

BAB II

TINJAUAN PUSTAKA

2.1 TEORI UMUM

Landasan dasar pada teori umum sebagai pengetahuan dalam membangun Sistem *E-Commerce* Berbasis *Codeigneter Framework* Pada Umkm Thrift M.S

2.1.1 Sistem

Sistem adalah pengelompokan dua bagian atau lebih yang berinteraksi dan berhubungan satu sama lain untuk mencapai tujuan tertentu. Dia juga berpendapat bahwa bisnis adalah sistem yang terdiri dari berbagai divisi yang berfungsi sebagai subsistemnya. Namun, yang lain berpendapat bahwa sistem adalah kumpulan proses yang saling berhubungan yang bersatu untuk melakukan tugas atau mencapai tujuan tertentu (Putri Primawanti & Ali, 2022)

secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. teori sistem umum peratama kali diuraikan oleh kanneth boluding, terutama menekankan pentingnya perhatian terhadap setiap bagian yang memebentuk sebuah sistem. (sutabri, p. 2018)

2.1.2 Informasi

Informasi menurut para ahli berikut dibawah ini; informasi menurut para ahli berikut dibawah ini; (Irviani, 2019) Menurut Gordon B. Davis :

Pengertian informasi adalah “data yang telah diolah menjadi bentuk yang nyata dan bermanfaat bagi penerimanya”.




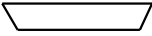
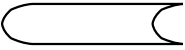
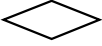
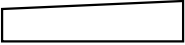

Stephen A. Moscovice menegaskan bahwa informasi harus dilihat sebagai fakta atau dalam format yang berguna yang dapat diterapkan pada pengambilan keputusan bisnis. Singkatnya, informasi dibuat dengan mengubah data menjadi bentuk yang lebih bermanfaat bagi mereka yang menerimanya dan dapat digunakan sebagai alat untuk pengambilan keputusan.

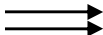
2.1.3 Sistem informasi

(Tukino & Amrizal, 2017) Sistem informasi perusahaan atau organisasi adalah sekelompok bagian yang berkaitan dengan pembuatan dan pergerakan informasi. TI hanyalah salah satu bagian dari bisnis dalam hal ini. Prosedur, struktur organisasi, sumber daya manusia, produk, klien, mitra, dan sebagainya merupakan komponen tambahan. Agar informasi yang dapat digunakan (benar, andal, terperinci, cepat, relevan, dll.) dihasilkan dan disebarluaskan ke lembaga yang bersangkutan, hubungan timbal balik antara komponen sistem informasi dalam suatu organisasi sangat penting.

2.1.4 Aliran sistem informasi

Information System Flow (ASI) adalah diagram yang menggambarkan dan mengontrol aliran kerja di seluruh sistem (Ismael, 2017). Model yang dapat diterapkan pada aliran sistem informasi ditunjukkan pada grafik di bawah ini yang menggambarkan setiap urutan program dan langkah-langkah sistem:

No	Simbol	Nama	Keterangan
1		<i>Document</i>	Formulir yang digunakan untuk merekam data yang menunjukkan input dan output, baik untuk proses manual maupun komputer
2		<i>Processing symbol</i>	Proses yang menggunakan komputer dimana pengelolah data dilakukan secara online
3		Arsip	Menggambarkan penyimpanan data, baik dalam bentuk arsip atau file komputer
4		<i>Manual operation</i>	Menggambarkan kegiatan manual atau pekerjaan yang dilakukan tidak dengan komputerisasi
5		<i>Disket</i>	Menggambarkan input dan output dengan menggunakan disket
6		<i>Decision</i>	Menggambarkan pemilihan proses berdasarkan kondisi yang benar
7		Manual <i>input</i>	Menggambarkan pemasukan data secara manual melalui keyboard
8		Simbol penghubung	Simbol untuk menunjukkan sambungan dari bagian alir yang terputus dihalaman yang masih sama atau

			halaman lainnya
9		<i>Flow direction</i>	Simbol untuk menghubungkan antar simbol lainnya

Gambar 2. 1 Alur sistem informasi

2.1.5 UML (*Unified Modelling Language*)

(M Teguh Prihandoyo, 2018) Teknik pemodelan visual yang digunakan dalam pembuatan perangkat lunak berorientasi objek disebut *Unified Modeling Language*. Standar penulisan atau cetak biru yang disebut UML terdiri dari proses bisnis dan kursus menulis dalam bahasa tertentu. Beberapa diagram UML sering digunakan saat mengembangkan sistem. (Tedyyana, Ratnawati, & Kurniati, 2019) Selain itu, untuk mendefinisikan, memvisualisasikan, membuat, dan mendokumentasikan artefak sistem perangkat lunak (informasi yang digunakan atau dihasilkan selama proses pembuatan perangkat lunak; artefak ini dapat berupa model, deskripsi, atau perangkat lunak). Notasi yang ditawarkan oleh UML memudahkan untuk memodelkan sistem dari berbagai sudut. (Tedyyana et al., 2019) Diagram adalah salah satu komponen utama UML. Diagram adalah representasi grafis dari simbol elemen model yang dirangkai untuk mengilustrasikan komponen atau fitur sistem tertentu. Jenis diagram termasuk yang berikut ini :

1. Diagram *use case*

Use case berfungsi sebagai deskripsi dari tujuan yang dimaksudkan sistem dan sebagai representasi dari interaksi antara aktor dan aktivitas sistem. Sejumlah besar pemain eksternal dan use case untuk sistem dijelaskan dalam use case oleh aktor, yang juga memberikan deskripsi individu atau sistem yang membentuk sistem. Lebih baik untuk memiliki pemahaman yang jelas tentang elemen dinamis dari sistem karena kasus penggunaan hanya dapat dipahami dalam hal apa yang dilihat oleh aktor eksternal atau keadaan lingkungan sistem seperti yang dirasakan oleh pengguna dari diagram kasus penggunaan fungsional dari sistem. . Memanfaatkan diagram kasus penggunaan, persyaratan sistem dikumpulkan dari sumber internal dan eksternal. (Munawar, 2018). Tujuan umum diagram use case dapat diringkas sebagai berikut:

1. Diterapkan untuk mengumpulkan persyaratan sistem.
2. Untuk mendapatkan perspektif orang luar.
3. Tentukan variabel yang memiliki dampak baik di dalam maupun di luar sistem untuk menunjukkan bagaimana aktor sistem berinteraksi





Tujuan dari skenario use case adalah untuk mendeskripsikan apa yang dicapai oleh sistem atau subsistem. Contoh salah satunya ditunjukkan pada tabel di bawah ini.:

No	<i>User case name</i>	<i>Name use case</i>
1.	<i>Goal</i>	Tujuan dari <i>use case</i>
2.	<i>Pre-condition</i>	Kondis yang harus terpenuhi sebelum <i>use case</i> tersebut di eksekusi <i>use case</i>
3.	<i>Post- condition</i>	Adalah kondisi yang harus terpenuhi setelah <i>use case</i> tersebut di eksekusi <i>use case</i>
4.	<i>Failed end condition</i>	Actor utama yang berpartisipasi apabila terjadi kegagalan dalam eksekusi <i>use case</i>
5.	<i>Primary actor</i>	Adalah <i>actor</i> utama yang berpartisipasi dalam <i>use case</i>
6.	<i>Flow/basic path</i>	Arus kerja utama dengan asumsi semua berjalan dengan baik
7.	Alternatif flow	Arus kerja alternatif dari arus kerja utama

Gambar 2. 2 Skenario *use case*

Penjelasan gambaran dan simbol beserta keterangan diagram *use case* ada

Pada gambaran berikut:




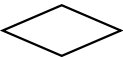

No	Simbol	Keterangan
1.	<i>Actor</i> 	Menspesifikasin peran pengguna mainkan waktu berinteraksi di <i>use case</i>
2.	<i>Dependency</i>▶	Perubahan elemen mandiri yang akan mempengaruhi elemen lain yang bergantung pada elemen ini.
3.	<i>Extends</i> - -▶	Menspesifikasikan <i>use case</i> bahwa untuk memperluas target perilaku dari <i>use case</i> sumber pada satu titik yang telah diberikan
4.	<i>Association</i> ↔	Untuk menghubungkan antara objek satu dengan objek lain
5.	<i>Sistem</i> 	Mengspesifikasikan paket yang menampilkan sistem secara terbatas
6.	<i>Use case</i> 	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
7.	<i>Note</i> 	Bagian fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Gambar 2. 3 Diagram use case

2. Activity Diagram

Aliran peristiwa dan perspektif umum dari sistem aktif. Komponen dinamis dari sistem yang menyoroti pentingnya UML adalah diagram aktivitas. Untuk. Tujuan dari diagram aktivitas adalah untuk menggambarkan aliran pesan dari satu aktivitas ke aktivitas lainnya untuk menggambarkan perilaku dinamis suatu sistem. (Munawar, 2018). Penjelasan berikut menjelaskan tujuan umum dari diagram aktivitas:

1. menjelaskan aliran aktivitas dari sistem.
2. Jelaskan tahap kegiatan dari satu kegiatan ke kegiatan lainnya.
3. Menjelaskan paralelisme sistem, percabangan, dan aliran paralel.

No	Simbol	Nama	Keterangan
1.		<i>Activity</i>	Dengan cara masing-masing kelas memperlihatkan antarmuka dan berinteraksi satu sama lain
2.		<i>Initial node</i>	Bagaimana objek dibentuk atau diawali
3.		<i>Activity final node</i>	Bagaimana objek dibentuk dan dihancurkan
4.		<i>Decision</i>	Pilihan untuk pengambilan keputusan
5.		<i>Fork node</i>	Menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu

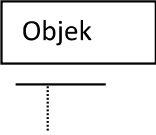


Gambar 2. 4 Activity diagram


3. Sequence Diagram

Menjelaskan hubungan antara item di dalam dan di sekitar sistem yang dijelaskan sepanjang waktu dalam bentuk pesan. Diagram urutan sering menampilkan garis waktu pengiriman pesan dari satu objek ke objek lainnya. (Munawar, 2018).

Secara khusus, berikut ini dapat dikatakan tentang fungsi diagram urutan:

1. Model interaksi dapat mendukung antar item yang aktif dalam suatu sistem.
2. Model interaksi mendukung kerja sama antara instance objek untuk menyelesaikan *use case*.
3. Kerja sama yang melakukan operasi dapat mendukung model interaksi antar objek.
4. Ditampilkan sebagai jenis interaksi umum (hanya menampilkan semua jalur melalui interaksi). atau kejadian tertentu dari pertemuan.

No	Nama	Simbol	Keterangan
1.	Objek aktor		Sebuah objek yang berasal dari kelas, atau dapat dinamai dengan kelasnya saja, aktor termasuk objek. Garis putus-putus menunjukkan garis hidup suatu objek
2.	Aktivitasi		Memperlihatkan masa hidup dari objek
3.	Pesan		Interaksi antara satu objek dengan objek lainnya. Objek dapat mengirimkan pesan ke objek lain.

			Interaksi antara objek ditunjukkan pada bagian operasi pada diagram kelas
4.	<i>Return</i>		Pesan kembalian dari komunikasi antar objek


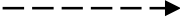
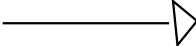

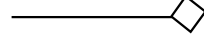
Gambar 2. 5 *Sequence Diagram*

4. *Class diagram*

Perincian struktur, bersama dengan deskripsi setiap kelas, paket, dan elemen yang terhubung seperti asosiasi dan pewarisan. Diagram kelas menurut (Suendri, 2018) terdiri dari 3 (tiga) bagian utama, yaitu:

- sebuah. Nama kelas: Nama yang diberikan ke kelas terletak di sini.
- Integritas ditambahkan sebagai atribut baru ke kelas. nilai kelas yang hanya dapat memproses sejumlah atribut tertentu.
- Operasi: tugas-tugas yang dapat dilakukan oleh kelas Individu atau kelas lainnya.

No	Nma	Simbol	Keterangan
1.	Kelas	<div style="background-color: #4b6121; color: white; padding: 2px;"><i>Class name</i></div> <div style="padding: 2px;">+atribut</div> <div style="padding: 2px;">+operasi</div>	Kelas pada struktur sistem
2.	Antarmuka/ <i>interface</i>	<div style="background-color: #4b6121; color: white; padding: 2px;"><<<i>interface</i>>></div> <div style="background-color: #4b6121; color: white; padding: 2px;"><i>Interfacename</i></div> <div style="padding: 2px;">+membernama</div>	Sama dengan konsep didalam pemograman berorientasi objek

3.	<i>Directed assosiation</i>		Relasi antar kelas dengan makna kelas satu digunakan dengan kelas lainnya, <i>assosiasi</i> juga disertai dengan <i>multiplicity</i>
4.	<i>Depedency</i>		Relasi antar kelas dengan makna kebergantungan antar kelas
5.	<i>Generalization</i>		Relasi antar kelas dengan makna <i>generalisasi spesialisasi</i> (umum-khusus)
6.	<i>Assosiation</i>		Relasi antar kelas dengan makna umum, <i>asosiastion</i> biasanya sekaligus disertai <i>simplicity</i>
7.	<i>agregasi</i>		Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

Gambar 2. 6 Class diagram

2.1.6 Metode Waterfall

Metode *waterfall* atau metode air terjun, juga dikenal sebagai *Classic Life Cycle* (*Classic Life Cycle*). Nama model ini sebenarnya adalah "*Linear Sequential Model*", yang menggambarkan pendekatan pengembangan perangkat lunak yang sistematis dan berurutan. , yang dimulai dengan spesifikasi kebutuhan yang dilalui

pengguna, melewati fase perencanaan, pemodelan, pembangunan, dan pengiriman sistem ke pengguna, dan diakhiri dengan dukungan perangkat lunak yang ditulis secara lengkap Model air terjun pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970, sehingga sering dianggap usang, tetapi merupakan model yang paling banyak digunakan dalam rekayasa perangkat lunak (SE).

Model *waterfall* adalah model pengembangan perangkat lunak yang umum digunakan saat ini. Model pengembangan ini mengikuti pendekatan yang sistematis dan berurutan. Tahapan yang dilaluinya disebut air terjun karena menunggu tahap sebelumnya selesai dan harus dijalankan satu per satu. Model pengembangan ini bersifat linear dari tahap awal pengembangan sistem, tahap perencanaan, hingga tahap akhir pengembangan sistem, tahap pemeliharaan. Tahap selanjutnya tidak terjadi sampai tahap sebelumnya selesai, dan tahap sebelumnya tidak dapat dikembalikan atau diulang.(Wahid, 2020)

2.2 TEORI KHUSUS

Ada beberapa landasan teori khusus untuk membangun rancang bangun Sistem *E-Commerce* Berbasis *Codeignater Framework* Pada Umkm Thrift M.S yang akan dibuat yaitu:

2.2.1 Ecommerce

Promosi, distribusi, dan pertukaran barang dan jasa melalui jaringan elektronik seperti internet, televisi, atau jaringan komputer lainnya dikenal sebagai perdagangan elektronik, atau *e-Commerce*. Transfer keuangan elektronik, pertukaran data, manajemen inventaris terkomputerisasi, dan pengumpulan data

adalah semua komponen *e-commerce* yang mungkin. *Electronic commerce*, atau *e-commerce*, adalah istilah yang relatif baru yang umumnya mengacu pada pertukaran barang, jasa, dan informasi di seluruh jaringan informasi seperti internet. Selain penggunaan teknologi *mobile commerce* saat ini yang seluruhnya dilakukan dengan smartphone yang sudah kita miliki, juga dikembangkan dengan menggunakan komputer dan laptop. Oleh karena itu, semua tugas dapat diselesaikan secepat satu langkah jari. Sedangkan strategi adalah teknik untuk mendapatkan kemenangan (*victory*) pencapaian tujuan (*to achieve goals*). Istilah strategi sendiri berasal dari kata Yunani. untuk ahli militer atau memimpin pasukan. Menurut *bussinesdictionary*, Gagasan strategi adalah rencana atau metode yang digunakan untuk mencapai masa depan yang diinginkan, seperti mencapai tujuan atau menemukan solusi untuk suatu masalah; strategi adalah seni dan ilmu perencanaan dan pengalokasian sumber daya untuk penggunaan yang paling efektif dan efisien. Dengan demikian, secara teori, strategi *E-Commerce* adalah cara atau pendekatan untuk mencapai tujuan perusahaan melalui penggunaan sistem jaringan komputer(Putra, 2020).

E-commerce akan mengubah semua strategi pemasaran sekaligus mengurangi biaya overhead yang terkait dengan aktivitas perdagangan (perdagangan). Berikut adalah prosedur dalam *e-commerce*(Putra, 2020) :

- a. Pemesanan secara langsung dan tersedianya tagihan.

- b. Otomatisasi akun pelanggan secara aman (baik nomor rekening maupun nomor Kartu Kredit).
- c. Pembayaran yang dilakukan secara Langsung (*online*)
- d. Presentasi elektronik (pembuatan *website*) untuk produk dan layanan.

Dalam *e-commerce*, ada berbagai jenis transaksi, seperti bisnis-ke-bisnis, bisnis-ke-konsumen, konsumen-ke-konsumen, dan konsumen-ke-bisnis. Dengan demikian, frasa "*e-commerce*" dan "*e-business*" selalu digunakan secara bergantian.(Meliyawati & Iwan Krisnadi, 2019);

1. *Business-to-Business* (B2B)

Semua transaksi elektronik barang atau jasa antar bisnis termasuk dalam *B2B e-commerce*. Jenis *e-commerce* ini biasanya digunakan oleh produsen dan pedagang tradisional. *EDI (Electronic Data Interchange)* dan email biasanya digunakan dalam jenis *e-commerce* ini untuk keperluan pembelian barang dan jasa, memberikan informasi dan konsultasi, atau mengirim dan meminta proposal bisnis. Pertukaran data elektronik, atau EDI, adalah metode untuk mentransfer data terstruktur secara elektronik antara sistem komputer dalam format standar. Bizzy dan Ralali adalah dua situs *web e-niaga* B2B. Di Indonesia, Bizzy merupakan platform *e-commerce* pertama dengan konsep B2B atau *business-tobusiness*. Bizzy menawarkan bantuan kepada bisnis yang mengalami kesulitan memperoleh persediaan dan layanan terkait bisnis. Bizzy menawarkan berbagai barang, antara lain *pantry*, perlengkapan kantor (ATK), dan *elektronik*. Salah satu *bisnis e-commerce B2B* di Indonesia yang menjual produk MRO (*maintenance*,

repair, and operational) adalah Ralali. Dengan PT Sejak 2013, Laju Lintang Giant telah menjalankan bisnis. Ralali menjual berbagai macam peralatan listrik, antara lain GPS, alat ukur, dan perlengkapan otomotif.

2. *Business-to-Consumer* (B2C)

E-commerce antara bisnis dan pengguna akhir dikenal sebagai B2C. Ini mirip dengan komponen ritel *e-commerce*, yang biasanya dikelola oleh bisnis ritel konvensional. Jenis ini bisa lebih sederhana dan lebih dinamis, tetapi juga bisa berhenti atau lebih tidak merata. *E-commerce* semacam ini berkembang pesat berkat dukungan yang diberikan oleh berbagai toko *virtual online* bahkan mal yang menjual berbagai barang dan jasa. Sementara itu, sudah banyak contoh *e-commerce* retail online yang sukses di negara maju seperti Amerika Serikat. Jika dibandingkan dengan transaksi ritel konvensional, pelanggan biasanya memiliki akses ke lebih banyak informasi, membayar lebih sedikit, dan dapat mengantisipasi proses pembelian dan penjualan yang lebih cepat hingga pengiriman. Bhinneka, Berrybenka, dan Tiket.com adalah tiga website Indonesia yang menggunakan *e-commerce* semacam ini. Penjual atau produsen yang serius menjalankan bisnis dan mengalokasikan sumber daya untuk mengelola situs web mereka sendiri biasanya menggunakan *e-commerce* semacam ini.

3. *Consumer-to-Consumer* (C2C)

Semua pertukaran elektronik barang atau jasa antara pelanggan termasuk dalam kategori *e-commerce* "bisnis-ke-konsumen" (C2C). Transaksi ini biasanya dilakukan melalui perusahaan ketiga yang menawarkan platform *online* untuk

melaksanakannya. Tokopedia, Bukalapak, dan Lamido adalah beberapa *website* di Indonesia yang menggunakan C2C. Penjualan langsung produk melalui situs web aktif diizinkan. Namun, situs web tertentu, termasuk Blanja dan Elevenia, menggunakan model C2C dan meminta vendor melalui prosedur verifikasi terlebih dahulu.

4. *Consumer-to-Business (C2B)*

C2B adalah bentuk *e-commerce* yang sepenuhnya membalikkan proses konvensional untuk bertukar atau membeli dan menjual barang. Adik-adik, *ecommerce* semacam ini cukup lazim dalam proyek-proyek yang memanfaatkan beberapa sumber daya. Untuk bisnis yang mencari layanan atau produk ini, sejumlah besar orang menawarkan barang atau layanan mereka. Sebagai ilustrasi, pertimbangkan situs web di mana perancang web menawarkan banyak kemungkinan logo, hanya memilih yang mereka anggap paling sesuai. *Platform* yang menawarkan gambar, gambar, media, dan komponen desain bebas royalti, seperti www.istockphoto.com, adalah platform lain yang sering menggunakan gaya e-niaga ini. *URL* yang disediakan oleh MBT memungkinkan pembuat template mengunggah kreasi mereka sebagai *template*, setelah itu MBT akan membagikan dan menjual *template* yang diunggah.

Berikut ini bentuk aktivitas ecommers pada thrift ms meliputi :

- a. Pemesanan : terdiri dari keranjang dan selesai belanja
- b. Pembayaran : pembayaran yang dilakukan oleh customer usai menerima konfirmasi dari admin atau pihak penjual

- c. Pengiriman : pengiriman dilakukan oleh admin dengan jasa pengiriman yang telah dipilih oleh customer.

2.2.2 Bootstrap

Salah satu *framework HTML, CSS, dan JS* yang paling populer adalah *Bootstrap*. Templat desain dengan fungsionalitas tambahan disebut *Bootstrap*. Kembangkan tata letak halaman yang mudah diedit dalam tampilan HTML dasar sehingga Anda cukup membuat halaman web yang menyertakan komponen tambahan dengan menyiapkan alat yang diperlukan. (Yuniva; Syafi'i, 2018).

Kerangka kerja untuk membuat desain web yang responsif disebut *Bootstrap*. Artinya apakah pengguna menggunakan browser desktop, tablet, atau seluler, tampilan web yang dihasilkan oleh bootstrap akan sesuai dengan ukuran layar. Bergantung pada preferensi kami, kami dapat mengaktifkan atau menonaktifkan fitur ini. Tampilan web yang saya rancang tidak sesuai dengan layar saat diproduksi oleh *browser seluler*, sehingga kami hanya dapat menggunakan web dalam tampilan desktop. Bootstrap dapat digunakan untuk membuat website statis maupun dinamis. Untuk membangun antarmuka yang estetis, sederhana, dan ringan, *Bootstrap* menawarkan kumpulan komponen antarmuka yang mendasar. Selain elemen antarmuka kelas, Bootstrap juga menyertakan fitur kisi. Ini sangat sederhana dan cepat digunakan, dan digunakan untuk menata halaman web. Selain itu, Bootstrap memberi kita kebebasan untuk menggunakannya untuk menyesuaikan tampilan situs web yang kita buat. Untuk mengubah tampilan *bootstrap*, tambahkan kelas dan kode *CSS*. (Nurjaman & Yasin, 2020)

1. Instalasi Bootstrap

Download file framework di <https://getbootstrap.com/>



Gambar 2. 7 Instalasi bootstrap

Sesudah itu ekstrak filenya dan copy folder css dan js ke dalam website kita.

2.2.3 Code-ignater

Kerangka kerja untuk membuat desain web yang responsif disebut *Bootstrap*. Artinya apakah pengguna menggunakan browser desktop, tablet, atau seluler, tampilan web yang dihasilkan oleh bootstrap akan sesuai dengan ukuran layar. Bergantung pada preferensi kami, kami dapat mengaktifkan atau menonaktifkan fitur ini. Tampilan web yang saya rancang tidak sesuai dengan layar saat diproduksi oleh browser seluler, sehingga kami hanya dapat menggunakan web dalam tampilan desktop. Bootstrap dapat digunakan untuk membuat website statis maupun dinamis. Untuk membangun antarmuka yang estetis, sederhana, dan ringan, Bootstrap menawarkan kumpulan komponen antarmuka yang mendasar. Selain elemen antarmuka kelas, Bootstrap juga

menyertakan fitur kisi. Ini sangat sederhana dan cepat digunakan, dan digunakan untuk menata halaman web. Selain itu, Bootstrap memberi kita kebebasan untuk menggunakannya untuk menyesuaikan tampilan situs web yang kita buat. Untuk mengubah tampilan bootstrap, tambahkan kelas dan kode CSS.(Erinton, R. Negara, R. Sanjoyo, 2017)

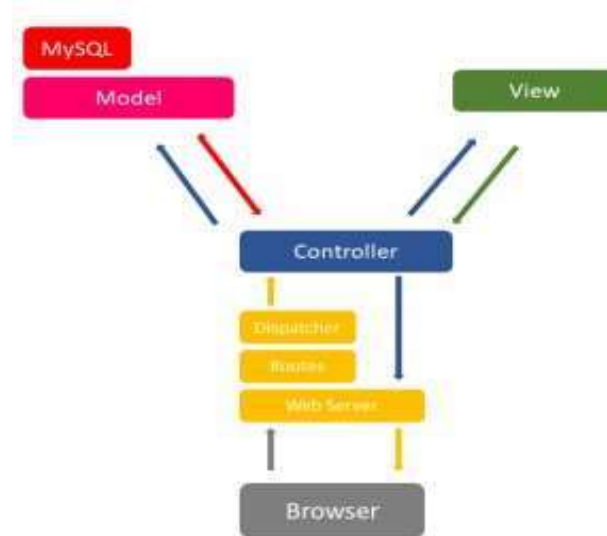
Framework adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah kompleks. Istilah ini sering digunakan antara lain dalam bidang perangkat lunak untuk menggambarkan suatu desain sistem perangkat lunak yang dapat digunakan kembali.

1. Komponen *MVC pattern*

Dijelaskan dalam buku (Yosef Murya Kusuma Ardhana, 2018, halaman 13–14) menjelaskan ide yang banyak digunakan dalam pengembangan aplikasi web. Ada tiga jenis komponen saat membangun pola *MVC*:

- a. Model yang terhubung ke database atau layanan web digunakan untuk merepresentasikan bentuk data dari perangkat lunak database dan data lainnya dengan memanipulasinya dengan cara berikut: Konfirmasi pengontrol faktor dapat diperbaiki dengan *insert*, *update*, *delete*, dan *search* , tetapi mungkin tidak terkait langsung dengan bagian tampilan yang biasanya menampilkan perintah kueri SQL terkait.
- b. Lihat, pengguna yang akan menerima semuanya. Biasanya berupa rss, javascript, atau halaman web. Tampilan hanya digunakan untuk menampilkan pengontrol dan data model. Bagian model tidak dapat diakses langsung dari bagian ini.

- c. Interaksi *model-view* dikendalikan oleh *controller*. Pengontrol bertanggung jawab atas variabel yang ditampilkan dalam tampilan, menerima permintaan dan data pengguna, dan membantu memutuskan apa yang dilakukan aplikasi Pengontrol.



Gambar 2. 8 Proses MVC

2. Installasi CodeIgniter

CodeIgniter yang penulis gunakan untuk merancang aplikasi adalah *CodeIgniter3*. Tahap pengistalan *framework codeigniter 3* dapat di uraikan sebagai berikut:

- a. *Download file Codeigniter*, Silahkan klik link <https://codeigniter.com/> lalu download framework codeigniter terbaru.
- b. Ekstrak dan *Install Codeigniter Framework* silahkan ekstrak file tersebut setelah itu akan muncul folder bernama CodeIgniter- - -, silahkan rename menjadi codeigniter lalu salin folder tersebut ke *folder root*, yaitu di *htdocs*
- c. Setelah menyelesaikan penginstalan *codeigniter*, buka file *application/config/config.php* di editor teks favorit Anda dan lanjutkan konfigurasi url dasar di sana. lalu di bagian kode di bawah ini:

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

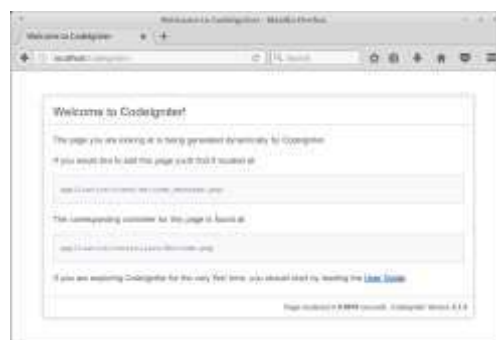
Code language: PHP (php)

Ubahlah menjadi dibawah ini:

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

Code language: PHP (php)

Nilai tersebut sesuai dengan alamat.halaman folder yang ditentukan ketika meyalin file codeignataer ke dalam folder root web server.



gambar 2. 9 *framework CI*

- d. *Framework* berhasil diinstal.

2.2.4 Data base

Salah satu konsep kunci dalam jaringan komputer adalah *database*, atau yang dikenal dengan *database* dalam terminologi teknis. Basis data adalah kumpulan semua data yang saat ini tersedia di dalam perusahaan atau pengaturan serupa lainnya. Basis data sering disimpan di *server*, yang dapat digunakan kapan pun diperlukan untuk tujuan tertentu. Para ahli dan spesialis basis data dan jaringan komputer telah memberikan definisi basis data sebagai berikut: Basis data, menurut Everest, adalah kumpulan atau kumpulan data yang dipisahkan secara mekanis, ditentukan secara teknis, dan diatur. Sistem basis data berada di bawah kendali terpusat dan sering dimiliki dan dipertahankan oleh suatu organisasi.

Basis data, dalam definisi Toni Fabbri Date, adalah sekelompok "data operasional" yang secara sadar disimpan dan digunakan oleh sistem aplikasi dari suatu organisasi. Selanjutnya, menurut Date, database berisi tiga kategori data yang berbeda: data operasional, output, dan input.

Definisi lain dari database ditawarkan oleh Chou. Menurut Chou, database adalah kumpulan bahan bermanfaat yang telah disusun menurut proses tertentu sehingga dapat digunakan untuk berbagai tujuan organisasi.

Menurut Rogayah dalam modulnya tentang sistem basis data, sistem basis data adalah suatu sistem yang dapat menyusun dan mengelola catatan dengan menggunakan komputer untuk menyimpan atau merekam dan memelihara semua

data operasional suatu organisasi atau perusahaan agar dapat memberikan informasi kepada pengguna. informasi terbaik yang diperlukan untuk proses pengambilan keputusan(Putri, 2020)

Hampir setiap perusahaan dapat memperoleh manfaat besar dari data. Masalah pengelolaan data secara efisien menjadi sangat signifikan dalam pengembangan sistem informasi manajemen sebagai akibat dari kelimpahannya. Oleh karena itu, berikut adalah tujuan dari manajemen data(Duggan, Roderick, & Sieburg, 1970) ;

- a. Sebagai metode input data untuk mempermudah tugas operator dan juga memperhitungkan waktu yang dibutuhkan user untuk mendapatkan data dan hak yang dimiliki atas data yang ditangani;
- b. Sebagai penyimpanan data untuk penggunaan organisasi saat ini dan di masa mendatang.
- c. Perlindungan data terhadap potensi penambahan, modifikasi, pencurian, dan gangguan lainnya;
- d. kontrol data untuk setiap siklus agar data selalu *up to date* dan dapat mencerminkan perubahan tertentu yang terjadi pada setiap sistem.

2.2.5 My sql

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh duniac.(Vivian Siahaan & Rismon Hasiholan Sianipar, 2018)

RDBMS termasuk *MySQL* (Sistem Manajemen Basis Data Relasional). Tabel, kolom, baris, dan blok bangunan basis data lainnya digunakan oleh *MySQL*. Akibatnya, mekanisme database relasional akan digunakan untuk pengambilan data.

Berikut ini adalah beberapa manfaat dari *MySQL*, yang dibuktikan dengan fitur dan penggunaannya. (“arnandy didet,” 2021);

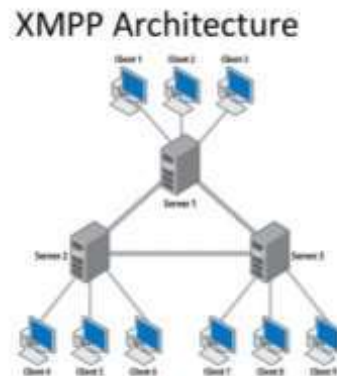
1. Dukungan untuk penggunaan banyak pengguna
2. Ini adalah Open Source, kedua.
3. Dapat beradaptasi dengan berbagai bahasa pemrograman.
4. Berbagai tipe data.
5. Sangat mudah untuk memahami bagaimana tabel diatur.
6. Sistem keamanan yang efektif.
7. Memerlukan memori dengan ukuran yang wajar.
8. Aplikasi seluler.
9. Saat melacak kesalahan, bekerja dengan baik dan menyertakan kemampuan yang mudah dipahami pengguna.

2.2.6 Xmpp

Pameran Pendidikan Internasional Turki membuat protokol IoT yang disebut *Extensible Messaging and Presence Protocol (XMPP)*, juga dikenal sebagai *protokol Jabber (IEFT)*. Karena menawarkan pola komunikasi requestresponse dan publish-subscribe, pesan kecil, latensi rendah, dan pengiriman data streaming, protokol ini dipandang mampu memenuhi kebutuhan

IoT. Arsitektur terdesentralisasi dan banyak ekstensi yang telah ditentukan sebelumnya disediakan oleh protokol XMPP yang sangat skalabel.(Kayal & Perros, 2017)

Arsitektur client-server terdesentralisasi dari teknologi XMPP sebanding dengan *World Wide Web* dan jaringan email. Tiga server dan tiga klien digambarkan dalam diagram yang disederhanakan pada Gambar di bawah ini. Jauh lebih mudah bagi organisasi untuk mengelola arsitektur *klien-server* terdesentralisasi daripada teknologi *peer-to-peer* murni karena sistem lengkap tidak memiliki titik kegagalan tunggal (siapa pun dapat menjalankan server *XMPP* mereka sendiri dan dengan demikian bergabung dengan jaringan), dan server dapat menegakkan kebijakan keamanan penting seperti otentikasi pengguna, enkripsi saluran, dan pencegahan spoofing alamat. Ini adalah salah satu keuntungan menggunakan arsitektur *client-server* yang terdesentralisasi. Terakhir, komunitas *XMPP* terus bekerja untuk menyederhanakan klien dan mentransfer kerumitan sebanyak mungkin ke server, memfasilitasi adopsi teknologi yang lebih luas. Namun, Web, email, dan Jabber semuanya memiliki perbedaan arsitektural yang signifikan.(BP, n.d.)



Gambar 2. 10 arsitektural Xmpp

2.2.7 *Php*

"*Hypertext Preprocessor*," juga dikenal sebagai *PHP*, adalah bahasa skrip yang kuat yang diinstal sebelumnya pada dokumen *HTML*. Sebagian besar sintaks *PHP* sebanding dengan *C*, *Java*, dan *Perl*, tetapi beberapa fungsinya lebih terspesialisasi. Sedangkan pembelajaran bahasa ini banyak dilakukan untuk membuat web designer dinamis dan mampu bekerja secara otomatis. (Laaziri, Benmoussa, Khouliji, & Kerkeb, 2019)

Bahasa scripting yang paling populer digunakan saat ini adalah *PHP*. Meskipun dapat digunakan untuk hal lain, *PHP* sering digunakan untuk memprogram website dinamis. Aplikasi *PHP* populer termasuk *forum (phpBB)* dan *MediaWiki* (perangkat lunak di belakang Wikipedia). Alternatif lain untuk *Microsoft ASP.NET/C#/VB.NET*, *ColdFusion* oleh *Macromedia*, *JSP* oleh *Java* oleh *Sun Microsystems*, dan *CGI* oleh *Perl* adalah *PHP*. *Mambo*, *Joomla!*, *Postnuke*, *Xaraya*, dan aplikasi berbasis *CMS* lainnya dengan tingkat kerumitan lebih tinggi yang dibuat menggunakan *PHP* adalah beberapa contohnya.

PHP awalnya adalah singkatan dari *Personal Home Page* (Situs Pribadi). Rasmus Lerdorf menemukan *PHP* pertama kali pada tahun 1995. *PHP* masih dikenal sebagai *FI* (*Form Interpreted*) pada saat itu, dan berupa kumpulan skrip yang digunakan untuk mengolah data *form* dari web. (Vivian Siahaan & Rismon Hasiholan Sianipar, 2018)

Rasmus juga menyediakan kode sumber untuk masyarakat umum dengan nama *PHP/FI*. Banyak pemrogram tertarik untuk membuat *PHP* karena kode sumbernya telah tersedia di bawah lisensi open source. Rilis *PHP/FI* 2.0 terjadi pada November 1997. *Interpreter PHP* telah diimplementasikan dalam program C untuk edisi ini. Modul ekstensi yang sangat meningkatkan *PHP*/kemampuan *FI* juga disertakan dalam edisi ini. Penerjemah *PHP* ditulis ulang pada tahun 1997 oleh sebuah perusahaan bernama Zend untuk menjadi lebih baik, lebih cepat, dan lebih bersih. Kemudian, pada bulan Juni 1998, bisnis meresmikan pengenalan juru bahasa *PHP* baru sebagai *PHP* 3.0 dan mengubah singkatan *PHP* menjadi akronim berulang *PHP: Hypertext Preprocessing*.