Horizontal form



Email:

Password:

☐ Remember me

Gambar 23. Form Horizontal

3. Tabel

Bootstrap memberikan kemudahan untuk membuat tabel tanpa harus mengetikkan berbaris-baris kode program. Kita hanya perlu memanggil class-class yang sudah ada di dalam file Bootstrap. Untuk menggunakan class dengan kombinasi class lainnya, gunakan class table ditambah dengan class lain. Contoh penggunaannya seperti di bawah ini:

```
# untuk membuat tabel stripped
<table class="table table-striped">

Atau

# untuk membuat tabel hover<table class="table table-hover">
```

Di dalam file Bootstrap sudah terdapat beberapa class yang didesain untuk kebutuhan pembuatan tabel, antara lain `“.table”`, `“.table-striped”`, `“.table-bordered”`, dan `“.table-hover”`.

- `“.table”`: class yang digunakan untuk membuat dan mendefinisikan tabel biasa/standar. Class ini hanya menambahkan lapisan yang kecil dan pembatas secara horizontal. Class ini menggunakan baris kode `<table class="table">`.
-

- `“.table-striped”`: class yang digunakan untuk membuat table dengan gaya yang zebra. Jadi warna setiap baris berselang seling. Class ini menggunakan baris kode `<table class=“table table-sripped”>`.
- `“.table-bordered”`: merupakan class yang digunakan untuk menampilkan border di dalam table. Jadi keseluruhan table akan dipasang sebuah border. Class ini menggunakan baris kode `<table class=“table table-bordered”>`.
- `“.table-hover”`: dapat dimanfaatkan untuk memberikan efek hover pada sebuah row. Efek ini akan berjalan ketika pengguna meletakkan mouse di row yang diberikan class ini. Class ini menggunakan baris kode `<table class=“table table-hover”>`.

Selain kumpulan class untuk mendesain table, terdapat beberapa class pewarnaan row tabel dan data menggunakan Bootstrap, yaitu `“.success”`, `“.danger”`, `“.info”`, `“.warning”`, dan `“.active”`. Untuk menggunakan class ini, selipkan di antara tag `tr`, seperti ini:

```
... <tr class=“info”>    <td>Info</td>    <td>Khafid</td>
    <td>july@example.com</td>    </tr>...
```

Menggunakan kumpulan class ini, kita dapat mendesain tabel yang menarik dan mudah dibaca oleh pengunjung website. Kita dapat menggunakan baris kode di bawah ini untuk mencoba menampilkan tabel dengan Bootstrap di browser.

4. Gambar

Dalam bootstrap dapat juga mendesain gambar. Membuat gambar yang responsive, berbentuk round, lingkaran, dan juga thumbnail sangat mudah menggunakan framework ini. Class `“.img-responsive”`, `“.img-rounded”`, `“.img-circle”`, dan `“.img-thumbnail”` adalah kumpulan class yang digunakan untuk memanipulasi gambar di Bootstrap. Kembangkan desain yang maksimal dan hadirkan gambar menarik di website.

- `“.img-responsive”`: class ini memungkinkan kita membuat gambar yang dapat dibuka di berbagai macam resolusi layar. Class `.img-responsive` membuka ukuran gambar dapat menyesuaikan dengan ukuran elemen inti. Kita dapat melihat efeknya ketika melakukan `resize` pada browser.

```

```

- `“.img-rounded”`: ketika kita ingin membuat gambar dengan sisi melengkung, gunakan class ini, maka kita akan memiliki gambar yang melengkung di bagian sisinya.

```

```

- `“.img-circle”`: terkadang ingin gambar yang berbentuk lingkaran, gunakan class ini dan gambar akan tampil dengan bentuk yang melingkar.

```

```

- `“.img-thumbnail”`: class ini digunakan untuk membuat gambar thumbnail dengan Bootstrap.

```

```

5. Alert

Bootstrap menyediakan class khusus untuk menampilkan alert dengan sangat sederhana dan mudah. Kita hanya perlu memanggil sebuah class di Bootstrap menggunakan satu perintah saja. Supaya tampilan alert dapat muncul, cukup panggil class `“alert”` untuk mendefinisikan tampilan alert di Bootstrap. Supaya tampilannya lebih bervariasi, tambahkan

class lain. Untuk membuat class ini caranya cukup mudah. Cukup buat tag <div> dengan isi peringatan yang ingin dibuat. Kemudian di tag pembuka <div> buat class alert untuk mendeklarasikan alert, berikut contohnya:

```
...<div class="container">
<h2>Alerts</h2>
<div class="alert alert-success">
  <strong>Success!</strong> Alert box ini dapat digunakan
  untuk mengindikasikan aksi sukses atau positif.
</div>
<div class="alert alert-info">
  <strong>Info!</strong> Alrt box ini dapat
  mengindikasikan informasi yang netral.
</div>
<div class="alert alert-warning">
  <strong>Warning!</strong> Alert box ini dapat untuk
  mengindikasikan pesan warning dan mungkin butuh
  perbaikan.
</div>
<div class="alert alert-danger">
  <strong>Danger!</strong> Alert box ini dapat
  mengindikasikan pesan penting dan kemungkinan efek
  buruk.
</div>
</div>...
```

Terdapat beberapa pilihan class untuk menampilkan pesan peringatan di Bootstrap, seperti “.alert-success”, “.alert-info”, “.alert-warning”, atau “.alert-danger”. Kumpulan class ini akan membuat tampilan alert lebih bervariasi.

- .alert-success: Class alert box ini dapat digunakan untuk mengindikasikan aksi sukses atau positif. Kita dapat memanfaatkan class ini ketika ada input yang berhasil. Misalnya ketika pengguna selesai mengisi data dan mengirimkannya. Untuk menggunakan class ini cukup masukkan kode:

```
<div class="alert alert-success">
```

- .alert-info: Class alert box ini dapat mengindikasikan informasi yang netral. Class ini dapat kita gunakan ketika ingin memberikan informasi tambahan kepada pengguna. Misalnya memberikan informasi kegunaan tombol atau fungsi tertentu dan cara penggunaannya. Untuk menggunakan class ini cukup masukkan kode :

```
<div class="alert alert-info">
```

- .alert-warning: Class alert box ini dapat untuk mengindikasikan pesan warning dan mungkin butuh perbaikan. Class ini dapat kita gunakan ketika ingin memberikan informasi yang sedikit mendesak atau penting. Misalnya ketika terjadi error pada suatu konfigurasi atau kesalahan pada saat pengisian data. Untuk menggunakan class ini cukup masukkan kode :

```
<div class="alert alert-warning">
```

- .alert-danger: Class alert box ini dapat mengindikasikan pesan penting dan kemungkinan efek buruk. Class ini dapat kita gunakan ketika terjadi suatu yang sangat serius. Misalnya ketika terjadi permasalahan pada server atau konfigurasi keamanan yang gagal. Penggunaan alert yang sesuai akan memberikan user experience yang menarik. Untuk menggunakan class ini cukup masukkan kode :

```
<div class="alert alert-danger">
```

Pengguna tidak akan kesulitan menemukan error yang terjadi sehingga problem solving pun dapat dilakukan dengan cepat.

6. Button

Tombol adalah komponen yang umum dan sering digunakan untuk membuat website. Inilah yang membuat Bootstrap juga menyediakan class untuk membuat tombol yang menarik. Kita hanya perlu memanggil salah satu class tombol dan tombol yang indah dapat kita tampilkan di halaman website. Supaya kita bisa menggunakan tombol di dalam website, cukup panggil class "btn". Selain itu, kita bisa menambahkan class lain untuk memberikan variasi tombol di dalamnya. Terdapat banyak variasi tombol yang disediakan oleh Bootstrap, seperti ".btn", ".btn-default", ".btn-primary", ".btn-success", ".btn-info", ".btn-warning", ".btn-danger", dan ".btn-link". Contoh penggunaan skrip seperti di bawah ini:

```
<div class="container">

<h2>Button Styles</h2>

<button type="button" class="btn">Basic</button>

<button type="button" class="btn btn-default">Default</button>

<button type="button" class="btn btn-primary">Primary</button>

<button type="button" class="btn btn-success">Success</button>

<button type="button" class="btn btn-info">Info</button>

<button type="button" class="btn btn-warning">Warning</button>

<button type="button" class="btn btn-danger">Danger</button>

<button type="button" class="btn btn-link">Link</button>

</div>
```

- `.btn`: Class ini untuk membuat tombol standar warna abu kehitaman.
- `.btn-default`: Class ini untuk membuat tombol default berwarna abu.
- `.btn-primary`: Class ini digunakan untuk membuat tombol warna biru. Biasanya digunakan untuk tombol login dan masuk ke dalam menu tertentu.
- `.btn-success`: Class ini digunakan untuk membuat tombol ketika berhasil menginputkan sesuatu atau memproses suatu program.
- `.btn-info`: Class ini digunakan untuk membuat tombol warna biru muda.
- `.btn-warning`: Class ini digunakan untuk membuat tombol warning dengan warna kuning.
- `.btn-danger`: Class ini digunakan untuk membuat tombol warning warna merah tua.
- `.btn-link`: Selain tombol persegi, Bootstrap juga menyediakan tombol untuk link. Anda bisa menggunakan class ini.

7. Panel

Panel adalah sebuah garis yang terbentuk dari garis dan padding. Di Bootstrap, sebuah panel terdiri dari tiga bagian; header, body, dan footer. Header berada di bagian atas. Body di bagian tengah atau inti. Sedangkan footer di bagian bawah. Untuk membuat panel dengan header, body, dan footer harus dideklarasikan terlebih dahulu menggunakan kode seperti di bawah ini.

```
<div class="panel panel-default">  
  <div class="panel-heading">Panel Heading</div>  
  <div class="panel-body">Panel Content</div>  
  <div class="panel-footer">Panel Footer</div>  
</div>
```

Terdapat banyak variasi panel yang dapat dibuat menggunakan Bootstrap, seperti “.panel-default”, “.panel-primary”, “.panel-success”, “.panel-info”, “.panel-warning”, dan “.panel-danger”.

- .panel-default: Class ini digunakan untuk membuat sebuah panel dengan warna biasa (abu).

```
<div class="panel panel-default">  
  <div class="panel-heading">Panel dengan class panel-  
  default</div>  
  <div class="panel-body">Isi Konten</div>  
</div>
```

- .panel-primary: Class ini digunakan untuk membuat panel dengan warna biru tua.

```
<div class="panel panel-primary">  
  <div class="panel-heading">Panel dengan class panel-  
  Primary  
  </div>  
  <div class="panel-body">Isi Konten  
  </div>  
</div>
```

- .panel-success: Class ini digunakan untuk membuat panel dengan dengan warna atas hijau.

```
<div class="panel panel-success">  
  <div class="panel-heading">Panel dengan class panel-  
  Success  
  </div>  
  <div class="panel-body">Isi Konten  
  </div>  
</div>
```

- .panel-info: Class ini digunakan untuk membuat panel dengan warna header biru muda.

```
<div class="panel panel-info">  
  <div class="panel-heading">Panel dengan class panel-  
  info class  
  </div>  
  <div class="panel-body">Isi Konten  
  </div>  
</div>
```

- .panel-warning: Class ini digunakan untuk membuat panel dengan warna header kuning muda.

```
<div class="panel panel-warning">  
  <div class="panel-heading">Panel dengan class panel-  
  info class  
  </div>  
  <div class="panel-body">Isi Konten  
  </div>  
</div>
```


- .panel-danger: Class ini digunakan untuk membuat panel dengan warna merah muda,

```
<div class="panel panel-danger">  
  <div class="panel-heading">Panel dengan class panel-  
  danger class  
</div>  
  <div class="panel-body">Isi Konten  
</div>  
</div>
```

Lima bagian di atas adalah contoh cara menggunakan Bootstrap. Selain itu masih banyak class lain yang dapat kita gunakan untuk membuat Bootstrap, seperti BS Dropdowns Collapse, Tabs/Pills, Navbar, Forms, Carousel, Modal, Tooltip, Popover, Scrollspy, dan masih banyak yang lainnya.



Rangkuman

Bootstrap adalah framework CSS yang dikhususkan untuk pengembangan front-end website. Framework ini mempunyai nama asli Twitter Blueprint. Ada kata 'Twitter' karena pada awalnya dikembangkan untuk sosial media Twitter yang sangat terkenal saat ini dengan pengguna hampir 326 juta orang. Sebelum Bootstrap muncul, sudah ada framework lain yang tersedia. Hanya saja tingkat konsistensi dalam proses pengembangan buruk. Selain itu juga butuh biaya perawatan mahal. Berdasarkan permasalahan ini, akhirnya para pengembang menemukan Bootstrap. Berbeda dengan framework sebelumnya, Bootstrap dikenal konsisten dan lebih sederhana. Bootstrap adalah salah satu framework yang memungkinkan developer dapat mengembangkan website

dengan mudah dan cepat. Developer hanya perlu memanggil class tertentu untuk membuat tombol, panel, tabel, pesan peringatan, dan lain sebagainya.

Kumpulan class yang sangat lengkap membuat Bootstrap pantas menjadi framework front end yang paling populer saat ini. Class-class yang tersedia terbukti dapat membuat website lebih menarik. Tidak hanya itu, Bootstrap memudahkan pengembang untuk membuat tombol, panel, alert, dan lain sebagainya hanya dengan satu perintah saja.



Latihan Soal

1. Apakah manfaat yang didapatkan ketika kita menggunakan bootstrap dalam pengembangan web?
2. Dengan menggunakan bootstrap saja apakah sudah cukup dalam membangun web? Berikan alasannya?
3. Bangunlah tampilan awal atau dashboard halaman web dengan menggunakan bootstrap?
4. Bangunlah halaman login dengan menggunakan salah satu form dalam bootstrap? (halaman login terhubung dengan dasborad)

BAB VII

CodeIgniter

7.1 Pengertian CodeIgniter

CodeIgniter adalah sebuah web application network yang bersifat open source yang digunakan untuk membangun aplikasi php dinamis. CodeIgniter menjadi sebuah framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, CodeIgniter juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih CodeIgniter sebagai framework pilihannya. Karena kelebihan-kelebihan yang dimiliki oleh CodeIgniter, pembuat PHP Rasmus Lerdorf memuji CodeIgniter di frOSCon (Agustus 2008) dengan mengatakan bahwa dia menyukai CodeIgniter karena “it is faster, lighter and the least like a framework.” CodeIgniter pertamakali dikembangkan pada tahun 2006 oleh Rick Ellis. Dengan logo api yang menyala, CodeIgniter dengan cepat “membakar” semangat para web developer untuk mengembangkan web dinamis dengan cepat dan mudah menggunakan framework PHP yang satu ini. Fungsi CodeIgniter ialah :

1. Mempercepat dan mempermudah kita dalam pembuatan website.
2. Menghasilkan struktur pemrograman yang sangat rapi, baik dari segi kode maupun struktur file phpnya.
3. Memberikan standar coding sehingga memudahkan kita atau orang lain untuk mempelajari kembali system aplikasi yang dibangun.

Serta kelebihan yang dimiliki oleh CodeIgniter sebagai berikut :

1. Berukuran sangat kecil. File download nya hanya sekitar 2MB, itupun sudah includedokumentasinya yang sangat

lengkap.

2. Dokumentasi yang bagus. Saat anda mendownloadnya, telah disertakan dengan dokumentasi yang berisi pengantar, tutorial, bagaimana panduan penggunaan, serta referensi dokumentasi untuk komponen-komponennya.
3. Kompatibilitas dengan Hosting. CodeIgniter mampu berjalan dengan baik pada hampir semua platform hosting. CodeIgniter juga mendukung database-database paling umum, termasuk MySQL.
4. Tidak ada aturan coding yang ketat. Terserah anda jika anda hanya ingin menggunakan Controller, tanpa View, atau tidak menggunakan Model, atau tidak salah satu keduanya. Namun dengan menggunakan ketiga komponennya adalah pilihan lebih bijak.
5. Kinerja yang baik. CodeIgniter sangat cepat bahkan mungkin bisa dibilang merupakan framework yang paling cepat yang ada saat ini.
6. Sangat mudah diintegrasikan. CodeIgniter sangat mengerti tentang pengembangan berbagai library saat ini. Karenanya CodeIgniter memberikan kemudahan untuk diintegrasikan dengan library-library yang tersedia saat ini.
7. Sedikit Konfigurasi. Konfigurasi CodeIgniter terletak di folder application/config. CodeIgniter tidak membutuhkan konfigurasi yang rumit, bahkan untuk mencoba menjalankannya, tanpa melakukan konfigurasi sedikitpun ia sudah bisa berjalan.
8. Mudah dipelajari. Disamping dokumentasi yang lengkap, ia juga memiliki berbagai forum diskusi.

Selain cepat dan mudah, codeigniter juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih codeigniter sebagai framework pilihannya. Karena kelebihan-kelebihan yang dimiliki oleh codeigniter, pembuat PHP Rasmus Lerdorf memuji codeigniter di frOSCon (Agustus 2008) dengan mengatakan bahwa dia menyukai codeigniter karena "it is faster, lighter and the least like a framework." Codeigniter pertamakali

dikembangkan pada tahun 2006 oleh Rick Ellis. Dengan logo api yang menyala, codeigniter dengan cepat “membakar” semangat para web developer untuk mengembangkan web dinamis dengan cepat dan mudah menggunakan framework PHP yang satu ini. Pada bulan Juli 2013, EllisLab mengumumkan bahwa mereka mencari pemilik baru untuk codeigniter karena internal mereka sendiri tidak memiliki cukup fokus untuk terus mengembangkan codeigniter. Akhirnya pada bulan Oktober 2014, kepemilikan CodeIgniter berpindah tangan kepada British Columbia Institute of Technology, salah satu sekolah tinggi teknologi di Kanada.

Sedangkan kekurangan yang dimiliki oleh CodeIgniter diantaranya : (a) CodeIgniter tidak ditujukan untuk pembuatan web dengan skala besar, (b) Library yang sangat terbatas. Hal ini dikarenakan sangat sulit mencari plugin tambahan yang terverifikasi secara resmi, karena pada situsnya CodeIgniter tidak menyediakan plugin-plugin tambahan untuk mendukung pengembangan aplikasi dengan CI, (c) Belum adanya editor khusus CodeIgniter, sehingga dalam melakukan create project dan modul-modulnya harus berpindah-pindah folder.

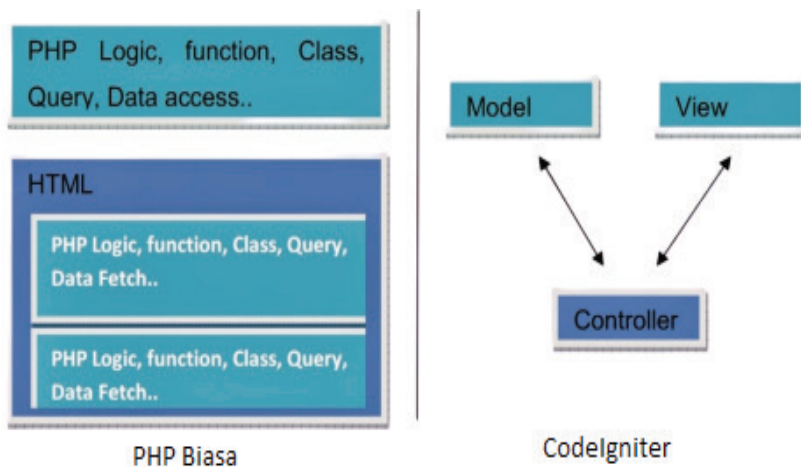
7.2 MVC (Model View Controller)

MVC adalah konsep dasar yang harus diketahui sebelum mengenal CodeIgniter. MVC (Model View Controller) merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman Small Talk, yang memisahkan bisnis logic (alur piker), data logic (penyimpanan data) dan presentation logic (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses. Ada 3 komponen yang membangun suatu MVC yaitu :

1. Model, biasanya berhubungan dengan data dan interaksi ke database atau webservice. Model juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun webservice. Biasanya didalam model akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data website. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian Model

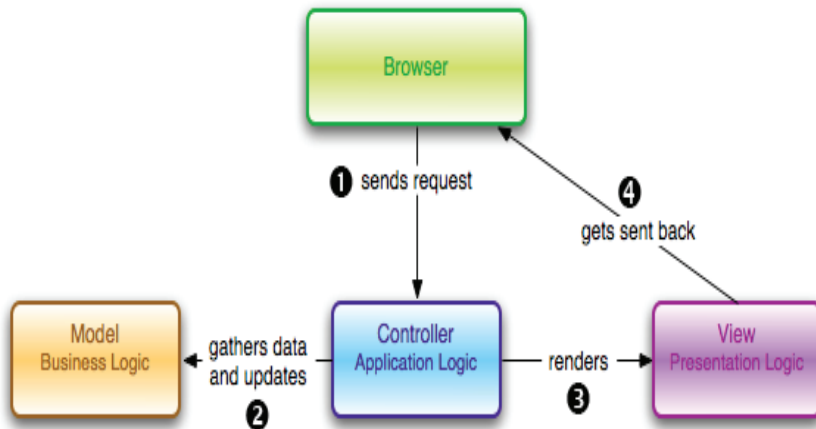
biasanya akan berhubungan dengan perintah-perintah query SQL.

2. View, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. View berfungsi untuk menerima dan merepresentasikan data hasil dari model dan controller kepada user. View tidak memiliki akses langsung terhadap bagian model.
3. Controller, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view. Pada controller terdapat class-class dan fungsi-fungsi yang memproses permintaan dari View ke dalam struktur data di dalam model. Controller juga tidak boleh berisi kode untuk mengakses basis data. Karena tugas mengakses data telah diserahkan kepada model. Tugas controller adalah menyediakan berbagai variable yang akan ditampilkan di view, memanggil model untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/error, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap input.



Gambar 24. Perbandingan PHP biasa dengan CodeIgniter

Pemisahan ini menghasilkan permintaan pengguna yang diproses sebagai berikut:

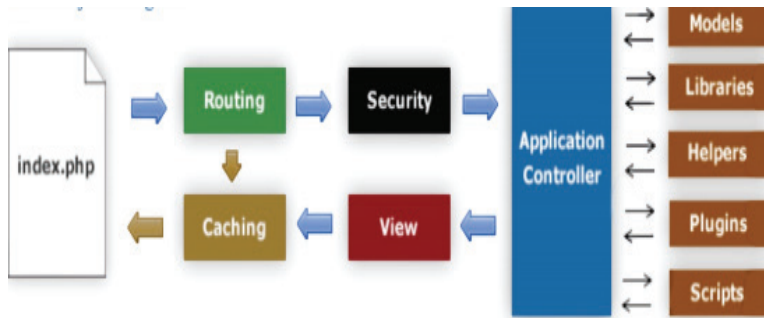


Gambar 25. Ilustrasi MVC

- a. Browser (pada klien) mengirimkan permintaan halaman ke controller di server.
- b. Pengontrol mengambil data yang dibutuhkan dari model untuk menanggapi permintaan.
- c. Kontroler memberikan data yang diambil ke tampilan.
- d. Tampilan diberikan dan dikirim kembali ke klien untuk ditampilkan browser.

Memisahkan aplikasi perangkat lunak ke dalam tiga komponen berbeda ini adalah ide yang bagus untuk sejumlah alasan, termasuk:

- Peningkatan skalabilitas - misalnya, jika aplikasi kita mulai mengalami masalah kinerja karena akses basis data lambat, kita dapat memutakhirkan perangkat keras yang menjalankan basis data tanpa komponen lain yang terpengaruh
- kemudahan pemeliharaan — karena komponen memiliki ketergantungan yang rendah satu sama lain, membuat perubahan pada satu (untuk memperbaiki bug atau mengubah fungsi) tidak memengaruhi yang lain
- reusability — model dapat digunakan kembali oleh banyak tampilan



Gambar 26. Alur kerja CodeIgniter

- **Index.php**
Index.php disini berfungsi sebagai file pertama dalam program yang akan dibaca oleh program.
- **The Router**
Router akan memeriksa HTTP request untuk menentukan hal apa yang harus dilakukan oleh program.
- **Cache File**
Apabila dalam program sudah terdapat “cache file” maka file tersebut akan langsung dikirim ke browser. File cache inilah yang dapat membuat sebuah website dapat di buka dengan lebih cepat. Cache file dapat melewati proses yang sebenarnya harus dilakukan oleh program codeigniter.
- **Security**
Sebelum file controller di load keseluruhan, HTTP request dan data yang disubmit oleh user akan disaring terlebih dahulu melalui fasilitas security yang dimiliki oleh codeigniter.
- **Controller**
Controller akan membuka file model, core libraries, helper dan semua resources yang dibutuhkan dalam program tersebut.
- **View**
Hal yang terakhir akan dilakukan adalah membaca semua program yang ada dalam view file dan mengirimkannya ke browser supaya dapat dilihat. Apabila file view sudah ada yang di “cache” maka file view baru yang belum ter-cache akan mengupdate file view yang sudah ada.

7.3 CodeIgniter Library dan Helper

Codeigniter menyediakan dua jenis sarana yang dapat digunakan untuk membantu proses pengembangan aplikasi, antara lain:

1. Library

Library adalah sekumpulan kelas dan fungsi yang dibuat untuk membantu pengembang aplikasi untuk dapat membangun aplikasi dengan lebih cepat dan lebih efisien. Pada umumnya saat kita membuat aplikasi web ada beberapa kelas yang hampir selalu digunakan, sehingga kelas-kelas tersebut dapat di atur supaya secara otomatis di-load oleh system dan dapat langsung digunakan. Pada CodeIgniter library dibagi menjadi 2 yaitu library yang bersifat global dan library yang dapat dibuat sendiri sesuai kebutuhan. Library global terdiri dari kelas dan fungsi-fungsi yang telah disediakan oleh CodeIgniter, dan terletak pada folder system/libraries. Sedangkan library yang kita buat sendiri sesuai dengan kebutuhan ditempatkan pada folder application/libraries. Beberapa library yang wajib diketahui oleh pengembang di antaranya adalah :

1. Database, library yang digunakan untuk mengakses database dan melakukan pengolahan data yang ada di dalam database. Database yang di dukung oleh CodeIgniter adalah mysql, mssql, oracle dan postgres. Sedangkan database yang tidak didukung secara langsung dapat dijembatani dengan driver odbc.
2. Input, library yang digunakan untuk menangani dan memproses data-data yang berasal dari form. Misalnya apabila kita menggunakan form untuk memasukan data maka library ini harus di-load supaya dapat melakukan pemrosesan data form.
3. File Uploading, library yang digunakan apabila kita akan membangun web yang dapat mengunggah (upload) file ke dalam web. Misalkan kita menginginkan supaya di dalam web kita ada fitur yang dapat digunakan untuk memasukkan file gambar ke dalam aplikasi web kita, maka digunakanlah library ini.
4. Session, library yang digunakan untuk memelihara

informasi status mengenai pengguna. Sebagai contoh misalkan kita membangun suatu website dimana pengunjung website tersebut harus melakukan proses login terlebih dahulu untuk masuk ke dalam suatu halaman, maka pada situasi seperti ini, library session harus di-load supaya kita dapat memelihara state dari pengunjung, sampai pengunjung tersebut logout.

5. URI Class, library ini berisi fungsi-fungsi yang membantu kita untuk mendapatkan informasi dari URI pada alamat web kita.
6. Validation, library ini digunakan untuk melakukan validasi terhadap form input yang ada pada aplikasi web kita.
7. Pagination, library ini berguna pada saat kita memiliki banyak data yang harus ditampilkan. Misalkan kita memiliki 100 data, dimana ke-100 data ini akan ditampilkan ke dalam 10 halaman (10 data / halaman). Untuk membuat 10 halaman yang masing-masing memuat 10 data dan masing-masing halaman terhubung satu sama lain, maka pagination merupakan library yang tepat untuk digunakan.
8. Benchmarking, Kelas benchmark selalu aktif, memungkinkan perbedaan waktu antara dua titik yang ditandai untuk dihitung.
9. Caching, library ini akan melakukan cache halaman, untuk dengan cepat mengakses kecepatan halaman.
10. Calendaring, dengan menggunakan library ini, kita dapat secara dinamis membuat kalender
11. Shopping Cart, Menggunakan library ini, kita dapat menambah atau menghapus item dari Keranjang Belanja. Item disimpan dalam sesi dan akan tetap aktif sampai pengguna menelusuri situs.
12. Config, Preferensi konfigurasi dapat diambil, menggunakan library ini. library ini diinisialisasi secara otomatis.
13. Email, library ini menyediakan fungsionalitas terkait email, seperti mengirim atau membalas email.

14. Encryption, library ini menyediakan fungsionalitas enkripsi data dua arah.
15. FTP, menyediakan berbagai fungsi terkait FTP seperti mentransfer file untuk menghapus server, memindahkan, mengganti nama, atau menghapus file di server.
16. Image Manipulation, Manipulasi gambar seperti mengubah ukuran, pembuatan thumbnail, memotong, memutar, menandai dapat dilakukan dengan bantuan library ini.
17. Language, library ini digunakan untuk internasionalisasi.
18. Loader, library ini memuat elemen seperti Lihat file, Driver, Pembantu, Model dll.
19. Migration, library ini menyediakan fungsionalitas yang terkait dengan migrasi basis data.
20. Output, library ini mengirimkan output ke browser dan juga, cache halaman web itu.
21. Template Parser, library Parser Templat dapat melakukan substitusi teks sederhana untuk variabel semu yang terdapat dalam file tampilan web. Itu dapat mem-parsing variabel sederhana atau pasangan tag variabel.
22. Security, library ini berisi fungsi terkait keamanan seperti Penyaringan XSS, CSRF dll.
23. HTML Table, library ini digunakan untuk menghasilkan tabel HTML secara otomatis dari hasil array atau database.
24. Typography, library Tipografi menyediakan metode yang membantu memformat teks.
25. Unit Testing, library ini menyediakan fungsionalitas untuk menguji unit aplikasi dan menghasilkan hasilnya.
26. User Agent, library Agen Pengguna menyediakan fungsi yang membantu mengidentifikasi informasi tentang browser, perangkat seluler, atau robot yang mengunjungi situs. Selain itu, dapat memperoleh informasi pengarah serta bahasa dan informasi kumpulan karakter yang didukung.
27. XML-RPC and XML-RPC Server, library XML-RPC CodeIgniter memungkinkan untuk mengirim permintaan ke server lain, atau mengatur server XML-RPC Anda sendiri untuk menerima permintaan.

28. Zip Encoding, library ini digunakan untuk membuat arsip zip dari data Anda.

Untuk menggunakan library yang ada pada folder system/libraries, ada dua cara yang dapat dilakukan, yaitu :

- Mengatur pada file system/config/autoload.php,

Contoh konfigurasi library

```
$autoload['libraries']=array('form_validation','database','session');
```

- Dengan melakukan loading terhadap library yang kita inginkan pada controller dimana library ini akan digunakan. Biasanya library ini di-load pada konstruktor dari controller yang bersangkutan. Berikut sintaknya:

```
$this->load->library('nama_library');
```

2. Helper

Helper juga berfungsi untuk membantu pengembang membangun aplikasi secara lebih cepat dan efisien. Setiap helper bisa terdiri dari beberapa fungsi, dimana setiap fungsi dari helper melakukan satu pekerjaan yang spesifik tanpa ada ketergantungan terhadap fungsi yang lain. Helper biasanya disimpan dalam folder system/helpers, atau di dalam folder system/application/helpers. CodeIgniter akan terlebih dulu mencari helper di dalam folder system/application/helpers, jika helper yang dicari tidak ditemukan pada folder tersebut, baru kemudian dicari pada folder system/helpers. Untuk menggunakan helper, ada dua cara yang dapat dilakukan, yaitu :

- Melalui konfigurasi pada file autoload.php. Konfigurasi pada file autoload.php untuk melakukan proses autoloading terhadap helper-helper yang akan digunakan adalah sebagai berikut :

```
$autoload['helper'] = array('url','form','file');
```

- Melakukan loading pada setiap controller yang akan menggunakan helper, dilakukan dengan sintak sebagai berikut :

```
$this->load->helper('nama_helper');
```

Contoh helper :

1. URL helper : membantu dalam pembuatan link.
2. Form helper : membantu untuk membuat element-element form.
3. Text helper : membantu untuk pekerjaan berformat text.
4. Cookie helper : membantu untuk penanganan cookies.
5. File helper : membantu untuk kerja dengan file.

7.4 Konfigurasi CodeIgniter

Konfigurasi dasar pada Codeigniter merupakan point penting dalam membangun website dengan codeigniter. Kita tidak bisa membuat website dengan Codeigniter jika tidak memahami konfigurasi dasar ini. Terdapat dua direktori penting di dalam CI: application dan system. Selain itu terdapat juga direktori user_guide dan beberapa file. Berikut ini penjelasannya:

- application berisi semua kode aplikasi. Di dalam direktori inilah kita akan menulis semua kode aplikasi kita.
- system berisi kode-kode inti dari Codeigniter. Jangan mengubah apapun di dalam direktori ini. Jika kita ingin upgrade versi, kita cukup me-replace direktori ini dengan yang baru.
- tests berisi kode untuk melakukan unit testing.
- user_guide berisi dokumentasi codeigniter. Kita bisa menghapus direktori ini saat web sudah jadi.
- .editor_config berisi konfigurasi untuk teks editor.
- .gitignore berisi daftar file dan folder yang akan diabaikan oleh Git.
- composer.json adalah file yang berisi keterangan project dan keterangan library yang digunakan. File ini dibutuhkan oleh composer.
- contributing.md adalah file yang berisi penjelasan cara berkontribusi di proyek CI. Kita bisa menghapus file ini, apabila web sudah jadi.

- `license.txt` adalah file yang berisi keterangan lisensi dari CI.
- `readme.rst` sama seperti file `contributing.md` file ini berisi penjelasan dan informasi tentang project CI. Kita juga bisa menghapus file ini saat web sudah selesai.
- `index.php` adalah file utama dari CI. File yang akan dibuka pertamakali saat kita mengakses web.

Folder `application/config` berisi file-file konfigurasi Codeigniter. File-file ini akan mempengaruhi kerja codeigniter.

- `cache` berisi cache dari aplikasi.
- `config` berisi konfigurasi aplikasi.
 - o `autoload.php` tempat kita mendefinisikan `autoload`;
 - o `config.php` konfigurasi aplikasi;
 - o `constants.php` berisi konstanta;
 - o `database.php` konfigurasi database aplikasi;
 - o `doctypes.php` berisi definisi untuk doctype HTML;
 - o `foreign_chars.php` berisi karakter dan simbol;
 - o `hooks.php` berisi konfigurasi hooks;
 - o `index.html` untuk mencegah direct access;
 - o `memcached.php` untuk berisi konfigurasi untuk `memcached`;
 - o `migration.php` konfigurasi untuk migrasi;
 - o `mimes.php` berisi definisi tipe file;
 - o `profiler.php` konfigurasi untuk profiler;
 - o `routers.php` tempat kita menulis route aplikasi;
 - o `smileys.php` berisi kode untuk emoji;
 - o `user_agents.php` berisi definisi untuk user agents.
- `controller` berisi kode untuk controller.
- `core` berisi kode untuk custom core.
- `helpers` berisi fungsi-fungsi helper.
- `hooks` berisi kode untuk script hook.
- `language` berisi string untuk bahasa, apabila web mendukung multibahasa.
- `libraries` berisi library.
- `logs` berisi logs dari aplikasi.
- `models` berisi kode untuk model.
- `thrid_party` berisi library dari pihak ketiga.
- `views` berisi kode untuk view.
- `index.html` file html untuk mencegah direct access.

Untuk memberikan gambaran penggunaan file-file konfigurasi

Codeigniter. Dibawah ini merupakan konfigurasinya :

1. Siapkan framework codeigniter, apabila kita belum memiliki maka silakan langsung download ke situs resmi codeigniter di <https://www.codeigniter.com/>.

2. Autoload.php

File ini menentukan, secara default, sistem mana yang harus dimuat. Untuk menjaga kerangka kerja seringan mungkin, hanya sumber daya minimal absolut yang dimuat secara default. Seseorang harus secara otomatis memuat sistem yang sering digunakan, daripada memuatnya di tingkat lokal, berulang kali. Berikut ini adalah hal-hal yang dapat kita muat secara otomatis :

- o Libraries, yang harus dimuat secara otomatis. Berikan daftar pustaka dalam array seperti yang ditunjukkan di bawah ini untuk dimuat secara otomatis oleh CodeIgniter. Dalam contoh ini, kami memuat database, email and session secara otomatis.

```
$autoload['libraries'] = array();
```

Menjadi

```
$autoload['libraries'] = array('database', 'email', 'session');
```

- o Drivers, Kelas-kelas ini terletak di sistem / perpustakaan / atau di direktori aplikasi / perpustakaan / Anda, tetapi juga ditempatkan di dalam subdirektori mereka sendiri dan mereka memperluas kelas CI_Driver_Library. Mereka menawarkan beberapa opsi driver yang dapat dipertukarkan. Berikut ini adalah contoh untuk memuat driver cache secara otomatis.

```
$autoload['drivers'] = array();
```

Menjadi

```
$autoload['drivers'] = array('cache');
```

- o Helper Files, helper files untuk diisikan otomatis. Berikan helper files dalam array, seperti yang ditunjukkan di

bawah ini, untuk dimuat secara otomatis oleh CodeIgniter. Dalam contoh yang diberikan, kami autoloading URL dan file helper.

```
$autoload['helper'] = array();
```

Menjadi

```
$autoload['helper'] = array('url','file');
```

- o Custom config files, File-file ini dimaksudkan untuk digunakan, hanya jika kita telah membuat file konfigurasi khusus. Jika tidak, biarkan kosong. Berikut ini adalah contoh cara memuat ulang otomatis lebih dari satu file konfigurasi.

```
$autoload['config'] = array();
```

Menjadi

```
$autoload['config'] = array('config1', 'config2');
```

- o Language files, file bahasa seharusnya dimuat secara otomatis. Lihatlah contoh yang diberikan di bawah ini. Berikan daftar bahasa dalam array seperti yang ditunjukkan di bawah ini untuk dimuat secara otomatis oleh CodeIgniter. Ingatlah bahwa tidak termasuk bagian “_lang” dari file yang kita miliki. Misalnya, “codeigniter_lang.php” akan dirujuk sebagai array (“codeigniter”);
- o Models, models yang harus dimuat secara otomatis. Berikan daftar model dalam array seperti yang ditunjukkan di bawah ini untuk dimuat secara otomatis oleh CodeIgniter. Berikut ini adalah contoh cara memuat secara otomatis lebih dari satu model file.

```
$autoload['model'] = array();
```

Menjadi

```
$autoload['model'] = array('first_model', 'second_model');
```


3. Config.php

File config.php berisi konfigurasi utama dari aplikasi yang sedang dibuat. Pada dasarnya config.php berisi definisi variabel bernama \$config. Variabel ini bertipe “associative array” dan memiliki index/key untuk masing-masing itemnya. Nama-nama index key tersebut menggambarkan data apa yang ditampungnya. Ada banyak item di config.php namun kita hanya akan membahas satu saja yaitu base_url. Base_url digunakan untuk mendefinisikan url website/aplikasi yang kita buat. Base_url ini merupakan url dasar dari suatu aplikasi berbasis web misalnya www.apple.com atau www.codeigniter.com. Untuk mengatur base_url, silahkan buka file config.php yang ada di dalam folder application/config dan cari kode seperti berikut:

```
$config['base_url'] = '';
```

Lalu ubah menjadi seperti ini:

```
$config['base_url'] = 'http://localhost/coba/';
```

Maksud dari \$config['base_url'] = ''; adalah untuk memberitahu bahwa direktori dari website yang akan digunakan ada dimana, maka kita isi dengan nama folder yang digunakan, jadi nanti jika ingin memanggil css, js atau lokasi gambar dengan format base_url (contoh) jika base_url ini tidak sesuai maka tidak akan terbaca nantinya.

4. Database.php

dapat dikonfigurasi dalam file application / config / database.php. Seringkali kita perlu mengatur basis data untuk lingkungan yang berbeda seperti pengembangan dan produksi. Dengan array multidimensi yang disediakan di CodeIgniter, kita dapat mengatur database untuk lingkungan yang berbeda. Pengaturan konfigurasi disimpan dalam array seperti yang ditunjukkan di bawah ini:

```
$db['default'] = array(  
    'dsn' => "",  
    'hostname' => 'localhost',  
    'username' => 'root',  
    'password' => "",  
    'database' => 'database_name',  
    'dbdriver' => 'mysqli',  
    'dbprefix' => "",  
    'pconnect' => TRUE,  
    'db_debug' => TRUE,  
    'cache_on' => FALSE,  
    'cachedir' => "",  
    'char_set' => 'utf8',  
    'dbcollat' => 'utf8_general_ci',  
    'swap_pre' => "",  
    'encrypt' => FALSE,  
    'compress' => FALSE,  
    'stricton' => FALSE,  
    'failover' => array()  
);
```

Kita dapat meninggalkan beberapa opsi ke nilai defaultnya kecuali nama host, nama pengguna, kata sandi, basis data, dan dbdriver.

- hostname - Tentukan lokasi basis data Anda di sini mis. localhost atau alamat IP
- username - Tetapkan nama pengguna dari basis data Anda di sini.
- password - Tetapkan kata sandi basis data Anda di sini.
- database - Tetapkan nama database di sini.
- dbdriver - Tetapkan jenis basis data yang Anda gunakan mis. MySQL, MySQLi, Postgre SQL, ODBC, dan MS SQL.

Dengan mengubah kunci array \$ db, Anda dapat mengatur konfigurasi database lainnya seperti yang ditunjukkan di bawah ini. Di sini, kami telah menetapkan kunci untuk 'test' untuk menetapkan basis data untuk lingkungan pengujian, dengan menjaga lingkungan basis data lainnya seperti apa adanya.

Silahkan atur sesuai dengan database yang telah dibuat sebagai contoh:

```
$db['test'] = array(  
    'dsn' => '',  
    'hostname' => 'localhost',  
    'username' => 'root',  
    'password' => '',  
    'database' => 'database_name',  
    'dbdriver' => 'mysqli',  
    'dbprefix' => '',  
    'pconnect' => TRUE,  
    'db_debug' => TRUE,
```

```
'cache_on' => FALSE,  
'cachedir' => '',  
'char_set' => 'utf8',  
'dbcollat' => 'utf8_general_ci',  
'swap_pre' => '',  
'encrypt' => FALSE,  
'compress' => FALSE,  
'stricton' => FALSE,  
'failover' => array()
```

5. Routes.php

Router pada Codeigniter bertugas untuk menentukan controller dan method/fungsi yang akan dieksekusi. Buka application\config\routes.php dan scroll ke bagian paling bawah maka akan terdapat code berikut :

```
$route['default_controller'] = 'welcome';  
  
$route['404_override'] = '';  
  
$route['translate_uri_dashes'] = FALSE;
```

Pada bagian default_controller link default website akan ditampilkan, di atas tertulis welcome yang artinya pada controller welcome akan muncul saat pertama kali pengguna membuka halaman.



Rangkuman

Framework secara sederhana dapat diartikan sebagai kumpulan dari fungsi-fungsi atau prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau class dari awal. Salah satu framework yang bisa digunakan adalah codeigniter. CodeIgniter adalah aplikasi open source yang berupa framework dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP. CodeIgniter memudahkan developer untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal.

Model View Controller merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman Small Talk, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, user interface, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC pattern dalam suatu aplikasi yaitu : (1) View, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. View berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model, (2) Model, biasanya berhubungan langsung dengan database untuk memanipulasi data (insert, update, delete, search), menangani validasi dari bagian controller, namun tidak dapat berhubungan langsung dengan bagian view. (3) Controller, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view, controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.



Latihan Soal

1. Apakah framework itu? dan apakah hubungan antara framework dengan CodeIgniter?
2. Jelaskan fungsi MVC dalam CodeIgniter?
3. Apakah perbedaan antara Bootstrap dan CodeIgniter?
4. Apakah fungsi library dan jelaskan beberapa library yang kamu ketahui?
5. Jelaskan dengan sederhana tahapan dalam mengkonfigurasi CodeIgniter?

BAB VIII

Implementasi Sistem

8.1 Latar Belakang

Toko adalah sebuah tempat tertutup yang di dalamnya terjadi kegiatan perdagangan dengan jenis benda atau barang yang khusus, misalnya toko buku, toko buah, toko kecil serba ada, toko sembako, toko kue, toko kelontong, dan sebagainya. Administrasi adalah seluruh proses kerja sama antara dua orang atau lebih dalam mencapai tujuan dengan memanfaatkan sarana dan prasarana tertentu secara berdaya guna dan berhasil guna. Administrasi juga berarti kegiatan yang meliputi catat-mencatat, suratmenyurat, pembukuan ringan, ketik-mengetik, agenda, dan sebagainya yang bersifat teknis ketatausahaan. Administrasi toko adalah segala bentuk kegiatan administrasi yang ada di toko.

Pada saat ini para pengusaha berusaha untuk meningkatkan strategi dalam menjalankan usahanya. Hal ini diakibatkan oleh perkembangan jaman serta teknologi yang terus bergulir. Di zaman modern ini, kemajuan di bidang teknologi informasi sudah berkembang semakin pesat dan semua menggunakan komputerisasi. Dahulu penjualan dilakukan dengan menggunakan pencatatan manual yang mengandalkan nota-nota yang dikumpulkan. Sedangkan untuk stok barangnya hanya menyesuaikan nota dari distributor barang dengan barang yang datang saja. Semua nota tersebut disimpan untuk menjadi bukti transaksi. Banyak sekali toko-toko yang dalam pencatatan transaksinya masih tergolong manual. Laporan keuangan tiap harinya juga masih dilakukan secara manual yaitu tulis tangan. Termasuk pencatatan stok barang, hutang toko, gaji pegawai, semuanya dilakukan secara manual oleh pemilik toko. Pencatatan yang dilakukan manual ini selain tidak efisien, resiko yang ditanggung pemilik toko juga besar. Apabila kehilangan nota pembelian dan penjualan barang, salah perhitungan toko, dan kehilangan catatan utang merupakan resiko terbesar dari pemilik toko karena dari nota itu dapat menentukan berapa pendapatan

atau keuntungan yang didapat. Selain itu apabila ingin mencari data nota lama mengalami kesusahan dan memakan waktu yang lama karena harus dicek manual satu persatu. Oleh karena itu kami berencana membuat sistem manual menjadi komputerisasi agar lebih efisien dan dibutuhkan sebuah program yang berguna untuk mempermudah pencatatan transaksi jual-beli dan stok barang yang ada di toko.

8.2 Ide Gagasan

Efektifitas dan efisiensi merupakan hal yang sangat dibutuhkan oleh manusia karena semakin banyak kegiatan manusia dalam melakukan proses bisnisnya. Hal ini tentunya semakin meningkatkan kebutuhan manusia akan teknologi yang lebih efektif dan efisiensi. Banyak bisnis yang telah memanfaatkan kemajuan teknologi, hal ini dapat diklarifikasi dari transaksi bisnis yang semakin banyak dilakukan secara komputerisasi. Karena pengguna teknologi dapat membantu bisnis untuk berjalan lebih efektif dan efisiensi.

Bedasarkan pernyataan tersebut kami mencoba untuk meningkatkan efektifitas dan efisiensi administrasi toko dengan membuat aplikasi web. Website ini diharapkan akan mempermudah semua proses administrasi yang terjadi dalam toko.

Dodolan adalah aplikasi administrasi untuk toko-toko kecil maupun besar untuk mengelola data agar lebih mudah dan efisien, baik itu mengatur data penjualan, pembelian, dan dapat juga mengatur laju keuntungan toko. Dalam aplikasi Dodolan ini terdapat berbagai macam fitur. Diantaranya yaitu, stok barang, daftar harga, grafik penjualan, keuntungan, arus modal, transaksi penjualan dan pembelian, dan fitur registrasi untuk akun-akun baru. Fitur-fitur tersebut sangat memudahkan bagi pemilik toko untuk mengatur administrasi maupun keuangan yang ada.

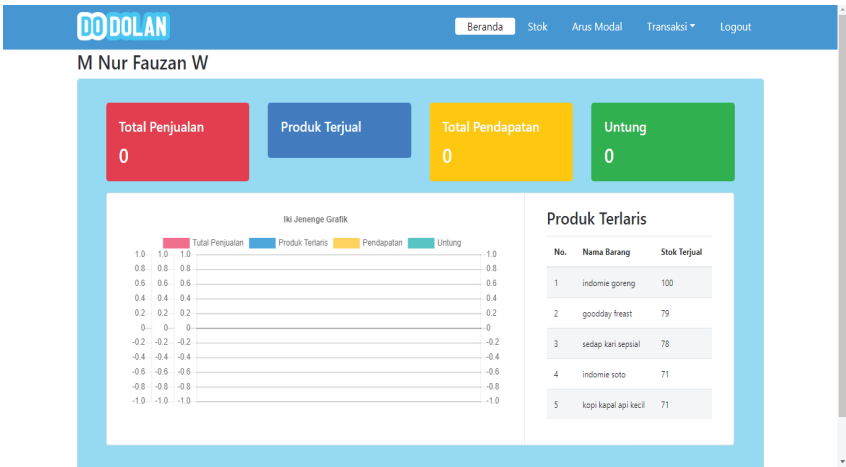
8.3 Perancangan Sistem

Perancangan sistem merupakan tahapan dalam melakukan blueprint atau kerangka kerja yang akan dilakukan dalam membangun sistem. Tahapan ini membantu pengembang dalam proses implementasi sistem. Dalam perancangan sistem ini dipilih desain mockup seperti pada gambar dibawah ini.



Gambar 27. Halaman Login

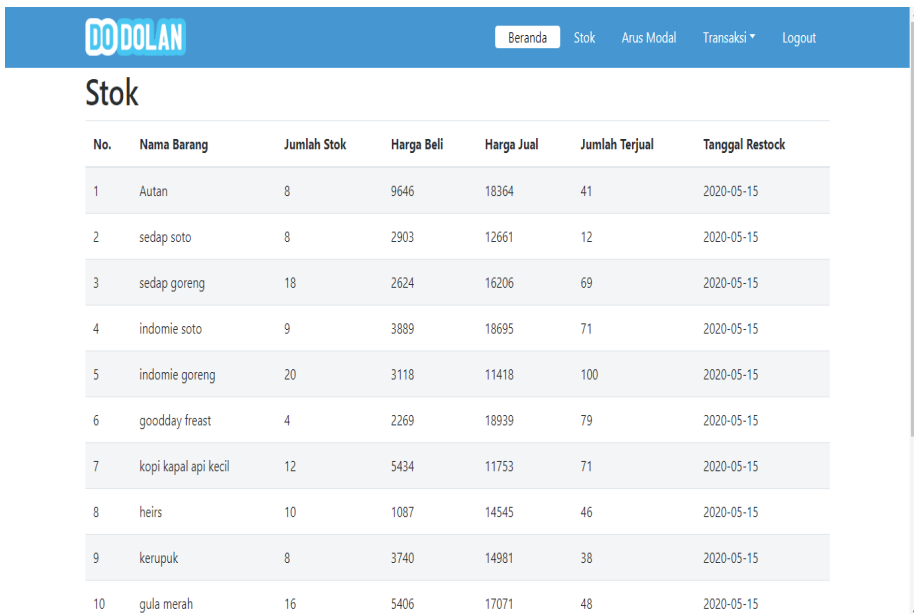
Halaman ini digunakan untuk Log In ke halaman utama atau beranda. Gambar 28 merupakan mockup halaman beranda. Pada halaman ini akan ditampilkan nama pengguna, grafik penjualan, total penjualan, produk terjual, total pendapatan, untung, juga daftar produk terlaris.



Gambar 28. Halaman Beranda

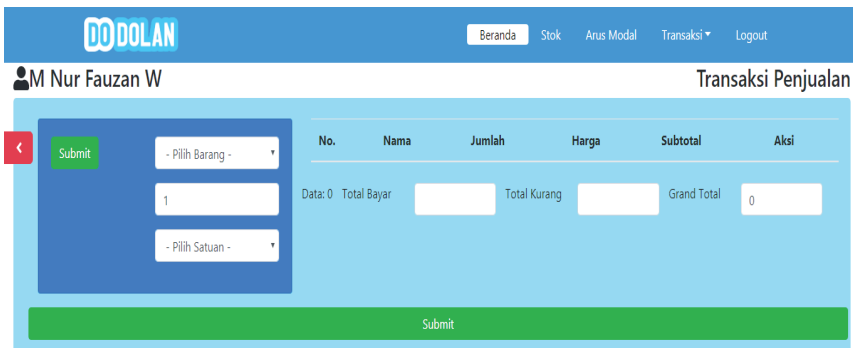
Pada Gambar 29 merupakan desain mocup untuk halaman stock dan daftar harga. Pada Halaman ini ditambahlkan stok ketersediaan


barang, daftar harga, jumlah terjual, serta tanggal restock dari barang tersebut.




No.	Nama Barang	Jumlah Stok	Harga Beli	Harga Jual	Jumlah Terjual	Tanggal Restock
1	Autan	8	9646	18364	41	2020-05-15
2	sedap soto	8	2903	12661	12	2020-05-15
3	sedap goreng	18	2624	16206	69	2020-05-15
4	indomie soto	9	3889	18695	71	2020-05-15
5	indomie goreng	20	3118	11418	100	2020-05-15
6	goodday freast	4	2269	18939	79	2020-05-15
7	kopi kapal api kecil	12	5434	11753	71	2020-05-15
8	heirs	10	1087	14545	46	2020-05-15
9	kerupuk	8	3740	14981	38	2020-05-15
10	gula merah	16	5406	17071	48	2020-05-15

Gambar 29. Halaman stock dan Daftar Harga




Beranda Stok Arus Modal Transaksi Logout

 M Nur Fauzan W
 Transaksi Penjualan

Submit

- Pilih Barang -

1

- Pilih Satuan -

No.	Nama	Jumlah	Harga	Subtotal	Aksi
Data: 0 Total Bayar <input type="text"/> Total Kurang <input type="text"/> Grand Total <input type="text"/>					

Submit

Gambar 30. Halaman Transaksi Penjualan

Gambar 30 merupakan desain mockup untuk tampilan halaman transaksi penjualan. Halaman ini digunakan untuk menginput barang yang terjual, dan akan disimpan otomatis ketika menekan

tombol submit. Sedangkan Gambar 31 adalah desain mockup untuk tampilan halaman transaksi pembelian. Halaman ini digunakan untuk menginput barang yang akan dijual, ataupun ketika melakukan restock barang.

DO DOLAN

BerandaStokAnus ModalTransaksi▼Logout

M Nur Fauzan W

Transaksi Pembelian

Submit

Nama Barang

Jumlah Barang

Satuan Barang

Harga Barang

Harga Jual

No.	Nama	Jumlah	Harga	Subtotal	Aksi
Data: 0 Total Bayar Total Kurang Grand Total 0					

Submit

Gambar 31. Halaman transaksi pembelian

Desain mockup selanjurnta ialah gambar desain mocup untuk tampilan halaman rekap. Halaman ini digunakan untuk melihat aktifitas pembelian dan penjualan.

DO DOLAN

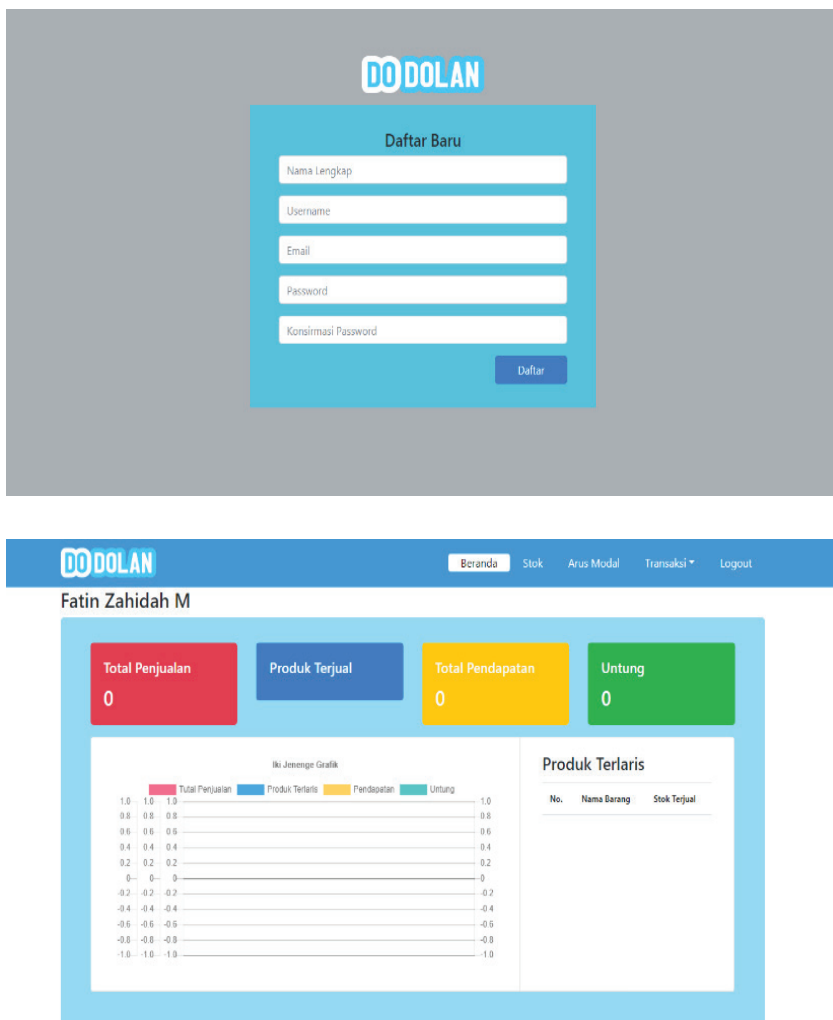
BerandaStokRekapTransaksi▼Logout

Rekap Pembelian dan Penjualan

Kode Transaksi	Tanggal	Total Harga	Total Bayar	Total Kurang	Catatan	Aksi
TRXBELI5764	2020-07-01	12500	0	0	kosong	

Gambar 32. Halaman Rekap

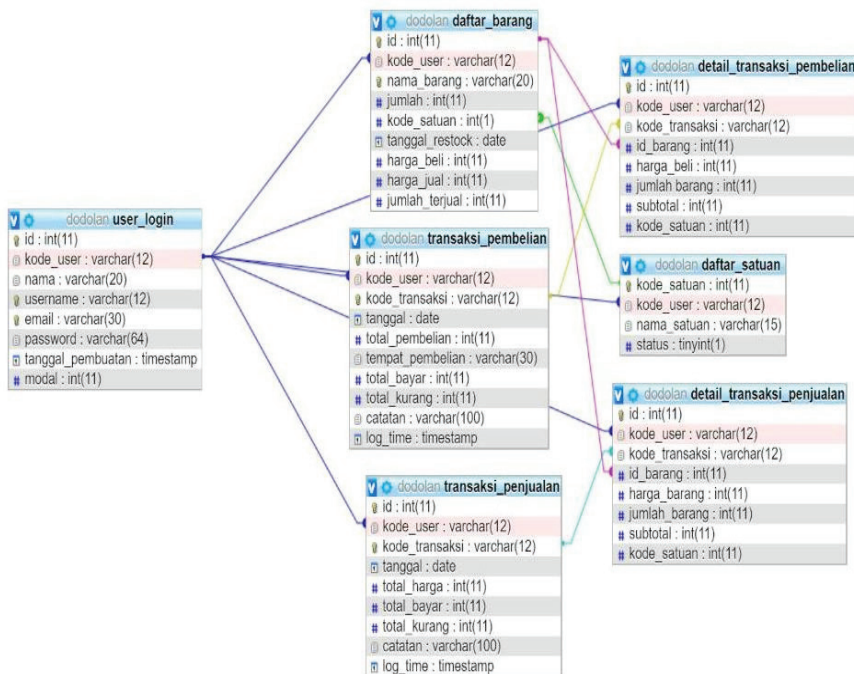
Dan yang terakhir untuk desain mockup halaman register. Halaman ini digunakan untuk menambah akun pada web.



Gambar 33. Halaman Register

Dalam membantu dalam proses penyimpanan data dalam sistem, kita perlu merancang database sehingga kita buat ERD atau entity relationship diagram. Didalam ERD tersusun atas entitas – entitas yang saling berhubungan. Implementasi program Dodolan ini menggunakan sistem CRUD. CRUD adalah akronim dari Create, Read, Update, dan Delete yang merupakan manipulasi data dasar untuk database. Create digunakan untuk membuat sebuah data baru dan menyimpannya ke database, seperti registrasi dan

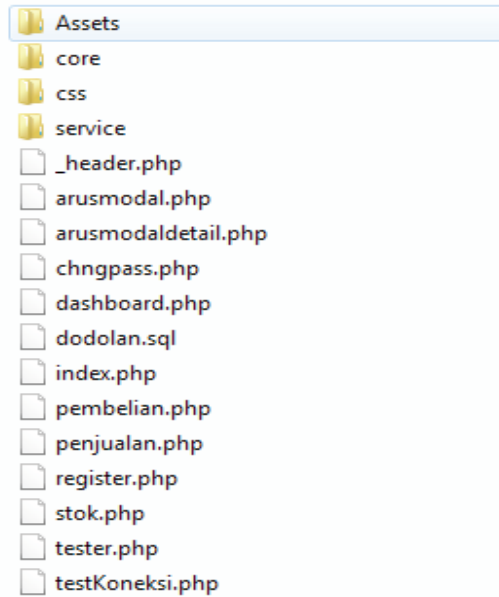
memasukkan data produk pada bagian transaksi pembelian. Read digunakan untuk membaca atau menampilkan suatu data yang tadinya berada didatabase MySQL, kemudian ditampilkan di WEB menggunakan bahasa pemrograman PHP. Update digunakan untuk mengedit sebuah data dari database seperti update dari stock terjual pada menu transaksi penjualan dan perubahan harga produk lewat menu transaksi pembelian. Delete memiliki fungsi yang hampir sama dengan Update akan tetapi proses ini adalah untuk melakukan penghapusan data di database melalui bahasa PHP, seperti pada menu arus modal. ERD dalam sistem ini dapat dilihat pada Gambar 34.



Gambar 34. ERD database sistem

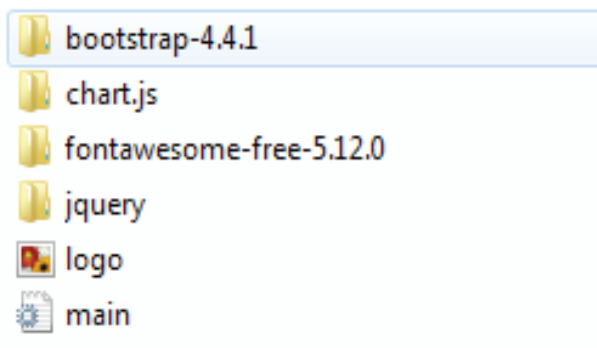
8.4 Implementasi

Sistem ini dibangun dengan menggunakan HTML, CSS, PHP, JavaScript, dan Bootstrap.



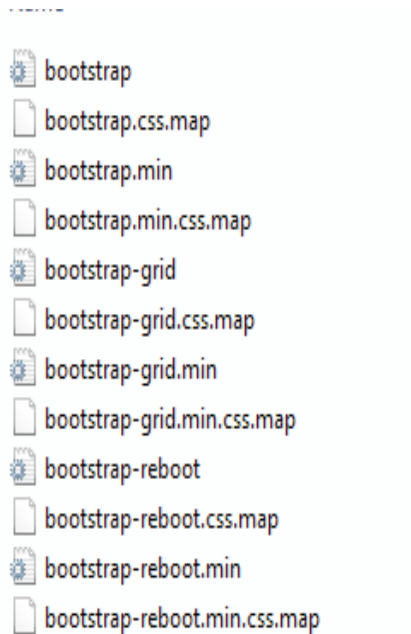
Gambar 35. File yang dibuat dalam pembangunan sistem

Pada gambar 35, dapat dilihat terdapat 17 file yang telah dibuat dalam membangun sistem ini yang terdiri dari 4 buah folder, 12 file .php dan 1 file .sql yang menyatakan databasenya. Dalam folder Assets didalamnya mengandung file bootstrap (lihat pada gambar 36). Dalam folder core terdapat file Database.php dan dalam folder css berisi file style.css. sedangkan dalam folder Service terdapat file ApiService.php dan dataDummy.php

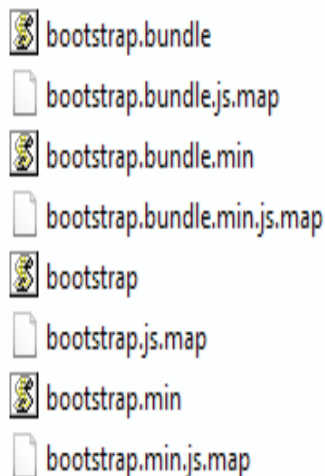


Gambar 36. Folder Assets

Dalam folder bootstrap terdapat 2 buah folder yakni css dan js. Dalam folder css terdapat 12 buah file yang dapat dilihat pada Gambar 37 sedangkan folder js berisi 8 buah file (Gambar 38)



Gambar 37. File dalam folder css



Gambar 38. File dalam folder js

Berikut penjabaran terkait pembangunan dan script yang dibuat:

- index.php

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-
scale=1.0">

  <meta http-equiv="X-UA-Compatible" content="ie=edge">

  <title>Document</title>

  <link rel="stylesheet" href="css/style.css">

  <link rel="stylesheet" href="Assets/bootstrap-4.4.1/css/bootstrap.
min.css">

  <script src="Assets/jquery/dist/jquery.js"></script>

  <script src="Assets/bootstrap-4.4.1/js/bootstrap.min.js"></script>

</head>

<body style="background: #A0A6AB !important;">

  <div class="headlogin">

  </div>

  <form id="loginForm" method="post" action="service/
ApiServices.php">

    <div class="login">
```



```

<div class="col-md-12">
    <div class="form-group">
        <input type="text" class="form-control" id="username"
name="username" tabindex="1"
        placeholder="Masukkan Email/Username" required
data-error="Please enter your email/username">
        <div class="help-block with-errors"></div>
    </div>
</div>
<div class="col-md-12">
    <div class="form-group">
        <input type="password" placeholder="Masukkan
Password" tabindex="2" id="password" class="form-control"
name="password"
        required data-error="Please enter your password">
        <div class="help-block with-errors"></div>
    </div>
</div>
<div class="col-md-12 row m-0">
    <div class="col-md-9 p-md-0">
        <?php if (@$_COOKIE['alert']){ ?>
            <span id="error" class="text-error text-
danger"><b>*<?=$_COOKIE['alert']?></b></span>
            <?php } ?>
            <span class="text-register">*Belum punya akun
? <a href="register.php" style="text-decoration: underline
!important">Register</a></span>
    </div>

```

```

        <input type="hidden" name="req" value="login">

        <button class="btn btn-primary col-md-3" tabindex="3"
id="submit" type="submit">Masuk</button>

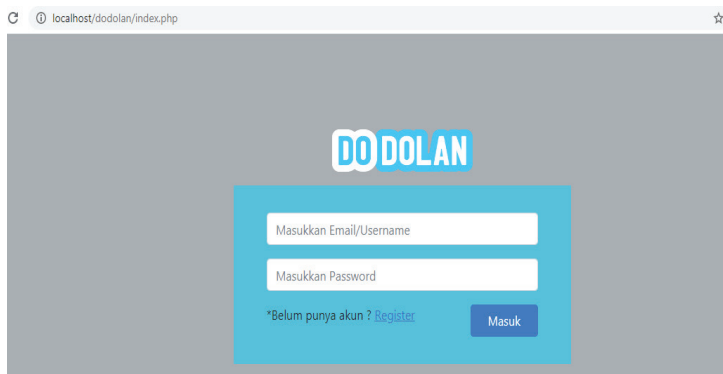
    </div>

</div>

</form>

<script>
    if ($("#error").length) {
        $(".text-register").css("display","none");
        $(".text-error").fadeOut(2000, 500).slideUp(500, function () {
            $(".text-register").slideToggle(500);
        });
    }
</script>
</body>
</html>

```



Gambar 39. Hasil tampilan index.php

- register.php

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-
scale=1.0">

  <meta http-equiv="X-UA-Compatible" content="ie=edge">

  <title>Document</title>

  <link rel="stylesheet" href="css/style.css">

  <link rel="stylesheet" href="Assets/bootstrap-4.4.1/css/bootstrap.
min.css">

  <script src="Assets/jquery/dist/jquery.js"></script>

  <script src="Assets/bootstrap-4.4.1/js/bootstrap.min.js"></script>

</head>

<body style="background: #A0A6AB !important;">

  <div class="headlogin mt-5">

  </div>

  <form id="loginForm" method="post" action="service/apiservices.
php">

    <div class="login">

      <h4 class="col-md-12 text-center">Daftar Baru</h4>

      <div class="col-md-12">
```

```

        <div class="form-group">
            <input type="text" class="form-control" id="nama"
name="nama" tabindex="1"
                placeholder="Nama Lengkap" required data-
error="Please enter your email/username">
            <div class="help-block with-errors"></div>
        </div>
    </div>
    <div class="col-md-12">
        <div class="form-group">
            <input type="text" placeholder="Username"
tabindex="2" id="username" class="form-control"
name="username"
                required data-error="Please enter your password">
            <div class="help-block with-errors"></div>
        </div>
    </div>
    <div class="col-md-12">
        <div class="form-group">
            <input type="email" placeholder="Email" tabindex="2"
id="email" class="form-control" name="email"
                data-error="Please enter your password">
            <div class="help-block with-errors"></div>
        </div>
    </div>
    <div class="col-md-12">
        <div class="form-group">

```

```

        <input type="password" placeholder="Password"
tabindex="2" id="password" class="form-control"
name="password"

        required data-error="Please enter your password">

        <div class="help-block with-errors"></div>

    </div>

</div>

<div class="col-md-12">

    <div class="form-group">

        <input type="password" placeholder="Konsfirmasi
Password" tabindex="2" id="konfirmasi_password" class="form-
control" name="konfirmasi_password"

        required data-error="Please enter your password">

        <div class="help-block with-errors"></div>

    </div>

</div>

<div class="col-md-12 row m-0">

    <div class="col-md-9 p-md-0">

        </div>

        <input type="hidden" name="req" value="register">

        <button class="btn btn-primary col-md-3" tabindex="3"
id="submit" type="submit">Daftar</button>

    </div>

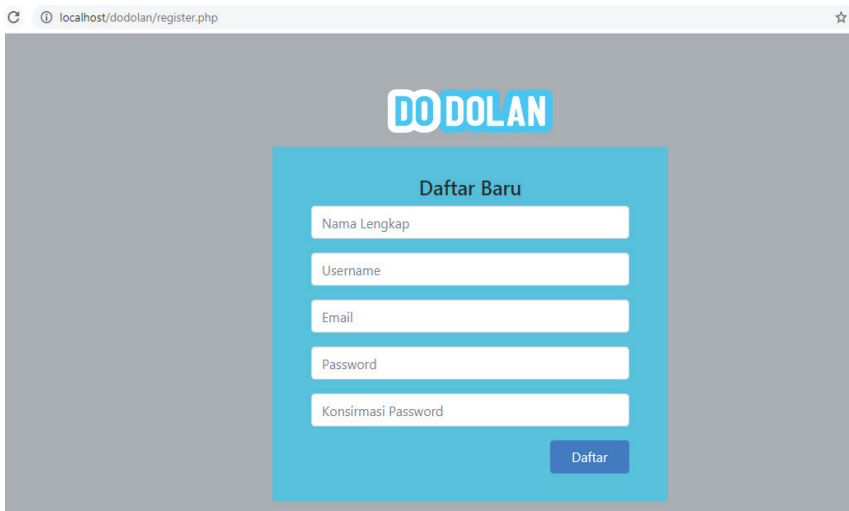
</div>

</form>

<script>

```

```
$("#submit").click(function(){  
    if ($("#password").val() != ($("#konfirmasi_password").val()) {  
        alert("Password Tidak Sama");  
        $("#konfirmasi_password").focus();  
        return false;  
    }  
    $("#loginForm").submit();  
})  
</script>  
</body>  
</html>
```

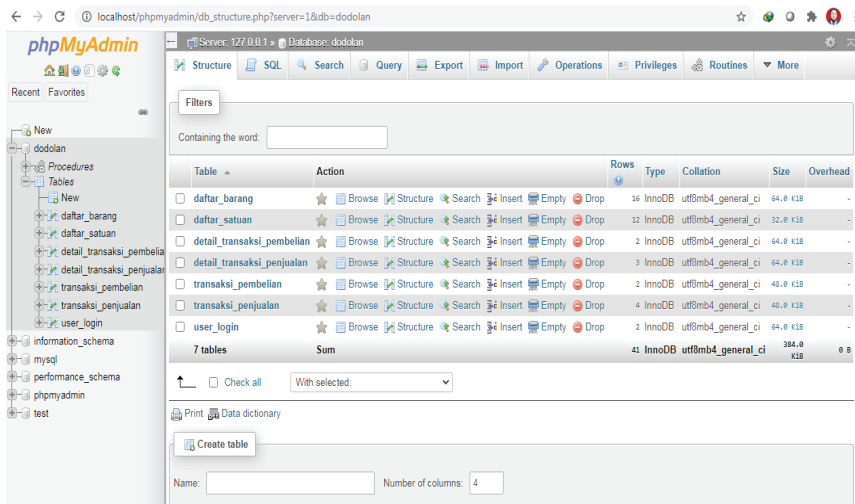


The screenshot shows a web browser window with the address bar displaying 'localhost/dodolan/register.php'. The page features a light blue header with the 'DODOLAN' logo. Below the header is a registration form titled 'Daftar Baru' (New Registration) on a teal background. The form contains five input fields: 'Nama Lengkap' (Full Name), 'Username', 'Email', 'Password', and 'Konfirmasi Password' (Confirm Password). A blue 'Daftar' (Register) button is located at the bottom right of the form.

Gambar 40. Register

Apabila sudah melakukan pendaftaran dalam form ini, pengguna akan dialihkan ke halaman dasbord melalui <http://localhost/dodolan/service/apiservices.php>. Pastikan bahwa sudah membuat database terlebih dahulu seperti ERD yang ditunjukkan pada

Gambar 33. Apabila telah memiliki database maka dalam <http://localhost/phpmyadmin/index.php> akan tampil database dodolan. Deskripsi database dodolan seperti pada Gambar dibawah.



Gambar 41. database dodolan

- _header.php

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Document</title>

    <link rel="stylesheet" href="css/style.css">

    <link rel="stylesheet" href="Assets/bootstrap-4.4.1/css/bootstrap.min.css">
```

```

<script src="Assets/jquery/dist/jquery.js"></script>

<script src="Assets/bootstrap-4.4.1/js/bootstrap.min.js"></script>

<script src="Assets/chart.js/dist/Chart.min.js"></script>

<script src="Assets/fontawesome-free-5.12.0/js/all.min.js"></script>

</head>

<body>

  <header class="top-navbar">

    <nav class="navbar navbar-expand-lg navbar-light">

      <div class="container">

        <a class="navbar-brand" href="dashboard.php">

          </a>

          <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbars-rs-food"
aria-controls="navbars-rs-food" aria-expanded="false" aria-
label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

          </button>

          <div class="collapse navbar-collapse" id="navbars-rs-
food">

            <ul class="navbar-nav ml-auto">

              <li class="nav-item active"><a class="nav-link"
href="dashboard.php">Beranda</a></li>

```



```

        <li class="nav-item"><a class="nav-link"
href="stok.php">Stok</a></li>

        <li class="nav-item"><a class="nav-link"
href="arusmodal.php">Arus Modal</a></li>

        <li class="nav-item dropdown">

            <a class="nav-link dropdown-toggle"
href="#" id="dropdown-a"

                data-toggle="dropdown">Transaksi</a>

            <div class="dropdown-menu" aria-
labelledby="dropdown-a">

                <a class="dropdown-item"
href="penjualan.php">Penjualan</a>

                <a class="dropdown-item"
href="pembelian.php">Pembelian</a>

            </div>

        </li>

        <li class="nav-item"><a class="nav-link"
href="service/apiservices.php?req=logout">Logout</a></li>

    </ul>

</div>

</div>

</nav>

</header>

```

- dasbord.php

```

<?php
include_once("_header.php");

```

```

require_once("core/Database.php");

session_start();

$db = new Database();

$totalPenjualan = $db->select("SELECT count(id) as jual FROM
transaksi_penjualan WHERE kode_user = '". $_SESSION['kode_
user']."' AND tanggal BETWEEN '".date('Y-m-01')." AND
"".date('Y-m-t')."");

$produkTerjual = $db->select("SELECT sum(a.jumlah_barang) as
jumlah FROM detail_transaksi_penjualan a, transaksi_penjualan
b WHERE a.kode_transaksi = b.kode_transaksi AND a.kode_user
= '". $_SESSION['kode_user']."' AND b.tanggal BETWEEN
"".date('Y-m-01')." AND '".date('Y-m-t')."");

$jumlahPendapatan = $db->select("SELECT sum(total_bayar)
as pendapatan FROM transaksi_penjualan WHERE kode_user
= '". $_SESSION['kode_user']."' AND tanggal BETWEEN
"".date('Y-m-01')." AND '".date('Y-m-t')."");

$totalUntung = $db->select("SELECT sum(a.subtotal - b.harga_beli
* a.jumlah_barang) AS untung FROM detail_transaksi_penjualan
a, daftar_barang b, transaksi_penjualan c WHERE a.kode_user
= '". $_SESSION['kode_user']."' AND a.id_barang = b.id AND
a.kode_transaksi = c.kode_transaksi AND c.tanggal BETWEEN
"".date('Y-m-01')." AND '".date('Y-m-t')."");

$dataGrafik = $db->select("SELECT COUNT(a.id) as jumlah_
transaksi, SUM(b.jumlah_barang) AS jumlah_barang, SUM(a.total_
bayar) AS pendapatan, sum(b.subtotal - c.harga_beli * b.jumlah_
barang) AS untung, MONTHNAME(a.tanggal) AS tanggal ,(a.
tanggal) as tgl

FROM transaksi_penjualan a, detail_transaksi_
penjualan b, daftar_barang c

WHERE a.kode_user = '". $_SESSION['kode_
user']."'

AND a.kode_transaksi = b.kode_transaksi

AND b.id_barang = c.id

GROUP BY MONTH(tgl)");

```

```
$produkTerlaris = $db->select("SELECT nama_barang,
jumlah_terjual FROM daftar_barang WHERE kode_user = '". $_SESSION['kode_user']."' ORDER BY jumlah_terjual desc limit 5 ");
```

```
?>
```

```
<div class="container">
```

```
  <div class="clearfix">
```

```
    <h3><?=@$_SESSION['nama']?></h3>
```

```
  </div>
```

```
  <div class="jumbotron" style="background: rgba(0, 204, 255,0.5)">
```

```
    <div class="container">
```

```
      <div class="col-md-12 pl-0 pr-0" style="margin-top:-30px">
```

```
        <div class="row">
```

```
          <div class="col-md-3">
```

```
            <div class="card border-danger mb-3">
```

```
              <div class="card-body bg-danger text-light">
```

```
                <h5 class="card-title">Total Penjualan</h5>
```

```
                <h3 class="card-text"><?=$totalPenjualan[0]
['jual']?></h3>
```

```
              </div>
```

```
            </div>
```

```
          </div>
```

```
        <div class="col-md-3">
```

```
      <div class="card border-primary mb-3">
```

```
        <div class="card-body bg-primary text-light">
```

```

        <h5 class="card-title">Produk Terjual</h5>
        <h3 class="card-text"><?=$produkTerjual[0]
["jumlah"]?></h3>
    </div>
</div>
</div>
<div class="col-md-3">
    <div class="card border-warning mb-3">
        <div class="card-body bg-warning text-light">
            <h5 class="card-title">Total Pendapatan</h5>
            <h3 class="card-text"><?=number_
format($jumlahPendapatan[0]["pendapatan"],0,"",",",".")?></h3>
        </div>
    </div>
</div>
<div class="col-md-3">
    <div class="card border-success mb-3">
        <div class="card-body bg-success text-light
warning">
            <h5 class="card-title">Untung</h5>
            <h3 class="card-text"><?=number_
format($totalUntung[0]['untung'],0,"",",",".")?></h3>
        </div>
    </div>
</div>
</div>

```

```

<div class="card col-md-8">
  <div class="card-body">
    <!-- <h4 class="card-title">Title</h4> -->
    <canvas id="canvas"></canvas>
  </div>
</div>

<div class="card col-md-4">
  <div class="card-body">
    <h4 class="card-title">Produk Terlaris</h4>
    <table class="table table-striped" style="font-
size:12px">
      <thead>
        <tr>
          <th scope="col">No.</th>
          <th scope="col">Nama Barang</th>
          <th scope="col">Stok Terjual</th>
        </tr>
      </thead>
      <tbody>
        <?php $no=1; foreach($produkTerlaris as $data)
{ ?>
          <tr>
            <td><?=$no?></td>
            <td><?=$data['nama_barang']?></td>
            <td><?=$data['jumlah_terjual']?></td>

```

```

</tr>

        <?php $no++; } ?>
    </tbody>
</table>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

<script>
    var lineChartData = {
        labels: [<?php foreach($dataGrafik as $data){ echo
        "".$data["tanggal"].","; }?>],
        datasets: [{
            label: 'Tatal Penjualan',
            borderColor: 'rgb(255, 99, 132)',
            backgroundColor: 'rgb(255, 99, 132)',
            fill: false,
            data: [
                <?php foreach($dataGrafik as $data){ echo
                "".$data["jumlah_transaksi"].","; }?>
            ],
            yAxisID: 'y-axis-1',

```

```

    }, {
        label: 'Produk Terlaris',
        borderColor: 'rgb(54, 162, 235)',
        backgroundColor: 'rgb(54, 162, 235)',
        fill: false,
        data: [
            <?php foreach($dataGrafik as $data){ echo
            "".$data["jumlah_barang"].""; }?>
        ],
        yAxisID: 'y-axis-2'
    }, {
        label: 'Pendapatan',
        borderColor: 'rgb(255, 205, 86)',
        backgroundColor: 'rgb(255, 205, 86)',
        fill: false,
        data: [
            <?php foreach($dataGrafik as $data){ echo
            "".$data["pendapatan"].""; }?>
        ],
        yAxisID: 'y-axis-3'
    }, {
        label: 'Untung',
        borderColor: 'rgb(75, 192, 192)',
        backgroundColor: 'rgb(75, 192, 192)',
        fill: false,

```

```

        data: [
            <?php foreach($dataGrafik as $data){ echo
            "".$data["untung"].""; }?>
        ],
        yAxisID: 'y-axis-4'
    ]
};

window.onload = function() {
    var ctx = document.
    getElementById('canvas').getContext('2d');
    window.myLine = Chart.Line(ctx, {
        data: lineChartData,
        options: {
            responsive: true,
            hoverMode: 'index',
            stacked: false,
            title: {
                display: true,
                text: 'Iki Jenenge Grafik'
            },
            scales: {
                yAxes: [{
                    type: 'linear',
                    display: true,

```



```

        position: 'left',
        id: 'y-axis-1',
    }, {
        type: 'linear',
        display: true,
        position: 'left',
        id: 'y-axis-2',
    }, {
        type: 'linear',
        display: true,
        position: 'left',
        id: 'y-axis-3',
    }, {
        type: 'linear',
        display: true,
        position: 'right',
        id: 'y-axis-4',
    },
    // grid line settings
    gridLines: {
        drawOnChartArea:
false, // only want the grid lines for one axis to show up
    },
    },
}
}

```

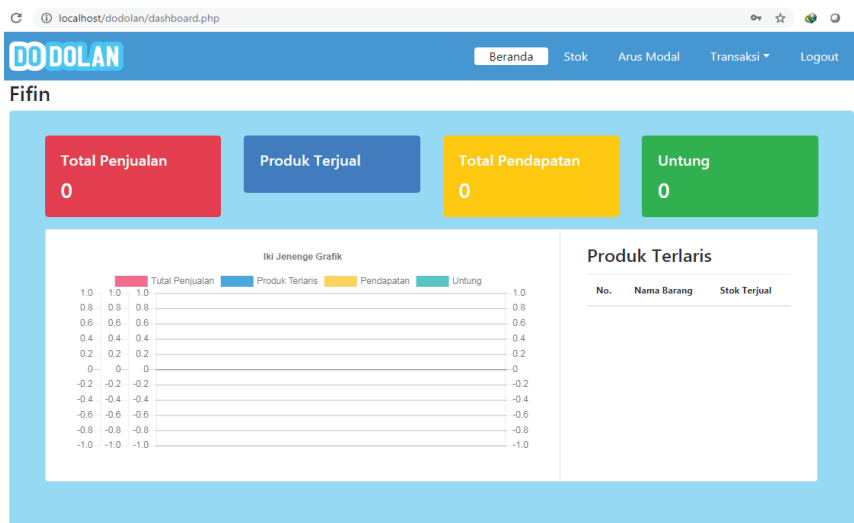
```

    });

};

</script>
</body>
</html>

```



Gambar 42. Dasbord

- stok.php

```

<?php
include_once("_header.php");
require_once("core/Database.php");
session_start();

$db = new Database();

$dataStock = $db->select("SELECT * FROM daftar_barang WHERE
kode_user = '".$_SESSION['kode_user']."' ORDER BY tanggal_

```

```

restock desc");
?>

<div class="container">
    <div class="stok">
        <h1>Stok</h1>
    </div>
    <table class="table table-striped">
        <thead>
            <tr>
                <th scope="col">No.</th>
                <th scope="col">Nama Barang</th>
                <th scope="col">Jumlah Stok</th>
                <th scope="col">Harga Beli</th>
                <th scope="col">Harga Jual</th>
                <th scope="col">Jumlah Terjual</th>
                <th scope="col">Tanggal Restock</th>
            </tr>
        </thead>
        <tbody>
            <?php $no=1; foreach($dataStock as $data){
                <tr>
                    <td scope="row"><?=$no?></td>

```

```

<td><?=$data["nama_barang"]?></td>

        <td><?=$data["jumlah"]?></td>
        <td><?=$data["harga_beli"]?></td>
        <td><?=$data["harga_jual"]?></td>
        <td><?=$data["jumlah_terjual"]?></td>
        <td><?=$data["tanggal_restock"]?></td>
    </tr>

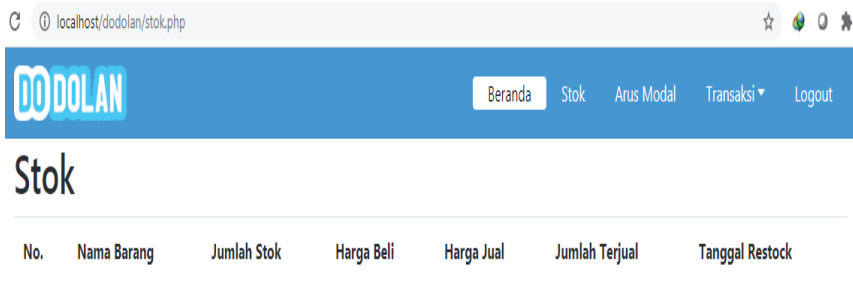
    <?php $no++; } ?>

</tbody>

</table>

</div>

```



Gambar 43. Stok

- arusmodal.php

```

<?php
include_once("_header.php");
require_once("core/Database.php");
session_start();
$db = new Database();

```

```

$kode = $_GET['kode'];

$tiipe = substr($kode,3,4);

$dataTable = "";

if ($tiipe == "BELI") {

    $dataTable = $db->select("SELECT a.kode_transaksi,a.harga_
    beli,a.jumlah_barang,a.subtotal,a.kode_satuan,b.nama_barang,c.
    nama_satuan FROM detail_transaksi_pembelian a, daftar_barang
    b, daftar_satuan c WHERE a.kode_transaksi = '$kode' AND a.id_
    barang = b.id AND a.kode_satuan = c.kode_satuan");

} else {

    $dataTable = $db->select("SELECT a.kode_transaksi,a.harga_
    barang as harga_beli,a.jumlah_barang,a.subtotal,a.kode_satuan,b.
    nama_barang,c.nama_satuan FROM detail_transaksi_penjualan
    a, daftar_barang b, daftar_satuan c WHERE a.kode_transaksi =
    '$kode' AND a.id_barang = b.id AND a.kode_satuan = c.kode_
    satuan");

}

?>

<div class="container">

    <div class="arus row">

        <button id="back" class="btn btn-dark btn-lg col-md-1"
        style="height:40px;padding-top:3px;margin-top:7px"><i class="fas
        fa-angle-left"></i></button><h1>Detail Arus Modal</h1>

    </div>

    <table class="table table-striped">

    <thead>

    <tr>

        <th scope="col">Kode Transaksi</th>

        <th scope="col">Nama Barang</th>

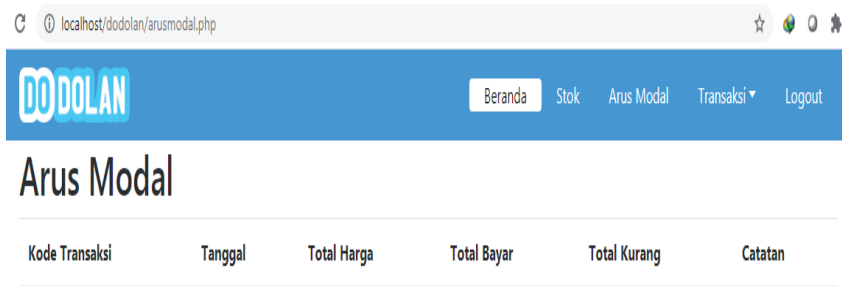
```

```

        <th scope="col">Jumlah Barang</th>
        <th scope="col">satuan</th>
        <th scope="col">Harga Barang</th>
        <th scope="col">Total Harga</th>
    </tr>
</thead>
<tbody>
    <?php foreach($dataTable as $row){?>
    <tr>
        <th scope="row"><?=$row['kode_transaksi']?></th>
        <td><?=$row['nama_barang']?></td>
        <td><?=$row['jumlah_barang']?></td>
        <td><?=$row['nama_satuan']?></td>
        <td><?=$row['harga_beli']?></td>
        <td><?=$row['subtotal']?></td>
    </tr>
    <?php } ?>
</tbody>
</table>
</div>

<script>
    $("#back").click(function(){
        history.back();
    })
</script>

```



Gambar 44. Arus Modal

- arusmodaldetail.php

```
<?php
include_once("_header.php");
require_once("core/Database.php");
session_start();
$db = new Database();
$kode = $_GET['kode'];
$tiipe = substr($kode,3,4);
$dataTable = "";
if ($tiipe == "BELI") {
    $dataTable = $db->select("SELECT a.kode_transaksi,a.harga_
    beli,a.jumlah_barang,a.subtotal,a.kode_satuan,b.nama_barang,c.
    nama_satuan FROM detail_transaksi_pembelian a, daftar_barang
    b, daftar_satuan c WHERE a.kode_transaksi = '$kode' AND a.id_
    barang = b.id AND a.kode_satuan = c.kode_satuan");
} else {
    $dataTable = $db->select("SELECT a.kode_transaksi,a.harga_
    barang as harga_beli,a.jumlah_barang,a.subtotal,a.kode_satuan,b.
    nama_barang,c.nama_satuan FROM detail_transaksi_penjualan a,
    daftar_barang b, daftar_satuan c WHERE
    a.kode_transaksi = '$kode' AND a.id_barang = b.id AND a.kode_
    satuan = c.kode_satuan");
```

```

}
?>
<div class="container">
    <div class="arus row">
        <button id="back" class="btn btn-dark btn-lg col-md-1"
        style="height:40px;padding-top:3px;margin-top:7px"><i class="fas
        fa-angle-left"></i></button><h1>Detail Arus Modal</h1>
    </div>
    <table class="table table-striped">
<thead>
    <tr>
        <th scope="col">Kode Transaksi</th>
        <th scope="col">Nama Barang</th>
        <th scope="col">Jumlah Barang</th>
        <th scope="col">satuan</th>
        <th scope="col">Harga Barang</th>
        <th scope="col">Total Harga</th>
    </tr>
</thead>
<tbody>
    <?php foreach($dataTable as $row){?>
    <tr>
        <th scope="row"><?=$row['kode_transaksi']?></th>
        <td><?=$row['nama_barang']?></td>
        <td><?=$row['jumlah_barang']?></td>
        <td><?=$row['nama_satuan']?></td>

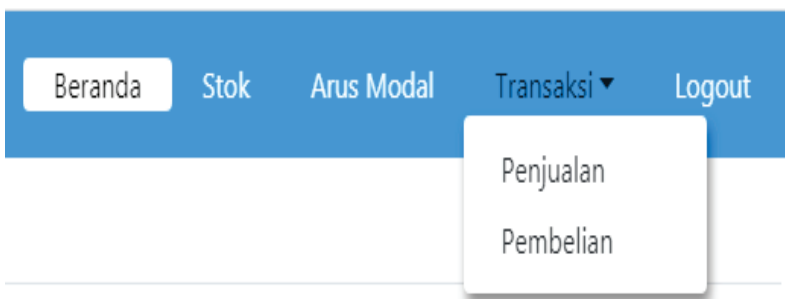
```



```
<td><?=$row['harga_beli']?></td>
    <td><?=$row['subtotal']?></td>
</tr>
<?php } ?>
</tbody>
</table>
</div>

<script>
    $("#back").click(function(){
        history.back();
    })
</script>
```

- Transaksi
Transaksi dalam sistem ini dibagi menjadi 2 yakni transaksi penjualan dan pembelian (Gambar 42)



Gambar 45. Transaksi

1. penjualan.php

```

<?php
include_once("_header.php");
require_once("core/Database.php");
session_start();

$db = new Database();

$listBarang = $db->select("SELECT a.id,a.nama_barang,a.jumlah,a.
harga_jual,a.kode_satuan FROM daftar_barang a WHERE a.kode_
user = '". $_SESSION['kode_user']."'");

$listSatuan = $db->select("SELECT kode_satuan, nama_satuan
FROM daftar_satuan WHERE kode_user = '". $_SESSION['kode_
user']."'");

?>

<button id="btnCollapse" class="btn btn-lg btn-danger pl-5
position-absolute"

    style="z-index:9999;left:-30px;margin-top:80px"><i id="icon"
class="fas fa-angle-left float-right"></i></button>

<div class="container-fluid">

    <div class="clearfix">

        <h3 class="float-left"><i class="fas fa-user"></i><?=@$_
SESSION['nama']?></h3>

        <h3 class="float-right">Transaksi Penjualan</h3>

    </div>

    <div class="jumbotron pl-4 pr-1 pt-4 mb-0 pb-3"
style="background: rgba(0, 204, 255,0.5)">

        <div class="col-md-12">

            <div class="row">

                <div class="col-md-4" id="tabInput">

```

```

<div class="card text-white bg-primary">
  <div class="card-body">
    <div class="float-right">
      <div class="form-group">
        <select id="nama_barang" name="nama_
barang" class="form-control">
          <option>- Pilih Barang </option>
          <?php foreach($listBarang as $data){ if
($data['jumlah'] == 0){?>
            <option label="<?=$data['nama_
barang']?> (Stock Kosong)" value="<?=$data['id']?>"
jumlah="<?=$data['jumlah']?>" harga="<?=$data['harga_jual']?>"
satuan="<?=$data['kode_satuan']?>" disabled="disabled">
              <?php } ?>
            <option label="<?=$data['nama_barang']?>"
value="<?=$data['id']?>" jumlah="<?=$data['jumlah']?>"
harga="<?=$data['harga_jual']?>" satuan="<?=$data['kode_
satuan']?>">
              <?php } ?>
            </select>
          </div>
          <div class="form-group">
            <input id="jumlah_barang" type="number"
name="jumlah_barang" class="form-control"
              placeholder="Jumlah Barang" value="1">
          </div>
          <div class="form-group">
            <select id="satuan" name="satuan"
class="form-control">

```

```

        <option>- Pilih Satuan -</option>

        <?php foreach($listSatuan as $data){ ?>
            <option label="<?=$data['nama_satuan']?>"
value="<?=$data['kode_satuan']?>">

            <?php } ?>

        </select>

    </div>

</div>

<div class="float-left">

    <button class="btn btn-success btn-md"
type="submit" id="submitBarang"> Submit</button>

</div>

</div>

</div>

</div>

<div id="tabTable" class="col-md-8 table-responsive">

    <form class="" action="service/apiservices.
php?req=transaksi_penjualan" method="post">

        <div class="table-responsive mb-2" style="max-
height:300px">

            <input type="hidden" id="counter" value="0">

            <table class="table table-striped overflow-auto">

                <thead>

                    <tr>

                        <th scope="col">No.</th>

                        <th scope="col">Nama</th>

```

```

        <th scope="col">Jumlah</th>
        <th scope="col">Harga</th>
        <th scope="col">Subtotal</th>
        <th scope="col">Aksi</th>
    </tr>
</thead>

<tbody>

</tbody>
</table>
</div>
<div id="pembayaran" class="col-md-13 row">
    <p id="totalData">Data: 0</p>
    <div class="form-group row col-md-4">
        <label class="col-md-6">Total Bayar</label>
        <input required type="text" name="total_bayar"
id="totBayar" class="form-control col-md-6">
    </div>
    <div class="form-group row col-md-4">
        <label class="col-md-6">Total Kurang</label>
        <input type="text" name="total_kurang"
id="totKurang" class="form-control col-md-6">
    </div>
    <div class="form-group row col-md-4">

```

```

        <label class="col-md-6">Grand Total</label>
        <input value="0" type="text" name="grand_total"
id="total" class="form-control col-md-6">

    </div>

</div>

</div>

    <button class="btn btn-block btn-success mt-3"
type="submit">Submit</button>

</form>

</div>

</div>

</div>

</div>

</div>

</div>

<script>

function deleteData(counter) {

    let count = counter;

    $(".btnDelete").click(function () {

        let id = $(this).data("id");

        $("##" + id).remove();

        count--

        $("#totalData").html("Data: " + count);

    })

}

$("#totBayar").keyup(function () {

```

```
$("#totKurang").val($("#total").val() - $(this).val());
});

$('#btnCollapse').on('click', function () {
    if ($(this).attr('data-click-state') == 1) {
        $(this).attr('data-click-state', 0);
        $("#tabInput").fadeIn(500);
        $(this).find($("#icon")).removeClass("fa-angle-right").
        addClass("fa-angle-left");
        $("#tabTable").removeClass("col-md-12").addClass("col-
        md-8");
        $("#pembayaran").removeClass("col-md-12").addClass("col-
        md-13");
    } else {
        $(this).attr('data-click-state', 1);
        $("#tabInput").fadeOut(500);
        $(this).find($("#icon")).removeClass("fa-angle-left").
        addClass("fa-angle-right");
        $("#tabTable").removeClass("col-md-8").addClass("col-
        md-12");
        $("#pembayaran").removeClass("col-md-13").addClass("col-
        md-12");
    }
});

$("#nama_barang").change(function(){
    let satuan = $("#nama_barang option[value='"+ $("#nama_
    barang").val() + "'").attr('satuan');
    $("#satuan").val(satuan);
```

```

});

$("#jumlah_barang").on("change keyup",function(e){

    let jumlah_barang = $("#nama_barang option[value='" +
    $('#nama_barang').val() + "'").attr('jumlah');

    let val = parseInt($(this).val());

    if (val > jumlah_barang) {

        $(this).val(jumlah_barang);

        alert("Stock Barang Tidak Mencukupi");

    }

})

$("#submitBarang").click(function () {

    let counter = $("#counter").val()

    counter++

    let grand_total = $("#total").val()

    let id_barang = $("#nama_barang").val();

    let nama_barang = $("#nama_barang option[value='" +
    $('#nama_barang').val() + "'").attr('label');

    let jumlah = $("#nama_barang option[value='" + $('#nama_
barang').val() + "'").attr('jumlah');

    let harga_barang = $("#nama_barang option[value='" +
    $('#nama_barang').val() + "'").attr('harga');

    console.log(id_barang);

    let jumlah_barang = $("#jumlah_barang").val();

    let satuan_value = $("#satuan").val();

```



```

let satuan = $("#satuan option[value='" + $('#satuan').val() +
    ""']").attr('label');

let data = `

    <tr id="data` + counter + `">

        <th>` + counter + `</th>

        <td>` + nama_barang + `</td>

        <input type="hidden" name="id_barang[]" value="` +
id_barang + `">

        <input type="hidden" name="nama_barang[]" value="` +
nama_barang + `">

        <td>` + jumlah_barang + ` ` + satuan + `</td>

        <input type="hidden" name="jumlah_barang[]" value="`
+ jumlah_barang + `">

        <input type="hidden" name="satuan[]" value="` +
satuan_value + `">

        <td>` + harga_barang + `</td>

        <input type="hidden" name="harga_barang[]" value="` +
harga_barang + `">

        <td>` + (jumlah_barang * harga_barang) + `</td>

        <input type="hidden" name="subtotal[]" value="` +
(jumlah_barang * harga_barang) + `">

        <td><button class="btn btn-sm btn-danger btnDelete"
data-id="data` + counter + `"><i class="fas fa-trash"></i></
button></td>

    </tr>

`;

$("#tbody").append(data);

$("#totalData").html("Data: " + counter);

deleteData(counter);

```

```

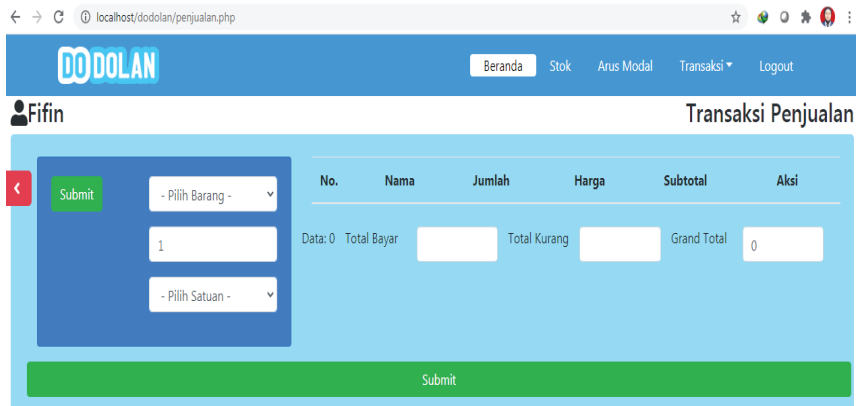
        $("#counter").val(counter);

        $("#total").val(parseInt(grand_total) + (jumlah_barang * harga_
barang));

    });
</script>
</body>

</html>

```



Gambar 46. Transaksi Penjualan

2. pembelian.php

```

<?php
include_once("_header.php");
require_once("core/Database.php");
session_start();

$db = new Database();

$listBarang = $db->select("SELECT

```

```
a.id,a.nama_barang,a.jumlah,a.harga_jual,a.kode_satuan FROM
daftar_barang a WHERE a.kode_user = ".$_SESSION['kode_
user'].");
```

```
$listSatuan = $db->select("SELECT kode_satuan, nama_satuan
FROM daftar_satuan WHERE kode_user = ".$_SESSION['kode_
user'].");
```

```
?>
```

```
<button id="btnCollapse" class="btn btn-lg btn-danger pl-5
position-absolute"
```

```
style="z-index:9999;left:30px;margin-top:80px"><i id="icon"
class="fas fa-angle-left float-right"></i></button>
```

```
<div class="container-fluid">
```

```
<div class="clearfix">
```

```
<h3 class="float-left"><i class="fas fa-user"></i><?=@$_
SESSION['nama']?></h3>
```

```
<h3 class="float-right">Transaksi Pembelian</h3>
```

```
</div>
```

```
<div class="jumbotron pl-4 pr-1 pt-4 mb-0 pb-3"
style="background: rgba(0, 204, 255,0.5)">
```

```
<div class="col-md-12">
```

```
<div class="row">
```

```
<div class="col-md-4" id="tabInput">
```

```
<div class="card text-white bg-primary">
```

```
<div class="card-body">
```

```
<div class="float-right">
```

```
<div class="form-group">
```

```
<input id="nama_barang" class="form-control"
list="listBarang" name="nama_barang"
```

```

        placeholder="Nama Barang">

<datalist id="listBarang">

    <?php foreach($listBarang as $data){ if
($data['jumlah'] == 0){?>

        <option value="<?=$data['nama_
barang']?> (Stock Kosong)" label="<?=$data['id']?>"
jumlah="<?=$data['jumlah']?>" harga="<?=$data['harga_jual']?>"
satuan="<?=$data['kode_satuan']?>" disabled="disabled">

        <?php } ?>

        <option value="<?=$data['nama_barang']?>"
label="<?=$data['id']?>" jumlah="<?=$data['jumlah']?>"
harga="<?=$data['harga_jual']?>" satuan="<?=$data['kode_
satuan']?>">

        <?php } ?>

</datalist>

</div>

<div class="form-group">

    <input id="jumlah_barang" type="number"
name="jumlah_barang" class="form-control"

        placeholder="Jumlah Barang">

</div>

<div class="form-group">

    <input id="satuan" list="listSatuan"
class="form-control" name="satuan"

        placeholder="Satuan Barang">

<datalist id="listSatuan">

    <?php foreach($listSatuan as $data){ ?>

        <option value="<?=$data['nama_satuan']?>"
label="<?=$data['kode_satuan']?>">

```

```

        <?php } ?>
    </datalist>
</div>
<div class="form-group">
    <input id="harga_barang" type="number"
class="form-control" name="harga_barang"
        placeholder=" Harga Barang">
</div>
<div class="form-group">
    <input id="harga_jual" type="number"
class="form-control" name="harga_jual"
        placeholder=" Harga Jual">
</div>
</div>
<div class="float-left">
    <button class="btn btn-success btn-md"
type="submit" id="submitBarang"> Submit</button>
</div>
</div>
</div>
<div id="tabTable" class="col-md-8 table-responsive">
    <form class="" action="service/apiservices.
php?req=transaksi_pembelian" method="post">
        <div class="table-responsive mb-2" style="max-
height:300px">
            <input type="hidden" id="counter" value="0">

```

```

<table class="table table-striped overflow-auto">
  <thead>
    <tr>
      <th scope="col">No.</th>
      <th scope="col">Nama</th>
      <th scope="col">Jumlah</th>
      <th scope="col">Harga</th>
      <th scope="col">Subtotal</th>
      <th scope="col">Aksi</th>
    </tr>
  </thead>

  <tbody>

</tbody>
</table>
</div>
<div id="pembayaran" class="col-md-13 row">
  <p id="totalData">Data: 0</p>
  <div class="form-group row col-md-4">
    <label class="col-md-6">Total Bayar</label>
    <input type="text" name="total_bayar"
id="totBayar" class="form-control col-md-6">
  </div>
  <div class="form-group row col-md-4">

```

```

        <label class="col-md-6">Total Kurang</label>

        <input type="text" name="total_kurang"
id="totKurang" class="form-control col-md-6">

    </div>

    <div class="form-group row col-md-4">

        <label class="col-md-6">Grand Total</label>

        <input value="0" type="text" name="grand_
total" id="total" class="form-control col-md-6">

    </div>

</div>

</div>

    <button class="btn btn-block btn-success mt-3"
type="submit">Submit</button>

</form>

</div>

</div>

</div>

</div>

<script>
function deleteData(counter) {
    let count = counter;

    $(".btnDelete").click(function () {
        let id = $(this).data("id");
        $("##" + id).remove();
    });
}

```

```

        count--
        $("#totalData").html("Data: " + count);
    })
}

$("#totBayar").keyup(function () {
    $("#totKurang").val($("#total").val() - $(this).val());
});

$("#btnCollapse").on('click', function () {
    if ($(this).attr('data-click-state') == 1) {
        $(this).attr('data-click-state', 0);
        $("#tabInput").fadeIn(500);
        $(this).find($("#icon")).removeClass("fa-angle-right").
addClass("fa-angle-left");
        $("#tabTable").removeClass("col-md-12").addClass("col-
md-8");
        $("#pembayaran").removeClass("col-md-12").addClass("col-
md-13");
    } else {
        $(this).attr('data-click-state', 1);
        $("#tabInput").fadeOut(500);
        $(this).find($("#icon")).removeClass("fa-angle-left").
addClass("fa-angle-right");
        $("#tabTable").removeClass("col-md-8").addClass("col-
md-12");
        $("#pembayaran").removeClass("col-md-13").addClass("col-
md-12");
    }
}

```



```

});

$("#nama_barang").change(function(){
    let satuan = $("#nama_barang option[value='"+ $("#nama_barang").val() + "']").attr('satuan');
    $("#satuan").val(satuan);
});

$("#submitBarang").click(function () {
    let counter = $("#counter").val()
    counter++
    let grand_total = $("#total").val()
    let nama_barang = $("#nama_barang").val();
    let id_barang = $("#listBarang option[value='"+ $("#nama_barang").val() + "']").attr('label');
    let jumlah_barang = $("#jumlah_barang").val();
    let satuan = $("#satuan").val();
    let satuan_value = $("#listSatuan option[value='"+ $("#satuan").val() + "']").attr('label');
    let harga_barang = $("#harga_barang").val();
    let harga_jual = $("#harga_jual").val();
    let data = `
        <tr id="data` + counter + `">
            <th>` + counter + `</th>
            <td>` + nama_barang + `</td>
            <input type="hidden" name="id_barang[]" value="` + id_barang + `">

```

```

<input type="hidden" name="nama_barang[]" value="` + nama_
barang + "`">

        <td>` + jumlah_barang + ` ` + satuan + `</td>

        <input type="hidden" name="jumlah_barang[]" value="` +
jumlah_barang + "`">

        <input type="hidden" name="satuan[]" value="` + satuan_
value + "`">

        <td>` + harga_barang + `</td>

        <input type="hidden" name="harga_barang[]" value="` +
harga_barang + "`">

        <input type="hidden" name="harga_jual[]" value="` +
harga_jual + "`">

        <td>` + (jumlah_barang * harga_barang) + `</td>

        <input type="hidden" name="subtotal[]" value="` +
(jumlah_barang * harga_barang) + "`">

        <td><button class="btn btn-sm btn-danger btnDelete" data-
id="data` + counter + "`"><i class="fas fa-trash"></i></button></td>

    </tr>

    `;

    $("tbody").append(data);

    $("#totalData").html("Data: " + counter);

    deleteData(counter);

    $("#counter").val(counter);

    $("#total").val(parseInt(grand_total) + (jumlah_barang * harga_
barang));

    });

</script>

</body>

</html>

```

The screenshot shows a web browser at the URL `localhost/dodolan/pembelian.php`. The application has a blue header with the logo 'DODOLAN' and navigation links: Beranda, Stok, Anus Modal, Transaksi (dropdown), and Logout. The user is logged in as 'Fifin'. The main section is titled 'Transaksi Pembelian' and contains a form with a sidebar on the left with a 'Submit' button. The form fields are: Nama Barang, Jumlah Barang, Satuan Barang, Harga Barang, and Harga Jual. Below the form is a summary table:

No.	Nama	Jumlah	Harga	Subtotal	Aksi
Data: 0	Total Bayar	<input type="text"/>	Total Kurang	<input type="text"/>	Grand Total
					0

At the bottom of the form is a green 'Submit' button.

Gambar 47. Transaksi Pembelian

- tester.php

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cara Pakai</title>
  <link rel="stylesheet" href="Assets/bootstrap-4.4.1/css/bootstrap.min.css">
</head>
<body>

<div class="awal">Belajar JQuery</div>

<div class="row">
  <button class="col-3 bg-success m-2 text-center text-white"

```

```

id="Test1">Test 1</button>

  <button class="col-3 bg-success m-2 text-center text-white"
id="Test2">Test 2</button>

  <button class="col-3 bg-success m-2 text-center text-white"
id="Test3">Test 3</button>

</div>

<div class="row">

  <div class="col-12 bg-primary" style="height:100px;display:none"
id="buka1">&nbsp;</div>

  <div class="col-12 bg-warning"
style="height:100px;display:none" id="buka2">&nbsp;</div>

  <div class="col-12 bg-danger" style="height:100px;display:none"
id="buka3">&nbsp;</div>

</div>

<!-- JQuery harus diatas bootstrap -->

<script src="Assets/jquery/dist/jquery.js"></script>

<script src="Assets/bootstrap-4.4.1/js/bootstrap.min.js"></script>

<script>

// Tulis Script JS disini

$(document).ready(function(){

  $("#Test1").click(function(){

    $("#buka1").slideToggle()

  });

  $("#Test2").click(function(){

    $("#buka2").fadeToggle()

  });

```

```
$("#Test3").click(function(){  
    $("#buka3").show()  
});  
});  
</script>  
</body>  
</html>
```

- chngpass.php

```
<?php  
include_once("_header.php");  
require_once("core/Database.php");  
session_start();  
$db = new Database();  
?>  
  
<body bgcolor="#f1f1f1">  
    <div class="headpass">  
        <h2>  
            <b>Ganti Password</b>  
        </h2>  
    </div>  
    <form id="passForm" method="post" action="service/ApiServices.  
php">  
        <div class="pass">
```

```

<div class="col-md-12">

  <div class="form-group">

    <input type="password" placeholder="Password
lama" tabindex="2" id="password" class="form-control"
name="password"

      required data-error="Please enter your password">

    <div class="help-block with-errors"></div>

  </div>

</div>

<div class="col-md-12">

  <div class="form-group">

    <input type="password" placeholder="Password
baru" tabindex="2" id="password" class="form-control"
name="password"

      required data-error="Please enter your password">

    <div class="help-block with-errors"></div>

  </div>

</div>

<div class="col-md-12 row m-0">

  <div class="col-md-9 p-md-0">

    <?php if (@$_COOKIE['alert']){ ?>

      <span id="error" class="text-error text-danger">*<?=@$_
COOKIE['alert']?>!/</span>

      <?php } ?>

      <span class="text-register">*Lupa Password?
<a href="register.php" style="text-decoration: underline
!important">Contact Admin</a></span>

    </div>

```

```
<input type="hidden" name="req" value="login">
<button class="btn btn-common col-md-3"
tabindex="3" id="submit" type="submit">Ganti</button>

</div>

</div>

</form>

<script>
    if ($("#error").length) {
        $(".text-register").css("display","none");
        $(".text-error").fadeOut(2000, 500).slideUp(500, function
    () {
        $(".text-register").slideDown(500);
    });
    }
</script>
</body>
</html>
```

- testKoneksi.php

```
<?php

require_once("core/Database.php");

$koneksi = new Database();

$test = $koneksi->select("SELECT * FROM daftar_satuan");
```

```
foreach($test as $data){  
    echo $data["nama_satuan"]."<br>";  
}  
  
// $test = $koneksi->insert("daftar_satuan",["nama_satuan"=>"renceng","status"=>1]);  
  
// $test = $koneksi->update("daftar_satuan",["status"=>1],["kode_satuan"=>6]);  
  
// $test = $koneksi->delete("daftar_satuan",["kode_satuan"=>6]);  
  
var_dump($test);
```

- database.php, file ini terdapat dalam folder core

```
<?php  
date_default_timezone_set("Asia/jakarta");  
  
class Database  
{  
    public $koneksi;  
  
    public function __construct()  
    {  
        $host = "localhost";  
        $user = "root";  
        $password = "";  
        $db = "dodolan";
```



```
$this->koneksi = mysqli_connect($host,$user,$password,$db);

if (!$this->koneksi) {
    throw new Exception("Tidak dapat Konek ke Database",
E_USER_ERROR);
}
}

public function select($sql)
{
    $query = mysqli_query($this->koneksi,$sql);
    if (!$query) {
        $error = mysqli_error($this->koneksi);
        throw new Exception($error);
    }else{
        $arrayReturn = array();
        foreach ($query as $data) {
            $arrayReturn[] = $data;
        }
        mysqli_free_result($query);
        return $arrayReturn;
    }
}

public function insert($table,$arrayData)
{

```

```
$fields = array_keys($arrayData);  
$values = array_values($arrayData);  
  
$escapeData = [];  
foreach ($values as $val) {  
    if (!is_numeric($val)) {  
        $val = "" . mysqli_escape_string($this->koneksi,$val) . "";  
    }  
    $escapeData[] = $val;  
}  
  
$sql = "INSERT INTO $table (";  
$sql .= join(" ", $fields);  
$sql .= ") VALUES(";  
$sql .= join(" ", $escapeData).")";  
  
$query = mysqli_query($this->koneksi, $sql);  
if (!$query) {  
    $error = mysqli_error($this->koneksi)."\n".$sql;  
    throw new Exception($error);  
    return False;  
}  
return True;  
}
```

```
public function update($table, $arrayData, $arrayCondition)
{
    $dataUpdate = [];
    foreach ($arrayData as $field => $val) {
        $val = "" . mysqli_escape_string($this->koneksi,$val) . "";
        $dataUpdate[] = "$field = $val";
    }

    $dataCondition = [];
    foreach ($arrayCondition as $field => $val) {
        if (!is_numeric($val)) {
            $val = "" . mysqli_escape_string($this->koneksi,$val) . "";
        }
        $dataCondition[] = "$field = $val";
    }

    $sql = "UPDATE $table SET ";
    $sql .= join(", ", $dataUpdate);
    $sql .= " WHERE ".join(" AND ", $dataCondition);
    $query = mysqli_query($this->koneksi, $sql);
    if (!$query) {
        $error = mysqli_error($this->koneksi)."\n".$sql;
        throw new Exception($error);
        return False;
    }
}
```

```

    }

    return True;
}

public function delete($table, $arrayCondition)
{
    $dataCondition = [];
    foreach ($arrayCondition as $field => $val) {
        if(!is_numeric($val)){
            $val = "" . mysqli_escape_string($this->koneksi, $val) . "";
        }
        $dataCondition[] = "$field = $val";
    }

    $sql = "DELETE FROM $table WHERE ".join(" AND
", $dataCondition);

    $query = mysqli_query($this->koneksi, $sql);
    if (!$query) {
        $error = mysqli_error($this->koneksi). "\n" . $query;
        throw new Exception($query);
        return False;
    }
    return True;
}

public function query($sql)

```

```
{  
    $query = mysqli_query($this->koneksi, $sql);  
    if (!$query) {  
        $error = mysqli_error($this->koneksi). "\n" . $query;  
        var_dump($error);  
        throw new Exception($query);  
    }  
    return $query;  
}  
}
```

- ApiService.php, File ini terdapat dalam folder service.

```
<?php  
require_once("../core/Database.php");  
session_start();  
$db = new Database();  
$Request = @$_REQUEST['req'];  
  
function createKodeUser(){  
    $kode = "USR".date("dHis");  
    return $kode;  
}  
  
function createKodeTransaksi($tipe){  
    $kode = "TRX".$tipe.random_int(1000,9999);
```

```
    return $kode;
}

if ($Request == "daftar_barang") {
    $getData = $db->select("SELECT * FROM daftar_barang");
}

elseif ($Request == "login") {
    $username = $_POST['username'];

    $auth=$db->select("SELECT * FROM user_login WHERE username
= '". mysqli_real_escape_string($db->koneksi,$username)."");

    if (count($auth) > 0) {
        $password = password_verify($_POST['password'],$auth[0]
['password']);

        if ($password) {
            $tokenID = md5($username.random_int(1,999));

            // $db->update("user_login",["token"=>$tokenID],$auth[0]
['id']);

            $_SESSION["token"] = $tokenID;

            $_SESSION["username"] = $username;

            $_SESSION["nama"] = $auth[0]['nama'];

            $_SESSION["kode_user"] = $auth[0]['kode_user'];

            header("location: ../dashboard.php");

        }else {
            setcookie("alert","Password Salah!", time() + (15),"");

            header("location: ../index.php");
        }
    }
}
```

```
    }else {  
        setcookie("alert","Username Tidak Ditemukan", time() + (15),"/");  
        header("location: ../index.php");  
    }  
}  
elseif ($Request == "register") {  
    $nama = $_POST['nama'];  
    $username = $_POST["username"];  
    $password = $_POST["password"];  
    $email = $_POST['email'];  
    $kode_user = createKodeUser();  
    $data = [  
        "kode_user" => $kode_user,  
        "nama" => $nama,  
        "username" => $username,  
        "password" => password_hash($password,PASSWORD_  
BCRYPT),  
        "email" => $email  
    ];  
    $db->insert("user_login",$data);  
    $_SESSION["username"] = $username;  
    $_SESSION["nama"] = $nama;  
    $_SESSION["kode_user"] = $kode_user;  
    header("location: ../dashboard.php");  
}
```

```
elseif ($Request == "tambah_barang") {  
    $data = [  
        "kode_user" => $_SESSION['kode_user'],  
        "nama_barang" => $_POST['nama_barang'],  
        "jumlah" => $_POST['jumlah'],  
        "kode_satuan" => $_POST['kode_satuan'],  
        "tanggal_restock" => date("Y-m-d"),  
        "harga_beli" => $_POST['harga_beli'],  
        "harga_jual" => $_POST['harga_jual']  
    ];  
    return $db->insert("daftar_barang",$data);  
}  
elseif ($Request == "tambah_satuan") {  
    $data = [  
        "kode_user" => $_SESSION['kode_user'],  
        "nama_satuan" => $_POST['nama_satuan']  
    ];  
    return $db->insert("daftar_satuan",$data);  
}  
elseif ($Request == "transaksi_pembelian") {  
    $kode_transaksi = createKodeTransaksi("BELI");  
    $dataTransaksi = [  
        "kode_user" => $_SESSION['kode_user'],  
        "kode_transaksi" => $kode_transaksi,  
        "tanggal" => date("Y-m-d"),  
        "total_pembelian" => $_POST['grand_total'],  
        "total_bayar" => $_POST['total_bayar'],
```



```
"total_kurang" => $_POST['total_kurang'],
"catatan" => "kosong"
];
$result = $db->insert("transaksi_pembelian", $dataTransaksi);
foreach ($_POST['id_barang'] as $key => $val) {
    if ($_POST["id_barang"][$key] == "undefined") {
        $namaBarang = $_POST["nama_barang"];
        $dataBarang = [
            "kode_user" => $_SESSION["kode_user"],
            "nama_barang" => $_POST["nama_barang"][$key],
            "jumlah" => $_POST["jumlah_barang"][$key],
            "kode_satuan" => $_POST["satuan"][$key],
            "tanggal_restock" => date("Y-m-d"),
            "harga beli" => $_POST["harga_barang"][$key],
            "harga_jual" => $_POST["total_jual"][$key]
        ];
        $db->insert("daftar_barang", $dataBarang);
        $getID = $db->select("SELECT id FROM daftar_barang WHERE
nama_barang LIKE '%" . $namaBarang . "%'");

        $dataDetailTransaksi = [
            "kode_user" => $_SESSION["kode_user"],
            "kode_transaksi" => $kode_transaksi,
            "id_barang" => $getID,
            "harga beli" => $_POST["harga beli"][$key],
```

```

        "jumlah_barang" => $_POST["jumlah_barang"][$key],
        "subtotal" => $_POST["subtotal"][$key],
        "kode_satuan" => $_POST["satuan"][$key]
    ];

    $insertDetail = $db->insert("detail_transaksi_pembelian",
    $dataDetailTransaksi);

    }else{

        $dataDetailTransaksi = [

            "kode_user" => $_SESSION["kode_user"],
            "kode_transaksi" => $kode_transaksi,
            "id_barang" => $_POST["id_barang"][$key],
            "harga_beli" => $_POST["harga_barang"][$key],
            "jumlah_barang" => $_POST["jumlah_barang"][$key],
            "subtotal" => $_POST["subtotal"][$key],
            "kode_satuan" => $_POST["satuan"][$key]
        ];

        $insertDetail = $db->insert("detail_transaksi_pembelian",
        $dataDetailTransaksi);

        $cek = $db->query("CALL SyncPembelian('".$_POST["id_
        barang"][$key]."', '".$_SESSION["kode_user"]."', '".$_date("Y-
        m-d")."', '".$_POST["jumlah_barang"][$key]."', '".$_POST["harga_barang"][$key]."',@a)");

        }

    }

    header("location:../arusmodal.php");
}

elseif ($Request == "transaksi_penjualan") {

```

```
$kode_transaksi = createKodeTransaksi("JUAL");

$dataTransaksi = [
    "kode_user" => $_SESSION['kode_user'],
    "kode_transaksi" => $kode_transaksi,
    "tanggal" => date("Y-m-d"),
    "total_harga" => $_POST['grand_total'],
    "total_bayar" => $_POST['total_bayar'],
    "total_kurang" => $_POST['total_kurang'],
    "catatan" => "Kosong"
];

$cek = $db->insert("transaksi_penjualan",$dataTransaksi);
var_dump($cek);

foreach ($_POST['id_barang'] as $key => $val) {
    $dataDetailTransaksi = [
        "kode_user" => $_SESSION["kode_user"],
        "kode_transaksi" => $kode_transaksi,
        "id_barang" => $_POST["id_barang"][$key],
        "harga_barang" => $_POST["harga_barang"][$key],
        "jumlah_barang" => $_POST["jumlah_barang"][$key],
        "subtotal" => $_POST["subtotal"][$key],
        "kode_satuan" => $_POST["satuan"][$key]
    ];

    $insertDetail = $db->insert("detail_transaksi_penjualan",
    $dataDetailTransaksi);
```

```

        $cek = $db->query("CALL SyncPenjualan(@a,@b,'" . @$__
POST["id_barang"][$key]."', '" . @$__SESSION["kode_user"]."',
'" . @$__POST["jumlah_barang"][$key]."'");

    }

    header("location: ../arusmodal.php");
}

elseif($Request == "logout"){
    session_destroy();
    header("location:../index.php");
}

```

- dataDummy.php, file ini terdapat dalam file service

```

<?php
require_once("../core/Database.php");
$db = new Database();

function daftarSatuan($db)
{
    $kodeUser = "#USER3108001";

    $dataSatuan =
['sachet','kg','g','mg','liter','bungkus','biji','renceng','lusin',
    gros','kodi','rim'];

    foreach ($dataSatuan as $satuan) {
        $data = [
            "kode_user" => $kodeUser,
            "nama_satuan" => ucfirst($satuan),

```

```
        "status" => 1
    ];
    $db->insert("daftar_satuan",$data);
}
}

function dummyBarang($db)
{
    $kodeUser = "#USER3108001";
    $listBarang = [
        'Autan',
        'Rokok Samsu',
        'Beras',
        'Gula',
        'telur',
        'gula merah',
        'kerupuk',
        'heirs',
        'kopi kapal api kecil',
        'goodday freast',
        'indomie goreng',
        'indomie soto',
        'sedap goreng',
        'sedap soto',
        'sedap kari sepsial'
    ];
    foreach ($listBarang as $val ) {
```

```
$data = [  
    "kode_user" => $kodeUser,  
    "nama_barang" => $val,  
    "jumlah" => rand(1,20),  
    "kode_satuan" => rand(1,12),  
    "tanggal_restock" => date("Y-m-d"),  
    "harga_beli" => rand(1000,10000),  
    "harga_jual" => rand(10000,20000),  
    "jumlah_terjual" => rand(10,100)  
];  
$db->insert("daftar_barang",$data);  
}  
}  
  
function createDataPembelian($db)  
{  
    $dataBarang = $db->select("SELECT * FROM daftar_barang");  
}
```

Daftar Pustaka

1. Anjani, Putri Dewi. 2013. "Apa itu internet? Pengertian Jaringan Internet Menurut Para Ahli Teknnologi Informasi". Tersedia dalam: <http://enetter.blogspot.com/2012/12/apa-itu-internet-pengertian-jaringan-internet.html>, [diakses tanggal 08 Juni 2020].
2. Desainweb. 2014. "Pengertian website apa itu website". Tersedia dalam: <http://desainweb.com/site/pengertian-website-apa-itu-website/> [diakses tanggal 13 maret 2020].
3. Puspitosari, Heni A. "Pemrograman Web Database dengan PHP dan MySQL Tingkat Lanjut". Penerbit : Skripta.Malang, Juli 2010
4. Madcomps, "Aplikasi Program PHP & MySQL untuk Membuat Website Interaktif", Yogyakarta : Andi Offset dan Madcomps, 2004.
5. Jogiyanto, HM, "Analisis Perancangan Sistem Informasi". Andi Offse, Yogyakarta. 2001.
6. http://hans.polinpdg.ac.id/html/html_4/index.html
7. <http://rizkyramadhansttg.wordpress.com/>
8. <http://agungpm.ashatec.com/2009/06/09/dasar-html/>,
9. <http://www.bakawan.com/log/tag/html/>
10. Saputra, Agus. 2011. Pemrograman CSS Untuk Pemula. Jakarta : PT. Gramedia.
11. Mulyana. Y.B. 2004. Trik Membangun Situs Menggunakan PHP dan MySQL. Jakarta: Elex Media Komputindo.
12. Hartama, Dedy., Efrizal. 2012. Pemrograman Web dengan HTML, CSS, dan Javascript. Yogyakarta : Penerbit ANDI. (Javascript.137-138, CSS.97-98, HTML.3)
13. Kadir, Abdul. 2005. Dasar Pemrograman Web Dinamis Menggunakan PHP. Yogyakarta : Informatika Bandung
14. http://id.wikipedia.org/wiki/Optimisasi_mesin_pencari
15. <http://www.rudinazar.com/apa-itu-seo/>
16. <http://yerlangga.blogspot.com/2012/04/pengertian-dan-contoh-program-xml.html>
17. www.w3school.com
18. Ari, Rosihan. Dasar Javascript. Diakses melalui www.rosihanari.net

19. Goodman, Danny with Michael Morrison. Javascript 5 Bible. Wiley Pubhlying,inc.
20. Spurlock, J. (2013). *Bootstrap: Responsive Web Development*. Sebastopol: O'Reilly Media, Inc.
21. <https://www.susantokun.com/konfigurasi-awal-menggunakan-codeigniter/>