**PROPOSAL PENELITIAN**

**SISTEM PEMESANAN FOOD COURT MENGGUNAKAN QR CODE**

****

**HASMITA**

**216 280 200**

**FAKULTAS TEKNIK**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH PAREPARE**

**2020**

**LEMBAR PERSETUJUAN**

**SISTEM PEMESANAN FOOD COURT MENGGUNAKAN QR CODE**

****

**HASMITA**

**NIM.** **216 280 200**

Setelah diperiksa/diteliti maka dokumen ini dapat diajukan untuk mengikuti seminar proposal.

Disetujui oleh :

**Pembimbing I Pembimbing II**

**Syahirun Alam, S.T.,M.T Mughaffir Yunus ST, MT**

**NBM. 883 314 NBM. 125 2809**

**Mengetahui**

**Ketua Program Studi**

**Syahirun Alam, S.T.,M.T**

**NBM. 883 314**

**KATA PENGANTAR**



Dengan segala kerendahan hati penulis panjatkan puji dan syukur kehadirat Allah SWT., atas segala limpahan rahmat dan hidayah-Nya, serta shalawat dan Salam, senantiasa tercurahkan kepada junjungan Nabi Muhammad SAW., yang telah menyelamatkan manusia dari dunia Jahiliyah, menuju dunia terdidik yang diterangi dengan cahaya keilmuan sehingga penulis dapat menyelesaikan proposal ini dengan judul “Sistem Pemesanan *Food Court* Menggunakan *Qr Code*”.

Proposal ini diajukan untuk memenuhi salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) pada Jurusan Teknik Informatika Fakultas Tenik, Universitas Muhammadiyah Parepare (UM Parepare).

Penulis menyadari bahwa proposal ini masih banyak kekurangan baik isi maupun susunannya. Semoga proposal ini bermanfaat bagi penulis khususnya serta bagi para pembaca.

Parepare, Oktober 2019

Penyusun

HASMITA

**DAFTAR ISI**

HALAMAN SAMPUL i

[LEMBAR PERSETUJUAN](#_Toc452198982) ii

[[KATA PENGANTAR](#_Toc452198982) iii](#_Toc452198982)

DAFTAR ISI iv

BAB I PENDAHULUAN

1. [Latar Belakang](#_Toc452198983) 1
2. [Rumusan Masalah](#_Toc452198984) 3
3. Batasan Masalah 4
4. [TujuanDan Manfaat](#_Toc452198986) 4
5. [Sistematika Penulisan](#_Toc452198986) 5

**BAB II TINJAUAN PUSTAKA**

1. PKL (Praktek Kerja Lapangan) 7
2. Geolocation 8
3. Monitoring 11
4. Web Mapping 11
5. HTML 13
6. PHP 16
7. JavaScript 23
8. XAMPP [25](#_Toc452198991)
9. [Sublime](#_Toc452198991) Text 27
10. Boostrap 31
11. Flowchart 32
12. Kerangka Pikir 37

**BAB III METODE PENELITIAN**

1. Lokasi Dan Waktu Penelitian 38
2. Jenis Penelitian 38
3. Metode Pengumpulan Data 38
4. Tahapan Penelitian 39
5. Metode Pengujian 39
6. Jenis Data 39
7. Alat dan Bahan Penelitian 40
8. Desain Sistem 41
9. Sistem Yang Sedang Berjalan 41
10. Sistem Yang Diusulkan 42

**DAFTAR PUSTAKA** 43

**BAB I**

**PENDAHULUAN**

1. **Latar Belakang**

Perekonomian di Indonesia berkembang sangat pesat, terbukti dari maraknya pembangunan *mall* (pusat perbelanjaan) dikota-kota besar, hal tersebut tentu saja mempengaruhi gaya hidup masyarakat perkotaan yang cenderung konsumtif. Oleh karena itu, para pengembang berlomba-lomba untuk memberikan fasilitas-fasilitas yang terbaik demi kenyamanan pengunjung. *Food court* *center* merupakan salah satu fasilitas yang disediakan oleh pihak pengembang. Hampir semua pusat perbelanjaan menyediakan gerai-gerai makanan terpusat di lantai tertentu dalam gedung atau dikenal dengan istilah *food court*. Makanan yang disediakan masing-masing gerai berbeda sehingga pembeli bebas membeli makanan sesuai dengan selera masing-masing.

Pengelola *food court* pada umumnya menerapkan konsep *self service* pada para pembeli. Artinya para pembeli datang ke gerai, memilih makanan siap saji yang disukainya, membayar sejumlah uang pada kasir dan menunggu makanan yang akan diantar pramusaji gerai di meja makan yang telah tersedia dalam area *food court.* Akan tetapi sering kita jumpai antrian yang terjadi pada loket kasir, hal tersebut tentu saja dapat mengurangi kenyamanan pelanggan dan mengakibatkan kelalaian petugas kasir. Terdapat pula beberapa food court yang sistem pembayarannya tidak terpusat pada loket kasir tetapi melakukan pembayaran langsung pada counter makanan. Hal tersebut dapat menimbulkan masalah apabila pihak counter makanan melakukan kecurangan dengan memanipulasi laporan keuangan, sehingga pada saat pembagian pendapatan, pihak pengelola tidak mendapatkan apa yang menjadi haknya. Untuk mengatasi masalah tersebut maka dibuatlah sistem transaksi pembayaran food court yang dapat mendukung kenyamanan pelanggan pada saat melakukan transaksi pembayaran dan juga memberikan kemudahan dalam manajemen pengelolaan food court.

Sistem ini akan menerapkan penggunaan media kartu elektronik, atau disebut E-card yang berbasis barcode. Karena pada sistem ini, E-card akan menggantikan uang tunai sebagai alat bantu pembayaran, sehingga kita tidak perlu membawa uang tunai dalam jumlah yang cukup banyak sekaligus dapat mempercepat transaksi pembayaran karena pelanggan tidak perlu mengantri di loket kasir. Sistem transaksi pembayaran *food court* akan dibangun dengan menggunakan *Apache Cordova (Phonegap)* dan sistem basis data *MySQL.*

Berdasarkan permasalahan diatas, maka peneliti tertarik untuk mengadakan penelitian tentang bagaimana sistem pemesanan pada *food court* menggunakan *Qr Code* yang hasilnya dituangkan dalam judul penelitian **“SISTEM PEMESANAN FOOD COURT MENGGUNAKAN QR CODE”.**

1. **Rumusan Masalah**

Berdasarkan latar belakang diatas, maka dirumuskan masalah sebagai berikut:

1. Bagaimana membangun sistem pemesanan *food court* dengan menggunakan *Qr Code.*
2. **Batasan Masalah**

Berdasarkan rumusan masalah diatas, penulis membatasi permasalahan sebagai berikut:

1. Pembayaran menggunakan saldo yang ada pada aplikasi
2. Menggunakan sistem pembayaran E-card dengan barcode
3. Menggunakan Apache Cordova (Phonegap)
4. **Tujuan Penelitian**

Berdasarkan masalah yang telah diidentifikasi maka dapat dirumuskan tujuan dari penelitian ini adalah untuk membangun system pemesanan food court dengan menggunakan *Qr Code*

1. **Manfaat Penelitian**

Adapun manfaat dalam penelitian ini adalah sebagai berikut :

1. Pembaca

Sebagai bahan referensi untuk pengembangan penelitian selanjutnya.

1. Penulis

Menambah pemahaman tentang pengaplikasian *Qr Code* dalam berbagai bidang terutama dibidang kuliner.

1. Customer

Memudahkan customer dalam melakukan proses pemesanan makanan sampai proses pembayaran.

1. Pemilik *Food Court*

Membuka peluang pengembangan dunia kuliner dan meningkatkan sistem pelayanan yang inovatif.

1. **Sistematika Penulisan**

Sistematika mengenai keseluruhan isi tulisan, dapat kita rinci dengan menguraikan inti tiap bab sebagai berikut :

**BAB I PENDAHULUAN**

Bab ini menguraikan tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

**BAB II TINJAUAN PUSTAKA**

Bab ini berisikan tentang uraian teoritis mengenai cara membuat aplikasi Sistem Pemesanan *Food Court* Menggunakan *Qr Code.*

**BAB III METODE PENELITIAN**

Bab ini membahas tentang lokasi dan waktu penelitian, jenis penelitian, metode pengumpulan data, alat dan bahan penelitian, tahapan penelitian, metode pengujian dan desain sistem.

**BAB IV PEMBAHASAN**

Bab ini merupakan analisis sistem yang berjalan dan sistem yang akan di usulkan, serta analisis pada perangkat yang akan digunakan untuk membangun sistem dan implementasi hasil program aplikasi yang dirancang, yaitu berupa tampilan program.

**BAB V PENUTUP**

Bab ini berisikan tentang kesimpulan hasil penelitian dan saran pengembangan lebih lanjut.

**BAB II  
TINJAUAN PUSTAKA**

1. **Tinjauan Penelitian Sebelumnya**
2. David Timothy Ivander, dkk Universitas Bina Nusantara melakukan penelitian dengan judul “*Aplikasi Pemesanan Makanan dan Layanan Restoran pada Smartphone Berbasis Android Menggunakan Qr Code*”. Penelitian ini bertujuan untuk memanfaatkan smartphone sebagai media untuk melakukan pemesanan makanan pada restoran. Penelitian ini ditujukan untuk mendukung dan menunjang teknologi khususnya dibidang industri restoran dalam mengelola kegiatan sehari-hari yang dilakukan Pada penelitian ini aplikasi yang digunakan hanya mendukung *smartphone* dengan sistem operasi android..
3. Yori Zulhanif, Universitas Andalas Fakultas Teknik Program Studi Sistem Komputer melakukan penelitian dengan judul “Perancangan *Smart Food Court* Menggunakan IoT“ hasil yang diperoleh dari perancangan sistem ini adalah pembeli dapat mengambil makanan yang ada pada *food court* dengan menggunakan aplikasi *mobile* serta pembayaran dilakukan secara otomatis, sehingga tidak adanya antrian pembayaran pada kasir. Pada sistem ini tidak menyediakan sistem admin pada aplikasi dan tampilan antarmuka yang kurang *user friendly*.
4. **Food Court**

Food court merupakan sebuah tempat makan yang terdiri dari counter-counter makanan yang menawarkan aneka menu yang bervariasi. Food court aadalah area makan yang terbuka dan bersifat informal. Area ini biasanya terdapat di mall-mall atau pusat perbelanjaan, perkantoran modern, universitas, atau sekolah-sekolah yang modern. Pemilik gedung biasanya mempekerjakan beberapa orang yang bertugas untuk mengelola dan menjalankan food court di gedung miliknya. Untuk mengelolanya, pemilik gedung juga bisa memberikan penawaran kepada sebuah perusahaan pengelolaan properti yang berpengalaman dalam mengelola food court.

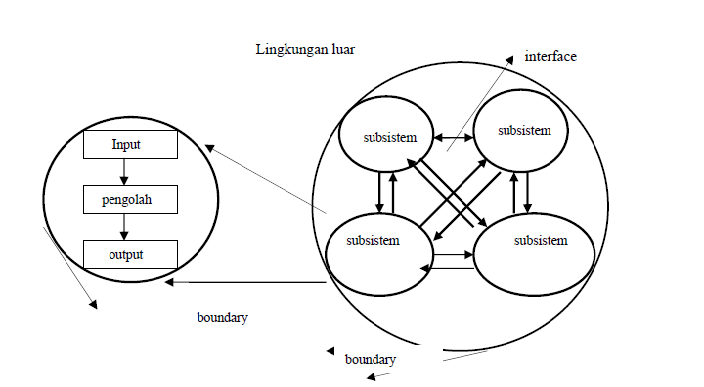
Selain mampu untuk menarik perhatian banyak pelanggan, food court juga bisa menjadi salah satu ajang untuk membuka pusat keramaian yang sangat efektif.

1. **Sistem**

Sistem adalah seperangkat bagian-bagian yang saling berhubungan erat satu dengan lainnya untuk mencapai tujuan bersama-sama. Subsistem sebenarnya adalah sistem di dalam suatu sistem, sebagai contoh, pesawat terbang adalah suatu sistem yang terdiri dari sistem-sistem bawahan seperti mesin, sistem badan pesawat dan sistem rangka. Masing-masing sistem ini terdiri dari sistem tingkat yang lebih rendah lagi, misal sistem mesin adalah kombinasi dari sistem karburator, sistem bahan bakar dan seterusnya. Istilah subsistem digunakan untuk memudahkan analisis dan pengkomunikasian.

Berikut adalah karakter atau sifat-sifat tertentu yang dimiliki sistem:

* Mempunyai komponen *(component)*
* Bats sistem *(boundary)*
* Penghubung sistem *(interface)*
* Masukan sistem *(input)*
* Keluaran sistem *(output)*
* Pengolahan sistem *(proces)*
* Lingkungan luar sistem *(environments)*
* Sasaran suatu tujuan *(goal)*



**Gambar 1. Karateristik sistem**

1. **Reservasi**

Pemesanan dalam bahasa Inggris adalah *Reservation* yang berasal dari kata *“to reserve”* yaitu menyediakan atau mempersiapkan tempat sebelumnya. Sedangkan *reservation* yaitu pemesanan suatu tempat fasilitas. Pengertian reservasi adalah sebuah proses perjanjian berupa pemesanan sebuah produk baik barang maupun jasa dimana pada saat itu telah terdapat kesepahaman antara konsumen dengan produsen mengenai produk tersebut namun belum ditutup oleh sebuah transaksi jual beli. Pada saat reservasi berlangsung biasanya ditandai dengan adanya proses tukar menukar informasi antara konsumen dan produsen agar kesepahaman mengenai produk dapat terwujud (Suartana, 1987).

Layaknya perjanjian lainnya, reservasi akan dinyatakan batal apabila terjadi beberapa hal berikut (1) Terjadi sesuatu di luar kendali manusia (bencana alam, perang, dll) sehingga produk tidak dapat dibuat atau diselenggarakan oleh produsen, (2) Konsumen melakukan pembatalan sebelum batas waktu, (3) Terdapat kesalahan pengisian data penting mengenai konsumen pada saat proses reservasi sehingga produsen harus membatalkan reservasi (catatan: tergantung kebijakan masing-masing perusahan), (4) Tidak dilakukan transaksi terhadap produk yang dipesan sampai *time limit* yang telah ditentukan, (5) Baik produsen maupun konsumen melakukan hal-hal yang menyebabkan reservasi harus dibatalkan, tergantung perjanjian yang dibuat/ditentukan (Suartana, 1987)

1. **Teknologi *Quick Response Code* (QR *Code)***

QR Code adalah barcode dua dimensi yang dapat menyimpan data. QR Code dikembangkan oleh Denso Corporation, Jepang dan dapat digunakan secara gratis., bahkan untuk keperluan komersial. Besaran data yang dapat disimpan bervariasi, tergantung pada versi QR Code, ukuran QR Code dan tingkat *Error Correction Capability-*nya.

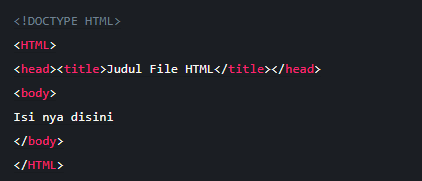
Berikut tabel yang menunjukkan variasi besaran data yang dapat dimuat QR Code :

|  |  |  |  |
| --- | --- | --- | --- |
| Versi QR Code | Modul | Tingkat Error Correction Capability | Jumlah Data (bits) |
| 1 | 21x21 | L | 152 |
| M | 128 |
| Q | 104 |
| H | 72 |
| 2 | 25x25 | L | 272 |
| M | 224 |
| Q | 176 |
| H | 128 |
| 3 | 29x29 | L | 440 |
| M | 352 |
| Q | 272 |
| H | 208 |

QR Code dapat berisikan konten yang umum, seperti URL ke sebuah halaman web atau unduhan file, business card, dan lain-lain. QR Code sejenis ini dapat di-scan dengan QR-*scanner* apapun. Namun demikian, terdapat QR Code yang dikhususkan untuk di-scan menggunakan aplikasi tertentu. Jika QR Code jenis ini di-scan menggunakan sembarang QR-Scanner, akan menghasilkan pesan yang tidak dimengerti

1. **HTML**

HTML merupakan singkatan dari *Hypertext Markup Language*. HTML digunakan untuk membangun suatu halaman web. HTML bukan suatu bahasa pemograman melainkan suatu bahasa markup. HTML merupakan bahasa standar untuk membuat suatu dokumen HTML (halaman web) yang terdiri dari kode-kode singkat tertentu, dimana dengan kode-kode tersebut akan memerintahkan web browser bagaimana untuk menampilkan halaman web yang terdiri dari dari berbagai macam format file seperti teks, grafik, animasi, *link* maupun audio-video. Adapun struktur kode HTML yaitu :



Keterangan :

1. <!DOCTYPE HTML> : tag ini berfungsi untuk menginformasikan pada browser bahwa dokumen yang sedang dibuka adalah dokumen HTML. Tag ini perlu dicantumkan disetiap dokumen HTML yang akan dibuat.
2. <HTML> … </HTML> : Awal dari isi dokumen HTML dimulai dari sini, semua kode HTML yang akan dibuat akan ditulis di dalam tag ini, perhatikan juga bahwa setiap tag harus memiliki tag penutup.
3. <head> … </head> : Tag Head akan digunakan untuk menyimpan berbagai informasi tentang dokumen HTML. Apa yang terdapat di dalam tag ini, tidak akan ditampilkan secara langsung pada web browser.
4. <title> … </title> : Salah satu contoh informasi yang terdapat di dalam tag Head adalah title. Title akan menampilkan teks pada judul browser atau pada tab browser.
5. <body> … </body> : tag body merupakan tag pembuka dari badan dokumen HTML.

**Kelebihan dan kekurangan HTML:**

**Kelebihan:**

1. Bahasa yang digunakan secara luas dan memiliki banyak sumber serta komunitas yang besar.
2. Dijalankan secara alami di setiap web browser.
3. Memiliki *learning curve* yang mudah.
4. *Open-source* dan sepenuhnya gratis.
5. Bahasa markup yang rapi dan konsisten.
6. Standard web yang resmi di-*maintain* oleh *World Wide Web Consortium* (W3C).
7. Mudah diintegrasikan dengan bahasa *backend*, seperti PHP dan Node.js.

**Kekurangan:**

1. Paling sering digunakan untuk halaman web statis. Untuk fitur dinamis, Anda bisa menggunakan JavaScript atau bahasa backend, seperti PHP.
2. HTML tidak memungkinkan user untuk menjalankan *logic*. Alhasil, semua halaman web harus dibuat terpisah meskipun menggunakan elemen yang sama, seperti header dan footer.
3. Fitur-fitur baru tidak bisa digunakan secara cepat di sebagian browser.
4. Terkadang perilaku browser susah untuk diprediksi (misalnya, browser lama tidak selalu bisa render tag yang lebih baru).
5. **PHP**

****

**Gambar 2.1**

PHP adalah singkatan dari *Hypertext Preprocessor.* PHP adalah bahasa pemprograman yang dipakai untuk membuat web dinamis, meskipun juga bisa dipakai untuk membuat program lain. PHP juga dikenal sebagai pemprograman *script server side,* sebab PHP di proses pada komputer server.

Variabel adalah suatu simbol untuk tempat penyimpanan sementara yang nilainya dapat berubah-ubah. Data yang tersimpan pada variabel akan hilang setelah program selesai dijalankan dan nilai pada variabel akan di *reset* menjadi *default*.. Adapun cara penulisan variabel yaitu:

* + - Untuk memberi nama variabel hanya boleh menggunakan huruf, angka dan garis bawah ( \_ ).
    - Nama variabel harus diawali dengan tanda **$**.
    - Tidak boleh menggunakan spasi pada nama variabel.
    - Variabel bersifat case-sensitive atau sensitif terhadap huruf besar dan kecil.



1. **Tipe data PHP :**
   1. **Integer**

Tipe data yang berbentuk angka bilangan asli atau bilangan bulat. Untuk penulisan tipe data integer tidak perlu menambahkan tanda petik karena akan dibaca sebagai tipe data string jika ditambahkan tanda petik pada tipe data integer.



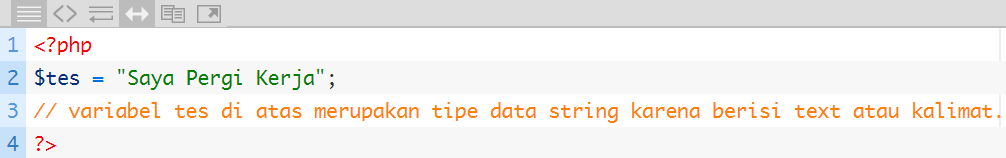
* 1. **Float**

Tipe data float dapat di katakan juga bilangan berkoma (,) atau bilangan desimal. tapi dalam pemerogramannya tidak menggunakan koma melainkan titik (.).



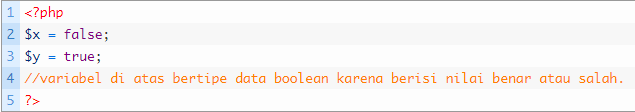
* 1. **String**

Tipe data string adalah tipe data yang berbentuk teks atau kalimat. Cara penulisan tipe data string yaitu diletakkan di tengah-tengah tanda petik. Di awali dengan tanda petik dan di akhiri dengan tanda petik.



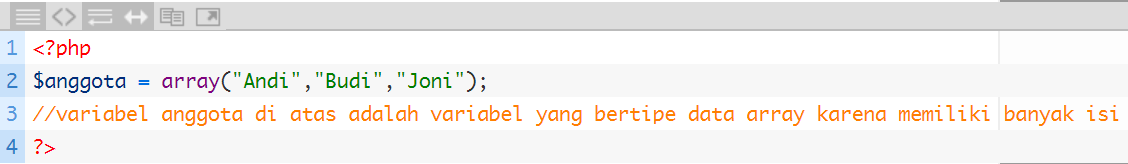
* 1. **Boolean**

Tipe data Boolean merupakan tipe data yang bernilai *True* atau *False.* Tipe data boolean sering digunakan untuk memeriksa ketersediaan atau memeriksa nilai kebenaran pada sebuah data.



* 1. **Array**

Array merupakan kumpulan dari tipe data, didalamnya memiliki banyak nilai. Jadi dapat dikatakan array merupakan tipe data yang terdiri dari kumpulan tipe data.



1. **Sintaks PHP beserta fungsinya**

|  |  |
| --- | --- |
| **Sintaks** | **Fungsi** |
| [<!-- ... -->](http://eliastutik.blogspot.com/2011/12/comment-tag.html) | Digunakan untuk memberi sebuah komentar atau keterangan. Kalimat yang terletak pada kontiner ini tidak akan terlihat pada browser |
| [<!DOCTYPE>](http://eliastutik.blogspot.com/2011/12/deklarasi-doctype_30.html) | Mendefinisikan informasi tipe dokumen |
| <a> | Mendefinisikan sebuah anchor, tetapi lebih tepat jika diartikan sebagai tautan dikarenakan tautan ini digunakan untuk saling menautkan antara satu dokumen HTML ke dokumen HTML yang lain |
| <abbr> | Digunakan untuk menguraikan satu ungkapan yang disingkat dan Anda dapat memberikan informasi bermanfaat kepada penelusur-penulusur page source(halaman sumber kode) / pembaca layar, sistem terjemahan dan mesin pencari yang berasal dari singkatan yang sudah diuraikan, tetapi saat di browser uraian tersebut tidak akan tampil dan hanya sebagai informasi saja |
| <acronym> | Mendefinisikan akronim / fungsi tag ini kurang lebih sama dengan tag <abbr> |
| <address> | Mendefinisikan informasi kontak untuk penulis/pemilik dokumen |
| <applet> | Digunakan untuk memasukan file java kedalam dokumen HTML |
| <area /> | Mendefinisikan daerah yang dapat diklik (link) pada peta gambar |
| <b> | Membuat teks tebal |
| <base /> | Mendefinisikan URL dasar/target untuk semua URL relatif dalam dokumen |
| <basefont /> | Membuat atribut teks default, seperti warna, ukuran, jenis font untuk semua teks dalam dokumen |
| <bdo> | Digunakan untuk menimpa arah teks |
| <big> | Memperbesar ukuran teks sebesar satu point dari defaultnya |
| <blink> | Membuat teks berkedip |
| <blockquote> | Mendefinisikan sebuah kutipan panjang. Pada saat di browser teks akan tampil menjorok kedalam |
| <body> | Mendefinisikan body/isi dokumen HTML, berfungsi untuk menentukan bagaimana isi suatu dokumen ditampilkan di web browsernya. Isi dokumen tersebut dapat berupa teks, gambar, animas, link dan seterusnya |
| <br /> | Memberi baris baru/pindah baris |
| <button> | Mendefinisikan sebuah tombol diklik |
| <caption> | Membuat caption pada tabel |
| <center> | Untuk perataan tengah terhadap teks atau gambar |
| <cite> | Mendefinisikan kutipan |
| <code> | Mendefinisikan sebuah bagian dari kode komputer |
| <col /> | Mendefinisikan nilai atribut dari satu kolom atau lebih dalam sebuah tabel |
| <colgroup> | Menentukan kelompok dari satu kolom atau lebih dalam sebuah tabel untuk performatan |
| <dd> | Mendefinisikan deskripsi dari item dalam daftar definisi |
| <del> | Untuk memberi garis tengah pada teks/mencoret teks |
| <dfn> | Mendefinisikan sebuah istilah definisi |
| <dir> | Mendefinisikan sebuah daftar direktori |
| <div> | Mendefinisikan sebuah section dalam dokumen |
| <dl> | Mendefinisikan sebuah daftar definisi |
| <dt> | Mendefinisikan istilah (item) dalam daftar definisi |
| <em> | Membuat teks miring. Fungsi tag ini sama dengan tag <i> tetapi tag <em> yang lebih dianjurkan/ditekankan pada penggunaan untuk teks miring |
| <embed> | Digunakan untuk memasukkan file video atau file musik |
| <fieldset> | Untuk mengelompokkan elemen-elemen yang terkait dalam form / membuat seperti frame-box di dalam form |
| <font> | Mendefinisikan jenis font, warna dan ukuran untuk teks |
| <form> | Mendefinisikan sebuah form HTML untuk input form |
| <frame /> | Mendefinisikan frame dalam fremeset |
| <frameset> | Mendefinisikan satu set frame |
| <h1> to <h6> | Digunakan untuk menunjukkan awal dari suatu header/judul dari dokumen HTML tersebut. |
| <head> | Digunakan untuk memberikan informasi tentang dokumen tersebut |
| <hr /> | Membuat garis horisontal |
| <html> | Mendefinisikan root dari suatu dokumen HTML |
| <i> | Membuat teks miring |
| <iframe> | Mendefinisikan sebuah inline frame |
| <img /> | Berfungsi untuk menampilkan gambar pada dokumen HTML |
| <input /> | Mendefinisikan input field pada form |
| <ins> | Membuat teks bergaris bawah |
| <kbd> | Mendefinisikan teks yang di input dari keyboard |
| <label> | Mendefinisikan label untuk sebuah elemen <input> |
| <legend> | Mendefinisikan sebuah caption untuk elemen <fieldset> |
| <li> | Digunakan untuk menampilkan informasi dalam bentuk item daftar |
| <link /> | Mendefinisikan hubungan antara dokumen dan sumber eksternalnya |
| <listing> | Fungsi tag ini sama dengan tag <pre> dan dianjurkan menggunakan tag <pre> karena tag <listing> tidak layak/diprotes |
| <map> | Mendefinisikan client-side peta gambar |
| <marquee> | Membuat teks berjalan secara vertikal atau horizontal |
| <menu> | Mendefinisikan sebuah daftar menu |
| <meta /> | Mendefinisikan metadata tentang sebuah dokumen HTML |
| <nobr> | Mencegah ganti baris pada teks atau gambar |
| <noframes> | Jika browser user tidak mendukung frame |
| <noscript> | Jika browser user tidak mendukung client-side scripts |
| <object> | meletakkan embed sebuah objek |
| <ol> | Mendefinisikan daftar dalam format penomoran |
| <optgroup> | Menampilkan beberapa pilihan yang sudah dikelompokkan dalam bentuk sebuah daftar drop-down |
| <option> | Menampilkan beberapa pilihan yang berbentuk dalam sebuah daftar drop-down |
| <p> | Membuat sebuah paragraf |
| <param /> | Mendefinisikan sebuah parameter untuk objek |
| <pre> | Membuat teks dengan ukuran huruf yang sama |
| <q> | Mendefinisikan sebuah kutipan singkat |
| <s> | Untuk memberi garis tengah pada teks/mencoret teks, fungsi tag ini sama dengan tag <del> tetapi tag <s> tidak dianjurkan sebagai gantinya menggunakan tag <del> |
| <samp> | Mendefinisikan contoh keluaran dari program komputer |
| <script> | Mendefinisikan client-side script |
| <select> | Membuat daftar drop-down |
| <small> | Memperkecil ukuran teks dari ukuran defaultnya |
| <span> | Mendefinisikan sebuah section dalam dokumen |
| <strike> | Untuk memberi garis tengah pada teks/mencoret teks, fungsi tag ini sama dengan tag <del> |
| <strong> | Membuat teks tebal, fungsi tag ini sama dengan tag <b> |
| <style> | Mendefinisikan informasi style untuk dokumen HTML |
| <sub> | Memberikan efek subscript pada teks |
| <sup> | Memberikan efek superscript pada teks |
| <table> | Membuat tabel |
| <tbody> | Untuk mengelompokkan isi body di dalam sebuah tabel |
| <td> | Mendefinisikan sel di dalam sebuah tabel |
| <textarea> | Mendefinisikan sebuah kontrol input multiline |
| <tfoot> | Untuk mengelompokkan isi footer di dalam sebuah tabel |
| <th> | Mendefinisikan sel header di dalam  sebuah tabel |
| <thead> | Untuk mengelompokkan isi header di dalam sebuah tabel |
| <title> | Membuat judul untuk dokumen HTML |
| <tr> | Membuat baris di dalam sebuah tabel |
| <tt> | Mendefinisikan teletype text |
| <u> | Membuat teks bergaris bawah, fungsi tag ini sama dengan tag <ins> tetapi tag <u> tidak dianjurkan untuk kategori HTML text formatting melainkan termasuk kategori HTML Style |
| <ul> | Mendefinisikan daftar dalam format bullet |
| <var> | Mendefinisikan sebuah variabel |

1. **JavaScript**

JavaScript adalah [bahasa](https://id.wikipedia.org/wiki/Bahasa) pemrograman [tingkat tinggi](https://id.wikipedia.org/wiki/Bahasa_pemrograman_tingkat_tinggi) dan [dinamis](https://id.wikipedia.org/wiki/Bahasa_pemrograman_dinamis). [Kode](https://id.wikipedia.org/wiki/Kode) JavaScript dapat disisipkan dalam [halaman](https://id.wikipedia.org/wiki/Halaman) [web](https://id.wikipedia.org/wiki/Web) menggunakan [tag](https://id.wikipedia.org/wiki/Tag) script. JavaScript merupakan salah satu teknologi inti [*World Wide Web*](https://id.wikipedia.org/wiki/World_Wide_Web)(WWW)  selain [HTML](https://id.wikipedia.org/wiki/HTML) dan [CSS](https://id.wikipedia.org/wiki/Cascading_Style_Sheets). JavaScript membantu membuat halaman web interaktif dan merupakan bagian aplikasi web yang esensial.

Untuk mulai menerapkan program JavaScript dalam web, ada beberapa hal yang harus diketahui oleh seorang perancang web yaitu:

1. Perancang harus mengetahui cara menggunakan HTML dan mengedit­­­­ dokumen HTML.
2. Perancang harus menggunakan browser yang sudah mendukung pemprograman JavaScript, misalnya internet explorer versi 4 atau versi diatasnya, netscape versi 3 atau diatasnya, dll.

Dengan JavaScript, sebuah halaman web akan lebih menjadi dinamis dan interaktif terhadap user karena halaman web mampu berfungsi sebagai sebuah program aplikasi yang dapat memproses masukan yang diberikan user dan memberikan hasil sesuai dengan yang telah diprogramkan. Penulisan program JavaScript harus di awali dengan tag <script language=”JavaScript”> dan diakhiri dengan tag </script>.

|  |
| --- |
| <script type=”text/JavaScript”>  \*\* kode skrip ditulis disini\*\*  </script> |

Adapun kelebihan dan kekurangan JavaScript sebagai berikut :

Kelebihan :

1. Lebih praktis dan mudah sebab bahasa pemprograman JavaScript memiliki sedikit sintaks.
2. Koneksi cepat sebab peletakan program terdapat disisi *client,* berukuran file sangat kecil dan dapat langsung dijalankan di *browser.*

Kekurangan :

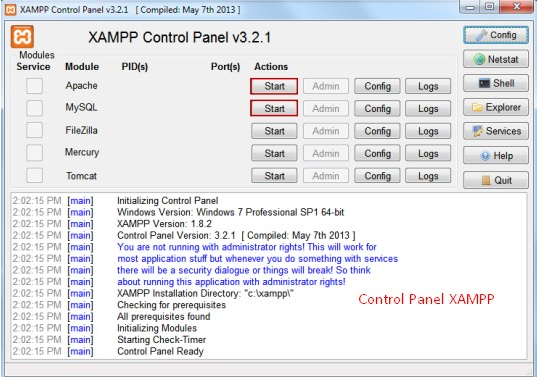
1. Pengelolaan objek dalam JavaScript sangat terbatas.
2. Penggunaan Script dalam di *copy* langsung melalui sebuah web browser, sehingga setiap orang dapat menggunakan program JavaScript yang telah dibuat.
3. **XAMPP**

XAMPP merupakan program aplikasi pengembang yang berguna untuk pengembangan website berbasis PHP dan MySQL. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

Fungsi XAMPP adalah sebagai server yang berdiri sendiri (localhost) yang terdiri atas program Apache HTTP Server, MySQL database dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Adapun bagian-bagian penting pada XAMPP adalah sebagai berikut :

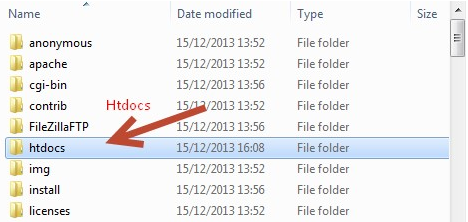
1. **Control Panel**

Control panel merupakan halaman yang di ibaratkan sebagai kunci karena control panel ini berguna untuk menghidupkan dan menghentikan layanan XAMPP.



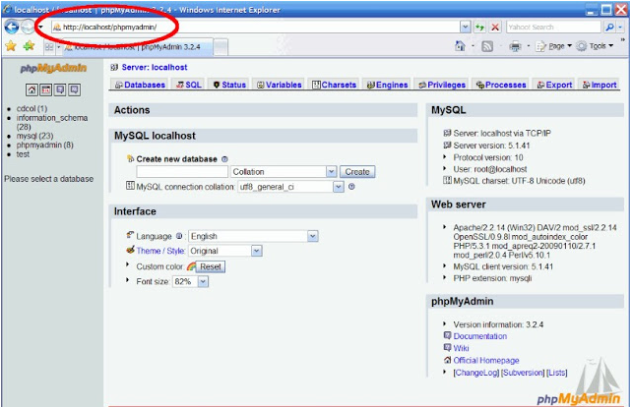
1. **Htdocs**

Htdocs yaitu folder tempat meletakkan berkas-berkas yang akan dijalankan. Di Windows, folder ini berada di C:/xampp.



1. **PhpMyAdmin**

PHPMyAdmin merupakan bagian untuk mengelola database. berfungsi untuk membuat, menyimpan, menambah ataupun menghapus database dari halaman web. Untuk mengakses PhpMyAdmin yaitu dengan membuka browser di internet lalu ketikkan alamat [*http://localhost/phpMyadmin*](http://localhost/phpMyadmin) maka akan muncul tampilannya..



Di dalam bagian ini ada beberapa menu yang berfungsi sebagai berikut:

* + - **Tab database**, berguna untuk membuat database halaman sebuah web.
    - **Tab SQL**, berguna untuk membuat database SQL, sehingga tidak perlu untuk merubah file ke dalam format lain, jika file sudah berektensi SQL, yang artinya hanya perlu dimasukkan kedalam database.
    - **Tab status**, untuk menampilkan status.
    - **Tab users**, berguna untuk mengedit nama pengguna dan password.
    - **Tab export**, berguna dalam pengunduhan kode SQL dalam database.
    - **Tab import**, berguna untuk mengimpor kode SQL.
    - **Tab Settings**, yaitu berguna dalam mengelola dan mengatur jalannya aplikasi XAMPP.

1. **MySQL**

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 dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, di mana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, di mana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius. ( Andrea Adelheid dan Khairil Nst,2013)

1. **Apache Cordova (PhoneGap)**

****

**Gambar 2.**

Apache Cordova adalah platform untuk membangn *native mobile applications* menggunakan HTML, CSS dan JavaScript. *Native Mobile Applications* yang didukung antara lain Android, iOS, Windows Phone dan Blackberry. Apache Cordova berisi sekumpulan API *(Application Programming Interface)* untuk mengakses *device* dari perangkat mobile. *Device* itu antara lain kamera, GPS *(Global Positioning System),* storage dan lain-lain. Dengan menggunakan UI *(User Interface) framework* seperti *jQuery Mobile, Dojo Mobile,* atau *Sencha Touch*, maka kita dapat mengakses API ini. Dengan kata lain kita dapat membangun aplikasi hanya menggunakan HTML, CSS, dan JavaScript.

Dengan menggunakan API dari Cordova, kita tidak perlu membangun aplikasi menggunakan native code seperti Java, Objective-C dan lainnya. Kita hanya menggunakan teknologi web dengan bahasa pemrograman web dan diinstall pada perangkat mobile yang bersangkutan. Karena menggunakan JavaScript, aplikasi yang kita bangun pada suatu platform mobile dapat digunakan di platform lain dengan sedikit atau tanpa perubahan.

Aplikasi yang dihasilkan dari Cordova adalah dikemas dalam aplikasi menggunakan SDK masing-masing platform dan dapat diterapkan ke platform lain menggunakan SDK platform tersebut.

Cordova cocok untuk

1. Seorang programmer untuk perangkat mobile yang multi platform, cukup membuat sekali coding tanpa harus melakukan implementasi ulang pada setiap platform
2. Seorang web developer di mana ada aplikasi yang perlu dijalankan pada perangkat mobile untuk mengakses aplikasi yang dibuatnya
3. Seorang programmer perangkat mobile yang tertarik membangun plugin dalam *WebView*
4. **Perancangan Basis Data**

Dalam perancangan basis data adalah bagaimana merancang struktur logika dan fisikal dari satu atau lebih basis data untuk memenuhi kebutuhan informasi yang diperlukan oleh pemakai sesuai dengan aplikasi-aplikasi yang telah ditentukan. Oleh karena itu diperlukan tahapan proses perancangan basis data yang dapat diharapkan memperoleh hasil yang sesuai dengan tujuan yaitu :

Koleksi dan analisis persyaratan

Peancangan konsepsual basis data

Pemilihan SMBD (Sistem Manajemen Basis Data)

Perancangan logika basis data

Perancangan fisikal basis data (pemetaan model data)

Implementasi sistem basis data

Dalam pelaksanaan peancangan basis data terdapat dua kegiatan yang secara paralel harus dilakukan yaitu :

1. Perancangan struktur dan isi data atau disebut juga analisis data
2. Perancangan pemrosesan data dan program aplikasi (transaksi data) ataudisebut juga analisis fungsional.

Menurut Abdul Kadir (2000) dalam tahap perancangan sistem dibagi menjadi dua bagaian yaitu :

1. Perancangan basis data
2. Perancangan proses

Perancangan basis data merupakan langkah untuk menentukan basis data yang diharapkan dapat mewakili seluruh kebutuhan pengguna. Pada perancangan basis data secara konseptual terdiri atas tiga langkah berikut :

1. Penentuan entitas pada basis data
2. pendefinisian hubungan antar entitas dan,
3. Penerjemahan hubungan ke dalam entitas.

Adapun penjelasan dari komponen-komponen perancangan basis data adalah sebagai berikut :

1. Entitas : terkadang disebut tipe entitas atau kelas entitas. Entitas menyatakan objek atau kejadian.
2. Atribut : adalah item data yang menjadi bagian dari suatu entitas. Istilah lain atribut adalah property.
3. Hubungan : adalah asosiasi atau kaitan antara dua entitas.
4. Kekangan : digunakan untuk melindungi integritas data
5. Domain : adalah himpunan nilai yang berlaku bagi suatu atribut.
6. Integritas referensial : adalah aturan-aturan yang mengatur hubungan antara kunci primer dengan kunci tamu milik table-tabel yang berada dalam suatu basis data relasional untuk menjaga konsistensi data. Tujuan integritas referensial adalah untuk menjamin agar elemen dalam suatu tabel yang menunjuk ke suatu nilai yang memang ada.

Menurut Abdul Kadir (2000) dalam beberapa literatur, diagram struktur data juga dianggap sebagai diagram E-R. Perlu diketahui bahwa tak ada standar diagram E-R sehingga ditemukan adanya berbagai variasi.

Setelah hubungan antar entitas didefinisikan, hubungan akan diterjemahkan kedalam table. Hal ini dilakukan melalui tiga langkah :

1. Penentuan kunci untuk entitas
2. Penerjemahan hubungan kedalam kunci tamu, dan
3. Penormalisasian basis data.

Pada model relasional entitas akan diterjemahkan menjadi table atau relasi. Tabel adalah koleksi rekaman. Secara khusus rekaman disebut *tupel* (baris), dan atribut atau properti suatu entitas akan disebut atribut (kolom) pada model relasional.

Kunci tidak sekedar sebagai metode untuk mengakses suatu baris tertentu, tetapi sekaligus juga dapat menjadi pengenal unik (tidak ada yang kembar) terhadap table. Untuk diketahui bahwa tidak semua kunci dapat menjadi pengenal yang unik.

* 1. Kunci primer : adalah kunci kandidat yang dipilih sebagai kunci utama untuk mengidentifikasi baris dalam table.
  2. Kunci alternatif : adalah semua kunci kandidat yang tidak bertindak sebagai kunci primer
  3. Kunci tamu adalah : sebrang atribut yang menunjuk ke kunci primer pada table lain.

1. **UML(Unified Modelling Language)**

UML (Unified Modelling Language) menurut menurut Adi nugroho (2010:6), “*Unified Modelling Language* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berpradigma berorientasi objek”. Pemodelan (*modeling*) sesungguhnya  digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Metode *Unified Modelling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan sistem atau perangkat lunak yang akan dikembangkan, yaitu:

1. Sesuatu (*things*)

Ada empat *things* dalam *Unified Modelling Language* (UML):

1. *Structural things*, bagian yang relatif statis dapat berupa elemen-elemen yang bersifat fisik maupun konseptual.
2. *Behavorial things,* bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang waktu
3. *Grouping things,* bagian pengorganisasian dalam UML. Dalam penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta subsitem-subsistem.
4. *An notational things,* merupakan bagian yang meperjelas model UML. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap element dalam model UML
5. Relasi (*relationship*)

Ada empat *relationship* (hubungan) dalam *Unified Modelling Language* (UML):

1. Ketergantungan (d*ependency*) adalah hubungan dimana perubahan yang terjadi pada suatu elemn independent akan mempengaruhi elemen yang bergantung padanya.
2. Asosiasi adalah apa dan bagaimana yang mengubungkan antara objek satu dengan yang lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.
3. Generalisasi adalah hubungan dimana objek anak berbagai perilaku dan struktur data dari objek yang ada diatasnya (objek induk). Arah dari objek induk ke objek anak dinamakan spesialisasi sedangkan arah sebaliknya dinamakan generalisasi..
4. Realisasi adalah operasi yang benar-benar dilakukan oleh suatu objek.
5. Diagram, *Unified Modelling Language* (UML) menyediakan Sembilan jenis diagram yang dapat dikelompokkan berdasarkan sifatnya (statis dan dinamis).

diagram *use case*

Diagram *use case* menyajikan interaksi antara *use case* dan aktor, dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang di bangu. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai.( *Sholiq, 2006*)

Adapun simbol-simbol Use Case Diagram antara lain :

Tabel 1. Simbol Use Case Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Actor* | Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
| 2 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
| 3 |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 4 |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara *eksplisit*. |
| 5 |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| 6 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

Tabel 2. Simbol Class Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Generalization* | Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 2 |  | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | *Collaboration* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 5 |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 6 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 7 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya |

Tabel 3. Simbol Sequence Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *LifeLine* | Objek *entity*, antarmuka yang saling berinteraksi. |
| 2 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |
| 3 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |

Tabel 4. Simbol StateChart Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *State* | Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek. |
| 2 |  | *Initial Pseudo State* | Bagaimana objek dibentuk atau diawali |
| 3 |  | *Final State* | Bagaimana objek dibentuk dan dihancurkan |
| 4 |  | *Transition* | Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya |
| 5 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 6 |  | *Node* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

Tabel 5. Simbol Actifity Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| **1** |  | *Actifity* | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| **2** |  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| **3** |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| **4** |  | *Actifity Final Node* | Bagaimana objek dibentuk dan dihancurkan |
| **5** |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

1. **Kerangka Pikir**

Untuk memperjelas alur penelitian ini dibuatkan kerangka pikir sebagai berikut:

Sistem pemesanan food court adalah sistem transaksi yang mendukung kenyamanan pelanggan dan memberikan kemudahan dalam manajemen pengelolaan food court

Sistem pemesanan saat ini dilakukan dengan mendatangi kasir dan melakukan transaksi secara manual yang menyebabkan antrian serta manajemen pengelolaan food court kurang efisien

Sistem pemesanan food court menggunakan Qr Code ini akan dijadikan solusi untuk mengatasi permasalahan tersebut

Diharapkan dengan adanya sistem ini dapat memberikan kenyamanan bagi pelanggan dalam melakukan transaksi pembayaran dan memberikan kemudahan dalam manajemen pengelolaan food court

**Gambar. 2.3**

**BAB III**

**METODE PENELITIAN**

1. **Lokasi dan waktu penelitian**

Penelitian ini dilakukan di Universitas Muhammadiyah Parepare dan waktu penelitian yang dilakukan ± 2 bulan.

1. **Jenis Penelitian**

Jenis penelitian yang digunakan adalah penelitian deskriptif dimana memberikan gambaran mengenai apa yang sesungguhnya terjadi. Dalam pembuatan proposal ini digunakan metode deskripsif yang menggambarkan fakta-fakta dan informasi secara sistematis, faktual dan akurat.

Penelitian ini dilakukan melalui internet yang dapat memberikan sumber data dan pengetahuan mengenai sistem yang diteliti, kemudian mencocokkan dengan kemungkinan yang terjadi dalam usaha penyelesaian masalah.

1. **Metode Pengumpulan data**
2. Studi literatur

Studi literatur dengan melakukan pencarian buku, karya ilmiah maupun jurnal, dan artikel yang berhubungan dengan judul, baik yang terdapat di perpustakaan maupun internet.

1. Wawancara

Wawancara adalah suatu teknik pengumpulan data untuk mendapatkan informasi yang digali dari sumber data langsung melalui percakapan atau tanya jawab.

1. **Tahapan Penelitian**

Tahapan penelitian yang dimaksud dalam penelitian ini ada beberapa tahapan yaitu persiapan penelitian, pengumpulan data, analisis perancangan, pengujian dan implementasi. Adapun Uraian dari tahapan tersebut adalah sebagai berikut :

1. Persiapan Penelitian

Pada tahapan ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topik penelitian serta software yang digunakan selama penelitian.

1. Pengumpulan Data

Pada tahap ini peneliti melakukan observasi dengan peninjauan, pencatatan dan pengamatan langsung di tempat penelitian.

1. Analisis

Pada tahap analisis, peneliti melakukan analisa terhadap sistem yang di terapkan sekarang berdasarkan kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

1. Perancangan

Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah.

1. Pengujian

Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

1. Implementasi

Setelah pada perancangan tidak terdapat kekurangan maka aplikasi siap untuk di gunakan oleh user.

1. **Jenis Data**

Data yang digunakan dalam penelitian ini berupa data-data yang telah dikumpulkan melalui Penelitian Pustaka (*Library Research*). Adapun jenis data primer dan data sekunder yang relevan dengan masalah yang akan dibahas.

* + - 1. Data primer

Data Primer adalah data yang berasal atau data yang diperoleh langsung dari sumber data dan pengetahuan.

* + - 1. Data Sekunder

Data sekunder adalah data yang diperoleh tidak secara langsung dari objek penelitian. Peneliti mendapatkan data yang sudah jadi dari internet, website dan jurnal.

1. **Metode Pengujian Aplikasi**

Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi, query, atau jalur navigasi yang mewakili penggunaan sistem yang *tipical,* kritis atau abnormal. Isu kunci pada pengembangan sistem adalah pemilihan sekelompok *test-case* yang cocok, sekecil dan secepat mungkin, untuk meyakinkan perilaku sistem secara detail. Pengujian harus mencakup *unit testing*, yang mengecek validasi dari prosedur dan fungsi-fungsi secara independen dari komponen sistem yang lain. Kemudian modul testing harus menyusul dilakukan untuk mengetahui apakah penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan.

Jika struktur kendali antar modul sudah terbukti bagus, maka pengujian yang tak kalah pentingnya adalah pengujian unit. Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan unit testing, yaitu :

1. White Box Testing

Uji coba *white box testing* merupakan metode perancangan *testcase* yang mengunakan struktural untuk mendapatkan *testcase*, test ini digunakan untuk meramal cara kerja perangkat lunak secara rinci kepada *logic path* ( jalur logika ), perangkat lunak di tes dengan kondisi dan perulangan secara fisik.

Contoh pengujian white box testing ini merupakan peringatan ketika user menginputkan password user yang salah, untuk kesalahan semacam ini akan memberikan suatu informasi kepada user mengenai kesalahan yang di lakukan.

1. Black Box

Berdasarkan hasil uji coba yang dilakukan, seluruh navigasi dan tombol fasilitas program lainnya serta proses yang di jalankan tidak terjadi kesalahan, tetapi aplikasi mempunyai aturan-aturan yang sudah di tetapkan dan harus di ikuti karena apabila di hiraukan maka sistem akan menolak perintah yang tidak sesuai seperti kesalahan ketika user belum menginput data yang harusnya di input sesuai ketentuan sistem yang di jalankan dan sistem memberikan informasi kepada user karena data yang ingin diproses belum lengkap atau tidak memenuhi ketentuan untuk proses selanjutnya.

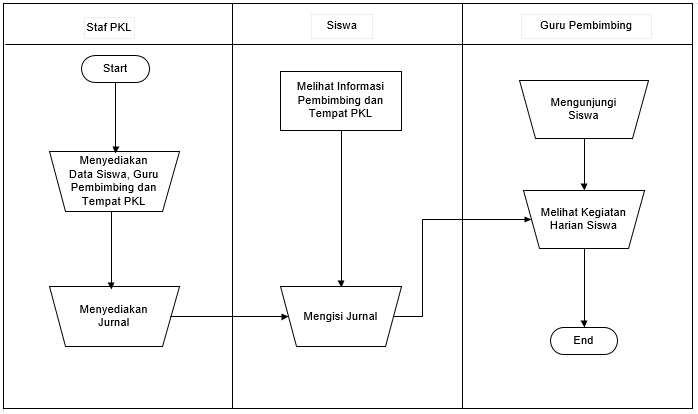
1. **Alat dan Bahan Penelitian**
2. *Hardware* (Perangkat Keras)

Perangkat keras yang digunakan adalah :

1. Laptop Asus AMD Radeon™ R2 Graphics
2. Prosesor AMD Quad Core E2-6110 1.5GHz
3. RAM 4 GB
4. Harddisk 500 GB
5. *Software* (Perangkat Lunak)

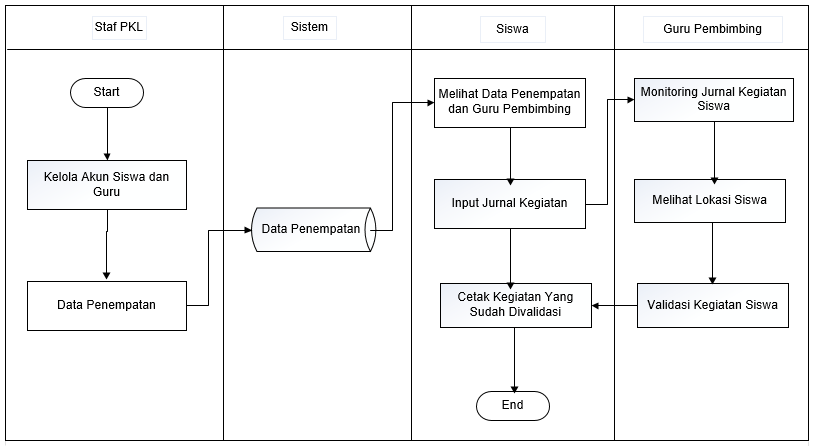
Adapun perangkat lunak yang digunakan adalah :

1. Sistem Operasi Windows .
2. Apache Cordova
3. Xampp
   * 1. **Desain Sistem**
4. **Rancangan data**

****

**Gambar. 3.1**

1. **Sistem Yang Diusulkan**

****

**Gambar 3.2**

**DAFTAR PUSTAKA**

King, Kevin F. 2009. *Geolocation and Federalism on the Internet: Cutting Internet Gambling’s Gordian Knot.Geolocation*, United States Court of Appeals for the Fourth Circuit.

Mercy. 2005. *Design, Monitoring and evaluation guidebook*.

*Wikipedia. 2019. Pelatihan Kerja Lapangan.*

*https://id.m.wikipedia.org/wiki/Pelatihan\_Kerja\_Lapangan* diakses 01 Oktober 2019.

Robi’in,Bambang. 2008. *Sistem Informasi Geografis Sumber Daya Alam Indonesia Berbasis Web.* Jurnal Informatika*.*Yogyakarta.

Budiutomo, Nanang. 2017. Simbol Flowchart beserta fungsinya. [*https://bukubiruku.com/simbol-flowchart-dan-fungainya/*](https://bukubiruku.com/simbol-flowchart-dan-fungainya/) diakses 01 Oktober 2019.

Juradin, dkk. 2018. *Sistem Informasi Monitoring Praktek Kerja Industri Berbasis Web*. Makassar: Universitas Negeri Makassar.

Susanto,Gito Aro,dkk. 2016 *Sistem Monitoring Praktek Kerja Lapangan (Pkl) Pada SMK Assa’idiyya.* Kudus: Universitas Muria Kudus.

Wikipedia. 2019. *JavaScript.* [*https://id.wikipedia.org/wiki/JavaScript*](https://id.wikipedia.org/wiki/JavaScript) diakses pada 30 september 2019.

Andre. 2019. *PHP*. *https://www.duniailkom.com/pengertian-dan-fungsi-php.*