

BAB III

METODE PENELITIAN

Didalam penyusunan tugas akhir yang berjudul **perancangan sistem produksi dan pengemasan sambal pecel berbasis mikrokontroler** ini, penulis menggunakan metode kepustakaan dan studi literatur. Dengan menggunakan metode kepustakaan penulis berusaha untuk mendapatkan dan mengumpulkan data-data, informasi, konsep-konsep yang bersifat teoritis dari buku, materi kuliah, artikel dan *source code* di *internet*.

Sedangkan studi literatur penulis gunakan untuk mengumpulkan dan mempelajari serta melakukan wawancara-wawancara kepada pengusaha kecil yang berkecimpung dalam bidang pembuatan sambal pecel.

Hasil dari pengumpulan data-data tersebut, akan penulis gunakan sebagai bahan dan dasar untuk menyelesaikan permasalahan-permasalahan yang timbul dalam pembuatan tugas akhir ini.

Perancangan yang dilakukan dalam tugas akhir ini bersifat desain, yaitu perancangan yang mengarah pada pembuatan simulasi perangkat keras maupun perangkat lunak dengan tujuan nantinya dapat dikembangkan dan direalisasikan kedalam peralatan yang sesungguhnya.

Pada bab ini akan dibahas mengenai masalah-masalah yang berhubungan dengan perancangan dan pembuatan simulasi perangkat keras maupun perangkat lunak pada bagian proses produksi dan pengemasan sambal pecel. Dari kedua

bagian perangkat tersebut akan dipadukan agar keduanya dapat bekerja sama dalam menjalankan sistem dengan baik.

Perangkat yang terdapat pada alat ini dibagi menjadi dua, yaitu rangkaian elektronik (perangkat keras) dan perangkat lunak. Perancangan rangkaian elektronik terdiri dari rangkaian minimum sistem, dimana pada alat ini, penulis menggunakan mikrokontroler AT89S52 yang bertindak sebagai pengatur dan penggerak dari perangkat yang terdapat pada alat ini. Pada pembuatan alat ini, penulis menggunakan dua buah rangkaian mikrokontroler. Rangkaian mikrokontroler yang pertama berfungsi sebagai pengontrol *inputan* yang berupa (sensor) dan mengontrol *outputan* yang berupa motor dari semua proses yang terjadi pada proses produksi sambal pecel. Sedangkan untuk rangkaian mikrokontroler yang kedua digunakan sebagai pengontrol *input* dan *output* pada proses pengemasan sambal pecel.

3.1. Perancangan Perangkat Keras

Perancangan sistem dilakukan terlebih dahulu sebelum proses pembuatan sistem, hal ini dilakukan dengan tujuan agar pekerjaan selanjutnya lebih mudah dan terarah. Perancangan perangkat keras ini meliputi kontroler dan mekanik.

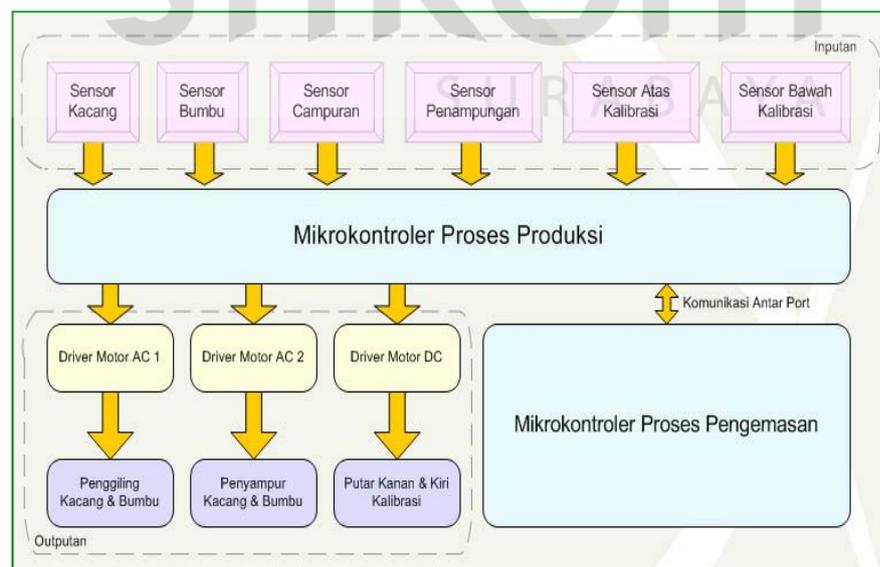
3.1.1. Perancangan Kontroler

Sesuai dengan kebutuhan dalam perancangan alat ini, penulis membutuhkan dua buah mikrokontroler yang diimplementasikan pada masing-

masing bagian untuk melakukan kontrol terhadap semua proses yang terjadi pada mesin pengolahan dan pengemasan sambal pecel.

A. Rangkaian Sistem Mikrokontroler Pertama

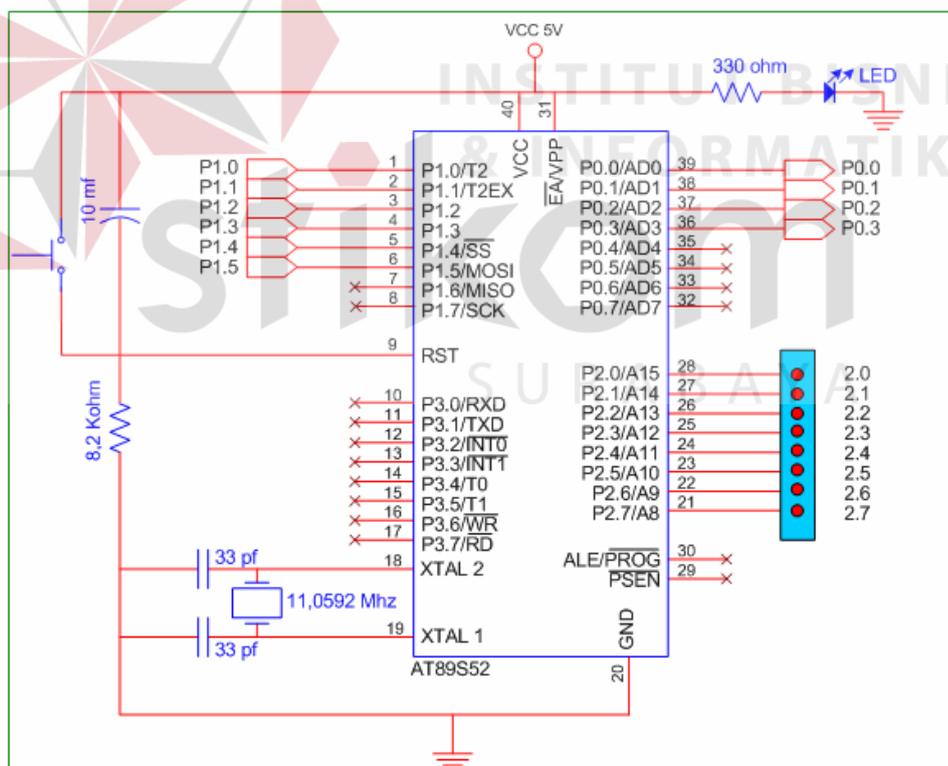
Rangkaian sistem mikrokontroler pertama berfungsi sebagai pengatur dan pengendali semua proses yang terjadi pada mesin produksi, antara lain pembacaan sensor yang terpasang pada masing-masing bagian dengan fungsi yang telah ditentukan (sensor kacang, sensor bumbu, sensor penyampuran, sensor penampungan dan sensor kalibrasi) selain pembacaan sensor mikrokontroler bertugas untuk mengendalikan mesin penggerak yang terdapat pada mesin penggiling kacang dan bumbu, mesin penyampur kacang dan bumbu serta pada mesin kalibrasi. Gambar 3.1 merupakan blok diagram dari rancangan mikrokontroler pertama.



Gambar 3.1. Blok Diagram Mikrokontroler Pertama

Perancangan *hardware* untuk realisasi tugas akhir ini dengan modul utama berupa minimum sistem AT89S52. Pada desain *board* yang telah dibuat diberikan konektor *port* sebagai *input* dan *output* untuk sistem yang telah dirancang. *Port* 0 pada mikrokontroler digunakan sebagai *port output* untuk menghidupkan motor AC dan motor DC, *port* 1 pada mikrokontroler digunakan sebagai *port inputan* untuk sensor (detektor) sedangkan *port* 2 difungsikan sebagai *port* komunikasi secara dua arah dengan mikrokontroler kedua.

Rangkaian skematik dari mikrokontroler pertama seperti yang ditunjukkan pada Gambar 3.2. berikut.



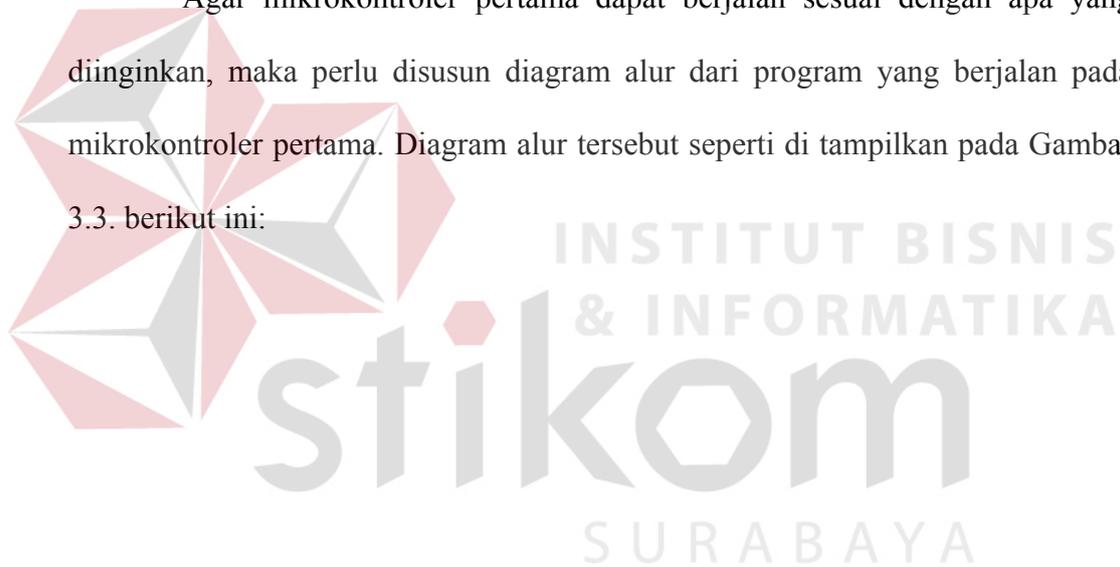
Gambar 3.2. Rangkaian Minimum Sistem Mikrokontroler Pertama

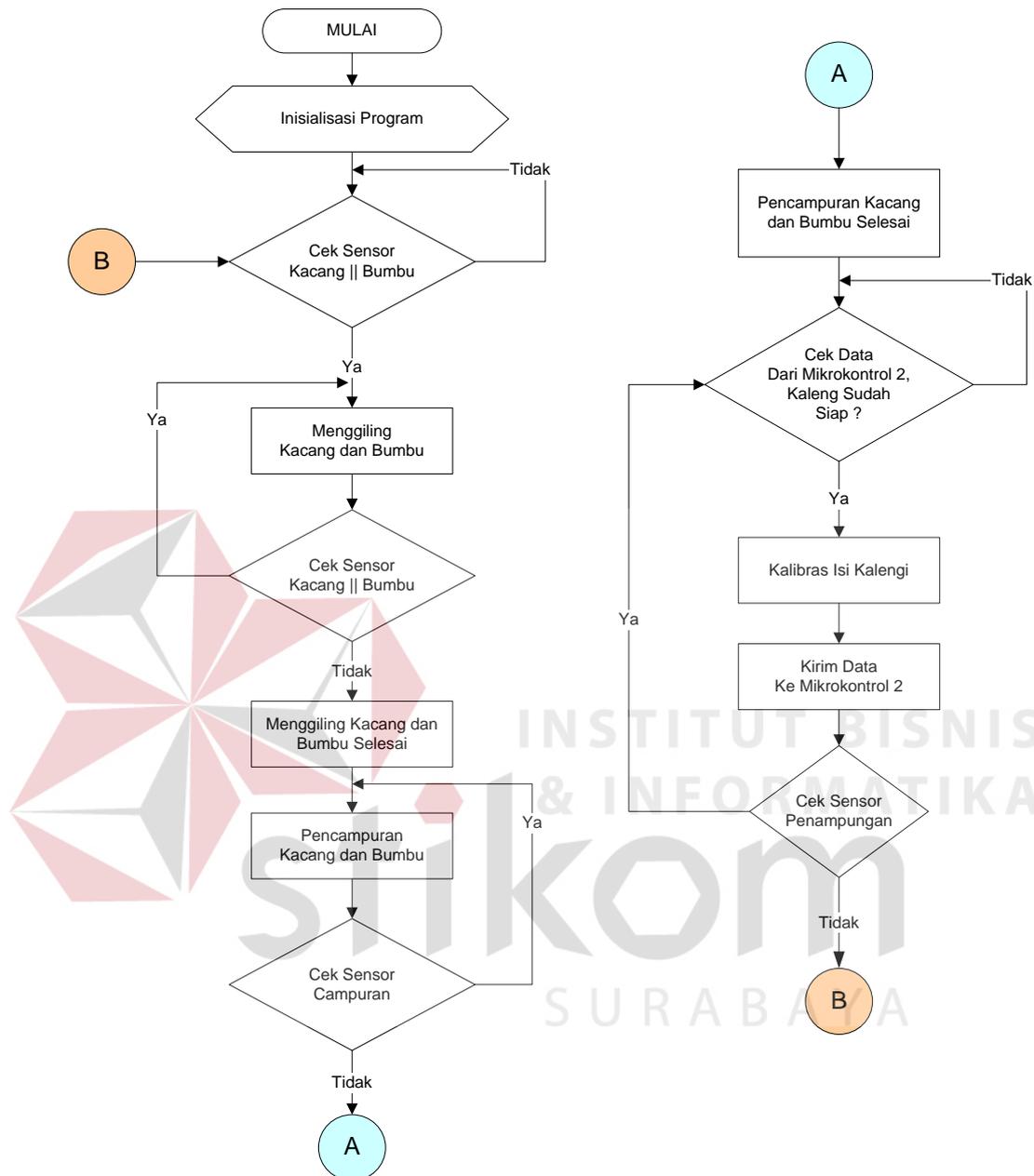
Pengaturan dan fungsi dari masing-masing *pin* pada mikrokontroler pertama adalah seperti yang terlihat pada Tabel 3.1. berikut.

Tabel. 3.1. Pengaturan *Pin* Pada Mikrokontrler Pertama

Nomor <i>Pin</i>	Fungsi	Keterangan
P1.0	Photodiode	Sensor Kacang
P1.1	Photodiode	Sensor Bumbu
P1.2	Photodiode	Sensor Campuran Kacang dan Bumbu
P1.3	Photodiode	Sensor Sambal pada Penampungan
P1.4	<i>Limit switch</i>	Sensor Batas Atas Kalibrasi
P1.5	<i>Limit switch</i>	Sensor Batas Bawah Kalibrasi
P0.0	Motor AC	Penggerak Gilingan Kacang dan Bumbu
P0.1	Motor AC	Penggerak Campuran Kacang dan Bumbu
P0.2-P0.3	Motor DC	Penggerak Naik Turun Kalibrasi
P2.0-P2.7	Koneksi	Komunikasi dengan Mikrokontroler Kedua

Agar mikrokontroler pertama dapat berjalan sesuai dengan apa yang diinginkan, maka perlu disusun diagram alur dari program yang berjalan pada mikrokontroler pertama. Diagram alur tersebut seperti di ditampilkan pada Gambar 3.3. berikut ini:

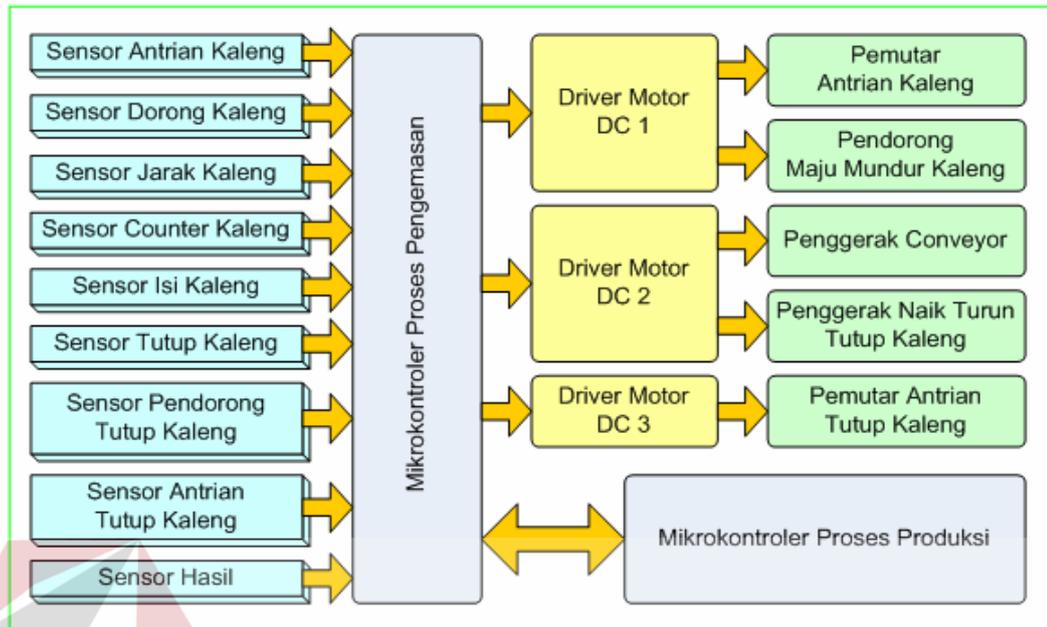




Gambar 3.3. Diagram Alur Mikrokontroler Pertama

B. Rangkaian Sistem Mikrokontroler Kedua

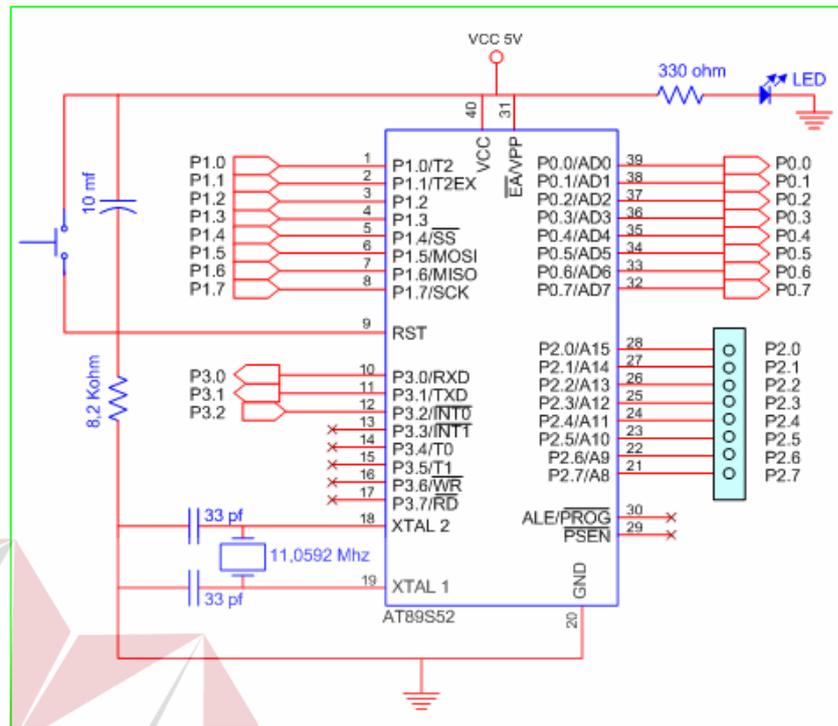
Mikrokontroler kedua ini dibuat untuk melakukan kontrol dan kendali pada mesin pengemasan. Adapun blok diagram dari rancangan mikrokontroler kedua dapat dilihat pada Gambar 3.4. berikut ini:



Gambar 3.4. Blok Diagram Mikrokontroler Kedua

Seperti halnya dengan mikrokontroler pertama, mikrokontroler kedua ini terdiri dari *input* dan *output*. *Port 0* dan *Port 3* pada mikrokontroler digunakan sebagai *port output* untuk mengontrol motor DC, *port 1* digunakan sebagai *port input* untuk sensor (*detektor*) dan *port 2* digunakan sebagai *port* komunikasi dengan mikrokontroler pertama.

Untuk lebih jelasnya dapat dilihat pada Gambar 3.5. skematik berikut ini.



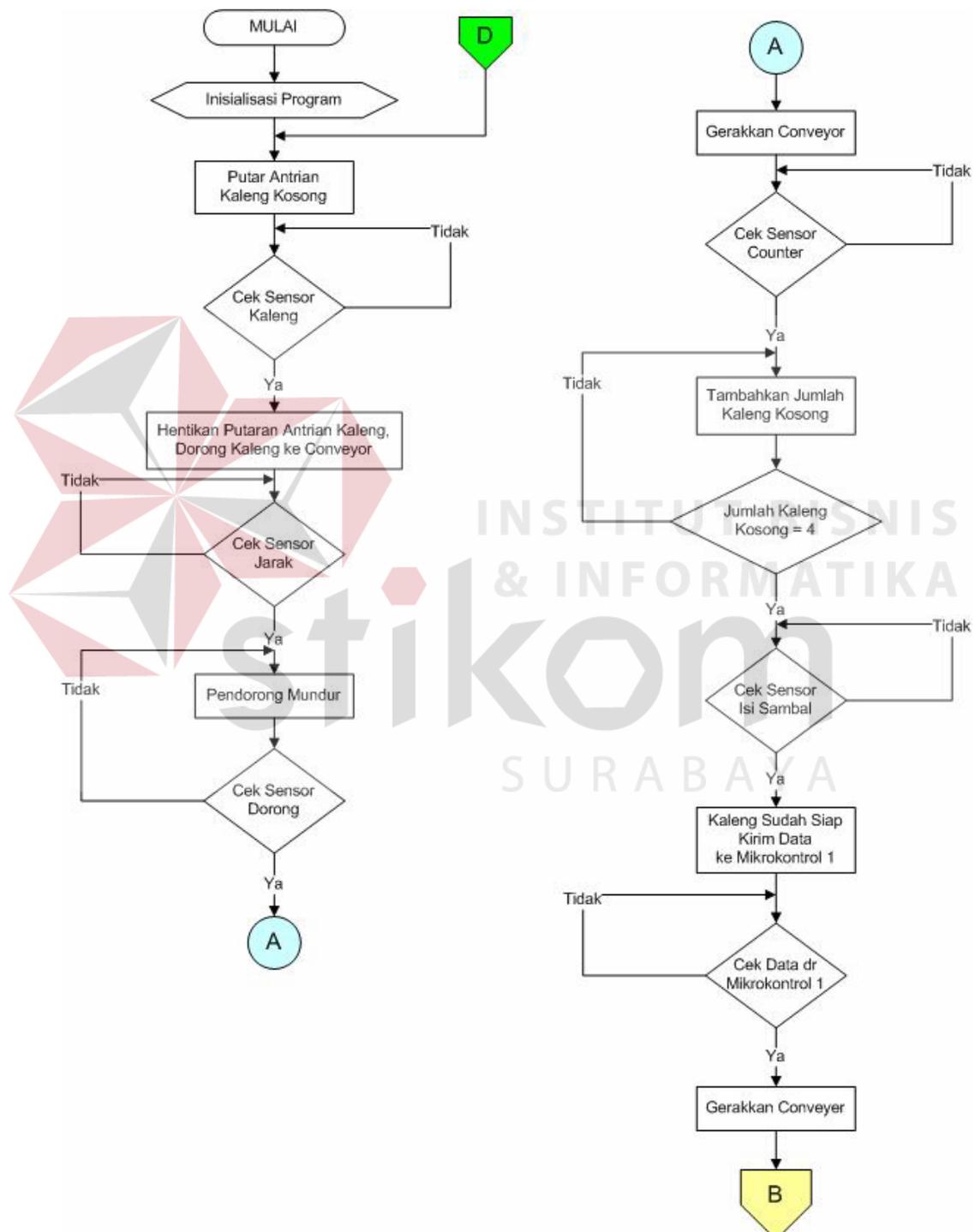
Gambar 3.5. Rangkaian Minimum Sistem Mikrokontroler Kedua

Pengaturan dan fungsi dari masing-masing *pin* pada mikrokontroler kedua dapat dilihat pada Tabel 3.2. berikut.

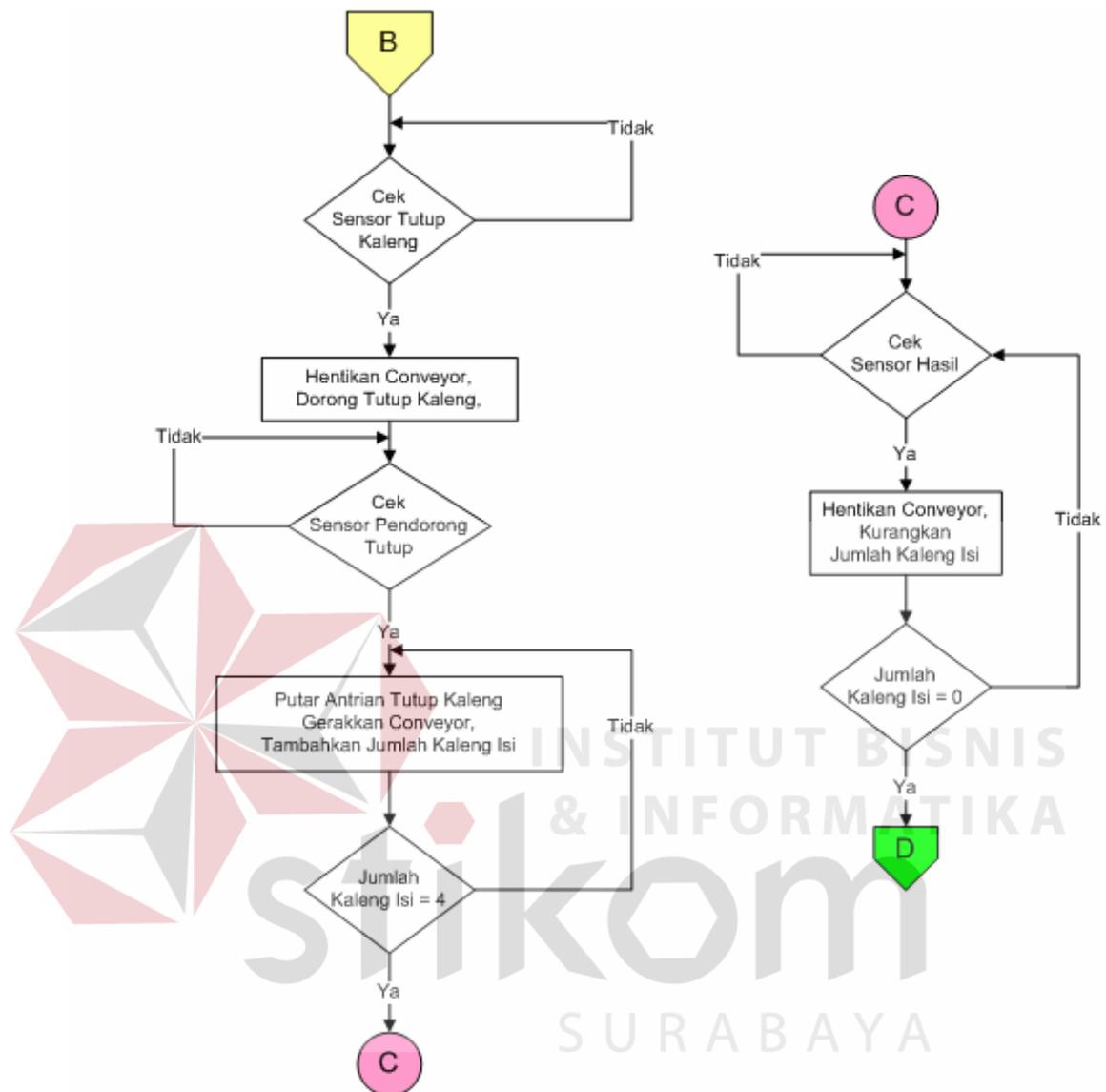
Tabel 3.2 Pengaturan *pin* Mikrokontroler Kedua

Nomor <i>Pin</i>	Fungsi	Keterangan
P1.0	Photodiode	Sensor Antrian Kaleng
P1.1	Photodiode	Sensor Jarak Antar Kaleng
P1.2	Photodiode	Sensor Counter Kaleng
P1.3	Photodiode	Sensor Isi Kaleng
P1.4	<i>Limit switch</i>	Sensor Tutup Kaleng
P1.5	<i>Limit switch</i>	Sensor Pendorong Tutup Kaleng
P1.6	<i>Limit switch</i>	Sensor Antrian Tutup Kaleng
P1.7	<i>Limit switch</i>	Sensor Hasil
P3.2	<i>Limit switch</i>	Sensor Dorong Kaleng
P0.0-P0.1	Motor DC	Pemutar Antrian Kaleng Kosong
P0.2-P0.3	Motor DC	Pendorong Maju Mundur Kaleng Kosong
P0.4-P0.5	Motor DC	Penggerak <i>Conveyor</i>
P0.6-P0.7	Motor DC	Penggerak Naik Turun Tutup Kaleng
P3.0-P3.1	Motor DC	Pemutar Antrian Tutup Kaleng
P2.0-P2.7	Koneksi	Komunikasi dengan Mikrokontroler Pertama

Adapun cara kerja dari mikrokontroler kedua, dapat dilihat pada Gambar 3.6 (a) dan 3.6 (b) diagram alur berikut ini:



Gambar 3.6. (a) Alur Diagram Mikrokontroler Kedua



Gambar 3.6. (b) Alur Diagram Mikrokontroler Kedua

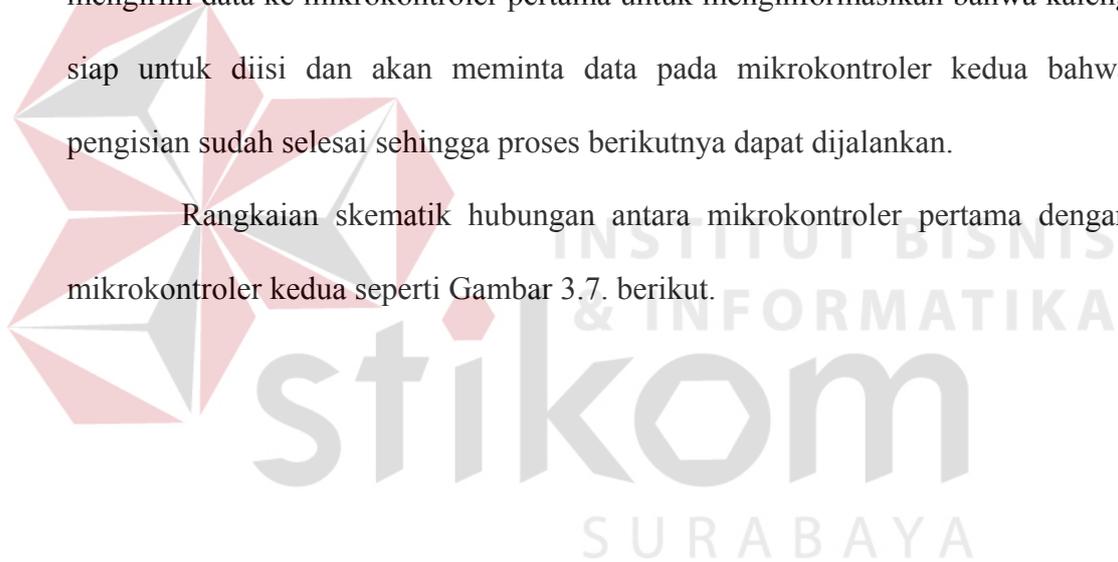
C. Komunikasi Antar Port Mikrokontroler Pertama dan Kedua

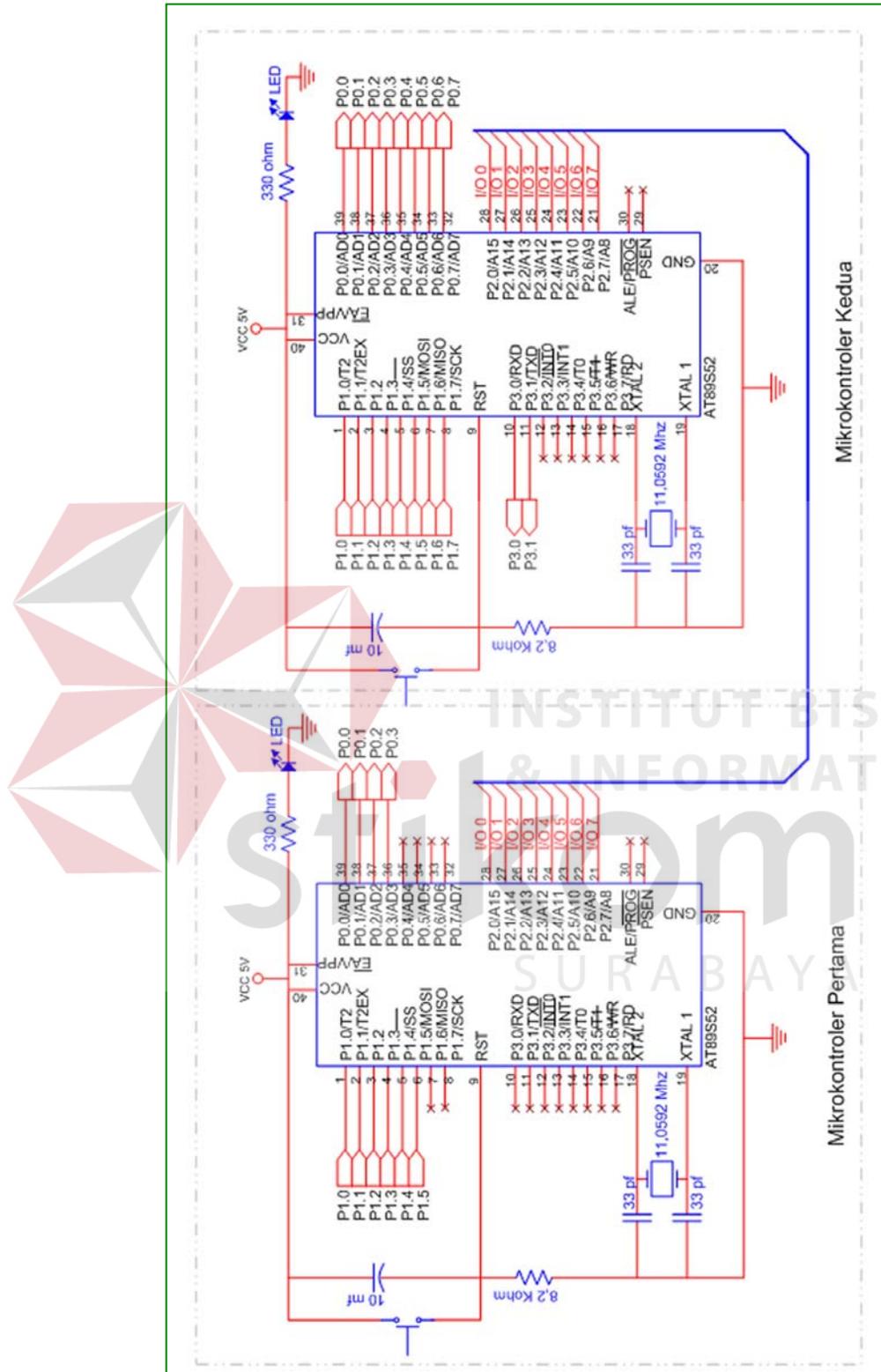
Proses produksi dan pengemasan sambal pecel ini akan dapat berjalan bila mikrokontroler pertama dan mikrokontroler kedua bisa berhubungan dengan baik.

Komunikasi antar *port* dilakukan secara dua arah dimana pertama-tama *port 2* pada mikrokontroler pertama akan berfungsi sebagai *inputan* untuk

menerima data dari mikrokontroler kedua yang menginformasikan bahwa kaleng kosong sudah berada pada posisinya dan siap untuk dilakukan proses pengisian sambal pecel. Setelah mikrokontroler pertama melakukan pengisian maka proses selanjutnya adalah *port 2* pada mikrokontroler pertama akan berfungsi sebagai *port output* yang akan mengirim informasi ke mikrokontroler kedua bahwa proses pengisian telah selesai. Begitupun juga dengan mikrokontroler kedua, *port 2* pada mikrokontroler kedua berfungsi sebagai *port output* dan *input*, dimana akan selalu mengirim data ke mikrokontroler pertama untuk menginformasikan bahwa kaleng siap untuk diisi dan akan meminta data pada mikrokontroler kedua bahwa pengisian sudah selesai sehingga proses berikutnya dapat dijalankan.

Rangkaian skematik hubungan antara mikrokontroler pertama dengan mikrokontroler kedua seperti Gambar 3.7. berikut.





Gambar 3.7. Skematik Komunikasi Antar Port Mikrokontroler Pertama dengan Mikrokontroler Kedua.

D. Rangkaian Driver Motor

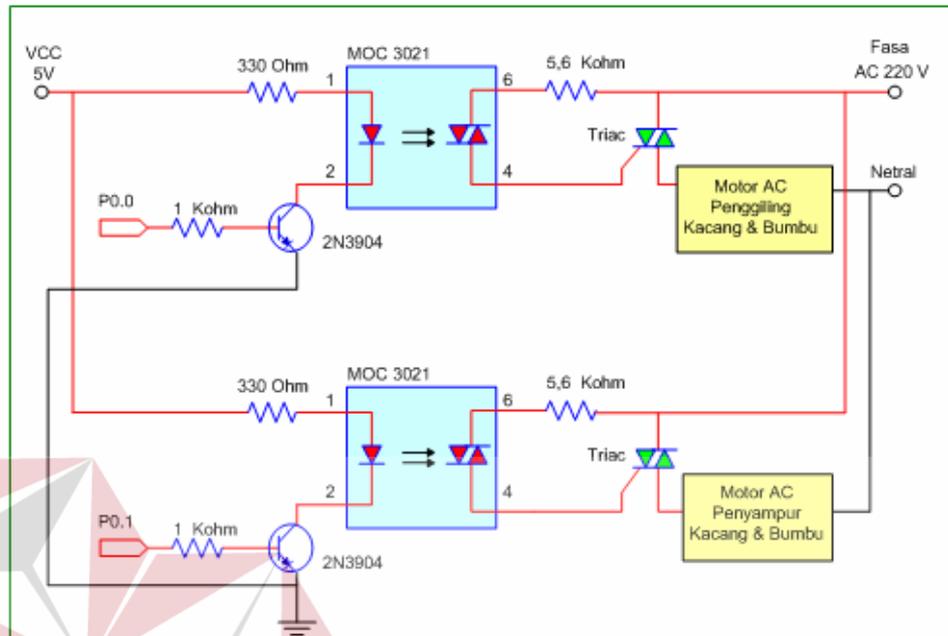
Rangkaian driver motor dirancang dengan tujuan agar dapat mengendalikan sejumlah motor penggerak, dalam perancangan ini dibutuhkan dua buah model driver motor untuk mengendalikan motor AC dan motor DC.

D.1. Rangkaian Driver Motor AC

Rangkaian driver motor AC digunakan untuk mengendalikan mesin penggiling kacang dan bumbu serta mesin penyampur kacang dan bumbu. Oleh karena itu dua buah IC MOC3021 (dimana setiap IC hanya bisa menangani satu buah motor AC) digunakan sebagai kontrol dari putaran motor satu arah tersebut.

Faktor terpenting yang harus diperhatikan pada *interface* I/O terutama untuk beban yang menggunakan tegangan AC 220 Volt adalah masalah sistem *isolator* dimana sistem digital menggunakan level tegangan +5 volt DC sedangkan beban menggunakan tegangan AC 220 Volt. Perbedaan tegangan ini sudah cukup untuk menyebabkan sistem kontrol digital, seperti halnya mikrokontroler untuk rusak jika *port* pada mikrokontroler ini menerima tegangan imbas dari beban 220 Volt tersebut, oleh sebab itulah rangkaian yang dapat menangani semua masalah tersebut adalah salah satunya dengan menggunakan IC MOC 3021.

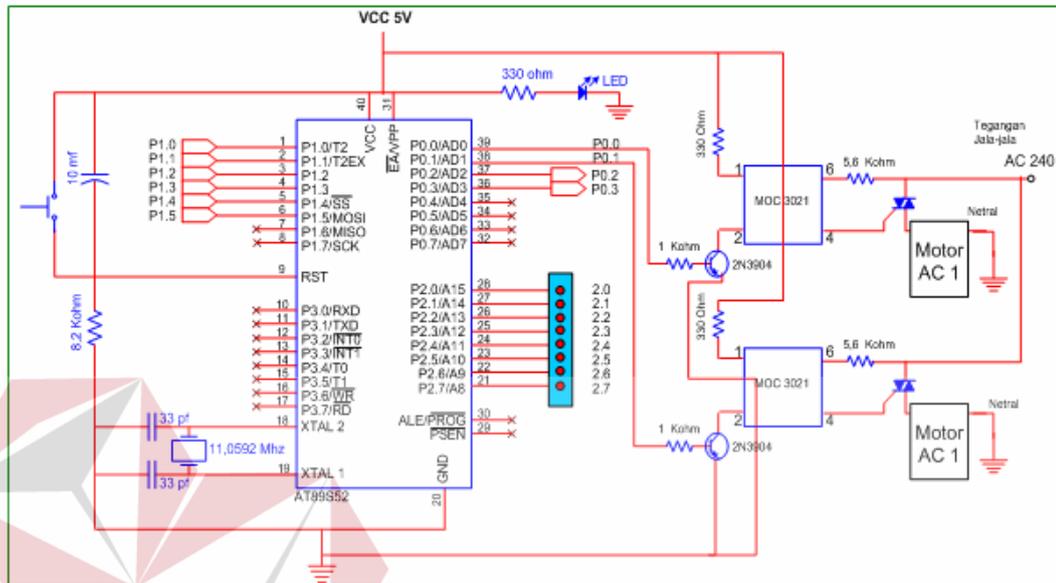
Bagian saklar pengatur daya dibentuk dengan kombinasi rangkaian R, ISO1 MOC3021, TRIAC dan Motor AC sebagai beban. Bagian ini langsung berhubungan dengan sumber tegangan jala-jala listrik 220 Volt, agar tegangan jala-jala terpisah dari bagian lainnya, dipakai Opto Isolator MOC3021 untuk menghubungkan AT89S52 dengan TRIAC. Rangkaian driver motor AC dilihat pada Gambar 3.8. berikut.



Gambar 3.8. Rangkaian Driver Motor AC

Gambar diatas, menunjukkan tentang rangkaian driver motor AC triac MOC 3021 yang merupakan rangkaian opto isolator dari dua buah rangkaian DC dan AC. Pada MOC 3021 *pin* 1 dan 2 merupakan rangkaian untuk menghidupkan led. Led pada rangkaian opto isolator tersebut akan menyala bila basis (*input*) yang terhubung pada *pin* P0.0 dan *pin* P0.1 pada mikrokontroler diberi arus lebih dari 0,7A. Resistor 1 K Ω berfungsi sebagai penahan arus yang besar sebelum masuk ke basis transistor 2N3904. Jika basis dari transistor tersebut sudah terbias maka led akan menyala dan akan memberi cahaya pada rangkaian opto DIAC pada *pin* 6 dan 4 MOC3021, sehingga DIAC akan aktif. Aktifnya DIAC akan mengaktifkan *gate* pada TRIAC, jika TRIAC aktif maka tegangan AC 220V (Tegangan Jala-Jala PLN) akan mengaktifkan *Motor AC* penggiling dan penyampur kacang dan bumbu.

Skemantik hubungan mikrokontroler, driver motor dan Motor AC dapat dilihat pada Gambar 3.9. berikut.



Gambar 3.9. Skematik Hubungan Mikrokontroler, Driver Motor dan Motor AC

D.2. Rangkaian Driver Motor DC

Cara kerja dari motor DC hampir sama dengan motor AC, kelebihan dari motor DC adalah gerakan putaran dari motor DC dapat diatur secara dua arah (searah atau melawan arah jarum jam) sehingga motor DC sangat cocok digunakan untuk menggerakkan sistem kendali dua arah (seperti alat pendorong kaleng ke *conveyor* dan lain sebagainya).

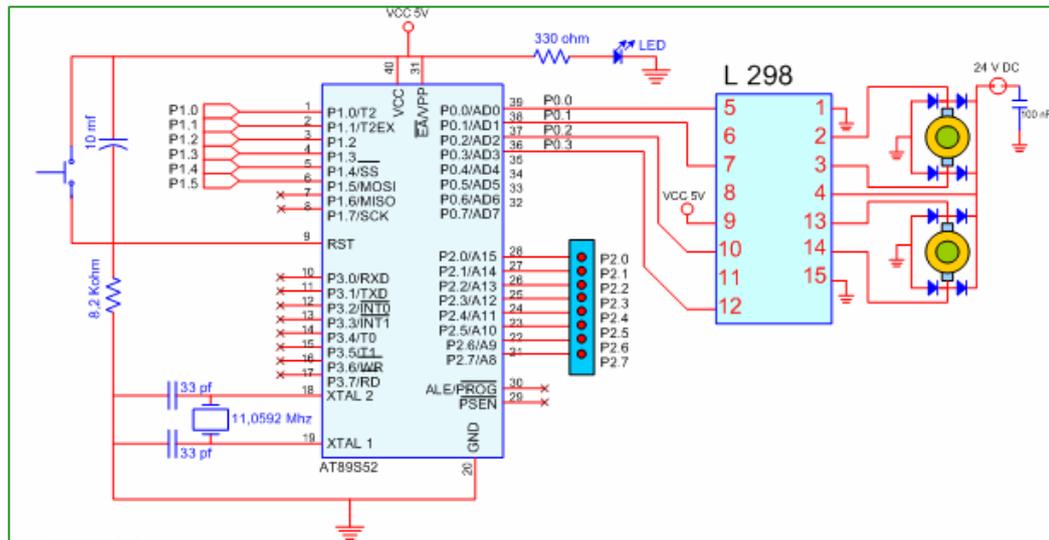
Motor DC dalam pembuatan alat ini difungsikan sebagai penggerak (*aktuator*) untuk menjalankan mekanik dari proses produksi dan pengemasan dimana dalam sistem ini memerlukan enam buah motor DC yang masing-masing digunakan untuk menggerakkan sistem kalibrasi, memutar antrian kaleng,

pendorong kaleng ke *conveyor*, penggerak *conveyor*, pendorong tutup kaleng dan pemutar antrian tutup kaleng.

Agar motor DC dapat di atur dan dikontrol maka diperlukanlah sebuah rangkaian driver. L298 merupakan salah satu IC driver motor yang menggunakan sistem *H-Bridge* sehingga kecepatan dari motor DC dapat diatur dengan memakai sistem *PWM (Pulse Width Modulation)*. Selain itu L298 dapat berfungsi sebagai relay otomatis untuk mengatur komunikasi dan menjaga terjadinya hubungan singkat yang mungkin terjadi karena perbedaan tegangan antara mikrokontroler dengan tegangan *penyuplay* motor DC (misal bila motor DC yang digunakan membutuhkan tegangan 46 Volt berdasarkan daya konsumsi motor, sedangkan tegangan yang dibutuhkan pada *port* mikrokontroler adalah 5 Volt), maka perlu adanya pengaturan tegangan antara mikrokontroler dengan motor DC.

Satu buah IC L298 dapat digunakan untuk mengontrol dua buah motor DC sehingga dalam perancangan alat ini dibutuhkan empat buah IC L298 sebagai rangkaian driver motor. Dari pengendalian motor tersebut, dapat diasumsikan bahwa pengendalian suatu motor DC dapat terrealisasikan.

Rangkaian skematik driver motor DC L298 dapat dilihat seperti Gambar 3.10. berikut:



Gambar 3.10. Skematik Driver Motor DC

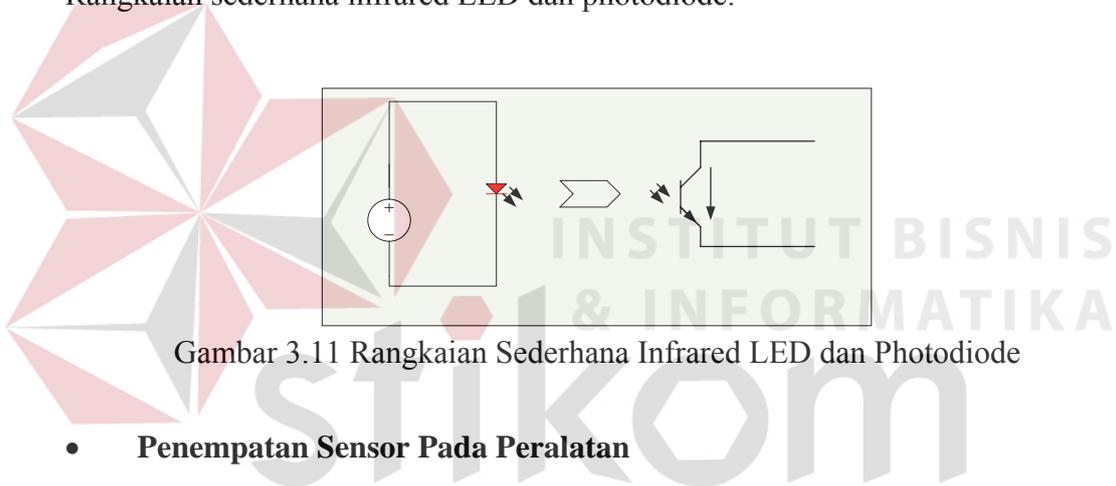
E. Rangkaian Sensor

Sensor merupakan alat *inputan* mikrokontroler yang digunakan untuk melakukan monitoring terhadap kejadian yang ada. Dalam perancangan alat ini digunakan dua buah model sensor yaitu jenis sensor photodiode dan sensor *limit switch*.

E.1. Rangkaian Sensor Photodiode

Rangkaian sensor *photo diode* ini difungsikan sebagai *proximity sensor* (sensor yang digunakan untuk mendeteksi benda tanpa harus menyentuh bagian fisik dari benda yang dideteksi), yaitu untuk mendeteksi ada benda atau tidak ada benda, misalnya pada proses pengisian sambal pecel kedalam kaleng yang terjadi pada proses pengemasan, kaleng yang berada diatas *conveyor* harus terdeteksi oleh sensor photodiode (sensor_isi kaleng) terlebih dahulu baru kemudian proses pengisian kedalam kaleng dapat dilakukan.

Komponen utama dari sensor photodiode ini adalah infrared LED yang berfungsi sebagai pemancar (*transmitter*) dan photodiode yang berfungsi sebagai penerima (*receiver*). Cara kerja dari sensor ini adalah ketika infrared LED diberi tegangan 5 V maka infrared akan memancarkan sinar infra merah, *photodeode* yang mempunyai resistansi (R) tinggi akan mengecil ketika terkena sinar infra merah sehingga resistansi (R) photodiode akan berubah-ubah ketika infra merah yang dipancarkan oleh infrared LED dihidupkan atau dimatikan. Gambar 3.11. Rangkaian sederhana infrared LED dan photodiode.



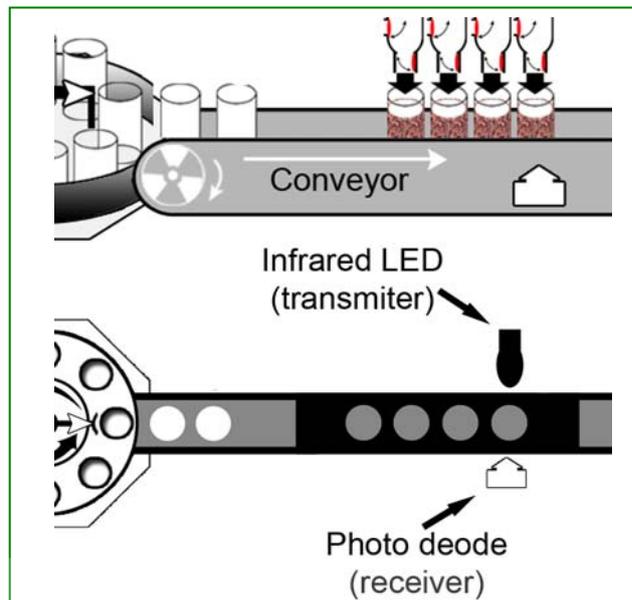
Gambar 3.11 Rangkaian Sederhana Infrared LED dan Photodiode

- **Penempatan Sensor Pada Peralatan**

Cara penempatan infrared LED dan photodiode ini sedikit banyak akan mempengaruhi kepekaan dari hasil sensor itu sendiri, namun itu semua tergantung pada tempat dan ruang serta benda yang akan dideteksi.

Dalam perancangan alat ini infrared LED ditempatkan secara berhadapan dengan photodiode, hal ini ditujukan untuk memberikan hasil yang maksimal terhadap perubahan yang terjadi.

Contoh model penempatan sensor photodiode dapat dilihat pada Gambar 3.12. berikut.



Gambar 3.12. Tampak Samping dan Atas Penempatan Sensor

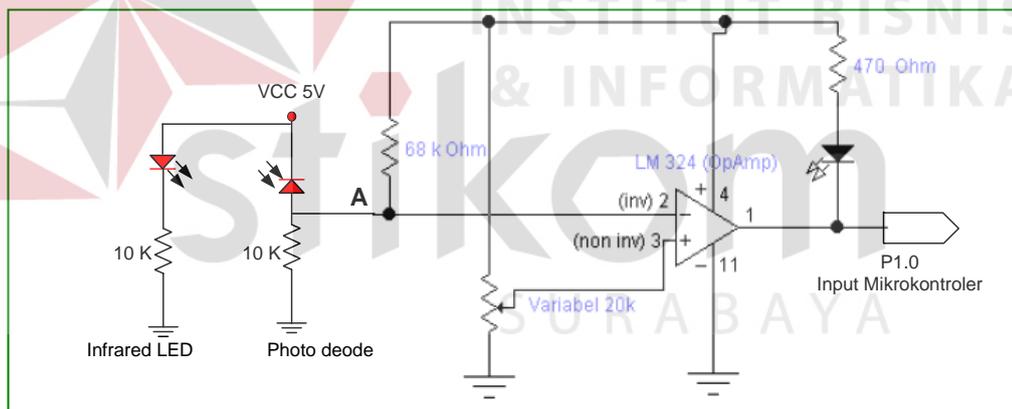
Namun ada juga sensor photodiode dalam perancangan alat ini yang didesain antara infrared LED dan photodiode diletakkan secara berdampingan (seperti yang terdapat pada sensor pendorong kaleng kosong ke *conveyor*), cara kerjanya adalah photodiode akan mengambil perubahan sinar yang dipantulkan oleh benda yang disinari oleh infrared LED, cara penempatan ini hanya dilakukan apabila infrared LED tidak memungkinkan untuk diletakkan secara sejajar (berhadapan) dengan photodiode dikarenakan tempat dan cara kerja dari benda yang dideteksi.

- **Comparator OpAmp (*Operational Amplifier*)**

Pada rangkaian komporator ini, IC yang digunakan adalah IC tipe LM 324 dimana setiap satu IC LM 324 mampu menangani empat buah sensor photodiode, sensor photode yang dibutuhkan dalam prancangan alat ini sebanyak delapan buah dimana empat buah sensor berada di mikrokontroler pertama dan empat buah yang lain berada di mikrokontroler

kedua, sehingga komparator yang dibutuhkan sebanyak dua buah. Alasan pemakaian komparator disebabkan jumlah kondisi yang diterima mikrokontroler hanya dua, yaitu aktif dan tidak aktif.

Bagian ini adalah rangkaian pembanding, menggunakan sebuah OpAmp yang mempunyai dua masukan yaitu *inverting* dan *non inverting*. Prinsipnya adalah membandingkan antara masukan *inverting* dan *non inverting*, jika kedua masukan bernilai (bertegangan) sama, maka *output* OpAmp akan bernilai nol dan begitu sebaliknya. Sebuah masukan dijadikan patokan dan diberi variabel tegangan (untuk menentukan kepekaan daya tangkap) dalam hal ini, masukan *non inverting* lah yang menjadi patokan.



Gambar 3.13. Rangkaian Comparator OpAmp menggunakan IC LM 324

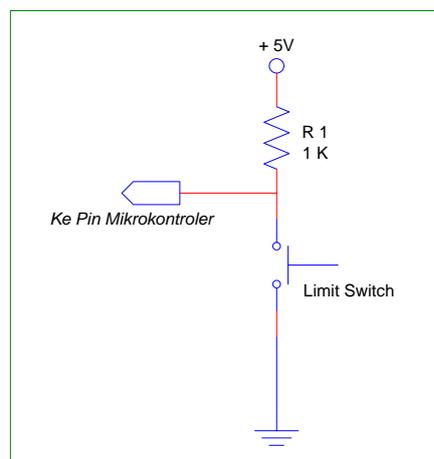
Cara kerja dari rangkaian ini adalah ketika photodiode tidak terkena cahaya, kita set R vareabel 20K hingga masukan *inverting* dan *non inverting* pada OpAmp sama, ditandai dengan LED menyala (bernilai 0). Pada saat photodiode terkena cahaya, maka resistansinya akan mengecil sehingga terjadi drop tegangan pada titik A, dropnya tegangan tersebut akan mengubah masukan *inverting* OpAmp sehingga menyebabkan *output*

OpAmp bertegangan tidak nol lagi. *Output* dari OpAmp inilah yang nantinya akan digunakan sebagai masukan mikrokontroler sebagai alat pendeteksi adanya benda.

E.2. Rangkaian Sensor *Limit switch*

Limit switch atau yang sering kita kenal dengan nama saklar adalah sebuah perangkat yang digunakan untuk memutus dan menghubungkan arus listrik. Sistem kerja dari saklar mirip dengan gerbang yang dapat membuka dan menutup.

Limit switch merupakan salah satu contoh saklar khusus yang digunakan sebagai pendeteksi barang yang mana barang yang akan dideteksi tersebut harus dapat disentuh (dilakukan kontak) atau juga *limit switch* dapat digunakan sebagai respon batas dari suatu peralatan mekanik. Aktif dan non-aktifnya *limit switch* tergantung pada bagian dari peralatan mekanik yang menekan (mengaktifkan) saklar tersebut. Gambar 3.14. berikut adalah skematik dari sensor *limit switch*.



Gambar 3.14. Skematik Sensor *Limit switch*

Dalam perancangan alat ini sensor *limit switch* diatur *active low* (ketika aktif bernilai logika '0') pada kondisi awal (tidak ada benda yang menyentuh sensor) sensor bernilai satu (tidak aktif) karena posisi *limit switch* dalam keadaan open sehingga arus 5 Volt dari VCC akan mengalir ke *input* mikrokontroler. Ketika benda yang dideteksi menyentuh sensor dan terjadi kontak fisik maka switch akan menutup dan secara otomatis arus 0 Volt dari ground akan masuk dan diteruskan ke *input* mikrokontroler, kemudian mikrokontroler akan melakukan konversi data kedalam bentuk logika 0 dan 1 (logika 0 berarti sensor aktif dan logika 1 berarti sensor pasif).

3.1.2. Perancangan Mekanik

Mekanik merupakan salah satu bagian komponen yang sekaligus merupakan representasi dari penerapan aplikasi perancangan sistem ini. Oleh karena tugas akhir yang penulis buat ini bersifat rancang bangun (desain), maka pembahasan berikut ini berisi tentang gambaran skematik dari semua unsur komponen yang dibutuhkan untuk membangun sebuah alat produksi dan pengemasan sambal pecel.

Rancangan skematik proses produksi dan pengemasan sambal pecel ini meliputi:

A. Desain Mesin Penggiling Kacang dan Bumbu

Mesin penggiling kacang dan bumbu merupakan tahapan awal dari proses produksi sambal pecel ini, mesin ini berfungsi sebagai alat penggiling

(penghancur) kacang dan bumbu. Dalam mesin ini terdapat dua buah sensor photodiode yang masing-masing terpasang pada tempat kacang dan bumbu untuk melakukan deteksi terhadap ada atau tidak adanya kacang dan bumbu yang akan digiling. Mesin penggerak dari bagian alat ini adalah sebuah motor AC yang mana cara kerjanya adalah, ketika sensor photodiode yang terpasang pada tempat kacang dan bumbu aktif (terdapat bahan kacang dan bumbu) maka selanjutnya mikrokontroler akan memerintahkan motor AC untuk melakukan proses penggilingan. Setelah kacang dan bumbu habis dan kedua sensor photodiode tidak aktif maka motor AC akan berhenti.

B. Desain Mesin Penyampur Kacang dan Bumbu

Pada mesin bagaian ini berfungsi untuk melakukan penyampurn antara kacang dan bumbu, mesin ini digerakkan oleh motor AC yang akan menyala bersamaan dengan mesin penggiling kacang dan bumbu pengaktifannya melalui kedua sensor potodeode yang terpasang pada tempat kacang dan bumbu diatas. Setelah proses penyampuran terjadi, selanjutnya kacang yang sudah tercampur dengan bumbu (menjadi sambal pecel) akan keluar dari tempat penyampuran dan akan dideteksi oleh sensor photodiode (sensor_penyampuran). Sensor inilah yang selanjutnya berfungsi sebagai penghenti dari mesin penyampuran ini.

Mesin penggiling dan mesin penyampur kacang dan bumbu didesain secara menyatu seperti yang terlihat pada Gambar 3.15 berikut ini.



Gambar 3.15. Realisasi Desain Mesin Penggilingan dan Penyampuran.

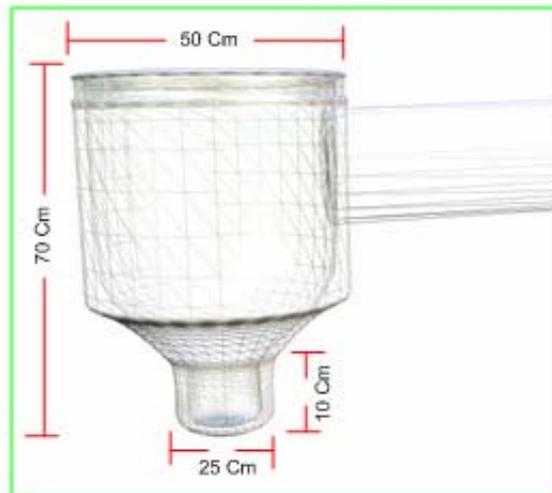
Keterangan Gambar 3.15. diatas:

1. Tempat sensor penyampuran dan keluarnya sambal pecel ke tempat penampungan.
2. Tabung penyampuran yang didalamnya terdapat mekanik yang berputar secara melingkar dan memlintir.
3. Ruang kendali mesin penyampuran.
4. Mesin penggiling kacang dan tempat sesnsor kacang ditempatkan

5. Mesin penggiling bumbu dan tempat sensor bumbu diletakkan
6. Tempat kacang diletakkan (penampungan bahan baku yang berupa kacang) sebelum proses penggilingan dilakukan.
7. Tempat penampungan bumbu sambal pecel.
8. Karet belt sebagai rantai penghubung gerakan motor AC dengan mesin penggilingan dan penyampuran.
9. Motor AC satu arah untuk menggerakkan mesin penggiling kacang dan bumbu.
10. Motor AC satu arah sebagai penggerak dari mesin penyampuran kacang dan bumbu.

C. Desain Tempat Penampungan Sambal

Setelah proses penyampuran kacang dan bumbu selesai maka dengan demikian kacang dan bumbu sudah menjadi sambal pecel dan siap untuk dilakukan proses berikutnya (pengemasan). Sebelum dilakukan proses pengemasan, sambal pecel ditampung terlebih dahulu kedalam tempat penampungan hal ini dilakukan dengan tujuan untuk mempermudah proses pengemasan dan sekaligus ketika nanti proses pengemasan berlangsung mesin penggiling kacang dan bumbu dapat dilakukan untuk melakukan proses penggilingan kembali Gambar 3.16 dan Gambar 3.17 berikut adalah ukuran dan realisasi dari tempat penampungan sambal pecel.



Gambar 3.16. Dimensi dan Ukuran Penampungan Sambal Pecel



Gambar 3.17. Realisasi Desain Penampungan Sambal Pecel

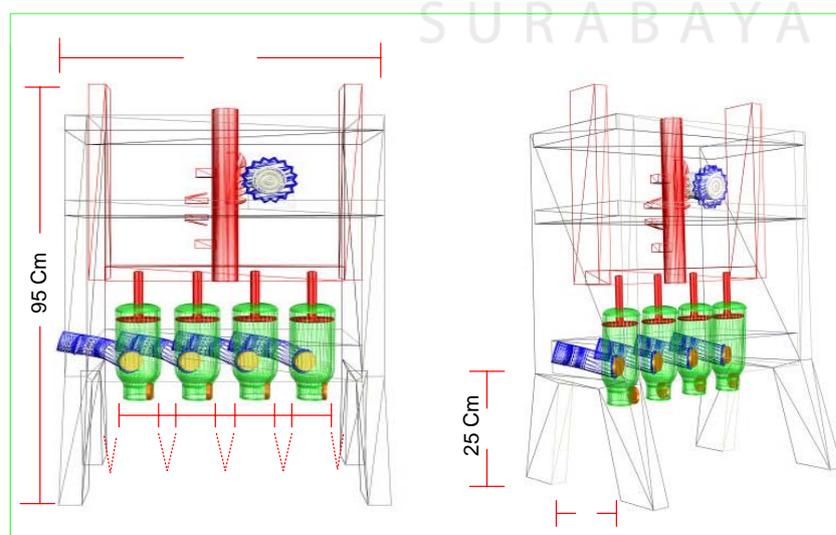
Keterangan gambar 3.17 desain penampungan:

1. Mesin penyampuran kacang dan bumbu
2. Tempat penampungan sambal pecel setelah proses penyampuran selesai
3. Sensor penampungan

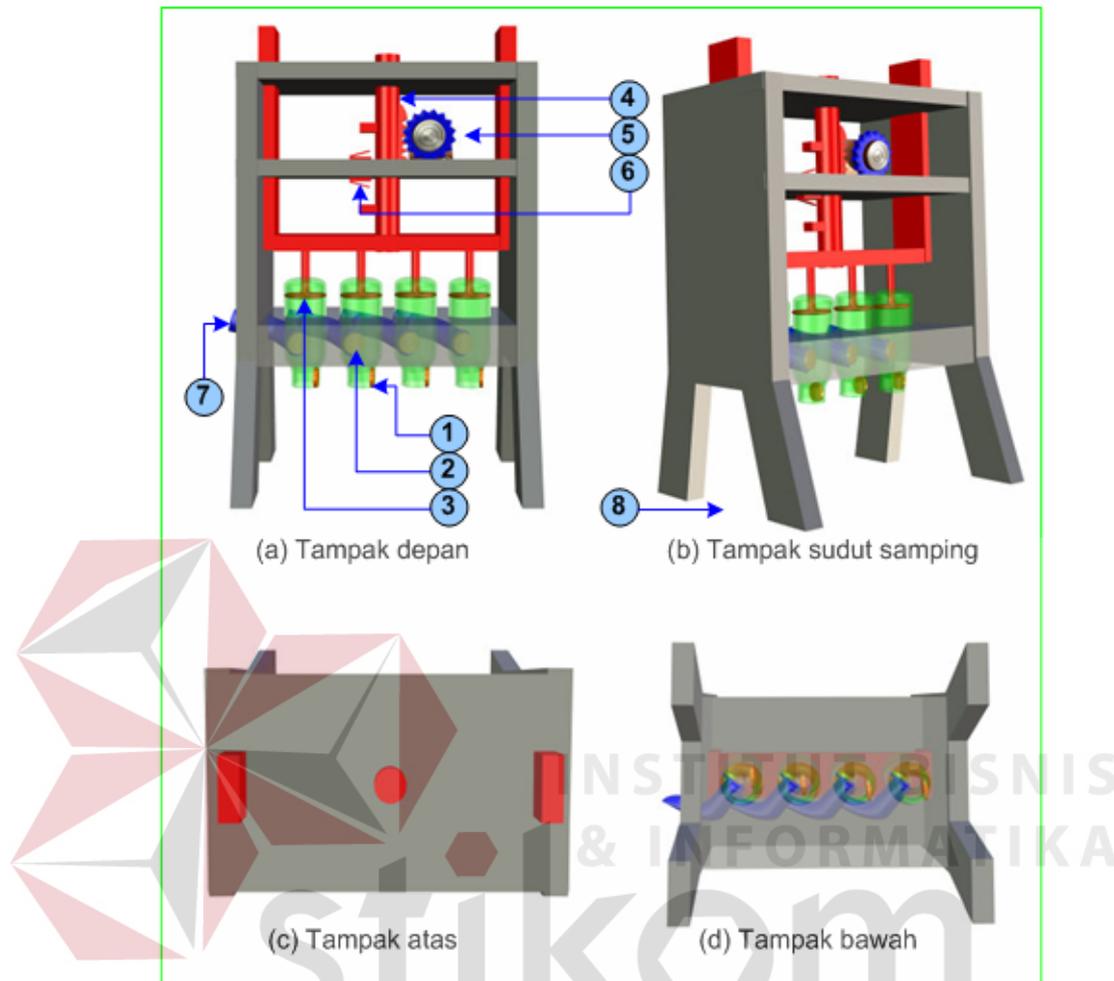
4. Nomor (4,5,6,7) pada gambar di atas, adalah pipa penghubung antara penampungan dengan kalibrasi yang nantinya akan dilewati sambal pecel untuk dilakukan proses pengemasan.

D. Desain Mesin Kalibrasi

Kalibrasi dalam alat ini diartikan sebagai suatu cara atau teknik pengukuran yang menggunakan media tertentu yang dimodifikasi sedemikian rupa dengan maksud dan tujuan agar hasil dari kalibrasi ini diharapkan sama ataupun mendekati sama (presisi) dengan hasil pengukuran yang sesungguhnya (pengukuran berat dalam satuan kilogram). Kalibrasi ini dilakukan agar proses pengemasan (pengisian sambal kedalam kaleng) dapat dilakukan dengan cepat dan tepat, karena dengan teknik yang cepat dan tepatlah hasil produksi yang maksimal dapat tercapai. Gambar 3.18. dan Gambar 3.19. berikut adalah ukuran dan realisasi dari mesin kalibrasi.



Gambar 3.18. Dimensi dan Ukuran Mesin Kalibrasi



Gambar 3.19. Realisasi Desain Mesin Kalibrasi

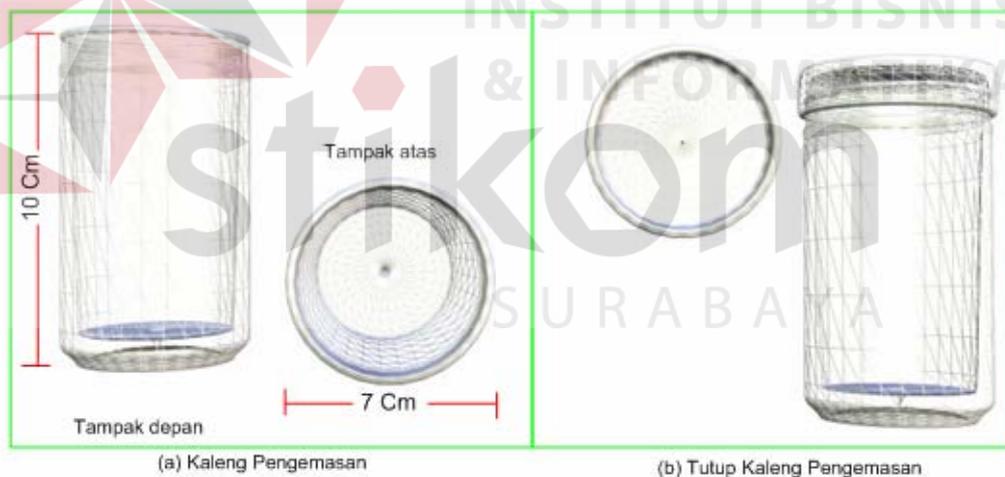
Keterangan Gambar 3.19. Realisasi mesin kalibrasi:

1. Katup pengatur keluarnya sambal pecel kedalam kaleng pengemasan.
2. Katup pengatur keluarnya sambal pecel dari tempat penampungan ke mesin kalibrasi.
3. Kelep pemompa dan pendorong sambal pecel.
4. Mekanik pengontrol gerakan pompa.
5. Motor DC 46 V penggerak naik turun pompa kalibrasi.
6. Sensor *limit switch* deteksi naik dan turun pompa kalibrasi.

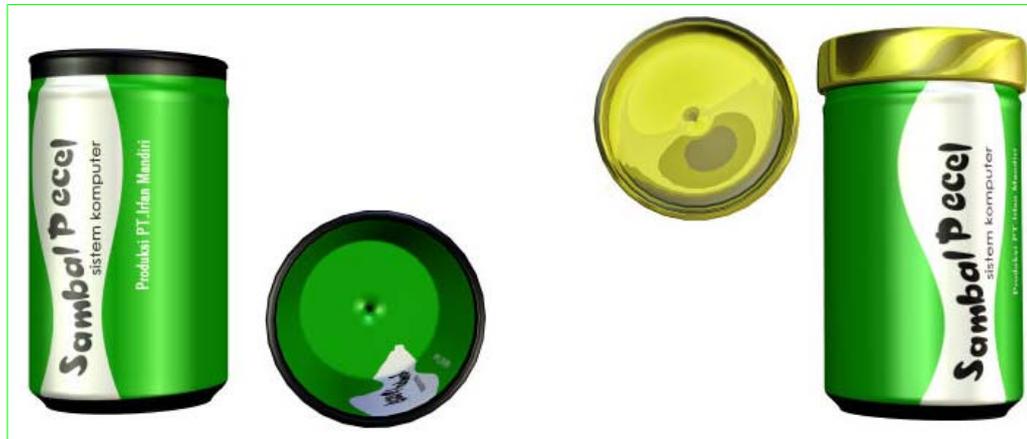
7. Pipa penghubung keluarnya sambal dari tempat penampungan ke mesin kalibrasi.
8. Tempat *conveyor* pembawa kaleng untuk proses pengemasan, tepat berada dibawah mesin kalibrasi.

E. Desain Kaleng Pengemasan

Dalam pengemasan ini setiap satu kaleng didesain mempunyai berat isi (berat sambal pecel) 1/4 Kilogram. Desain dan ukuran dari kaleng yang digunakan untuk mengemas sambal dapat dilihat pada Gambar 3.20. dan Gambar 3.21. berikut ini:



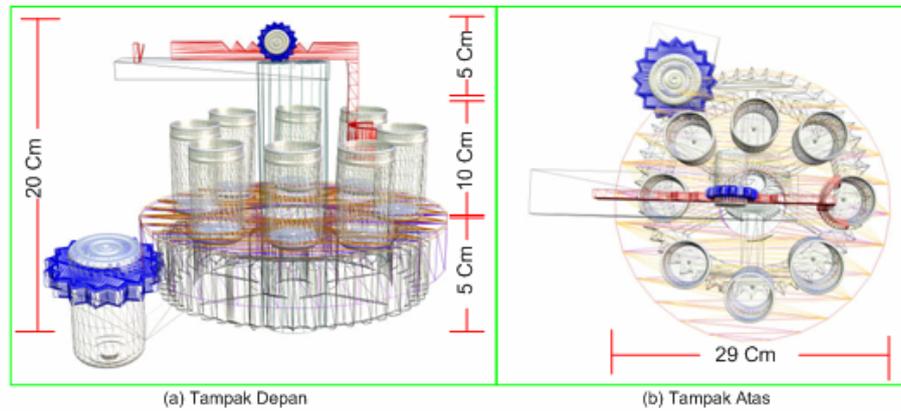
Gambar 3.20. Dimensi dan Ukuran Kaleng Pengemas Sambal Pecel



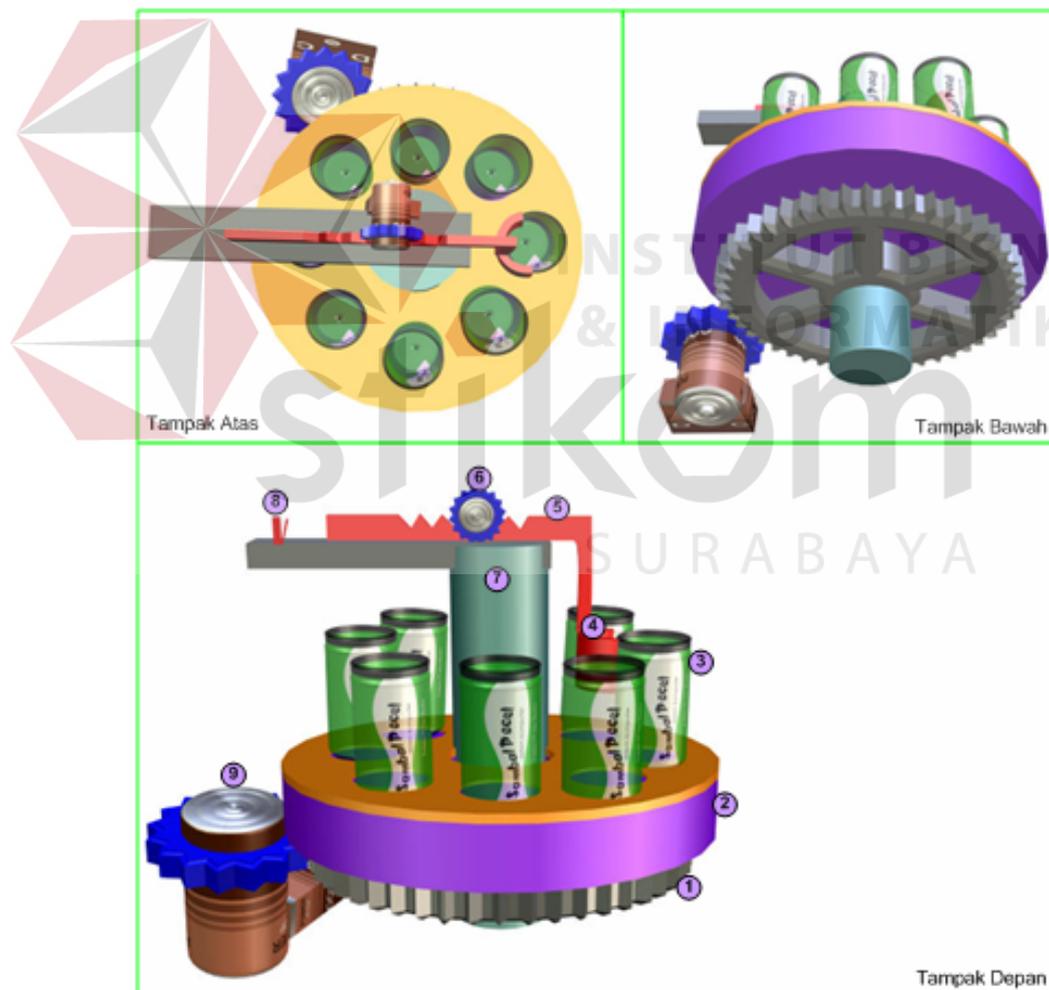
Gambar 3.21. Realisasi Desain Pengemas Sambal Pecel

F. Desain Mesin Antrian Kaleng Kosong

Sebelum dilakukan proses pengemasan sambal pecel, kaleng kosong di tempatkan pada mesin antrian kaleng. Mesin antrian kaleng berfungsi untuk mengatur keluarnya kaleng kosong, sehingga kaleng dapat keluar secara berurutan satu demi satu. Komponen penyusun dari mesin ini antara lain satu buah motor DC penggerak putaran antrian kaleng, satu buah motor DC pendorong maju dan mundur kaleng kosong ke *conveyor*, satu buah sensor photodiode yang digunakan untuk mendeteksi kaleng yang akan didorong serta untuk menghentikan putaran dari antrian kaleng kosong, dan satu buah sensor *limit switch* pengontrol pendorong kaleng kekonveyor. Ukuran dan realisasi desain mesin antrian dapat dilihat pada Gambar 3.22. dan Gambar 3.23.



Gambar 3.22. Dimensi dan Ukuran Mesin Antrian Kaleng



Gambar 3.23. Realisasi Mesin Antrian Kaleng Kosong

Keterangan Gambar 3.23. Mesin antrian kaleng kosong:

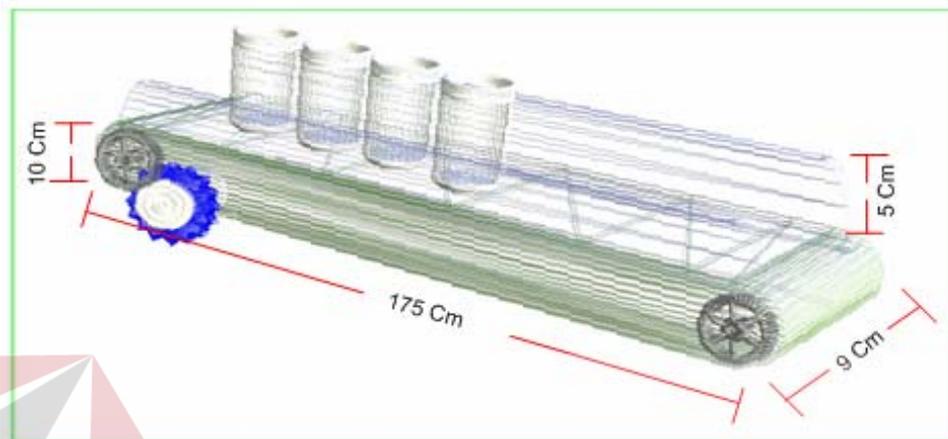
1. *Gear bok* penggerak piringan antrian kaleng (*main drive*).
2. *Main drive* tempat antrian kaleng yang bergerak berputar.
3. Antrian kaleng kosong.
4. Penampang pendorong botol yang didalamnya terdapat sensor pendeteksi barang (kaleng).
5. Mekanik penggerak maju mundur pendorong kaleng ke *conveyor*.
6. Motor DC penggerak mekanik pendorong kaleng ke *conveyor*
7. As (poros) *main drive* dan tempat tumpuan mekanik pendorong
8. Sensor *limit switch* untuk menghentikan gerakan mundur motor DC penggerak mekanik pendorong kaleng ke *conveyor*
9. Motor penggerak putaran *main drive* antrian kaleng kosong.

G. **Desaian Conveyor**

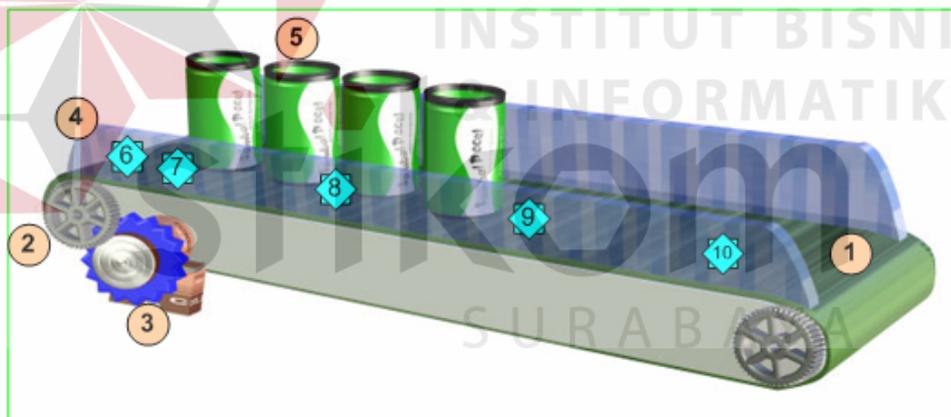
Conveyor berasal dari kata “convoy” yang artinya berjalan bersama dalam suatu grup besar. Fungsi dari suatu *conveyor*, yaitu mengangkut suatu barang dalam jumlah tertentu dan dapat mengatasi jarak tertentu yang diberikan.

Conveyor yang digunakan dalam perancangan ini menggunakan *conveyor* jenis *pressless combain* yaitu *conveyor* yang dirancang dengan menggunakan komponen rantai yang didesain datar sehingga dapat membawa kaleng berjalan di atasnya. Rantai digerakkan oleh poros melalui *sprocket* yang telah didesain khusus. Putaran poros berasal dari motor DC yang dipasang pada

frame *conveyor*. Gambar 3.24. dan Gambar 3.25 berikut adalah ukuran dan realisasi desain *conveyor* proses pengemasan sambal pecel.



Gambar 3.24. Dimensi dan Ukuran *Conveyor*



Gambar 3.25. Realisasi Desain *Conveyor*

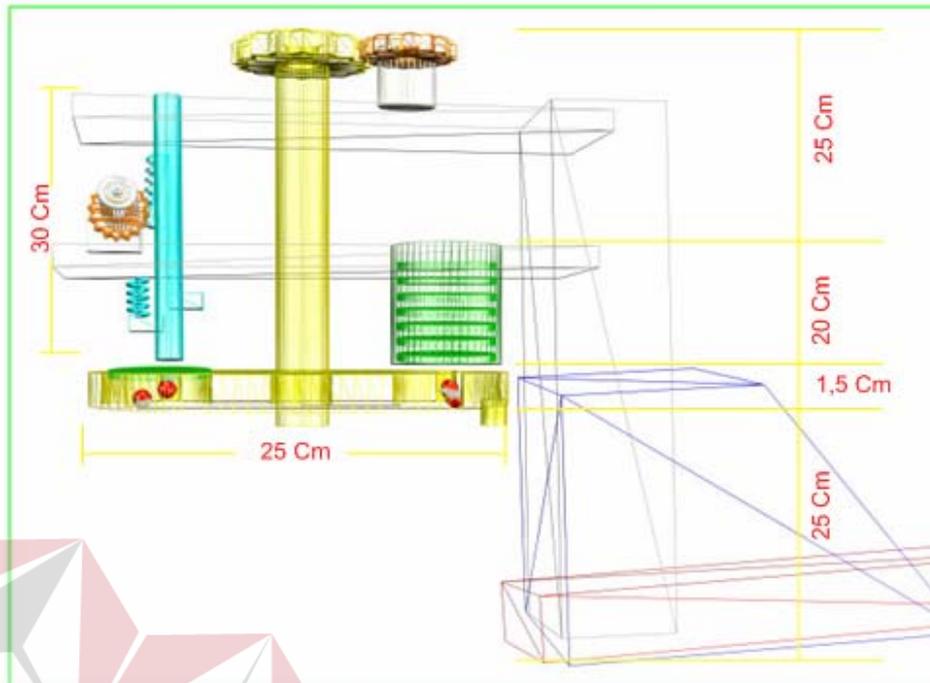
Keterangan Gambar 3.25. di atas adalah:

1. *Pressless combain* (bagian *conveyor* yang bergerak membawa kaleng)
2. Rangkaian *sprocket* yang terhubung dengan poros untuk menggerakkan *conveyor*.
3. Motor DC penggerak poros.

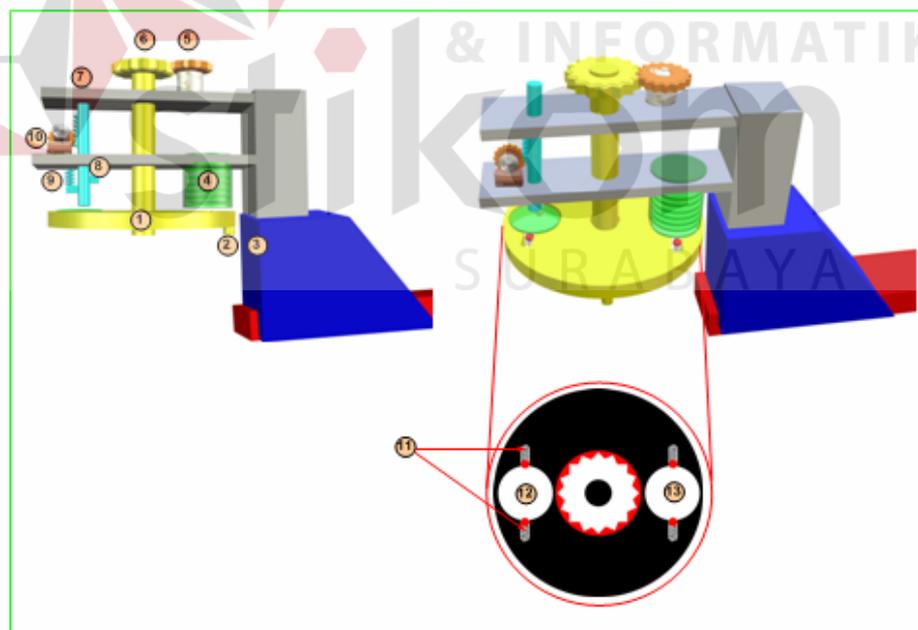
4. Dinding pagar kanan dan kiri untuk meletakkan sensor untuk mendeteksi dan melakukan proses-proses yang diinginkan.
5. Antrian kaleng yang siap untuk dilakukan proses-proses pengemasan.
6. Sensor jarak (untuk mengatur jarak antar kaleng)
7. Sensor Counter (untuk menghitung jumlah kaleng yang siap diproses, dalam hal ini kaleng yang siap untuk diproses sebanyak empat buah)
8. Sensor Isi (untuk mendeteksi bahwa kaleng siap untuk dilakukan pengisian)
9. Sensor tutup kaleng (untuk mendeteksi bahwa kaleng siap untuk dilakukan proses penutupan)
10. Sensor hasil sebagai penghitung hasil produksi, apakah kaleng yang keluar dari antrian sama dengan setelah proses pengemasan.

H. Desain Mesin Penutup (*Capping*) Kaleng.

Setelah proses pengisian selesai yang di tandai dengan adanya informasi dari mikrokontroler pertama ke mikrokontroler kedua, maka selanjutnya *conveyor* akan bergerak untuk membawa kaleng yang telah terisi sambal pecel untuk dilakukan penutupan. Mesin penutup kaleng terdiri dari sebuah sensor photodiode yang terletak pada *conveyor* untuk mendeteksi kaleng yang siap untuk dilakukan penutupan, dua buah sensor *limit switch* yang terletak pada mesin pendorong tutup kaleng dan pada antrian tutup kaleng, dua buah motor DC sebagai penggerak pendorong tutup kaleng dan penggerak antrian tutup kaleng. Ukuran dan realisasi desain mesin penutup kaleng dapat dilihat pada Gambar 3.26 dan Gambar 3.27. berikut.



Gambar 3.26. Dimensi dan Ukuran Mesin Penutup Kaleng



Gambar 3.27. Realisasi Desain Penutup Kaleng

Keterangan Gambar 3.27. Desain penutup kaleng:

1. *Main drive* antrian tutup kaleng
2. Pemicu sensor antrian tutup kaleng
3. Sensor *limit switch* antrian tutup kaleng (untuk memutar *main drive* 180°)
4. Tabung untuk menumpuk tutup kaleng
5. Motor DC Penggerak poros yang terhubung dengan *main drive* antrian tutup kaleng
6. Gear bok poros untuk memutar antrian tutup kaleng
7. Pendorong tutup kaleng yang telah siap untuk dilakukan penutupan.
8. sensor pendorong tutup kaleng
9. Pegas untuk menaikkan pendorong tutup kaleng setelah melakukan pendorongan terhadap tutup kaleng
10. Motor DC penggerak pendorong tutup kaleng
11. Rangkaian pegas dan gotri untuk mempermudah dan menahan tutup kaleng agar tidak jatuh sebelum disodok oleh penyodok
12. Tempat tutup yang telah siap
13. Tempat penumpukan tutup.

3.2. Perancangan Perangkat Lunak

Perancangan perangkat lunak dalam desain produksi dan pengemasan sambal pecel ini meliputi program pada mikrkontroler pertama yang mengontrol proses produksi dan perangkat lunak pada mikrokontroler kedua yang digunakan untuk mengontrol semua proses pengemasan.

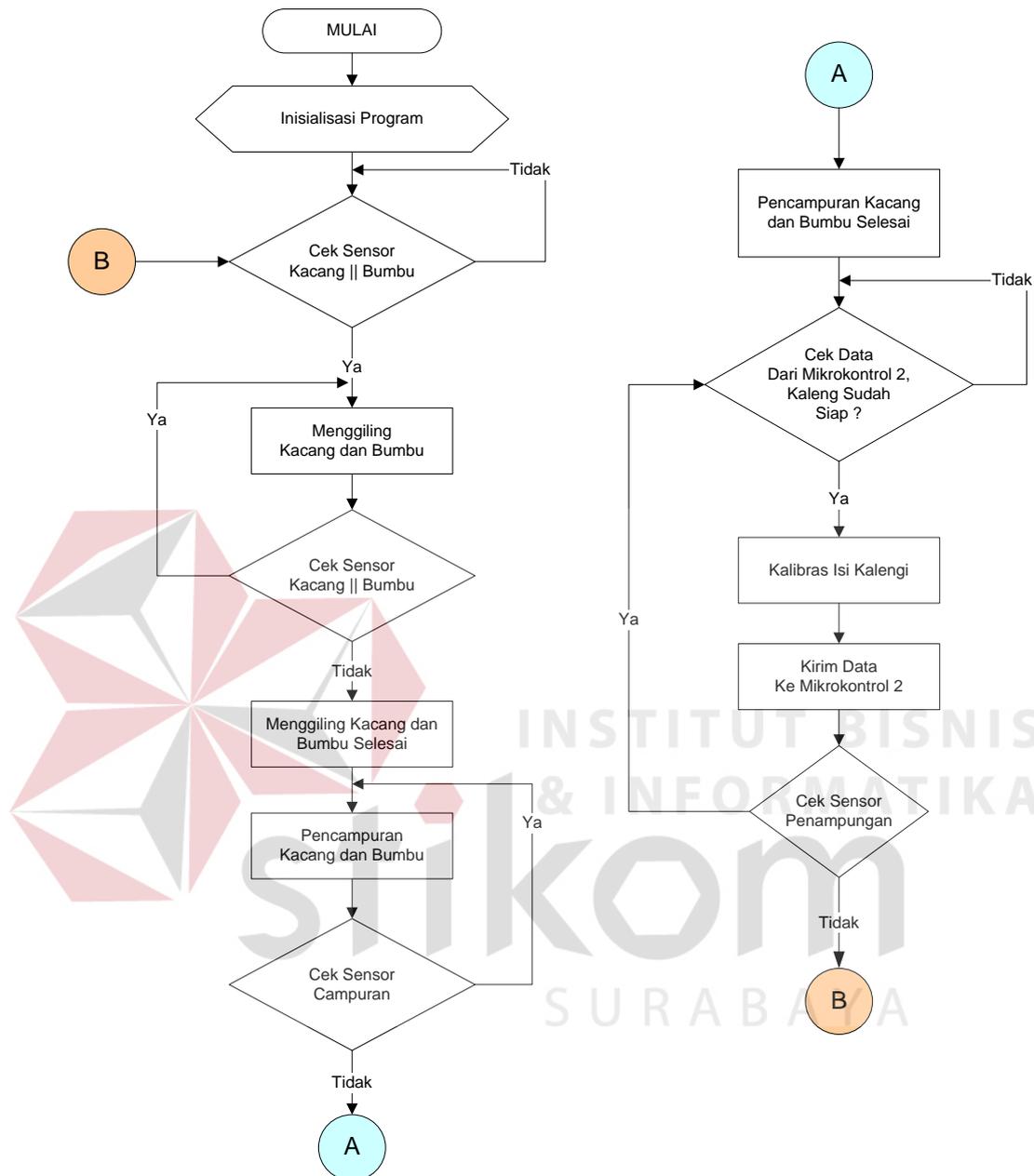
Bahasa pemrograman yang digunakan dalam sistem ini yaitu menggunakan bahasa pemrograman *Assembly* MCS-51, sedangkan *software* yang digunakan untuk pembuatan program adalah *ProView32*.

3.2.1. Perangkat Lunak Pada Mikrokontroler Pertama

Sebelum pembuatan program pada mikrokontroler pertama dilakukan, langkah awal yang perlu kita lakukan untuk mempermudah pembuatan program nantinya adalah pembuatan diagram alur.

Secara garis besar cara kerja dari mikrokontroler pertama adalah membaca *inputan* dari sensor, mengolah data dari sensor sesuai dengan apa yang kita inginkan dan kemudian mikrokontroler akan melakukan perintah untuk menjalankan motor AC ataupun motor DC dengan fungsi yang telah kita tentukan.

Tahapan-tahapan yang dilakukan oleh mikrokontroler pertama diatas dapat digambarkan kedalam sebuah alur program, seperti Gambar 3.28. Diagram alur berikut ini.



Gambar 3.28. Diagram Alur Mikrokontroler Pertama

Setelah kita dapat membuat diagram alur dan memahami tahapan-tahapan yang dilakukan oleh mikrokontroler pertama, maka selanjutnya kita dapat memulai pembuatan program.

Dibawah ini adalah tahap demi tahap dari program mikrokontroler pertama.

a. Inisialisasi *Port*.

```
ORG 100H
```

```

SENSOR_KACANG          BIT          P1.0
SENSOR_BUMBU           BIT          P1.1
SENSOR_CAMPURAN        BIT          P1.2
SENSOR_PENAMPUNGAN     BIT          P1.3
SENSOR_KAL_ATAS        BIT          P1.4
SENSOR_KAL_BAWAH       BIT          P1.5
MOTOR_AC_KACANG        BIT          P0.0
MOTOR_AC_CAMPURAN      BIT          P0.1
MOTOR_DC_KAL_NAIK      BIT          P0.2
MOTOR_DC_KAL_TURUN     BIT          P0.3

```

```
LJMP START
```

```

MOTOR_AC_KACANG_AKTIF :
    SETB    MOTOR_AC_KACANG
RET

```

```

MOTOR_AC_KACANG_MATI :
    CLR     MOTOR_AC_KACANG    ; kacang dan bumbu
RET

```

```

MOTOR_AC_CAMPURAN_AKTIF :
    SETB    MOTOR_AC_CAMPURAN
RET

```

```

MOTOR_AC_CAMPURAN_MATI :
    CLR     MOTOR_AC_CAMPURAN
RET

```

```

MOTOR_KALIBRASI_NAIK :
    SETB    MOTOR_DC_KAL_NAIK
    CLR     MOTOR_DC_KAL_TURUN
RET

```

```

MOTOR_KALIBRASI_TURUN :
    CLR     MOTOR_DC_KAL_NAIK
    SETB    MOTOR_DC_KAL_TURUN
RET

```

```

MOTOR_KALIBRASI_STOP :
    CLR     MOTOR_DC_KAL_NAIK
    CLR     MOTOR_DC_KAL_TURUN
RET

```

```

MOTOR_AC_KACANG_AKTIF :
    SETB    MOTOR_AC_KACANG    ; kacang dan bumbu
RET

```

```

RESET_PORT :
    MOV     P0 , #0
    MOV     P1 , #0
    MOV     P2 , #0
    MOV     P3 , #0
RET

```

```

DELAY:
    MOV     R0 , #08H
DELAY_2:
    MOV     R1 , #0FFH
DELAY_1:
    MOV     R2 , #0FFH
DELAY_0:
    DJNZ   R2 , DELAY_0
    DJNZ   R1 , DELAY_1
    DJNZ   R0 , DELAY_2
RET

```

- b. Menunggu data dari sensor pada *pin* 1.0 dan *pin* 1.1 untuk mengetahui apakah bahan berupa kacang dan bumbu sudah siap atau belum untuk digiling. Jika bahan sudah siap, data dikirim ke driver motor AC untuk menggerakkan motor AC melalui *pin* P0.0 untuk proses penggilingan kacang dan bumbu yang hasilnya akan ditampung untuk proses pencampuran.

```

START:
    CALL   RESET_PORT

CEK_SENSOR_KACANG :
    JNB    SENSOR_KACANG , CEK_SENSOR_BUMBU
    CALL   MOTOR_AC_KACANG_AKTIF

CEK_SENSOR_BUMBU :
    JNB    SENSOR_BUMBU , CEK_SENSOR_KACANG

CEK_SENSOR_KACANG2 :
    JB     SENSOR_KACANG , CEK_SENSOR_BUMBU2

CEK_SENSOR_BUMBU2 :
    JB     SENSOR_BUMBU , CEK_SENSOR_KACANG2

GILING_KACANG_SELESAI :
    CALL   MOTOR_AC_KACANG_MATI

```

- c. *Pin* P1.2 menunggu data dari sensor campuran kacang dan bumbu yang selanjutnya mengirim data ke driver motor AC untuk menggerakkan motor AC melalui *pin* P0.1 untuk melakukan proses pencampuran sampai selesai. Dari proses ini menghasilkan bumbu sambal pecel yang ditampung dan siap dilakukan proses pengemasan.

PROSES_PENCAMPURAN:

```
CALL    MOTOR_AC_CAMPURAN_AKTIF
JNB     SENSOR_CAMPURAN, $ ;loop sampai
        pencampuran selesai
```

PROSES_PENCAMPURAN_SELESAI:

```
JB      SENSOR_CAMPURAN, $
CALL    MOTOR_AC_CAMPURAN_MATI
```

CEK_SENSOR_PENAMPUNGAN:

```
JNB     SENSOR_PENAMPUNGAN, $
```

- d. Setelah proses pencampuran selesai yang diketahui dari sensor campuran kacang dan bumbu pada *pin* P1.2, dilanjutkan dengan proses pengisian sambal pecel ke dalam kaleng kemasan melalui proses kalibrasi dengan aturan kaleng kosong yang akan diisi sambal pecel harus sudah siap di bawah mesin kalibrasi yang diketahui dari *Port* 2. Pada sistem ini *port* 2 pada Mikrokontroler 1 melakukan komunikasi dengan *port* 2 pada Mikrokontroler 2.

TUNGGU_KALENG:

```
MOV     A, P2           ;ambil data dari mikro-2
LCALL   DELAY
CJNE    A, #11H, TUNGGU_KALENG
```

- e. Proses kalibrasi dilakukan oleh motor DC yang menggerakkan alat mekanik naik dan turun melalui *pin* P0.2 dan *pin* P0.3. Pada saat naik tabung kalibrasi akan terisi sambal pecel dari penampung melalui lubang pipa sampai sensor kalibrasi_atas aktif yang terhubung ke *pin* P1.4. Selanjutnya

mekanik bergerak turun untuk mengeluarkan sambal pecel ke kaleng kemasan melalui lubang bawah (lubang pengisian sambel ke kaleng) sampai sensor kalibrasi_bawah aktif yang terhubung ke *pin* P1.5. Pada proses ini Mikrokontroler 1 mengirim informasi ke Mikrokontroler 2 melalui *port* 2 bahwa pengisian selesai dan *conveyor* bergerak untuk membawa kaleng yang sudah terisi ke mesin penutup kaleng. Proses pengemasan ini selesai apabila sambal pecel dalam penampungan habis, yang diketahui melalui sensor penampungan pada *pin* P1.3

```

KALIBRASI_NAIK:
    CALL    MOTOR_KALIBRASI_NAIK
    JNB     SENSOR_KAL_ATAS, $    ; loop sampai
          terdeteksi sensor kal atas

KALIBRASI_TURUN:
    CALL    MOTOR_KALIBRASI_TURUN
    JNB     SENSOR_KAL_BAWAH, $    ; loop sampai
          terdeteksi sensor kal bawah

KALENG_SUDAH_TERISI:
    CALL    MOTOR_KALIBRASI_STOP

KIRIM_DATA_KE_MIKRO2:

    MOV     A, #22H
    MOV     P2, A
    LCALL   DELAY
    JNB     SENSOR_PENAMPUNGAN, KEMBALI_KEPROSES_AWAL
    SJMP    TUNGGU_KALENG

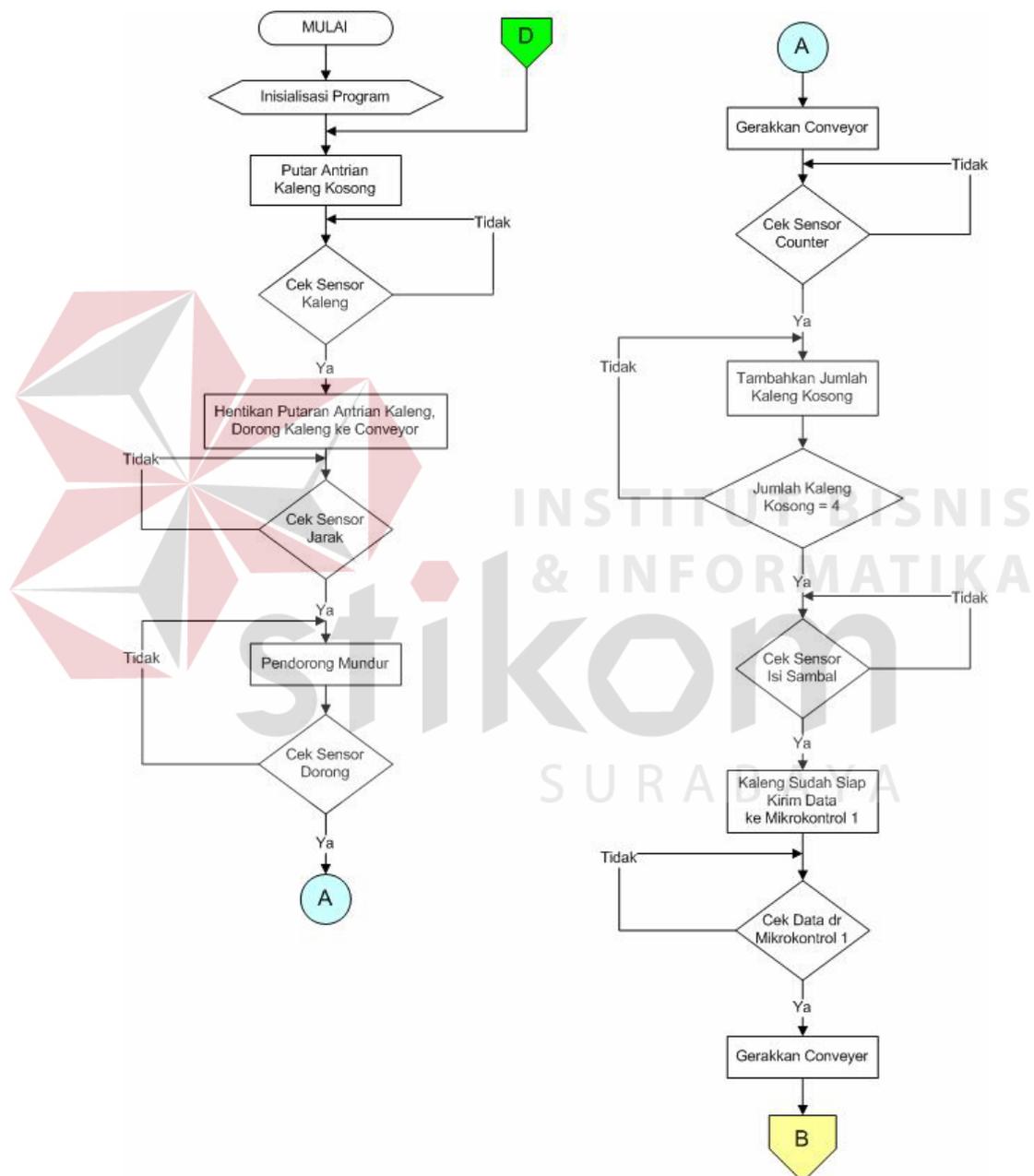
KEMBALI_KEPROSES_AWAL:
    LJMP    START
END

```

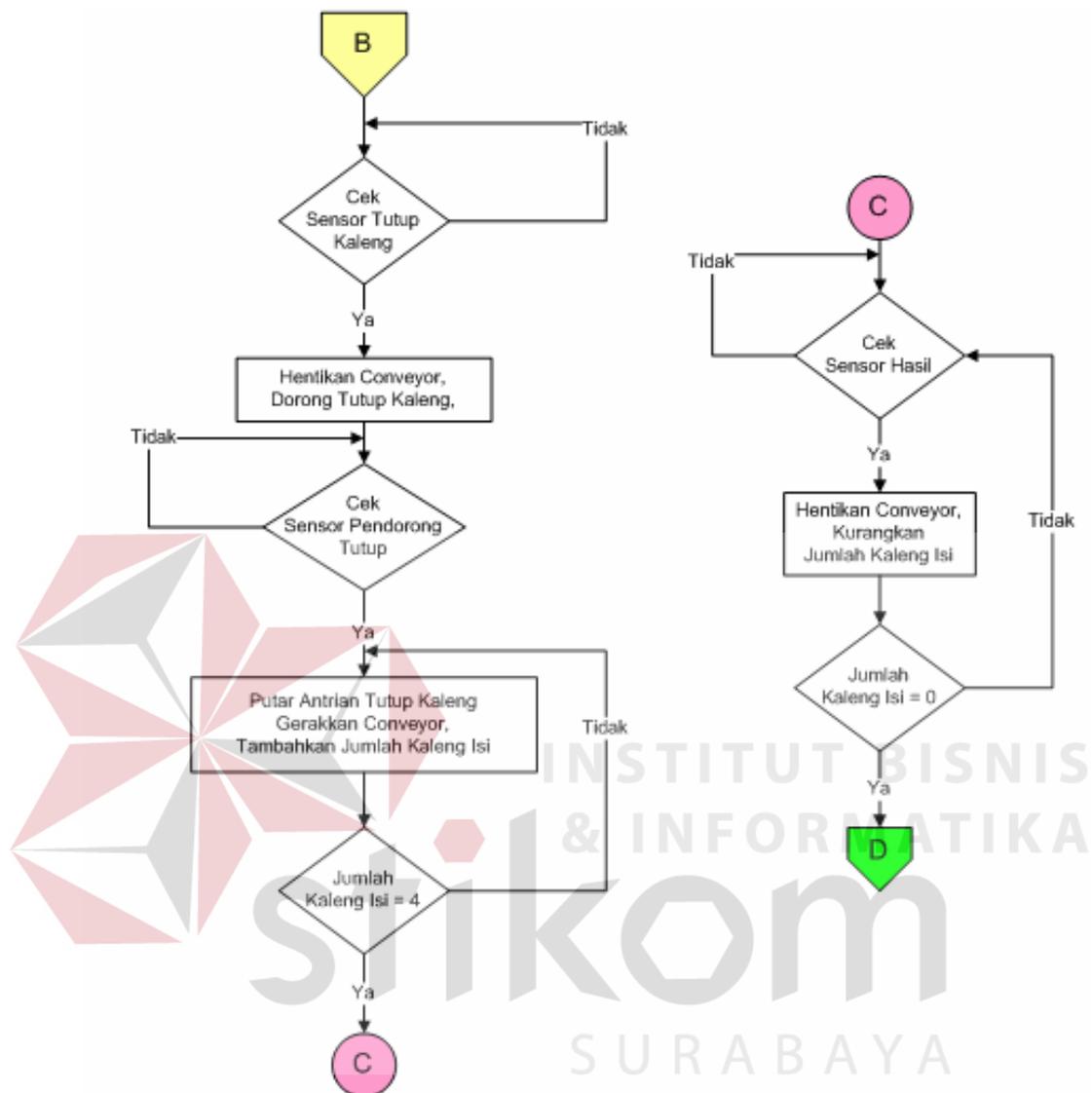
3.2.2. Perangkat Lunak Pada Mikrokontroler Kedua

Seperti halnya dengan Mikrokontroler pertama, Mikrokontroler kedua secara umum bertugas untuk membaca sensor, mengolah data dari sensor dan memberikan aksi untuk mengatur gerakan Motor DC.

Gambar 3.29 berikut adalah gambar diagram alur dari mikrokontroler kedua.



Gambar 3.29. (a) Diagram Alur Mikrokontroler kedua



Gambar 3.29. (b) Diagram Alur Mikrokontroler kedua

Pembuatan perangkat lunak dibuat secara tahap demi tahap sesuai dengan diagram alur diatas, hal ini dilakukan untuk mempermudah dalam pemahaman dan penyusunannya. Langkah-langkahnya adalah sebagai berikut:

a. Inisialisasi port

```
ORG 100H
```

```
SENSOR_DORONG_KALENG BIT P3.2
SENSOR_ANTRIAN_KALENG BIT P1.0
SENSOR_JARAK_KALENG BIT P1.1
SENSOR_COUNTER_KALENG BIT P1.2
SENSOR_ISI_KALENG BIT P1.3
SENSOR_TUTUP_KALENG BIT P1.4
SENSOR_DORONG_TUTUP BIT P1.5
SENSOR_ANTRIAN_TUTUP BIT P1.6
SENSOR_HASIL BIT P1.7
```

```
MOTOR_DC_ANTRIAN_KALENG1 BIT P0.0
MOTOR_DC_ANTRIAN_KALENG2 BIT P0.1
```

```
MOTOR_DC_KALENG_MAJU BIT P0.2
MOTOR_DC_KALENG_MUNDUR BIT P0.3
```

```
MOTOR_DC_CONVEYOR1 BIT P0.4
MOTOR_DC_CONVEYOR2 BIT P0.5
```

```
MOTOR_DC_TUTUP1 BIT P0.6
MOTOR_DC_TUTUP2 BIT P0.7
```

```
MOTOR_DC_ANTRIAN_TUTUP1 BIT P3.0
MOTOR_DC_ANTRIAN_TUTUP2 BIT P3.1
```

```
LJMP START
```

```
MOTOR_ANTRIAN_KALENG_AKTIF:
```

```
SETB MOTOR_DC_ANTRIAN_KALENG1
CLR MOTOR_DC_ANTRIAN_KALENG2
```

```
RET
```

```
MOTOR_ANTRIAN_KALENG_MATI:
```

```
CLR MOTOR_DC_ANTRIAN_KALENG1
CLR MOTOR_DC_ANTRIAN_KALENG2
```

```
RET
```

```
MOTOR_DORONG_KALENG_MAJU:
```

```
SETB MOTOR_DC_KALENG_MAJU
CLR MOTOR_DC_KALENG_MUNDUR
```

```
RET
```

```
MOTOR_DORONG_KALENG_MUNDUR:
```

```
CLR MOTOR_DC_KALENG_MAJU
SETB MOTOR_DC_KALENG_MUNDUR
```

```
RET
```

```
MOTOR_DORONG_KALENG_STOP:
```

```
CLR MOTOR_DC_KALENG_MAJU
CLR MOTOR_DC_KALENG_MUNDUR
```

```
RET
```

```
MOTOR_CONVEYOR_AKTIF:
```

```
SETB MOTOR_DC_CONVEYOR1
```

```

        CLR          MOTOR_DC_CONVEYOR2
RET

```

```

MOTOR_CONVEYOR_MATI :

```

```

        CLR          MOTOR_DC_CONVEYOR1
        CLR          MOTOR_DC_CONVEYOR2
RET

```

```

MOTOR_DORONG_TUTUP_AKTIF:

```

```

        SETB         MOTOR_DC_TUTUP1
        CLR          MOTOR_DC_TUTUP2
RET

```

```

MOTOR_DORONG_TUTUP_MATI :

```

```

        CLR          MOTOR_DC_TUTUP1
        CLR          MOTOR_DC_TUTUP2
RET

```

```

MOTOR_ANTRIAN_TUTUP_AKTIF:

```

```

        SETB         MOTOR_DC_ANTRIAN_TUTUP1
        CLR          MOTOR_DC_ANTRIAN_TUTUP2
RET

```

```

MOTOR_ANTRIAN_TUTUP_MATI :

```

```

        CLR          MOTOR_DC_ANTRIAN_TUTUP1
        CLR          MOTOR_DC_ANTRIAN_TUTUP2
RET

```

```

RESET_KALENG :

```

```

        MOV          R6, #0           ; increment/tambah jumlah kaleng kosong
        MOV          R7, #0           ; increment/tambah jumlah kaleng isi
RET

```

```

RESET_PORT :

```

```

        MOV          P0, #0
        MOV          P1, #0
        MOV          P2, #0
        MOV          P3, #0
RET

```

```

DELAY :

```

```

        MOV          R0, #08H

```

```

DELAY_2 :

```

```

        MOV          R1, #0FFH

```

```

DELAY_1 :

```

```

        MOV          R2, #0FFH

```

```

DELAY_0 :

```

```

        DJNZ         R2, DELAY_0

```

```

        DJNZ         R1, DELAY_1

```

```

        DJNZ         R0, DELAY_2

```

```

RET

```

- b. Motor DC memutar Antrian kaleng kosong melalui *pin* P0.0 dan *pin* P0.1 sampai terdeteksi sensor antrian_kaleng yang terhubung pada *pin* P1.0.

```

START:

```

```
CALL    RESET_KALENG
CALL    RESET_PORT
```

```
AKTIFKAN_ANTRIAN_KALENG:
```

```
CALL    MOTOR_ANTRIAN_KALENG_AKTIF
JNB     SENSOR_ANTRIAN_KALENG, $ ; loop sampai Sensor
        antrian kaleng on
```

```
STOP_ANTRIAN_KALENG:
```

```
CALL    MOTOR_ANTRIAN_KALENG_MATI
```

- c. Selanjutnya motor DC yang terhubung dengan *pin* P0.2 dan *pin* P0.3 menggerakkan pendorong maju untuk mendorong kaleng kesatu ke *conveyor* sampai terdeteksi oleh sensor jarak yang terhubung dengan *pin* P1.1.

```
DORONG_KALENG_KE_CONVEYOR:
```

```
CALL    MOTOR_DORONG_KALENG_MAJU
JNB     SENSOR_JARAK_KALENG, $ ; loop sampai sensor jrk on
```

```
CEK_SENSOR_JARAK:
```

```
JB      SENSOR_JARAK_KALENG, $ ;loop sampai snsr jarak tidak
        aktif, kaleng melewati snsr jarak
```

- d. Pendorong kembali mundur dan berhenti sampai terkena sensor pendorong yang terhubung dengan *pin* P3.2 dan motor DC yang terhubung dengan *pin* P0.4 dan *pin* P0.5 menggerakkan *conveyor* sampai kaleng kesatu terdeteksi oleh sensor counter yang terhubung dengan *pin* P1.2.

```
PENDORONG_KALENG_MUNDUR:
```

```
CALL    MOTOR_DORONG_KALENG_MUNDUR
JNB     SENSOR_DORONG_KALENG, $ ; loop sampai sns dorong
        kaleng on
```

```
STOP_PENDORONG_KALENG:
```

```
CALL    MOTOR_DORONG_KALENG_STOP
```

```
JALANKAN_CONVEYOR_UNTUK_COUNTER:
```

```
CALL    MOTOR_CONVEYOR_AKTIF
JNB     SENSOR_COUNTER_KALENG, $ ; loop sampai sensor
        counter aktif, tepat di depan sensor counter
```

```
CEK_SENSOR_COUNTER:
```

```
JB      SENSOR_COUNTER_KALENG, $ ; loop sampai sensor
        counter tidak aktif, kaleng melewati sensor counter
```

```
STOP_CONVEYOR_UNTUK_COUNTER:
```

```
CALL    MOTOR_CONVEYOR_MATI
```

```
HITUNG_KALENG:
```

```
INC     R6 ; tambahkan jumlah kaleng kosong
CJNE   R6, #4, AKTIFKAN_ANTRIAN_KALENG
```

- e. Antrian kaleng kosong berputar lagi dan berhenti jika kaleng kedua terdeteksi oleh sensor `antrian_kaleng`. Kaleng kedua didorong sampai mengenai sensor jarak kemudian pendorong bergerak mundur dan *conveyor* bergerak lagi sampai kaleng kedua terdeteksi oleh sensor counter, begitu seterusnya sampai empat buah kaleng. Setelah empat buah kaleng berada diatas *conveyor* selanjutnya *conveyor* bergerak membawa kaleng dan berhenti sampai kaleng pertama terdeteksi oleh sensor `isi_kaleng` yang terhubung dengan *pin* P1.3.

```
JALANKAN_CONVEYOR_UNTUK_PENGISIAN:
    CALL    MOTOR_CONVEYOR_AKTIF
    JNB     SENSOR_ISI_KALENG, $ ; loop sampai sensor pengisian
        aktif
```

```
STOP_CONVEYOR_UNTUK_PENGISIAN:
    CALL    MOTOR_CONVEYOR_MATI
```

- f. Setelah kaleng kosong berada di bawah mesin kalibrasi dan siap untuk dilakukan pengisian sambal pecel, mikrokontroler kedua mengirim informasi ke mikrokontroler pertama melalui *port* 2 untuk melakukan proses pengisian sambal pecel.

```
KIRIM_DATA_KE_MIKRO1:
    MOV     A, #11H
    MOV     P2, A
    LCALL   DELAY
```

```
TUNGGU_KALENG_TERISI:
    MOV     A, P2
    ; ambil data dari mikro-1
    LCALL   DELAY
    CJNE    A, #33H, TUNGGU_KALENG_TERISI
```

- g. Setelah kaleng terisi, Mikrokontroler kedua mendapat informasi dari Mikrokontroler pertama melalui *port* 2 yang kemudian *conveyor* bergerak membawa kaleng yang sudah terisi ke mesin penutup kaleng sampai kaleng kesatu yang sudah terisi terdeteksi oleh sensor `tutup_kaleng` yang terhubung dengan *pin* P1.4.

```
JALANKAN_CONVEYOR_UNTUK_TUTUP_KALENG:
    CALL    MOTOR_CONVEYOR_AKTIF
    JNB     SENSOR_TUTUP_KALENG, $ ; loop sampai sensor tutup
        kaleng aktif
```

- h. *Conveyor* berhenti dan motor DC yang terhubung dengan *pin* P0.6 dan *pin* P0.7 mendorong tutup kaleng bergerak turun mendorong tutup ke kaleng satu yang sudah terisi.

```
CEK_SENSOR_TUTUP:
    JB          SENSOR_TUTUP_KALENG, $ ; loop sampai sensor tutup
              kaleng pasif

STOP_CONVEYOR_UNTUK_TUTUP_KALENG:
    CALL       MOTOR_CONVEYOR_MATI

DORONG_TUTUP_KALENG:
    CALL       MOTOR_DORONG_TUTUP_AKTIF
    JNB       SENSOR_DORONG_TUTUP, $ ; loop sampai sensor dorong
              tutup aktif

STOP_DORONG_TUTUP:
    CALL       MOTOR_DORONG_TUTUP_MATI
```

- i. Pada saat pendorong tutup bergerak turun, sensor pendorong_tutup yang terhubung dengan *pin* P1.5 akan pasif (0), lalu naik lagi dan sensor pendorong_tutup akan aktif (1). Selanjutnya *conveyor* bergerak dan motor DC yang terhubung dengan *pin* P3.0 dan *pin* P3.1 memutar antrian tutup 180° dengan bantuan sensor antrian_tutup yang terhubung dengan *pin* P1.6 untuk menyiapkan tutup yang kedua.

```
JALANKAN_ANTRIAN_TUTUP:
    CALL       MOTOR_ANTRIAN_TUTUP_AKTIF
    JNB       SENSOR_ANTRIAN_TUTUP, $ ; loop sampai sensor antrian
              tutup aktif

STOP_ANTRIAN_TUTUP:
    CALL       MOTOR_ANTRIAN_TUTUP_MATI

HITUNG_KALENG_ISI:
    INC       R7
    CJNE     R7, #4, JALANKAN_CONVEYOR_UNTUK_TUTUP_KALENG
```

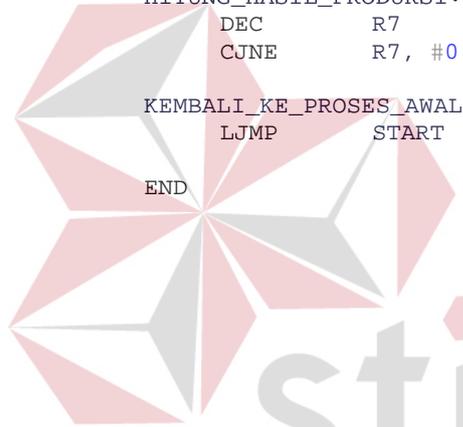
- j. Setelah kaleng kedua terdeteksi sensor tutup_kaleng, *conveyor* berhenti, dan proses penutupan kaleng dilakukan kembali, begitujuga seterusnya sampai keempat kaleng tertutup semua.
- k. Setelah empat buah kaleng tertutup semua, *conveyor* bergerak sampai kaleng terdeteksi oleh sensor hasil melalui *Pin* P1.7.

```
JALANKAN_CONVEYOR_UNTUK_PACKING:
    CALL       MOTOR_CONVEYOR_AKTIF
    JNB       SENSOR_HASIL, $ ; loop sampai sensor hasil aktif,
              untuk menghitung hasil produksi

CEK_SENSOR_HASIL:
    JB          SENSOR_HASIL, $ ; loop sampai sensor hasil pasif
```

1. Sensor hasil melakukan perhitungan mundur sampai ke 4 buah kaleng terdeteksi semua, hal ini dilakukan untuk memastikan bahwa kaleng yang masuk sama dengan kaleng yang keluar. selanjutnya *conveyor* berhenti dan siap untuk melakukan proses seperti semula yaitu Motor DC memutar antrian kaleng kosong seperti proses awal sampai bahan sambal pecel di penampungan pada proses produksi habis yang diketahui melalui komunikasi antar *port 2* mikrokontroler pertama dengan mikrokontroler kedua.

```
STOP_CONVEYOR_UNTUK_PACKING:  
    CALL        MOTOR_CONVEYOR_MATI  
  
HITUNG_HASIL_PRODUKSI:  
    DEC         R7  
    CJNE        R7, #0, JALANKAN_CONVEYOR_UNTUK_PACKING  
  
KEMBALI_KE_PROSES_AWAL:  
    LJMP        START  
  
END
```



INSTITUT BISNIS
& INFORMATIKA
stikom
SURABAYA