

BAB II KAJIAN PUSTAKA

A. Studi Pustaka

Pada penelitian ini mencari sumber atau *referensi* terhadap beberapa penelitian yang berkaitan dengan penelitian ini sebagai acuan dalam Pengujian *Robustness Testing* dan *Equivalence Partitioning* pada aplikasi *Panic Button*. Berikut ini adalah beberapa penelitian yang telah dilakukan oleh peneliti sebelumnya antara lain:

Tabel 2 1 Penelitian Terkait

No.	Peneliti	Keterangan Penelitian
1	Alvin Valerian, Delon Bernady, Ferry Wahyudi, Jonathan Pratama Dinatha, One Dhylan B. (2026)	Judul : <i>Implementasi Unit Testing, Integration Testing, System Testing, Dan Validation Testing Pada Aplikasi Berbasis Website.</i> Metode : <i>Unit Testing, Integration Testing, System Testing, Validation Testing, Black Box Testing, White Box Testing.</i> Hasil Penelitian : Berdasarkan hasil pengujian, sistem telah memenuhi standar kualitas melalui <i>Unit Testing, Integration Testing, System Testing, dan Validation Testing</i> . <i>Unit Testing</i> memastikan fitur CRUD berfungsi tanpa kesalahan, sementara <i>Integration Testing</i> membuktikan interaksi antar modul berjalan baik. <i>System Testing</i> menguji stabilitas sistem, dan <i>Validation Testing</i> menunjukkan kemudahan penggunaan serta keakuratan data. Hasil pengujian mengonfirmasi bahwa sistem memiliki integritas data yang baik dan beroperasi optimal[12].
2	Hasna Sarifatin M., Safiq Rosad, Edy Sulistyanto (2024)	Judul : <i>Implementasi Tombol Panik SOS Untuk Lansia Berbasis IoT Pada Panti Pelayanan Sosial Lanjut Usia Dewanata Cilacap</i> Metode : Berbasis Internet of Things Software : <i>Panic Button</i> Hasil Penelitian : Perancangan Tombol Panik SOS

No.	Peneliti	Keterangan Penelitian
		untuk Lansia Berbasis <i>Internet of Things</i> (IoT) memerlukan pertimbangan beberapa faktor, termasuk kemudahan penggunaan, keamanan, dan integrasi dengan teknologi lainnya. Tombol Panik SOS untuk Lansia Berbasis IoT dapat membantu meningkatkan keselamatan dan kesejahteraan lansia. Realibilitas perangkat Tombol Panik SOS terhadap lingkungan yang ada sangat penting untuk memastikan bahwa perangkat tersebut dapat berfungsi dengan baik dan efektif dalam situasi darurat[6].
3	Aufa Ardilla, Firza Prima Aditiawan, Yisti Vita Via (2024)	Judul : Penerapan Teknik Equivalence Partitioning Dan Boundary Value Analysis Dalam Pengujian Black Box Aplikasi Kediri Single Window For Investment (Studi Kasus : Dinas Penanaman Modal Dan Pelayanan Terpadu Satu Pintu Kota Kediri). Metode : Equivalence Partitioning Dan Boundary Value Analysis Hasil Penelitian : Hasil dari pengujian diperoleh nilai akhir efektivitas sistem yang dihitung dengan menjumlahkan nilai efektivitas dari masing-masing tabel dan dibagi dengan seluruh jumlah tabel sehingga hasil akhir diperoleh nilai sebesar 95,31%. Berdasarkan hasil perhitungan tersebut termasuk dalam kategori nilai sangat efektif. Dapat disimpulkan bahwa aplikasi Kediri Single Window for Investment milik Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu Kota Kediri ini sudah berjalan sesuai fungsionalitas dan dapat dikatakan sudah sangat baik. Namun, di samping itu masih terdapat beberapa kesalahan dalam aplikasi sehingga diperlukan perbaikan agar aplikasi berfungsi lebih optimal lagi. Untuk penelitian selanjutnya, diharapkan dapat

No.	Peneliti	Keterangan Penelitian
		dilakukan pengujian menggunakan metode lain atau bisa dengan menggunakan tools khusus untuk melakukan pengujian. Dengan demikian, hasil penelitian dapat digunakan sebagai bahan pembandingan dan dapat saling melengkapi sehingga membantu DPMPTSP Kota Kediri dalam meningkatkan efektivitas aplikasi tersebut[10].
4	Farra Wardah Gracillaria Suharyono, Kartini, Achmad Junaidi Program (2024)	Judul : Penerapan Metode Boundary Value Analysis Dan Equivalence Partitioning Dalam Pengujian Black Box Untuk Aplikasi Siadita. Metode : Equivalence Partitioning Dan Boundary Value Analysis Hasil Penelitian : Hasil pengujian Black Box Testing menggunakan metode Boundary Value Analysis dan Equivalence Partitioning pada aplikasi SIADITA memberikan beberapa kesimpulan yang akan dijelaskan dalam bagian ini. Kesimpulan yang pertama yaitu, terdapat enam kelemahan sistem pada aplikasi SIADITA yang berpengaruh terhadap nilai efektivitas aplikasi di antaranya yaitu, pada halaman login tidak ada pemberitahuan/alert mengenai pengguna jika tidak sengaja tidak mengisi email, tidak ada sensitive case pada input ketika pengguna tidak sengaja menambahkan spasi dan karakter yang tidak sesuai seperti karakter huruf kecil dan besar, pada halaman input email tidak ada sistem validasi alamat email, halaman pusat bantuan tidak dapat melakukan autentikasi menuju aplikasi WhatsApp, tidak ada fungsi untuk membatasi input pada nama lengkap yang terdapat di halaman edit profile dan halaman pengisian survey, dan yang terakhir yaitu tidak ada fungsi yang membatasi input pada form kritik dan saran pada halaman kritik dan saran. Kesimpulan

No.	Peneliti	Keterangan Penelitian
		selanjutnya yaitu, aplikasi SIADITA memperoleh nilai efektivitas sebesar 91,66% dari total keseluruhan 144 test case dengan 126 test case hasil valid dan 18 test case hasil invalid yang mana nilai tersebut masuk ke dalam kategori sangat sangat efektif berdasarkan pada standart acuan Litbang Depdagri (1991), namun perlu adanya upaya perbaikan dan pengembangan lebih lanjut pada sistem untuk mencapai nilai efektivitas terbaik, sehingga fungsionalitasnya dapat beroperasi secara optimal[2].
5	Taufik Maulana, Apriade Voutama	Judul : Black Box Testing Dengan Teknik Equivalence Partitions Pada Website Dinas Pengendalian Penduduk Dan Keluarga Berencana Karawang Metode : Hasil Penelitian : kelebihan dan kelemahan Black Box Testing dengan Teknik Equivalence Partitions pada website kbinten.id. Kelebihannya yaitu Efisien, Dengan teknik Equivalence Partitions, pengujian dapat dilakukan dengan efisien dan efektif karena hanya perlu memilih beberapa data input yang mewakili setiap partisi. Kelemahannya yaitu Terbatas pada kondisi partisi yang diuji: Teknik Equivalence Partitions hanya menguji beberapa kondisi partisi yang mewakili input yang mungkin digunakan pengguna. Ini mungkin tidak cukup untuk menguji keseluruhan fungsionalitas perangkat lunak[13].
6	Syita Luginawati, Ari Purno Wahyu (2023)	Judul : Pengujian Perangkat Lunak Menggunakan Metode Regression Testing Pada Aplikasi Pembelajaran Matematika Untuk Siswa Tingkat Smp Berbasis Android Metode : <i>Black Box Testing</i> Hasil Penelitian : Dari hasil pengujian yang telah dilakukan pada aplikasi cerMath berbasis mobile android

No.	Peneliti	Keterangan Penelitian
		mendapatkan kesimpulan bahwa metode <i>black box</i> dengan teknik regression ini mempermudah tester dalam mengetahui kesesuaian alur fungsi dengan hasil yang diharapkan dan dapat melihat hasil pada setiap prosesnya dan sangat berpengaruh pada pembelajaran matematika berbasis android[14].
7	Andika Pratama, Aditya Gema Pratama, Hanifah Permatasari (2023)	Judul : Pengujian Perangkat Lunak Sistem Informasi Perpustakaan Menggunakan Metode <i>Black Box Testing</i> Metode : <i>Black Box Testing</i> Software : <i>Google Form</i> Hasil Penelitian : Pengujian perangkat lunak menggunakan Metode <i>Black Box Testing</i> dengan Teknik <i>Equivalence Partition</i> cocok bagi pemula karena lebih sederhana caranya. Hasil pengujian menunjukkan output beberapa form dari system informasi perpustakaan yang dibuat sudah tidak ada error, dan sudah sesuai dengan yang diharapkan, pada perancangan dengan validitas 100% [15].
8	I Dewa Made Widia, Sovia Rosalin, Salnan Ratih Asriningtias, Elta Sonalita (2022)	Judul : Black Box Testing Menggunakan Boundary Value Analysis dan Equivalence Partitioning pada Aplikasi Pengadaan Bahan Baku Batik dengan Pendekatan Use Case Metode : Black Box Testing Hasil Penelitian : Penggabungan metode Boundary Value Analysis dan Equivalence Partitioning dengan pendekatan use case untuk desain test case pada metode Black Box Testing sangat efisien dalam menemukan kesalahan aplikasi. Hal ini ditunjukkan nilai DRE 0.45 yang berarti 45% test case yang dibangun tidak lolos uji atau dapat menemukan kesalahan dari aplikasi. Saat ini pengujian masih dilakukan dari sisi fungsional aplikasi,

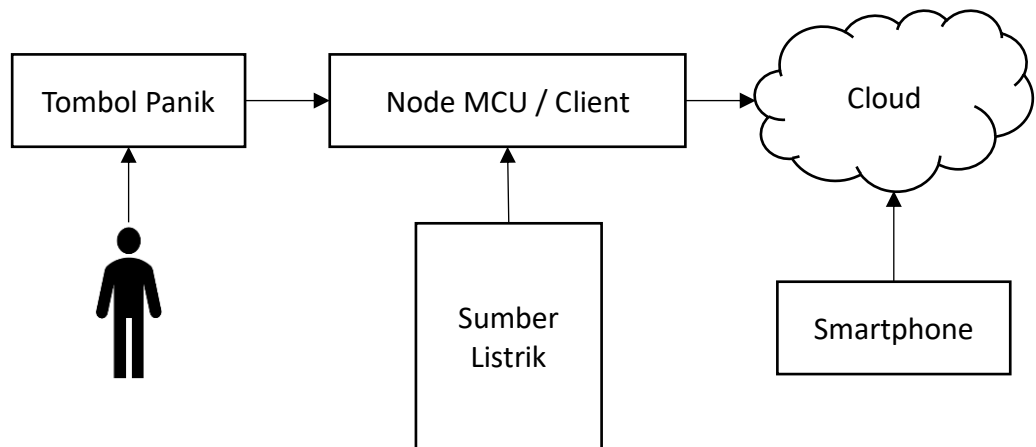
No.	Peneliti	Keterangan Penelitian
		maka untuk penelitian selanjutnya pengujian bisa dilakukan dari sisi non fungsional aplikasi. Selain itu bisa melakukan komparasi dengan metode Black Box Testing yang lainnya[16].
9	Nabil Fahlevi Abdi, S.R. Candra Nursari (2022)	Judul: Pengujian <i>Black Box</i> Pada Website Dengan Metode <i>Robustness Testing</i> Metode: <i>Robustness Testing</i> Hasil Penelitian : Berdasarkan hasil pengujian dengan menggunakan metode <i>Robustness Testing</i> pada website Eiger Adventure dapat disimpulkan dari 20 pengujian terdapat 19 pengujian yang berhasil dalam menangani kesalahan input atau gangguan pada sistem. Website ini dapat menangani kesalahan input yang tidak valid tersebut dan masih tetap berfungsi dengan baik, maka presentase keberhasilan pengujian ini adalah 95% [7].
10	Fadhila Tangguh Admojo1, M. Leo Adi Saputra (2022)	Judul : Analisis Sistem Keuangan Desa (SISKEUDes) di Kecamatan Muara Sugihan Menggunakan Metode <i>Black Box Testing</i> Metode : <i>Black Box Testing</i> Hasil Penelitian : Berdasarkan hasil pengujian black box dari 160 <i>test case</i> ditemukanya 24 defect, sehingga didapatkan nilai $DRE = \left(\frac{24}{160}\right) \times 100\% = 0,15 = 15\%$. Hal ini menunjukkan metode black bok dapat menemukan kesalahan atau defect pada aplikasi Siskeudes sebesar 15% [3].

B. Landasan Teori

1. Konsep dasar aplikasi *Panic Button SOS*

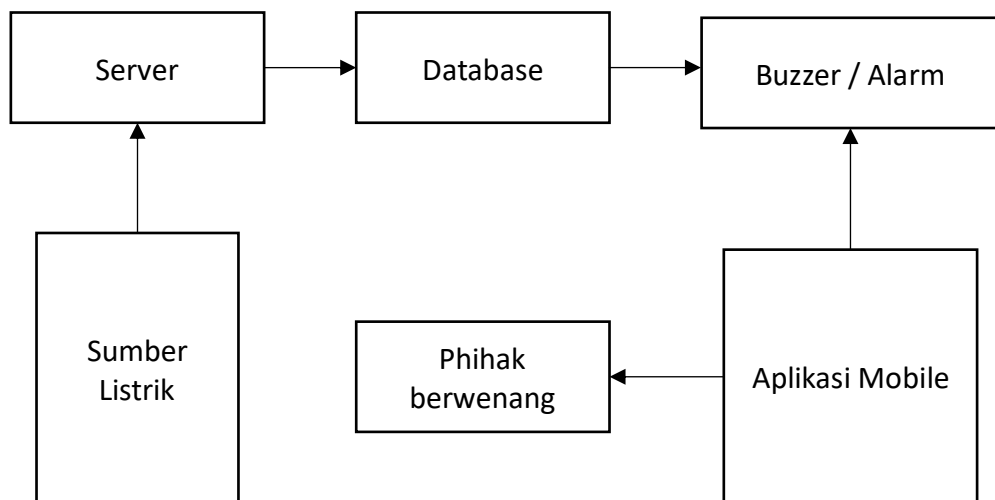
Panic Button SOS adalah sistem keamanan yang dirancang untuk memberikan tanggapan cepat dalam situasi darurat [17]. Fitur Panic Button SOS merepresentasikan lini pertahanan pertama dalam menghadapi situasi darurat. Dengan menekan tombol SOS, pengguna dapat langsung mengirimkan sinyal darurat ke pihak-pihak terkait yang telah ditentukan seperti admin atau pengguna lainnya [6]. Tujuan utamanya adalah untuk menjembatani waktu kritis antara insiden terjadi dan datangnya bantuan. Sistem ini bekerja dengan efisiensi, memungkinkan pengguna mengirim permintaan bantuan real-time tanpa perlu repot melakukan panggilan telepon.

Sistem keamanan pada jurnal [6] ini beroperasi secara end-to-end mulai dari perangkat keras hingga perangkat lunak, ketika pengguna mengaktifkan tombol panik, sinyal darurat segera ditangkap oleh unit mikrokontroler (Node MCU/Client) yang bertindak sebagai jembatan data. Selanjutnya, data tersebut diteruskan ke layanan Cloud untuk diproses dan diolah. Setelah diproses, Cloud bertanggung jawab mengirimkan notifikasi darurat secara real-time ke smartphone atau perangkat pihak berwenang atau pengguna yang telah ditentukan. Seluruh proses ini didukung oleh sistem baterai terpisah yang menyediakan daya kritis untuk tombol panik dan Node MCU, menjamin sistem tetap fungsional dan dapat diandalkan bahkan saat terjadi pemadaman daya utama. Secara keseluruhan, diagram ini memvisualisasikan alur sistem yang komprehensif, mulai dari inisiasi di perangkat keras hingga diterimanya peringatan darurat melalui infrastruktur cloud.



Gambar 2 1 Diagram Block Client

Pada bagian backend pada jurnal [6], Server berperan sentral dalam mengelola seluruh respons darurat. Setelah menerima sinyal dari client (Node MCU), Server bertindak cepat dengan melakukan tiga fungsi utama yaitu menyimpan data insiden secara aman ke dalam database, mengaktifkan buzzer atau alarm lokal sebagai peringatan dini, dan mengirimkan notifikasi push ke aplikasi mobile yang digunakan oleh admin, pihak berwenang atau tim tanggap darurat. Keberlanjutan operasional server ini sangat penting. Oleh karena itu, Powerbank (uninterruptible power supply) diintegrasikan untuk memastikan pasokan daya yang konstan, sehingga sistem tetap aktif dan responsif meskipun terjadi gangguan atau pemadaman listrik.



Gambar 2 2 Diagram Block Server

2. Pengujian Perangkat Lunak (*Software Testing*)

Pengujian Perangkat Lunak (*Software Testing*) adalah proses evaluasi sistematis yang krusial untuk memverifikasi dan memvalidasi apakah sebuah produk perangkat lunak berfungsi sesuai dengan persyaratan yang telah ditentukan, dan untuk mengidentifikasi adanya celah (*bug*) atau kekurangan [4]. Tujuannya melampaui sekadar menemukan kesalahan tetapi berfokus pada peningkatan kualitas, keandalan, dan kinerja aplikasi secara keseluruhan. *Software Testing* memastikan aplikasi mampu beroperasi dengan stabil dalam berbagai kondisi sehingga memenuhi ekspektasi pengguna dan mengurangi risiko kegagalan operasional saat digunakan [18].

Pengujian perangkat lunak dapat diklasifikasikan ke dalam beberapa dimensi untuk memberikan cakupan evaluasi yang komprehensif terhadap suatu system [12]. Secara umum, klasifikasi yang paling mendasar membaginya berdasarkan pengujian fungsional dan pengujian non-fungsional. Pertama, Pengujian Fungsional (*Functional Testing*) yang berfokus pada verifikasi apakah setiap fitur aplikasi bekerja sesuai dengan spesifikasi dan kebutuhan pengguna, Kedua, Pengujian Non-Fungsional (*Non-Functional Testing*) yang mengukur kualitas operasional sistem, termasuk aspek-aspek seperti kinerja, keamanan, kemudahan penggunaan, dan ketahanan (*robustness*). Selain itu, pengujian juga diklasifikasikan berdasarkan tahap implementasi (misalnya, *Unit Testing*, *Integration Testing*, *System Testing*, dan *Acceptance Testing*), memastikan kualitas aplikasi dievaluasi secara bertahap dari komponen terkecil hingga sistem secara keseluruhan [19].

3. Metode *Robustness Testing* (RT)

Robustness Testing adalah metode pengujian *black box* yang fundamental dan krusial yang berfokus pada evaluasi ketahanan dan stabilitas suatu sistem perangkat lunak ketika dihadapkan pada kondisi lingkungan yang tidak ideal atau masukan data yang tidak valid bahkan diluar batas kemampuan atau bahkan merusak (*malformed*) [7]. Tujuan utama pengujian ini adalah untuk memastikan bahwa aplikasi tidak *crash* atau gagal total melainkan mampu merespons *input* abnormal tersebut dengan cara yang anggun (*graceful*) dan

terkelola. Metode ini secara aktif mensimulasikan skenario terburuk seperti mencoba memasukkan karakter khusus yang tidak diharapkan ke dalam kolom angka atau menguji respons aplikasi saat koneksi jaringan terputus tiba-tiba, untuk mengidentifikasi titik kegagalan sistem [3].

Dalam konteks aplikasi *Panic Button*, *Robustness Testing* menjadi sangat penting karena sistem harus berfungsi dalam kondisi yang penuh tekanan dan tidak terduga. Pengujian ini berupaya memverifikasi fungsi penanganan kesalahan (*error handling*) sistem apakah aplikasi menampilkan pesan kesalahan yang jelas dan informatif, apakah aplikasi mampu kembali ke status normal setelah menerima *input* yang salah, dan apakah data sensitif (seperti lokasi) tetap terlindungi meskipun terjadi anomali *input*. Dengan memaksa aplikasi bekerja di luar batas spesifikasinya. Metode ini membantu menjamin bahwa sistem keamanan darurat tersebut akan tetap andal dan dapat diandalkan ketika pengguna paling membutuhkannya bahkan dalam situasi paling ekstrem.

4. Metode *Equivalence Partitioning* (EP)

Equivalence Partitioning (EP) adalah teknik desain kasus uji *black box* yang fundamental (segala sesuatu yang bersifat dasar atau pokok), dirancang untuk meningkatkan efisiensi dan cakupan pengujian dengan meminimalkan keadaan berlebih dari suatu hal (redundansi) [7]. Metode ini bekerja berdasarkan prinsip bahwa input data ke dalam sebuah aplikasi dapat dikelompokkan menjadi partisi atau kelas yang memiliki perilaku ekuivalen yang artinya jika satu kasus uji berhasil mendeteksi cacat dalam sebuah partisi, kemungkinan besar semua kasus uji lain dalam partisi yang sama juga akan menghasilkan hasil serupa. Oleh karena itu, *Equivalence Partitioning* membagi semua kondisi input yang mungkin menjadi dua kategori utama yaitu partisi valid (data yang diharapkan dan dapat diterima sistem) dan partisi tidak valid (data yang berada di luar batas atau format yang diharapkan) [2]. Prinsip kerja utama dari *Equivalence Partitioning* adalah mengasumsikan bahwa kondisi pemrosesan data oleh perangkat lunak akan identik untuk semua anggota yang berada dalam satu kelas masukan yang setara. Metode ini didasarkan pada premis statistik bahwa jika sebuah input yang diambil dari kelompok tertentu

(partisi) mengungkapkan adanya bug, maka semua input lain dari kelompok yang sama kemungkinan besar akan memicu bug yang sama. Sebaliknya, jika input tersebut berhasil, maka semua input lainnya juga akan berhasil [7].

Penerapan utama dari *Equivalence Partitioning* adalah mengurangi jumlah kasus uji secara drastis tanpa mengurangi efektivitasnya [13]. Sebagai contoh ketika menguji setiap kemungkinan input (misalnya, menguji seratus nomor telepon yang berbeda) penguji hanya perlu memilih satu atau dua nilai representatif dari setiap partisi yang telah ditentukan. Misalnya, untuk kolom yang menerima usia 18 hingga 65 tahun, partisi validnya adalah satu angka dalam rentang tersebut, dan partisi tidak validnya adalah angka di bawah 18 dan angka di atas 65. Dengan memfokuskan pengujian pada satu sampel dari setiap kelas *Equivalence Partitioning* memastikan bahwa pengujian menjadi lebih terorganisir, lebih cepat, dan lebih efektif dalam menemukan kesalahan yang berkaitan dengan pemrosesan data.

Selain contoh diatas penerapan *Equivalence Partitioning* dalam desain kasus uji dimulai dengan menganalisis setiap kolom input pada aplikasi untuk menetapkan kelas ekuivalen yang relevan [13]. Misalnya, pada aplikasi *Panic Button* jika kolom untuk nomor kontak darurat dispesifikasikan harus memiliki 10 hingga 13 digit maka penguji akan mendefinisikan tiga partisi utama yaitu satu partisi valid (misalnya, nomor 12 digit) dan dua partisi tidak valid (misalnya, nomor 9 digit sebagai *too short* dan nomor 14 digit sebagai *too long*). Kasus uji kemudian dirumuskan untuk mengambil satu nilai representatif dari masing-masing partisi tersebut. Pengujian ini bertujuan untuk menguji bahwa input valid diterima dan diproses, sementara input tidak valid ditolak dengan pesan kesalahan yang sesuai, sehingga memastikan validasi data pada antarmuka berfungsi efisien tanpa perlu menguji setiap kemungkinan kombinasi nomor telepon.